Reconstruction dans les systèmes complexes

Thèse

Charles Murphy

Doctorat en physique
Philosophiæ doctor (Ph. D.)

Québec, Canada

Reconstruction dans les systèmes complexes

Thèse

Charles Murphy

Sous la direction de:

Antoine Allard, directeur de recherche

Résumé

À l'ère des données massives, la science des systèmes complexes prend un virage important, basé sur le paradigme des réseaux complexes. Ces systèmes, composés d'un nombre important d'entités en interaction, sont modélisés comme des graphes, dont les noeuds et les liens représentent respectivement les entités et leurs interactions, faisant ainsi abstraction de leur nature. Depuis le début des années 2000, l'analyse des réseaux est guidée par une approche descriptive mettant à l'oeuvre des modèles simples, dans laquelle ils sont comparés qualitativement à des systèmes empiriques. Aujourd'hui, l'intégration des données massives guide une deuxième vague de la science des réseaux, où les modèles de réseaux sont complexifiés et *reconstruits* explicitement à partir de données elles-mêmes.

Dans cette thèse, nous étudions les problèmes de reconstruction de systèmes complexes, et empruntons la représentation des dynamiques sur réseaux, i.e., des processus stochastiques évoluant sur des graphes. Nous nous intéressons particulièrement à deux perspectives : la reconstruction de la structure et la reconstruction de la dynamique. Alors que la première vise à retrouver la structure d'un système complexe en évolution, la seconde cherche à déterminer les mécanismes d'évolution qui le gouvernent.

Notre exploration mène à trois contributions théoriques et numériques. D'abord, nous développons un formalisme informationnel pour quantifier la force de la relation entre la structure et la dynamique d'un système. De ce dernier, nous établissons une correspondance entre reconstructibilité et prévisibilité, i.e., la capacité théorique pour reconstruire et prédire l'évolution d'un système, respectivement. Ensuite, nous spécialisons ce formalisme pour étudier les limites de la reconstruction des réseaux complexes. Nous démontrons que les algorithmes de reconstruction sont limités par la reconstructibilité, laquelle est reliée au système sous-jacent, et établissons un protocole pour l'évaluer empiriquement. Enfin, nous proposons une approche numérique, basée sur l'apprentissage profond, pour reconstruire les mécanismes d'évolution de dynamiques de contagion. Nous démontrons l'étendue de la capacité de notre architecture à reconstruire des processus de complexité variable, incluant la propagation de la COVID-19 en Espagne observée à travers des données empiriques, et appliquons notre méthode pour étudier numériquement la criticalité de ces systèmes.

Abstract

In the era of big data, the science of complex systems is taking an important turn, based on the paradigm of complex networks. These systems, composed of a large number of interacting entities, are modeled as graphs, whose nodes and links respectively represent the entities and their interactions, thus abstracting their nature. Since the early 2000s, network analysis has been guided by a descriptive approach using simple models, in which they are qualitatively compared to empirical systems. Today, the integration of big data guides a second wave of network science, where network models are complexified and explicitly *reconstructed* from the data themselves.

In this thesis, we study the problems of reconstructing complex systems, and adopt the representation of dynamics on networks, i.e., stochastic processes evolving on graphs. We are particularly interested in two perspectives: the reconstruction of their structure and dynamics. While the first aims to find the structure of an evolving complex system, the second seeks to determine the underlying mechanisms that govern it.

Our exploration leads to three theoretical and numerical contributions. First, we develop an information-theoretic formalism to quantify the strength of the relationship between the structure and dynamics of a system. We establish a correspondence between reconstructability and predictability via this framework, i.e., the ability to reconstruct and predict the evolution of a system, respectively. Next, we specialize this framework to study the reconstruction limits of complex networks. We demonstrate that reconstruction algorithms are limited by a system's reconstructability and establish a protocol to empirically evaluate it. Finally, we propose a deep learning based numerical approach to reconstruct the evolution mechanisms of contagion dynamics. We demonstrate the ability of our architecture to reconstruct processes of varying complexity, including empirical data from the spread of COVID-19 in Spain, and apply our method to numerically study the criticality of these systems.

Table des matières

Liste des contributions

Les contributions énumérées ci-après ont été réalisées durant le doctorat ; les articles figurant dans la thèse à titre de chapitres sont indiqués d'un losange (◇).

Articles

◇ On the reconstruction limits of complex networks
C. Murphy, S. Lizotte, F. Thibault, V. Thibeault, P. Desrosiers, A. Allard
arXiv :2501.01437 (2025).

◇ Duality between predictability and reconstructability in complex systems
C. Murphy, V. Thibeault, A. Allard, P. Desrosiers
Nat. Commun. **15**, 4478 (2024).

◇ Deep learning of contagion on complex networks
C. Murphy, E. Laurence, A. Allard
Nat. Commun. **12**, 4720 (2021).

- Network comparison and the within-ensemble graph distance
H. Hartle, B. Klein, S. McCabe, A. Daniels, G. St-Onge, **C. Murphy**, L. Hébert-Dufresne
Proc. R. Soc. A **476**, 2243 (2020).
- Detecting structural perturbations from time series with deep learning
E. Laurence, **C. Murphy**, G. St-Onge, X. Roy-Pomerleau, V. Thibeault
arXiv :2006.05232 (2020).
- Phase transition in the recoverability of network history
J.-G. Young, G. St-Onge, E. Laurence, **C. Murphy**, L. Hébert-Dufresne, P. Desrosiers
Phys. Rev. X **9**, 041056 (2019).

Conférences (sélection)

- On the reconstructability of complex networks (présentation)
  F. Thibault, **C. Murphy**, S. Lizotte, V. Thibeault, P. Desrosiers, A. Allard
  International School and Conference on Network Science, Vienne, Autriche (2023).
- Optimal mesoscopic structure of general binary-state dynamics on networks (présentation)
  J. Lesage, **C. Murphy**, G. St-Onge, L. Hébert-Dufresne, A. Allard
  International School and Conference on Network Science, Vienne, Autriche (2023).
- Deep learning of contagion dynamics on complex networks (présentateur invité)
  C. Murphy, E. Laurence, A. Allard
  Séminaire Quantact, Montréal (Qc), Canada (2022).
- Information theory of dynamics on network (présentation en ligne)
  C. Murphy, V. Thibeault, A. Allard, P. Desrosiers
  International School and Conference on Network Science, Washington (DC), États-Unis (2021).
- Information theory of dynamics on network (présentation en ligne)
  C. Murphy, V. Thibeault, A. Allard, P. Desrosiers
  SIAM : Dynamical systems, Portland (OR), États-Unis (2021).
- Deep Learning of Epidemics Spreading on Complex Networks (présentation en ligne)
  C. Murphy, E. Laurence, A. Allard
  International School and Conference on Network Science, Rome, Italie (2020).
- Learning dynamical epidemic processes on complex network (présentation)
  C. Murphy, E. Laurence, A. Allard
  International School and Conference on Network Science, Burlington (VT), États-Unis (2019).

Liste des tableaux

Liste des figures

Liste des abbréviations et notations

Abbréviations

ER	Modèle **E**rdős-**R**ényi.
BA	Modèle de **B**arabási-**A**lbert.
CM	Modèle de configurations (***C**onfiguration **M**odel* en anglais).
SBM	Modèle stochastique par blocs (***S**tochastic **B**lock **M**odel* en anglais).
DAG	Graphe orienté et acyclique (***D**irected **A**cyclic **G**raph* en anglais).
SIS	Modèle de contagion **S**usceptible-**I**nfecté-**S**usceptible.
MCMC	Monte Carlo par chaîne de Markov (***M**arkov **C**hain **M**onte **C**arlo* en anglais).
IS	Échantillonnage par importance (***I**mportance **S**ampling* en anglais).
MF	Champ moyen (***M**ean-**F**ield* en anglais).
MLE	Estimateur du maximum de vraisemblance (***M**aximum **L**ikelihood **E**stimator* en anglais).
EAP	**E**spérance de la loi ***A P**osteriori*.
MAP	**M**ode de la loi ***A P**osteriori*.
KL	**K**ullback-**L**eibler.
ROC	Fonction d'efficacité du récepteur (***R**eceiver **O**perating **C**haracteristic* en anglais).
AUC	Aire sous la courbe (***A**rea **U**nder the **C**urve* en anglais).
MSE	Erreur quadratique moyenne (***M**ean **S**quared **E**rror* en anglais).
TAU	**T**héorème d'**A**pproximation **U**niverselle.
MLP	Perceptron multicouche (***M**ulti-**L**ayer **P**erceptron* en anglais).
GNN	Réseau de neurones sur graphe (***G**raph **N**eural **N**etwork* en anglais).

Symboles

$\mathcal{S}$	Ensemble et multi-ensemble, c'est-à-dire collection désordonnée sans et avec répétition (symbole majuscule caligraphique).
$\lvert\mathcal{S}\rvert$	Cardinalité de l'ensemble $\mathcal{S}$.
$\boldsymbol{x}$	Vecteur rangé.
$\boldsymbol{x}^\top$	Vecteur colonne.
$(x_i)_{i\in\mathcal{S}}$	Vecteur d'éléments x_i, où l'ensemble $\mathcal{S}$ est omis s'il n'est pas ambigu.
$[\boldsymbol{x}]_i$	Élément i du vecteur $\boldsymbol{x}$.
$(a_{ij})_{(i,j)\in\mathcal{T}}$	Matrice d'éléments a_{ij}, où l'ensemble $\mathcal{T}$ est omis s'il n'est pas ambigu.
$[\mathbf{a}]_{ij}$	Élément a_{ij} de la matrice $\mathbf{a}$.
X	Variable aléatoire (symbole majuscule).
x	Réalisation d'une variable aléatoire X (symbole minuscule).
$P(X)$	Distribution de probabilité de la variable aléatoire discrete X.
$\rho(X)$	Fonction de densité de probabilité de la variable aléatoire continue X.
$X \sim P(X)$	La variable aléatoire X suit la distribution $P(X)$.
$A \equiv B$	Le symbole A est défini par l'expression B.
$\mathbb{E}[f(X)]$, $\mathbb{E}_X[f(X)]$	Espérance de la variable aléatoire $f(X)$.
$\hat{f}$	Estimateur de f.
$\mathbb{I}[c]$	Fonction indicatrice, $\mathbb{I}[c] = 1$ si c est vrai et $\mathbb{I}[c] = 0$ autrement.
$\delta(i,j)$	Delta de Kronecker, $\delta(i,j) = 1$ si $i = j$ et $\delta_{ij} = 0$ autrement.
$\sum_{i\leq j}^{n}$	Sommation sur les paires (i,j) telles que $1 \leq i \leq j \leq n$ où n est omis s'il n'est pas ambigu.
$\prod_{i\leq j}^{n}$	Produit sur les paires (i,j) telles que $1 \leq i \leq j \leq n$ où n est omis s'il n'est pas ambigu.
$[n]$	Ensemble d'entiers contigus $\{1,...,n\}$.

Notation asymptotique

$f(x) = \mathcal{O}(g(x))$	$f(x)$ est du même ordre que $g(x)$, $\exists M > 0$ tel que $\lim_{x\to\infty} \frac{\lvert f(x)\rvert}{\lvert g(x)\rvert} \leq M$.
$f(x) = o(g(x))$	$f(x)$ est dominée par $g(x)$, $\lim_{x\to\infty} \frac{\lvert f(x)\rvert}{\lvert g(x)\rvert} = 0$.
$f(x) \simeq g(x)$	$f(x)$ converge vers $g(x)$, $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$.
$f(x) \sim g(x)$	$f(x)$ croît comme $g(x)$ à une constante près, $\exists c > 0$ tel que $\lim_{x\to\infty} \frac{f(x)}{g(x)} = c$.

À mon garçon, Laurier

Reife des Mannes : das heisst den Ernst wiedergefunden haben, den man als Kind hatte, beim Spiel.

Friedrich Nietzsche

Remerciements

Du plus loin que je me souvienne, la physique a toujours été pour moi source de mystères, de fascination, de frustration intellectuelle, de réussites, d'échecs ; une passion de laquelle j'ai toujours tirée une profonde satisfaction. La conclusion de mes études graduées dans ce magnifique domaine est ainsi l'aboutissement d'un long parcours académique, qui culmine avec plusieurs contributions scientifiques dont je suis particulièrement fier et une thèse de doctorat, dont la rédaction a, sans équivoque, demandé le meilleur de moi-même. Or, jamais je n'y serais parvenu sans de nombreuses personnes qui m'ont supporté dans ma tâche, toutes ces années. Je leur suis infiniment reconnaissant et j'aimerais les remercier sincèrement.

À commencer par mon directeur de thèse, le professeur Antoine Allard, que j'ai eu le plaisir de rencontrer durant ma maîtrise et qui, depuis, a brillamment repris le flambeau du groupe Dynamica. Antoine a été pour moi un mentor, un collègue, un ami, le grand frère que je n'ai jamais eu, un confident, et par-dessus tout, un modèle à suivre. Je me souviens du premier projet sur lequel nous avons travaillé ensemble, durant ma maîtrise. À l'époque, je sentais chez Antoine une certaine méfiance à l'égard de ma méthode de travail, qui je dois l'admettre, manquait de rigueur et de clarté. Avec le temps, Antoine et moi avons appris à travailler ensemble ; j'ai appris à mieux communiquer mes idées et structurer ma pensée, et Antoine a su me faire confiance et me supporter activement dans mes aventures les plus ambitieuses. Je lui serai éternellement reconnaissant pour sa confiance et son soutien.

Une thèse de doctorat n'est pas complète avant d'avoir été mise à l'épreuve. C'est pourquoi je remercie les membres de mon comité d'évaluation, les professeurs Jean-François Fortin, Nicolas Doyon et Giovanni Petri, d'avoir accepté d'y siéger. La nature de mes travaux, qui se veut celle d'un physicien théoricien, porte aussi une saveur interdisciplinaire—un aspect qui pourrait rendre la tâche d'évaluation plus ardue. J'espère donc que les chapitres qui précèdent nos contributions originales permettront de clarifier les motivations et les méthodes employées de nos travaux. Dans l'élan, je remercie mes évaluateurs pour leur ouverture d'esprit. Je souhaite également remercier le professeur Fortin pour sa prélecture attentive de ce document et pour ses commentaires constructifs.

Mon passage aux études graduées aura été marqué par deux grandes figures paternelles que

j'aimerais remercier. La première est celle de mon ancien directeur de recherche, le professeur Louis J. Dubé, avec qui j'ai complété la maîtrise et débuté le doctorat. Monsieur Dubé, quel personnage imposant et ô combien inspirant ! Je lui dois en grande partie mon intérêt pour la physique statistique, la physique numérique et, surtout, la physique de phénomènes étranges que le commun des physiciens n'aborde généralement pas mais qui représentent le pain quotidien des membres du groupe Dynamica. Ce que j'ai appris par-dessus tout de Monsieur Dubé est l'aspiration à l'excellence, la persévérance au travail et la joie que l'on a à marier les deux autour d'un bon café. Je lui suis éternellement reconnaissant pour son enseignement, son excellence et sa bienveillance. La deuxième figure paternelle que j'aimerais remercier est celle de Patrick Desrosiers, qui m'a accompagné tout au long de mes études graduées. L'enthousiasme de Patrick est contagieux, sa rigueur est sans égale et sa générosité, sans borne. Combien d'heures avons-nous discuté de problèmes conceptuels (principalement liés à la théorie de l'information), de trous dans un théorème, d'interprétations de nos résultats, de l'origine de la conscience et de la mécanique quantique ? Probablement pas assez, j'en aurais pris plus. Je le remercie pour sa patience, sa rigueur et sa disponibilité.

J'ai également eu la chance de côtoyer plusieurs membres et ex-membres du groupe Dynamica, qui ont contribué de manière significative à l'avancement de mes travaux. Commençons par les parrains du groupe ; Laurent, pour son ouverture d'esprit et sa créativité, et Jean-Gabriel, pour son pragmatisme et sa passion de la théorie Bayésienne. Leur éthique de travail m'a inspiré depuis le début de mes études graduées et je leur en suis reconnaissant. Ensuite, je tiens à remercier mes collègues immédiats—et amis— : Guillaume, Edward, Vincent et Xavier. Nos soupers bien arrosés, nos parties de basketball et leur excellence ont été pour moi une source inépuisable de motivation. Merci particulièrement à Vincent, auteur parmi plusieurs de mes projets de recherche, avec qui j'ai su aspirer à redevenir un enfant—dans le sens nietzschien du terme. Sans sa curiosité, sa rigueur mathématique et son impitoyable sens critique, nos travaux n'auraient jamais atteint la qualité qu'ils ont aujourd'hui. Puis, je remercie les plus récents membres du groupe : Simon, François et Jérémi, qui ont également contribué de manière substantielle à mes travaux. Avec eux, j'ai eu le plaisir d'apprendre à aimer la vocation de mentor ; leur ouverture et leur entrain m'ont rendu la tâche facile et je les en remercie. Enfin, je veux remercier les autres membres du groupe avec qui je n'ai pas eu le plaisir de travailler directement, mais qui ont contribué à l'ambiance chaleureuse et productive du groupe : Behnaz, Béatrice, Olivier, Antoine et les nombreux autres membres que je n'ai pas eu la chance de connaître davantage. Je vous remercie du fond du coeur.

En dehors du cercle académique, je tiens à remercier Martin, mon mentor chez Hectiq.ai, qui m'a donné la chance de terminer mes études tout en intégrant son équipe. Sans son soutien, sa bienveillance et sa confiance, je n'aurais jamais pu terminer mes études. Je tiens également à remercier Alexis, Antoine, Charles-D. et mes amis proches avec qui je vais à la pêche à chaque été ; mes coéquipiers de bateau-dragon Edouard, Marc, Catherine et plu-

sieurs autres ; la gang à Dario, Simon, Pierre-Luc et, bien sûr, Dario ; François, mon entraîneur de taekwondo ; la famille de Geneviève, Josée, Gervais, Gabriel et Valérie qui m'ont toujours accueilli chaleureusement chez eux. Ils m'ont tous aidé à m'échapper durant ces dures années et je leur en suis reconnaissant. Je dois également une grande partie de ma réussite à mes parents, Martine et Paul, à ma soeur et mon beau-frère, Frédérique et Jeff, qui ont toujours été présents et m'ont supporté inconditionnellement dans mes rêves. Merci pour votre amour et votre soutien.

Finalement, je dois mes derniers remerciements—les plus importants—à Geneviève, la mère de mes enfants Laurier et Cécile. Je vous dédie cette thèse et vous aime de tout mon coeur.

Avant-propos

Les articles qui suivent ont été publiés (ou sont en voie de l'être) en anglais dans des revues scientifiques et ont été directement intégrés à la thèse comme chapitre. De fait, leur contenu n'a pas été traduit en français, mais a tout de même été modifié par soucis de cohérence avec le contenu présenté dans la thèse et afin de se conformer au format exigé par la Faculté des études supérieures et postdoctorales de l'Université Laval. En tant que premier auteur, j'ai mené ces projets de recherche, apporté les idées originales, et contribué aux divers calculs théoriques et numériques. J'ai également été principal contributeur à la rédaction des manuscrits. Mes coauteurs ont participé à l'élaboration des projets et des techniques numériques, ont peaufiné les calculs analytiques et ont révisé les manuscrits.

- CHAPITRE 5
  Duality between predictability and reconstructability in complex systems
  C. Murphy, V. Thibeault, A. Allard, P. Desrosiers
  Nat. Commun. **15**, 4478 (publication : mai 2024).
- CHAPITRE 6
  On the reconstruction limits of complex networks
  C. Murphy, S. Lizotte, F. Thibault, V. Thibeault, P. Desrosiers, A. Allard
  arXiv :2501.01437
  En révision. *Sci. Adv.* (soumission : janvier 2025).
- CHAPITRE 8
  Deep learning of contagion on complex networks
  C. Murphy, E. Laurence, A. Allard
  Nat. Commun. **12**, 4720 (publication : août 2021).
- ANNEXE A
  Typed approximate master equations of binary-state dynamics on random graphs
  C. Murphy, J. Lesage, A. Allard
  en prépration.

Introduction

La physique des systèmes complexes

En mars 2020, le monde a été frappé par une pandémie causée par le coronavirus SARS-CoV-2, responsable de la maladie COVID-19. À ce jour, ce virus a provoqué 7 millions de décès parmi 675 millions d'individus infectés [330]. Le sérieux de la situation a forcé plusieurs gouvernements à imposer jusqu'à la fin de 2022 des mesures sanitaires strictes et des campagnes de vaccination de masse afin de contrôler la propagation du virus [41, 203]. En réponse à ces mesures sanitaires, des mouvements sociaux conspirationnistes, contestant leur efficacité et la véracité de la pandémie elle-même, ont émergé et se sont propagés à travers les médias sociaux, amplifiant leur portée [90]. Devenu un cas d'étude de premier plan, la pandémie de COVID-19 a exposé la complexité de notre société humaine, l'émergence de comportements collectifs et la difficulté que nous avons à les comprendre, les prédire et les contrôler.

Dans cette thèse, nous nous intéressons à des systèmes telles que les épidémies, qui appartiennent à la grande famille des *systèmes complexes*. Dans son célèbre essai de 1948 [324], Warren Weaver, l'un des physiciens fondateurs de la science de la complexité, décrit ces systèmes comme « [...] *dealing simultaneously with a sizable number of factors which are interrelated into an organic whole* »—une classe de phénomènes qu'il qualifie de *complexité organisée*. À l'époque, Weaver avait identifié une panoplie de systèmes appartenant à cette classe, incluant des systèmes biologiques, physiques, sociaux et économiques, et avait espoir qu'un jour une physique de la complexité organisée trouverait le même succès que celle des siècles qui l'ont précédée. C'est Philip Anderson, lauréat du prix Nobel de physique de 1977, qui introduit le concept d'émergence dans les systèmes complexes : des phénomènes non locaux issus d'une description macroscopique du système [52]. Dans son article *More is different* [10], Anderson soutient qu'une théorie fondamentale de la matière permettrait difficilement de prédire de tels phénomènes, mentionnant les exemples de la supraconductivité et de la superfluidité en mécanique quantique. Selon Anderson, une vision constructionniste de la science, dans laquelle tout phénomène peut ultimement être prédit par les lois fondamentales de la physique, est insuffisante pour expliquer l'émergence de la complexité, laquelle doit être décrite par une théorie propre—une physique des systèmes complexes.

Reconstruction dans la science des réseaux

Plus de 50 ans après la publication de l'essai d'Anderson, la science des systèmes complexes se veut une discipline plus mature. Ainsi, en 2010, James Crutchfield et Karoline Wiesner, des chercheurs de premier ordre en science de la complexité, revisitent la discussion de Weaver et Anderson dans leur article *Simplicity and complexity* [68]. Leur conclusion est claire : comprendre la structure des systèmes complexes est vital pour les caractériser et prédire leur comportement. C'est dans ce paradigme que s'inscrit la *science des réseaux complexes*. Un réseau est une abstraction mathématique tirée de la théorie des graphes [40] dans laquelle les nombreuses composantes d'un système sont reduites à des noeuds (représentant les composantes) et des liens (représentant leurs interactions). En somme, les réseaux permettent de définir des propriétés d'intérêt, reliées à des phénomènes collectifs, notamment la connectivité, la modularité, l'agglomération des noeuds et leur centralité [221]. Les réseaux permettent également de définir des processus dynamiques tels que la propagation d'information, d'épidémies ou de rumeurs [231]. La représentation réseau est universelle et se trouve partout [17] : les réseaux sociaux, l'internet et le *World-Wide Web*, les réseaux métaboliques, d'interactions de protéines et de gènes, les réseaux de transport et de communication, les réseaux de neurones, les réseaux de collaboration scientifique, lesquels sont des exemples parmi tant d'autres.

Dans ses premiers balbutiements, à l'aube des années 2000, la science des réseaux complexes pullule de découvertes empiriques et de modèles mathématiques simples pour les expliquer. Barabási et Albert [20], en 1999, démontrent empiriquement la quasi-omniprésence des réseaux *scale-free*, dans lesquels la distribution du nombre de liens par noeud est invariante d'échelle, ce qu'ils expliquent par un mécanisme d'attachement préférentiel. Par la suite, Pastor-Satorras et Vespignani [232] démontrent l'alarmante efficacité des réseaux *scale-free* pour la propagation d'épidémies. Quelques années plus tôt, Watts et Strogatz [323] introduisent le concept de *small-world* pour décrire les réseaux sociaux—ni complètement aléatoires, ni complètement réguliers, mais plutôt un mélange des deux. Puis, accompagné de Newman, ils étudient le modèle des configurations [223], qu'ils utilisent pour décrire plusieurs réseaux sociaux [217, 224]. Cette approche exploratoire qui caractérise cette première vague de la science des réseaux complexes se montre fructueuse encore à ce jour.

Jusqu'à récemment, la science des réseaux se voulait une discipline d'avantage descriptive que prédictive. Or, depuis quelques années, une deuxième vague de travaux s'est amorcée dont la méthodologie, plus rigoureuse, est de travailler directement avec des données empiriques afin de *reconstruire* les modèles de réseaux—i.e., les inférer—, plutôt que de les utiliser de manière *ad hoc* [338]. La détection de communauté [94] a connu une telle transformation, passant d'approches heuristiques basées sur la maximisation de la modularité [110] à des méthodes Bayésiennes [240] beaucoup plus robustes et fiables. Cette deuxième vague remet également en question la validité des graphes empiriques eux-mêmes [220], et introduit des

modèles de reconstruction de graphes qui modélisent et infèrent l'incertitude dans ces données [241, 339, 342]. L'avènement des base de données de plus en plus vastes couplé à ces nouvelles approches statistiques consolide les acquis de la science des réseaux et ouvre la voie vers de nouvelles découvertes et applications dans les systèmes complexes.

Limites dans les dynamiques sur réseaux

Les travaux originaux présentés dans la thèse s'inscrivent dans cette récente tendance, pour laquelle la reconstruction dans les *dynamiques sur réseaux* est mise de l'avant. Les dynamiques sur réseaux sont des outils puissants où un mécanisme d'évolution, faisant intervenir explicitement la structure d'un système, fait évoluer ses composantes dans le temps. Ces modèles se sont montrés capables de prédire mathématiquement une variété de phénomènes émergents : les transitions de phase dans la propagation des épidémies [231, 294, 296, 297], l'activité neuronale dans le cerveau [289], et la synchronisation d'oscillateurs couplés [14] sont des exemples canoniques qui témoignent de leur succès.

En plus d'être des modèles prédictifs, les dynamiques sur réseaux peuvent également être utilisées pour la reconstruction. Le cas échéant, on les utilise dans le problème inverse, i.e., pour inférer la structure du réseau à partir de données temporelles. La reconstruction de réseaux à partir de données temporelles est un problème bien connu [47], particulièrement en neurosciences [26] où les réseaux de neurones empiriques sont généralement incomplets. Dans ce contexte, la reconstruction permet de tirer parti de données d'activité neuronale pour inférer les connexions entre les neurones du cerveau, et enfin déterminer des cartes de connectivité fonctionnelle. Néanmoins, il est difficile de quantifier à quel point ces cartes nous informent de la connectivité anatomique—dite structurelle—du cerveau. Le processus de reconstruction pourrait être fondamentalement limité par la nature même des données, ce que les méthodes de reconstruction actuelles ne nous permettent pas de déterminer.

Lorsque des données temporelles issues d'un système et sa structure sont connue, il est possible de reconstruire les mécanismes d'évolution qui gouvernent le système. Cette approche s'avère fort utile lorsque ces mécanismes sont complexes et inconnus. Typiquement, les modèles dynamiques se voulaient des représentations simplifiées de phènomènes réels—donc mésadaptées pour la reconstruction—dans le but d'en extraire des intuitions. Le modèle susceptible-infecté-susceptible (SIS) est un exemple classique de dynamique simplifiée dans lequel un individu peut être dans l'un des deux états disponibles—*susceptible* ou *infecté*—et les transitions sont régies par des taux de transmission et de guérison constants via un réseau de contact statique. Des travaux récents ont tenté d'adapter ces modèles simples pour les rendre plus réalistes [316], en spécialisant la structure du réseau d'interaction [203] ou en introduisant des mécanismes de contagion plus sophistiqués [132]. Sur un autre front, certains travaux explorent la possibilité de construire des modèles dynamiques de toute pièce, à partir de données [49, 166, 235]. Les récents avancements en apprentissage profond [119] sont

à l'origine de ces travaux, dans lesquels les dynamiques sont représentées par des réseaux de neurones. Cette nouvelle approche prometteuse pourrait repousser les limites actuelles des modèles de dynamiques sur réseaux.

Les mécanismes d'évolution, la structure et les données temporelles forment les trois facettes des dynamiques sur réseaux. Dans chacun des cas discutés ci-haut—prédiction de l'évolution, reconstruction de la structure et reconstruction des mécanismes—deux des trois facettes sont connues et l'objectif est de déterminer la troisième. Dans cette thèse, nous nous intéressons à ces trois types de problèmes, et particulièrement aux deux derniers. Pour ce faire, nous adoptons une approche généraliste, basée sur des principes fondamentaux et étudions des cas d'espèce variés pour explorer les différents aspects de ces problèmes, principalement les limites de notre capacité à les résoudre.

Organisation de la thèse

La thèse est divisée en trois parties. La Partie I se veut une introduction générale et pédagogique aux processus dynamiques sur réseaux, introduisant principalement les concepts, la notation et les outils mathématiques et numériques utilisés dans la thèse. Les Parties II et III présentent nos contributions originales, principalement composées d'articles publiés dans des journaux scientifiques (les Chapitres 5, 6 et 8).

Partie I — La Partie I est elle-même divisée en trois chapitres. Le Chapitre 1 introduit les concepts de base en théorie des probabilités—la fondation mathématique de notre travail. Au Chapitre 2, nous introduisons la théorie des graphes et présentons plusieurs modèles de réseaux complexes utilisés dans nos travaux. Enfin, le Chapitre 3 présente les processus stochastiques sur graphes. Nous y discutons plus particulièrement les chaînes de Markov, la notion de localité dans les dynamiques sur réseaux et leur criticalité.

Partie II — La Partie II met l'accent sur le problème de reconstruction de réseaux à partir de séries temporelles. Le Chapitre 4 présente le cadre théorique nécessaire pour aborder ce problème, soit la théorie bayésienne et la théorie de l'information. Au Chapitre 5, nous présentons un formalisme informationnel pour décrire le problème de reconstruction. Nous démontrons l'existence d'une dualité entre notre capacité à reconstruire un réseau et notre capacité à prédire les séries temporelles issues de ce réseau. Au Chapitre 6, nous adaptons ce formalisme pour étudier les limites de la reconstruction des réseaux complexes.

Partie III — La Partie III se concentre sur la reconstruction des mécanismes dans les dynamiques sur réseaux. Pour ce faire, nous utilisons le paradigme de l'apprentissage profond et, plus précisément, les réseaux de neurones sur graphes : lesquels sont présentés au Chapitre 7. Le Chapitre 8 présente une méthode numérique pour reconstruire les dynamiques de contagion sur réseaux. Nous démontrons que cette méthode est capable de reconstruire des dynamiques sur réseaux à partir de données synthétiques et réelles.

Première partie

Éléments de structure et de dynamique

Chapitre 1

Fondation : La théorie des probabilités

La théorie que nous développons dans cette thèse repose sur une fondation solide—la théorie des probabilités. Les notions de probabilité et de variable aléatoire s'appliquent dans le développement de modèles statistiques, de méthodes numériques diverses, de techniques d'inférence ; la liste des applications est longue. En général, on utilise les variables aléatoires afin de modéliser une forme d'*incertitude* en rapport avec le système qui nous intéresse. Cette incertitude peut prendre la forme d'états cachés ou inconnus, d'incertitude dans les mesures du système, ou même sur les paramètres du modèle qu'on utilise pour représenter le système. En bref, la théorie des probabilités est tout à fait disposée pour la modélisation de systèmes complexes, et plus particulièrement pour leur reconstruction.

1.1 Espace de probabilité

L'espace de probabilité est le substrat fondamental derrière la théorie des probabilités, qui nous vient de la théorie de la mesure [15]. Intuitivement, l'espace de probabilité permet de décrire l'ensemble des issues possibles d'une expérience aléatoire en assignant pour chacune d'elles une *probabilité*, c'est-à-dire un scalaire compris inclusivement entre 0 et 1 qui, d'une certaine manière, quantifie sa fréquence d'apparition ou sa propension de survenir. Par exemple, si un certain événement survient avec une probabilité p donnée, alors on s'attendrait à ce que si on répétait l'expérience un grand nombre de fois, la fréquence avec laquelle cet événement surviendrait se rapprocherait de p. Cette interprétation, dite fréquentiste, du concept de probabilité est l'une parmi plusieurs autres [125].

Formellement, on utilise la théorie de la mesure pour définir correctement et en toute généralité l'espace de probabilité.

Définition 1.1 (Espace de probabilité)**.** *Un espace de probabilité* $(\Omega, \mathcal{A}, \mathbb{P})$ *est un espace dit* mesuré, *où*

1. Ω *est un ensemble appelé l'espace d'échantillonnage ;*

2. $\mathcal{A}$ *est une* σ*-algèbre de* Ω *appelée l'espace des événements ;* [1]

3. $\mathbb{P} : \mathcal{A} \to [0, 1]$ *est une mesure de probabilité.* [2]

Intuitivement, l'espace d'échantillonnage Ω représente l'ensemble des issues possibles et l'espace des événements $\mathcal{A}$, l'ensemble de toutes les combinaisons d'observations. Prenons l'exemple d'une pièce de monnaie qu'on lancerait une fois et pour laquelle la pièce tombe sur face (1) avec probabilité p et sur pile (0) avec probabilité $1 - p$. Alors, l'espace d'échantillonnage est $\Omega = \{0, 1\}$, l'espace des événements est l'*ensemble puissance* de Ω, c'est-à-dire $\mathcal{A} = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$, et la mesure de probabilité est

$$\mathbb{P}(\emptyset) = 0, \quad \mathbb{P}(\{1\}) = p, \quad \mathbb{P}(\{0\}) = 1 - p, \quad \mathbb{P}(\{0, 1\}) = 1 .$$

Les espaces de probabilités dont Ω est dénombrable (et potentiellement infini) sont des espaces *discrets*. Comme nous le verrons plus loin, les espaces de graphes et les processus stochastiques consultés dans nos travaux seront effectivement des espaces discrets. Dans ce cas, l'ensemble puissance de Ω comme espace des événements $\mathcal{A}$ est généralement adéquat pour définir une mesure de probabilité correctement normalisée. Notons que les cas continus—c'est-à-dire où Ω est non dénombrable, par exemple l'intervalle unité $\Omega = [0, 1]$—, nécessitent des considérations supplémentaires afin d'obtenir une mesure de probabilité non triviale. Nous laissons ces cas de côté dans cette thèse, par souci de simplicité.

Considérant des espaces de probabilité discrets, nous sommes en droit d'adopter un certain nombre de raccourcis. Par exemple, on pourra toujours considérer l'ensemble puissance de Ω, dénoté 2^Ω, comme espace des événements. [3] De ce fait, on omettra généralement de spécifier l'espace des événements, que l'on considérera comme 2^Ω. De plus, comme tous

1. Une σ-algèbre de Ω, aussi appelé tribu, est un ensemble de sous-ensembles de Ω possèdant les propriétés suivantes :

   a) $\emptyset \in \mathcal{A}$;

   b) Pour tout $A \in \mathcal{A}$, alors $A_c = \Omega \setminus A \in \mathcal{A}$ (fermeture sous la complémentarité);

   c) Pour tout $A, B \in \mathcal{A}$, alors $C = A \bigcup B \in \mathcal{A}$ (fermeture sous l'union dénombrable).

2. En théorie de la mesure, une mesure μ est une fonction qui agit sur un élément d'une σ-algèbre $\mathcal{A}$ et qui retourne un scalaire non-négatif ou l'infini, c'est-à-dire $\mu : \mathcal{A} \to \mathbb{R}_+ \wedge \{\infty\}$. Une mesure possède aussi les propriétés suivantes :

   a) $\mu(\emptyset) = 0$;

   b) Pour tout $A, B \in \mathcal{A}$, $\mu(A \bigcup B) = \mu(A) + \mu(B)$ (σ-additivité).

Pour un espace de probabilité, la mesure de probabilité est bornée entre 0 et 1.

3. Le nombre d'éléments dans l'ensemble puissance est donné par

$$|2^\Omega| = \sum_{n=1}^{|\Omega|} \binom{|\Omega|}{n} = \underbrace{1}_{\emptyset} + \underbrace{|\Omega|}_{\text{singletons}} + \underbrace{\frac{|\Omega|(|\Omega| - 1)}{2}}_{\text{paires}} + \cdots + \underbrace{1}_{\Omega} = 2^{|\Omega|} , \tag{1.1}$$

ce qui justifie en partie la notation 2^Ω.

les singletons $\{\omega\}$ de Ω seront éléments de 2^Ω, alors la probabilité de $A = \{\omega_1, \cdots, \omega_K\}$ s'exprimera toujours comme une somme sur les singletons, en vertue de la σ-additivité :

$$\mathbb{P}(A) = \sum_{\omega \in A} \mathbb{P}(\{\omega\})$$

où $\mathbb{P}(\{\omega\})$ est la probabilité de l'issue ω.

1.2 Variable aléatoire

L'espace de probabilité représente le substrat sur lequel nos expériences aléatoires reposent. En général, on sera d'avantage intéressé au comportement de *variables aléatoires* qui représentent des quantités d'intérêt. Une variable aléatoire X est une fonction qui envoie chaque élément ω de l'espace d'échantillonnage Ω vers $X(\omega)$, c'est-à-dire un élément d'un autre espace mesurable [15]. Formellement, on définit une variable aléatoire comme suit :

Définition 1.2 (Variable aléatoire)**.** *Une variable aléatoire $X : \Omega \to \mathcal{X}$ est une fonction inversible et mesurable sur l'espace de probabilité $(\Omega, \mathcal{A}, \mathbb{P})$ vers un espace mesurable $(\mathcal{X}, \mathcal{B})$ qui satisfait*

$$X^{-1}(B) = \{\omega : X(\omega) \in B\} \in \mathcal{A}\,, \quad B \in \mathcal{B}\,. \tag{1.2}$$

On appelle $P(X)$ la distribution (ou la loi) de X. La distribution de X est également une mesure, qui hérite des mêmes propriétés que $\mathbb{P}$. En principe, $P(X)$ peut s'écrire en termes de la mesure de probabilité comme suit :

$$P(X \in B) \equiv \mathbb{P}\Big(X^{-1}(B)\Big)\,, \tag{1.3}$$

Dans le cas de variables discrètes, de par la σ-additivité de $\mathbb{P}$, on peut réécrire $P(X \in B)$ comme une somme sur les éléments de B :

$$P(X \in B) = \sum_{x \in B} \mathbb{P}\Big(X^{-1}(\{x\})\Big) = \sum_{x \in B} P(X = x)\,. \tag{1.4}$$

où $P(X = x)$ est défini comme la probabilité que X retourne la valeur x.

Notons que $P(X)$ n'est pas une probabilité, mais définit une composition des fonctions P et X. Généralement, les mathémaciens préfèrent utiliser la notation P_X pour désigner la loi de X, et $P_X(x)$ pour la probabilité de x. Cette notation est plus standard, mais elle devient rapidement lourde lorsqu'on invoque plusieurs variables aléatoires. Ce sera bien sûr le cas dans cette thèse, c'est pourquoi on optera pour la notation $P(X)$. Nous utiliserons une notation similaire pour toute fonction f qui dépend de la valeur de X. On notera $f(X)$ la fonction f évaluée en X, qui sera comme $P(X)$ une composition des fonctions f et X.

À partir de la distribution, on peut définir des statistiques qui la caractérise. On définit l'espérance de X, un scalaire aussi appelé premier moment ou moyenne de X, comme suit :

Définition 1.3 (Espérance d'une variable aléatoire)**.** *L'espérance d'une variable aléatoire discrète X, notée $\mathbb{E}[X]$ (ou $\langle X \rangle$), est donnée par*

$$\mathbb{E}[X] = \langle X \rangle = \sum_{x \in \mathcal{X}} x\, P(X = x)\,. \tag{1.5}$$

L'espérance est une opération linéaire, de sorte que $\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$ pour tout $a, b \in \mathbb{R}$. Une autre statistique d'intérêt est la variance $\mathbb{V}[X]$, qui mesure la dispersion de X autour de sa moyenne :

$$\mathbb{V}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] \tag{1.6}$$

La variance peut s'exprimer en termes des deux premiers moments de X lorsqu'on développe l'expression ci-dessus :

$$\mathbb{V}[X] = \mathbb{E}\left[X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2\,. \tag{1.7}$$

où on a utilisé la linéarité de l'espérance pour écrire $\mathbb{E}[X\mathbb{E}[X]] = \mathbb{E}[X]^2$. On généralise naturellement l'espérance de X à toute fonction $f : \mathcal{X} \to \mathbb{R}$ qui dépend de X, comme $\mathbb{E}[f(X)]$ (ou $\langle f(X) \rangle$)[4], étant donné que $f(X)$ est aussi une variable aléatoire :

$$\mathbb{E}[f(X)] = \sum_{x \in \mathcal{X}} f(x) P(X = x)\,. \tag{1.8}$$

1.3 Équivalence et convergence des variables aléatoires

Il existe plusieurs notions d'équivalence entre des variables aléatoires. La raison est que deux variables aléatoires X et Y peuvent prendre plusieurs valeurs sur leur espace respectif. Ce faisant, il faut être capable de différentier les cas où X et Y ont toujours la même valeur, ou si leurs distributions sont identiques, par exemple. La notation d'équivalence la plus stricte est celle d'égalité presque sûre, définie comme suit :

Définition 1.4 (Égalité presque sûre de variables aléatoires)**.** *Deux variables aléatoires X et Y sont égales presque sûrement, c'est-à-dire $X = Y$, si et seulement si*

$$\mathbb{P}(\{\omega : X(\omega) \neq Y(\omega)\}) = 0\,. \tag{1.9}$$

Une notion d'équivalence moins forte que l'égalité presque sûre est l'égalité en distribution :

Définition 1.5 (Égalité en distribution)**.** *Deux variables aléatoires X et Y, respectivement mesurables sur $(\mathcal{X}, \mathcal{B})$ et $(\mathcal{Y}, \mathcal{C})$ sont égales en distribution, c'est-à-dire $X \simeq Y$, si et seulement si $\mathcal{B} \subseteq \mathcal{C}$ et*

$$P(X \in B) = P(Y \in B)\,, \quad \forall B \in \mathcal{B}\,. \tag{1.10}$$

4. On utilisera également les notations $\mathbb{E}_X[f(X)]$ et $\langle f(X) \rangle_X$ si le contexte n'est pas suffisant pour spécifier la variable aléatoire sur laquelle l'espérance est calculée.

Intuitivement, l'égalité en distribution signifie que les deux variables aléatoires partagent exactement la même distribution, et, donc, le même support. Conséquemment, pour tout $C \in \mathcal{C} \setminus \mathcal{B}$, $P(Y \in C) = 0$ puisque $P(X \in \mathcal{B}) = P(Y \in \mathcal{B}) = 1$.

De ces définitions, il est possible de définir la convergence en théorie des probabilités. Le concept de convergence intervient principalement dans le cadre des limites de suites de variables aléatoires. Notamment, elle apparaît dans la loi des grands nombres, le théorème central limite, et d'autres résultats importants en statistique (voir Réf. [141, Chapitre 7]). Voici une définition qui sera utile pour la suite :

Définition 1.6 (Convergence)**.** *Soit une séquence de variables aléatoires (X_n). On dit que X_n* ***converge presque sûrement*** *vers X, c'est-à-dire $X_n \longrightarrow X$, si et seulement si*

$$\mathbb{P}\Big(\Big\{ \omega \,:\, \lim_{n\to\infty} X_n(\omega) = X(\omega) \Big\} \Big) = \mathbb{P}(\Omega) = 1\,. \tag{1.11}$$

Similairement, on dit que X_n ***converge en probabilité*** *vers X, c'est-à-dire $X_n \xrightarrow{\mathbb{P}} X$, si et seulement si*

$$\lim_{n\to\infty} P(D_n > \epsilon) = 0\,, \quad \forall \epsilon > 0 \tag{1.12}$$

où $D_n = |X_n - X|$ est la norme entre X_n et X. De plus, on dit que X_n ***converge en distribution*** *vers X, c'est-à-dire $X_n \rightsquigarrow X$, si et seulement si, pour tout $B \in \mathcal{B}$*

$$\lim_{n\to\infty} P(X_n \in B) = P(X \in B)\,. \tag{1.13}$$

Finalement, on dit que X_n ***converge en moyenne*** *vers X, c'est-à-dire $X_n \xrightarrow{\mathrm{M}} X$, si et seulement si*

$$\lim_{n\to\infty} \mathbb{E}[X_n] = \mathbb{E}[X]\,. \tag{1.14}$$

Ces différentes formes de convergence sont ordonnées : une forme plus forte implique une forme plus faible. Par exemple, si $X_n \longrightarrow X$, alors $X_n \xrightarrow{\mathbb{P}} X$, donc $X_n \rightsquigarrow X$, et finalement, $X_n \xrightarrow{\mathrm{M}} X$. De ce fait, la convergence presque sûre est la forme la plus stricte, suivi de la convergence en probabilité, puis de la convergence en distribution et de la convergence en moyenne.

1.4 Plusieurs variables aléatoires

Le cas d'une variable aléatoire se généralise directement au cas de plusieurs variables aléatoires. Considérons une paire de variables aléatoires $X : \Omega \to \mathcal{X}$ et $Y : \Omega \to \mathcal{Y}$. Il est possible que ces variables aléatoires couvrent des sous-ensembles de Ω dont l'intersection est non nulle (voir Fig. 1.1). On appelle la probabilité de cette intersection, la *probabilité conjointe*.

Définition 1.7 (Distributions conjointe et marginale)**.** *La distribution conjointe $P(X, Y)$ de deux variables aléatoires X et Y évaluées sur (B, C), où $B \subseteq \mathcal{X}$ et $C \subseteq \mathcal{Y}$, est*

$$P(X \in B, Y \in C) \equiv \mathbb{P}\Big(X^{-1}(B) \bigcap Y^{-1}(C) \Big)\,. \tag{1.15}$$

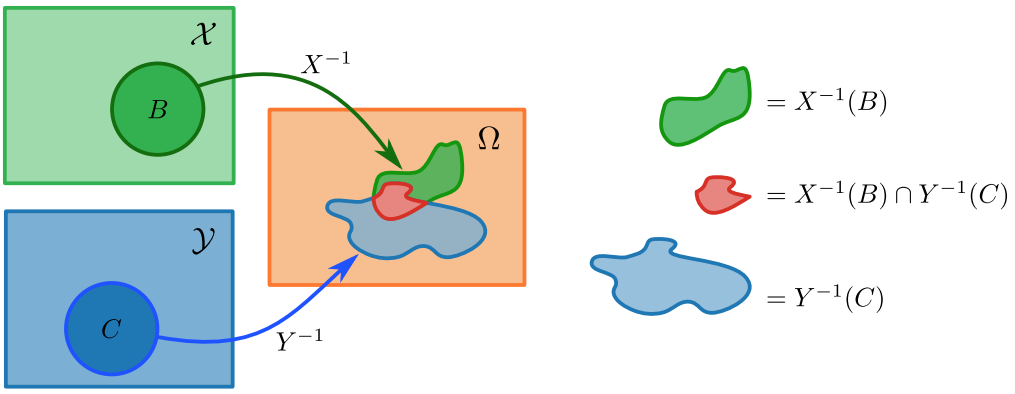


FIGURE 1.1 – Illustration de deux variables aléatoires X et Y respectivement mesurables sur les espaces $(\mathcal{X}, \mathcal{B})$ et $(\mathcal{Y}, \mathcal{C})$. Aux événements $B \in \mathcal{B}$ et $C \in \mathcal{C}$ correspondent des sous-ensembles de Ω, notés $X^{-1}(B)$ (en vert) et $Y^{-1}(C)$ (en bleu), dont l'intersection est $X^{-1}(B) \cap Y^{-1}(C)$ (en rouge). On appelle la mesure de cette intersection la probabilité conjointe de X et Y.

Dans le cas de plusieurs variables, on appelle $P(X)$ et $P(Y)$ les distributions marginales de X et Y, respectivement, que l'on relie à la distribution conjointe comme suit :

$$P(X \in B) = P(X \in B, Y \in \mathcal{Y}), \quad P(Y \in C) = P(X \in \mathcal{X}, Y \in C). \tag{1.16}$$

La dernière condition nous assure que $P(X \in \mathcal{X}) = P(Y \in \mathcal{Y}) = 1$. À partir des distributions conjointes et marginales, on définit la distribution conditionnelle :

Définition 1.8 (Distribution conditionnelle)**.** *La distribution conditionnelle $P(X|Y)$, qui se lit comme la probabilité de X sachant Y, est le ratio suivant :*

$$P\Big(X \in B \,\Big|\, Y \in C\Big) = \frac{P(X \in B, Y \in C)}{P(Y \in C)} \tag{1.17}$$

En bref, la distribution conditionnelle compare le volume de l'intersection entre X et Y au volume occupé par Y. En principe, lorsqu'on s'intéresse à deux variables aléatoires, il existe deux distributions conditionnelles, c'est-à-dire $P(X|Y)$ et $P(Y|X)$, lesquelles sont reliées par le *théorème de Bayes* :

Theorème 1.1 (Théorème de Bayes)**.**

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)}. \tag{1.18}$$

Démonstration. Partons de la définition de la distribution conditionnelle, avec laquelle on peut réécrire la distribution conjointe de deux manières :

$$P(X, Y) = P(X)\,P(Y|X) = P(Y)\,P(X|Y). \tag{1.19}$$

En divisant des deux côtés par $P(X)$, on obtient directement l'Éq. (1.18). Par symétrie, on peut également réécrire $P(Y|X)$ sous la forme de l'Éq. (1.18) :

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)}\,, \quad P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)}\,. \tag{1.20}$$

□

Le cas à n variables aléatoires s'obtient directement en généralisant le cas à deux variables : on considère un vecteur de variables aléatoires $\boldsymbol{X} = (X_1, \cdots, X_n)$, dont la distribution conjointe est $P(\boldsymbol{X}) = P(X_1, \cdots, X_n)$ et la distribution marginale de X_i est $P(X_i)$. De fait, on peut alors définir les distributions conditionnelles $P(\boldsymbol{X}_I|\boldsymbol{X}_J)$ pour toutes combinaisons de variables, lesquelles sont également reliées par le théorème de Bayes généralisé :

$$P(\boldsymbol{X}_I|\boldsymbol{X}_J) = \frac{P(\boldsymbol{X}_I)P(\boldsymbol{X}_J|\boldsymbol{X}_I)}{P(\boldsymbol{X}_J)}\,, \tag{1.21}$$

où $\boldsymbol{X}_I = (X_i)_{i\in\mathcal{I}}$ et $\boldsymbol{X}_J = (X_j)_{j\in\mathcal{J}}$.

1.5 Échantillonnage de variables aléatoires

L'échantillonnage est une étape critique lorsqu'on souhaite faire des calculs numériques. Par exemple, pour évaluer l'espérance μ d'une variable aléatoire X quelconque de variance σ^2, il est possible d'en faire une approximation via un estimateur *Monte-Carlo*, noté $\bar{X}_n$. D'abord, on génère un échantillon $(X_1, X_2, ..., X_n)$, où X_k est une réalisation aléatoire de X tirée selon la probabilité $P(X = X_k)$ *indépendamment* des autres réalisations. L'estimateur Monte-Carlo $\bar{X}_n$ est une variable aléatoire en soit, qui se calcule comme suit :

$$\bar{X}_n = \frac{1}{n}\sum_{k=1}^{n} X_k\,. \tag{1.22}$$

En vertu de la loi des grands nombres [178], cet estimateur converge en probabilité vers l'espérance de X :

$$\lim_{n\to\infty} P\Big(|\bar{X}_n - \mu| > \epsilon\Big) = 0\,. \tag{1.23}$$

En effet, dans cette limite, l'estimateur Monte-Carlo converge en distribution vers une variable normale, en vertu du théorème central limite [141], de moyenne μ dont la variance converge vers σ^2/n :

$$\mathbb{E}[\bar{X}_n] = \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}[X_k] = \frac{1}{n}\sum_{k=1}^{n}\mu = \mu \tag{1.24}$$

$$\mathbb{V}[\bar{X}_n] = \mathbb{E}\left[\bar{X}_n^2\right] - \mu^2 = \frac{\sigma^2}{n} + \mu^2 - \mu^2 = \frac{\sigma^2}{n}\,. \tag{1.25}$$

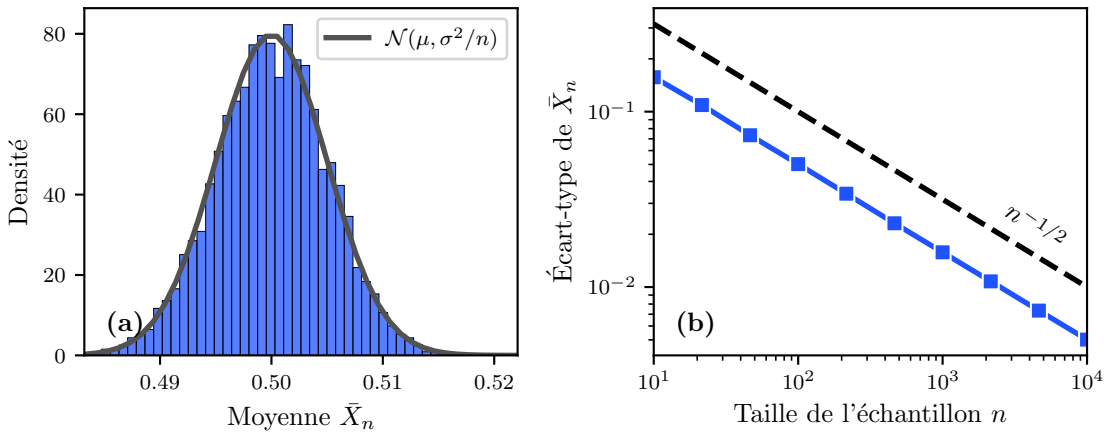


FIGURE 1.2 – Convergence de l'estimateur Monte-Carlo $\bar{X}_n$ pour une variable de Bernoulli X de paramètre $p = \frac{1}{2}$. On génère, pour plusieurs valeurs de n, 10 000 échantillons de $\bar{X}_n$ et on trace en (a) son histogramme pour $n = 10^4$. Ce dernier converge vers une distribution normale $\mathcal{N}(\mu, \sigma^2/n)$, de moyenne $\mu = p = \frac{1}{2}$ et de variance $\sigma^2/n = \frac{p(1-p)}{n} = \frac{1}{4n}$, que l'on illustre par la courbe qui suit l'histogramme. En (b), on montre comment l'écart-type de $\bar{X}_n$ décroit en fonction de n. On affiche la courbe de $n^{-1/2}$ en pointillé à titre comparatif.

Dans la dernière expression, on a utilisé le fait que

$$\mathbb{E}\left[\bar{X}_n^2\right] = \frac{1}{n^2}\mathbb{E}\left[\sum_{k=1}^{n} X_k^2 + 2\sum_{k<l} X_k X_l\right] = \frac{1}{n^2}\left(n\mathbb{E}\left[X^2\right] + n(n-1)\mathbb{E}[X]^2\right)$$
$$= \frac{1}{n}\left(\mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 + n\mathbb{E}[X]^2\right) = \frac{\sigma^2}{n} + \mu^2\,,$$

avec $\mathbb{E}[X_k X_l] = \mathbb{E}[X]^2 = \mu^2$ pour $k \neq l$ en vertu de l'indépendance des échantillons. Sur la Fig. 1.2, on présente un exemple d'évaluation de $\bar{X}_n$ pour une variable de Bernoulli de paramètre $p = \frac{1}{2}$, c'est-à-dire une variable binaire valant 1 avec probabilité p et 0 avec probabilité $1 - p$. Dans cet exemple, l'estimateur Monte-Carlo converge bien vers une distribution normale de moyenne $\mu = p$ et de variance $\frac{\sigma^2}{n} = \frac{p(1-p)}{n}$, tel que prédit par le théorème central limite.

Dans la pratique, il existe plusieurs algorithmes pour échantillonner des variables aléatoires. Par exemple, il est possible d'échantillonner une variable de Bernoulli, si on dispose d'un générateur de nombres aléatoires uniformes. Spécifiquement, on génère d'abord $U \sim \mathcal{U}(0,1)$, $\mathcal{U}(a,b)$ est une distribution uniforme de support $[a,b]$, puis on échantillonne X selon la règle suivante :

$$X = \begin{cases} 1 & \text{si } U \leq p \\ 0 & \text{sinon} \end{cases}\,. \tag{1.26}$$

Plusieurs algorithmes d'échantillonnage se basent sur de tels générateurs, on réfère à Réf. [105, Chapitres 1 à 4] pour plus de détails.

Nous verrons plus loin comment échantillonner des graphes aléatoires (Chapitre 2), et des processus stochastiques (Chapitre 3). Des techniques d'échantillonnage plus avancées sont nécessaires dans les cas où l'espace d'échantillonnage est de grande dimension, ou lorsque la distribution est complexe. Nous allons en aborder quelques-unes, notamment les méthodes de Monte-Carlo par chaînes de Markov, aux Chapitres 2 et 3. Pour plus de détails à propos des méthodes d'échantillonnage, on réfère le lecteur à Réf. [105].

Chapitre 2

Structure : La théorie des graphes

Les réseaux complexes [19, 221] sont aujourd'hui au centre de la science qui traite des systèmes complexes. Née à la fin des années 90 [20, 315, 323], la science des réseaux a révolutionné plusieurs domaines de recherche dont les neurosciences [21], les sciences sociales [173] et l'épidémiologie des populations [231], pour en nommer que quelques-uns. Dans ce nouveau paradigme, on suppose qu'un système peut être simplifié et étudié en termes de ses composantes—à la manière du réductionnisme qui a connu un succès retentissant en physique au 20e siècle—, à condition qu'on conserve les interactions. Les interactions elles-mêmes occupent un rôle clé dans la modélisation, et sont ainsi représentées par la structure d'un graphe, celui-ci permettant de les décrire mathématiquement.

Dans ce chapitre, nous introduisons la théorie des graphes et les propriétés structurelles qui les décrivent. Ce chapitre se conclut par les graphes aléatoires, utilisés pour générer et modéliser les réseaux complexes.

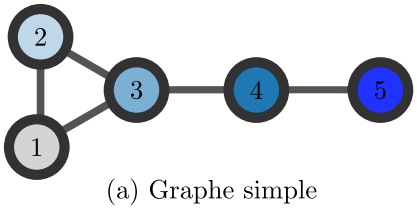
2.1 Le graphe

Le graphe est central dans la théorie des réseaux complexes. Il représente mathématiquement la structure d'un système, où les liens encodent les interactions entre les différentes composantes du systèmes. Commençons par le définir. [1]

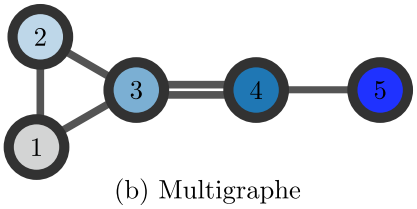
Définition 2.1 (Graphe)**.** *Un graphe $g = (\mathcal{V}, \mathcal{E})$ est un doublet, composé d'un ensemble de noeuds, dénoté $\mathcal{V}$, et d'une collection de liens $\mathcal{E}$.*

On note le nombre de noeuds par $N \equiv |\mathcal{V}|$ et le nombre de liens, par $E \equiv |\mathcal{E}|$. Il existe plusieurs types de graphes, lesquels sont utilisés dans des contextes différents.

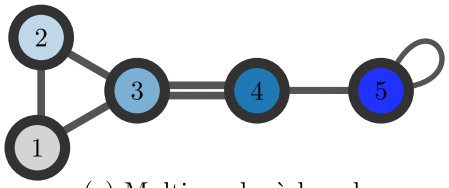
1. Les Définitions 2.1, 2.2 et 2.3 ne sont pas exhaustives. En effet, il existe encore plusieurs autres variantes des graphes—orientés, pondérés, signés, multiplexes, etc.—dont $\mathcal{V}$ et $\mathcal{E}$ vont être définis différemment (voir la Fig. 2.1(d)). Nous présentons notamment une analyse sur graphes pondérés et multiplexes au Chapitre 8. Cependant, cette définition sera suffisantes pour nos besoins. On réfère aux Réfs. [19, Chapitre 2] et [221, Chapitre 6] pour plus de détails à propos des autres variantes de graphes.



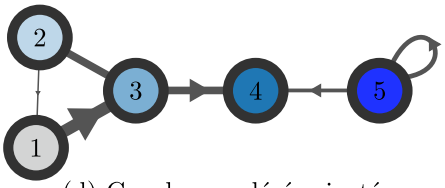
(a) Graphe simple



(b) Multigraphe



(c) Multigraphe à boucle



(d) Graphe pondéré orienté

FIGURE 2.1 – Exemples de graphes avec $\mathcal{V} = \{1,2,3,4,5\}$. En (a), on montre un graphe simple avec $\mathcal{E} = \{\{1,2\},\{1,3\},\{2,3\},\{3,4\},\{4,5\}\}$. En (b), on ajoute un lien supplémentaire entre les noeuds 3 et 4 au graphe, qui devient un multigraphe. En (c), on ajoute un boucle au noeud 5; on dit alors de g qu'il est un multigraphe à boucle. Finalement, en (d), on montre un graphe pondéré contenant une boucle et dont les liens sont orientés

Définition 2.2 (Graphe simple). *Pour un graphe simple g, $\mathcal{E}$ est un ensemble de la forme*

$$\mathcal{E} = \{\{i,j\} : i,j \in \mathcal{V} \wedge i \neq j\}\ , \tag{2.1}$$

où $\{i,j\}$ représente le lien non orienté entre les noeuds i et j.

Le graphe simple est utilisé comme une structure de réseau standard, c'est-à-dire sans spécificité. Comme on ne permet pas les multi-liens ou les boucles, on peut compter le nombre maximal de liens possibles :

$$E \leq \frac{N(N-1)}{2}\,. \tag{2.2}$$

Lorsqu'on permet qu'un lien apparaîsse plusieurs fois dans $\mathcal{E}$, on parle alors de multigraphes.

Définition 2.3 (Multigraphe). *Pour un multigraphe g, $\mathcal{E}$ est un multi-ensemble de la forme*

$$\mathcal{E} = \{\!\{\{i,j\} : i,j \in \mathcal{V} \wedge i \neq j\}\!\} \tag{2.3}$$

où $\{\!\{...\}\!\}$ représente un multi-ensemble, c'est-à-dire un ensemble où les éléments peuvent apparaître plusieurs fois.

Contrairement au graphe simple, la taille de $\mathcal{E}$ n'est pas bornée pour le multigraphe. Finalement, dans certains cas, on souhaitera inclure des *auto-connexions*, ou des *boucles*, auquel

cas $\mathcal{E}$ contient des multi-ensembles de la forme $\{\!\{i,j\}\!\}$ (admettant par exemple $\{\!\{i,i\}\!\}$). Sur la Fig. 2.1, on montre des exemples de graphes simple, mutligraphe et avec boucle.

La matrice d'adjacence, notée $\mathbf{a}$, est une représentation alternative commune pour les graphes, où l'élément $[\mathbf{a}]_{ij} = a_{ij} \geq 0$ indique le nombre de connexions entre les noeuds i et j. Par exemple, pour les graphes de la Fig. 2.1, les matrices d'adjacence sont, dans le même ordre que la figure (excluant le graphe pondéré en (d)), données par

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 1 & 2 \end{pmatrix}.$$

On note que $a_{34} = a_{43} = 2$ dans la deuxième matrice, et que $a_{33} = 2$ dans la troisième. Notons également que, par convention, toutes les entrées sur la diagonale sont des multiples de 2.

La liste d'adjacence $L = (\mathcal{N}_1, \cdots, \mathcal{N}_N)$ est une autre représentation alternative utile qui fait intervenir le voisinage des noeuds directement, où $\mathcal{N}_i = \{j : \{i,j\} \in \mathcal{E}\}$ est le voisinage du noeud i.

2.2 Propriétés structurelles

Les graphes peuvent être des objets complexes, s'ils sont assez volumineux ($N \gg 1$). C'est pourquoi on les regroupe souvent en termes de leurs propriétés structurelles, afin de les distiller ou de les comparer entre eux. En bref, une propriété structurelle est une fonction $f : \mathcal{G} \to \mathbb{R}$ qui prend un graphe $g \in \mathcal{G}$ et retourne un nombre réel. On retrouve quelques exemples de propriétés structurelles courantes au Tab. 2.1.

Les propriétés structurelles décrivent différents aspects d'un graphe. Par exemple, la densité ρ décrit la fraction des liens qui sont effectivement présents dans un graphe simple. Si $\rho = \mathcal{O}(1)$, une fraction significative des liens existent, autrement, par exemple si $\rho = \mathcal{O}\left(N^{-\delta}\right)$ pour $\delta \geq 1$, la plupart sont absents. Celle-ci permet de classifier les graphes en deux catégories dans la limite des tailles inifinies.

Définition 2.4 (Graphes parcimonieux et denses)**.** *Soit une séquence $(g_1, g_2, \cdots, g_N)$ de N graphes, où g_k est un graphe de k noeuds dont le nombre de liens $f(k)$ est une fonction du nombre de*

2. Cette formule est valide pour les graphes simples et les multigraphes. Par contre, dans le cas des multigraphes, le densité peut être plus grande que 1. Pour les graphes (multigraphes) avec boucles, le facteur de normalisation devient $\frac{N(N+1)}{2}$ afin de compter toutes les N boucles possibles.

3. Pour calculer la modularité Q, on doit inclure une information supplémentaire donnée par le vecteur $\boldsymbol{\sigma} \in [q]^N$, où on interprète l'élément σ_i comme le groupe auquel le noeud i appartient. Notons également que $\delta(x,y)$ est le delta de Kronecker.

Propriétés	Notation
Nombre de liens	$E = \sum_{i \leq j} a_{ij}$
Densité [2]	$\rho = \frac{2E}{N(N-1)}$
Degré du noeud i	$k_i = \sum_{j=1}^{N} a_{ij}$
Distribution des degrés	$p(k) = \frac{N_k}{N}$, où $N_k = \sum_{i=1}^{N} \delta(k_i, k)$
n-ième moments de $p(k)$	$\langle k^n \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i^n = \sum_{k=0}^{\infty} k^n p(k)$
Coefficient d'assortativité des degrés	$r = \frac{1}{E} \sum_{i \leq j}^{N} \left(a_{ij} - \frac{k_i k_j}{2E} \right) k_i k_j$
Modularité [3]	$Q = \frac{1}{E} \sum_{i \leq j}^{N} \left(a_{ij} - \frac{k_i k_j}{2E} \right) \delta(\sigma_i, \sigma_j)$

TABLEAU 2.1 – Quelques propriétés structurelles pour un graphe g à N noeuds, dont la matrice d'adjacence est **a**. Notons que chaque propriété est une fonction de g, même si cette dépendence est sous-entendu dans le tableau—par exemple on devrait lire $E(g)$ pour le nombre de liens.

noeuds. On dit que cette séquence de graphes est asymptotiquement ***parcimonieuse*** *(*sparse*) si*

$$\lim_{N \to \infty} \frac{f(N)}{N} < \infty \,. \tag{2.4}$$

À l'inverse, on dit qu'elle est ***dense*** *si*

$$\lim_{N \to \infty} \frac{f(N)}{N} = \infty \,. \tag{2.5}$$

On définit la limite parcimonieuse d'un graphe aléatoire comme la limite dans laquelle la condition Eq. (2.4) est respectée en moyenne.

Évidemment, les limites parcimonieuse et dense ne sont pas bien définies pour les graphes réels, étant donné que nous n'avons généralement pas accès à une séquence de graphes infinie. Pour les graphes réels, on doit donc assouplir le critère : un graphe à N noeuds et E liens est parcimonieux si son degré moyen $\langle k \rangle = \frac{2E}{N}$ est petit ; autrement, et si $\langle k \rangle$ est du même ordre de grandeur que N, il est dense. Dans la pratique, on mesure une majorité de réseaux réels parcimonieux [221]. La Fig. 2.2(a) montre des évidences numériques de ceci pour un grand nombre de réseaux réels. La relation entre le nombre de noeuds N et le nombre de liens E est montrée comme étant en majorité linéaire, ce qui est un indicateur de parcimonie.

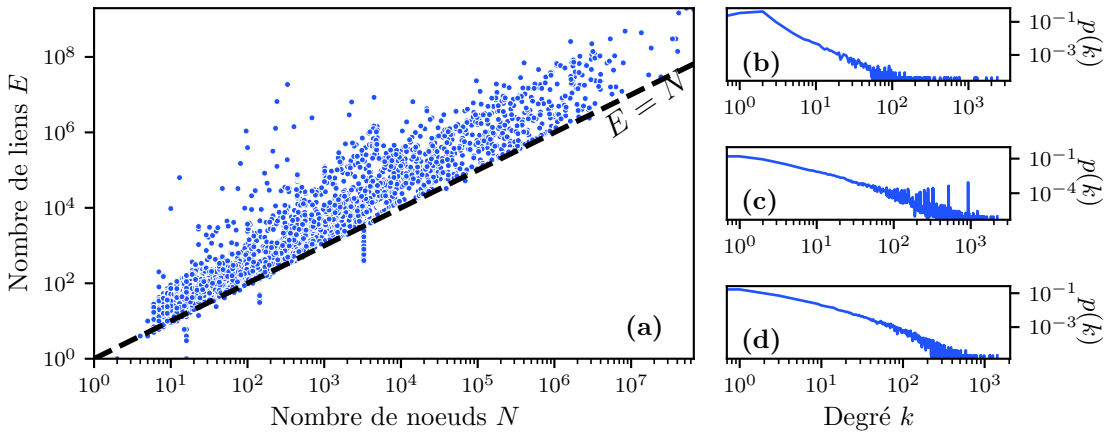


FIGURE 2.2 – Propriétés structurelles des réseaux réels. En (a), on montre la relation entre le nombre de noeuds N et le nombre de liens E pour 6,440 réseaux réels, lesquels sont récoltés de la plateforme *Netzschleuder* [244]. La ligne pointillée noire $E = N$ est tracée à titre indicatif. En (b–d), on montre la distribution des degrés $p(k)$ pour trois réseaux réels *scale-free* : (b) les systèmes autonomes de l'internet [149], (c) le réseau de collaborateurs appartenant au *Microsoft Adademic Graph* (MAG) [28] et (d) le réseau des publications sur facebook (circa 2009) [318].

Une autre propriété structurelle d'intérêt est la distribution des degrés, notée $p(k)$. Celle-ci compte la fraction des noeuds ayant un degré k, c'est-à-dire ayant exactement k voisins. Le degré en soit est un indication de la centralité des noeuds dans le graphe, et permet de les classifier selon leur importance. Ce faisant, $p(k)$ mesure l'hétérogénéité du graphe dans ses classes de degrés.

On observe une forte hétérogénéité chez un grand nombre de réseaux réels [19, 221]. Cette hétérogénéité se manifeste sous la forme d'une distribution $p(k)$ dont la queue décroit asymptotiquement comme une loi de puissance :

$$p(k) \sim k^{-\gamma}\,, \tag{2.6}$$

de sorte que γ, l'exposant caractéristique, prend typiquement des valeurs entre 2 et 3. On en montre trois exemples de réseaux à la Fig. 2.2(b–d). On dit de ces réseaux qu'ils sont *scale-free* puisqu'ils ne possèdent pas d'échelle caractéristique dans les classes de degrés, étant donné que $\langle k^2 \rangle$, le deuxième moment de $p(k)$, diverge pour $\gamma \in [2, 3]$.

2.3 Graphes aléatoires

Les graphes aléatoires sont les modèles statistiques de prédilection pour représenter les réseaux complexes. Formellement, un graphe aléatoire G est une variable aléatoire (définition 1.2) de distribution $P(G)$ et de support $\mathcal{G}$. Les graphes aléatoires trouvent un grand nombre d'applications en science des réseaux. Ils sont notamment au centre des méthodes

bayésiennes impliquant des modèles de graphes [240, 242, 339, 342]. On les utilise pour générer des graphes synthétiques dans le cadre d'analyses statistiques [95, 218, 219, 238], ou afin d'expliquer l'existence de certaines propriétés structurelles dans les réseaux réels comme leur hétérogénéité [3] ou leur auto-similarité [275].

2.4 Modèle Erdős-Rényi

La famille de modèles la plus souvent utilisée est celle des modèles qui maximisent l'*entropie*. Nous revisiterons le concept d'entropie plus particulièrement dans le Chapitre 4, mais pour l'instant notons que l'entropie est une mesure d'incertitude qui dépend de la distribution de probabilité $P(G)$. Un graphe aléatoire qui maximise l'entropie est donc une variable aléatoire dont l'issue est la plus incertaine possible, sous les contraintes désirées. Par exemple, maximiser l'entropie sous la contrainte que le nombre de liens E est fixe mène à un graphe aléatoire où $P(G)$ est une distribution uniforme :

$$P(G = g) = \begin{cases} |\mathcal{G}_{N,E}|^{-1} & \text{si } g \in \mathcal{G}_{N,E} \\ 0 & \text{autrement} \end{cases}, \tag{2.7}$$

où $\mathcal{G}_{N,E}$ est l'ensemble des graphes ayant N noeuds et E liens. Si $\mathcal{G}_{N,E}$ est un ensemble de graphes simples, alors

$$|\mathcal{G}_{N,E}| = \left(\begin{pmatrix} N \\ 2 \end{pmatrix} \right), \tag{2.8}$$

où $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ est le coefficient binomial, avec $n! = n \times (n-1) \times \cdots \times 1$ étant la factorielle de n. Ce modèle est mieux connu sous le nom de modèle Erdős-Rényi (ER), ayant été étudié en détails par Paul Erdős et Alfréd Rényi dans les années 60 [86] (on réfère à [40, Chapitre 7] pour plus de détails). Le modèle ER est un modèle simple qui permet de reproduire la densité des graphes réels, par le contrôle qu'il donne sur le nombre de liens, mais qui échoue généralement à reproduire d'autres propriétés.

2.5 Modèle des configurations

Pour aller plus loin, on doit complexifier le modèle en imposant des contraintes additionnelles. On pourrait par exemple introduire une contrainte sur le degré de chaque noeud, de sorte que la séquence des degrés complète, dénotée $\boldsymbol{k}$, soit fixe. Dans ce cas, sachant que $\mathcal{G}_{\boldsymbol{k}}$ est l'ensemble des graphes dont la séquence des degrés est $\boldsymbol{k}$, la distribution $P(G)$ qui maximise l'entropie est

$$P(G = g) = \begin{cases} |\mathcal{G}_{\boldsymbol{k}}|^{-1} & \text{si } g \in \mathcal{G}_{\boldsymbol{k}} \\ 0 & \text{autrement} \end{cases}. \tag{2.9}$$

Comme le modèle Erdős-Rényi, ce modèle fixe le nombre de liens par l'intermédiaire de la séquence des degrés. Cependant, il n'est pas aussi commode que le modèle Erdős-Rényi, puisque l'ensemble $\mathcal{G}_{\boldsymbol{k}}$ est difficile à manipuler. Notamment, calculer $|\mathcal{G}_{\boldsymbol{k}}|$ est un problème combinatoire difficile—bien connu sous le nom d'énumeration de graphes—qui ne connaît pas de solutions exactes pour grand N. Calculer la probabilité d'un graphe est par conséquent aussi problématique, et nous empêche notamment de générer efficacement des réalisations de l'ensemble.

Pour contourner ce problème, on fait intervenir le concept de configuration, dans laquelle les demi-liens, chacun attaché à un seul noeud, sont distinguables. Le nombre de demi-liens attachés à un noeud deviendra éventuellement son degré. Comptant un total de $2E$ demi-liens (et donc E liens), on génère alors une configuration aléatoire simplement en connectant aléatoirement des demi-liens deux à deux, jusqu'à ce qu'ils soient tous connectés. Le nombre d'appariements possibles des $2E$ demi-liens est donné par

$$\frac{(2E)!}{2^E E!}\,, \tag{2.10}$$

la probabilité uniforme d'obtenir une configuration est donc l'inverse du comptage, c'est-à-dire $\frac{2^E E!}{(2E)!}$. Ces appariements incluent les boucles et la répétition des liens ; ceci implique que notre ensemble sera ultimement composé de multigraphes. Une configuration génère un graphe unique, mais plusieurs configurations peuvent générer un même graphe. Ainsi, chaque multigraphe g est associé à un ensemble de configurations $C(g)$ qui peuvent le générer, et celui-ci compte

$$|C(g)| = \frac{\prod_{i=1}^{N} k_i}{\prod_{i<j} a_{ij}!\, \prod_{i=1}^{N} a_{ii}!!} \tag{2.11}$$

configurations, où $a_{ii}!! = a_{ii} \times (a_{ii}-2) \times \cdots \times 2$ est la double factorielle de a_{ii}, en supposant que a_{ii} est pair. On obtient alors la probabilité d'un multigraphe g en sommant sur toutes les probabilités des configurations qui le génèrent :

$$P(G=g) = \sum_{c \in C(g)} \frac{2^E E!}{(2E)!}\,, \tag{2.12}$$

$$= \frac{2^E E!}{(2E)!} \frac{\prod_{i=1}^{N} k_i}{\prod_{i<j} a_{ij}!\, \prod_{i=1}^{N} a_{ii}!!}\,. \tag{2.13}$$

Ce modèle est connu sous le nom de *modèle des configurations* (CM, pour *configuration model* en anglais). Le modèle des configurations ne maximise pas l'entropie, puisque sa distribution n'est pas uniforme. Cependant, on peut montrer qu'en moyenne le nombre de boucles et de multi-liens est asymptotiquement constant par rapport à N [223]. Dans la limite des grandes taille $N \to \infty$, la probabilité du modèle des configurations est approximativement uniforme, et il est souvent utilisé comme substitut au modèle qui maximise l'entropie pour générer des graphes aléatoires dont la séquence des degrés est fixe.

2.6 Techniques Monte Carlo d'échantillonnage de graphes

Pour échantillonner des réalisations du modèle Erdős-Rényi, on doit pouvoir échantillonner uniformément l'ensemble des liens possibles sans remplacement. Plusieurs techniques numériques existent pour y parvenir, mais la plus simple implique d'énumérer tous les liens possibles dans une liste, de générer une permutation aléatoire de cette liste, et de prendre les E premiers éléments. Dans le cas du modèle des configurations, on peut simplement énumérer tous les demi-liens et les apparier deux à deux aléatoirement, tel que discuté précédemment. Ces deux techniques d'échantillonnage sont simples et efficaces, mais ne permettent pas d'échantillonner tous les types de graphes aléatoires. Pour y parvenir, on emploie des techniques d'échantillonnage Monte Carlo par chaînes de Markov (MCMC).

Formellement, l'objectif de l'algorithme MCMC est d'échantillonner un graphe selon une distribution arbitraire $P(G)$ en appliquant des transitions successives d'un graphe initial G vers un autre G'. On doit donc proposer des transitions de G vers G', et les accepter avec une probabilité d'acceptation adéquate. La forme la plus standard est celle de Métropolis-Hastings [129], donnée par

$$\min\left(1, \frac{P(G')P(G' \to G)}{P(G)P(G \to G')}\right). \tag{2.14}$$

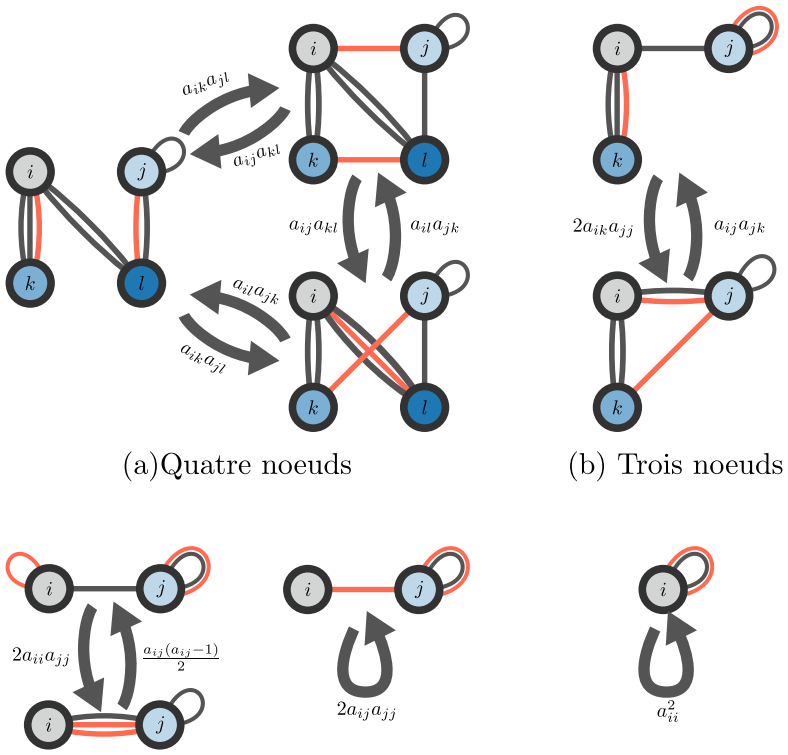
où $P(G \to G')$ est la probabilité de proposer G' ; $P(G' \to G)$ étant la probabilité de la proposition inverse. En effectuant des transitions de cette nature, on génère une chaîne de Markov, dont les états sont des graphes. On assure également que la chaîne de Markov est réversible et que sa distribution à l'équilibre est bien $P(G)$. Nous reverrons les chaînes de Markov et différents concepts qui s'y rattachent au Chapitre 3 (voir § 3.1 et § 3.4).

Notons que la probabilité d'acceptation à l'Eq. (2.14) évalue le ratio entre les probabilités respectives de G et G'. Ce faisant, l'algorithme MCMC permet d'échantillonner une distribution $P(G)$ même si on ne connait pas le facteur de normalisation de $P(G)$. Par exemple, on peut échantillonner le modèle uniforme contraint par la séquence des degrés (celui discuté précédemment) étant donné que

$$P(G = g) \propto 1\,, \quad \forall g \in \mathcal{G}_{\boldsymbol{k}}\,. \tag{2.15}$$

Il s'agit simplement d'avoir un modèle de propositions $P(G \to G')$ adapté. Dans ce cas-ci, un choix de proposition consiste à échanger deux liens aléatoirement (*double edge swap*). Cette procédure est illustrée à la Fig. 2.3. Certaines considérations sont de mises pour calculer la probabilité d'acceptation, lesquelles sont discutées à la Réf. [95].

Étant très général, l'algorithme MCMC permet d'échantillonner une grande variété de graphes aléatoires, dont plusieurs modèles de nature similaire au modèle des configurations (voir Réf. [95]). L'algorithme MCMC permet également d'effectuer des tâches d'inférence où la distribution ciblée est en réalité une distribution *a posteriori* $P(G|X)$, conditionnée sur des observations indirectes du graphe X. Nous y reviendrons en détail au Chapitre 4.



(a)Quatre noeuds (b) Trois noeuds

(c) Deux noeuds (d) Un noeud

FIGURE 2.3 – Illustration de tous les types de propositions du *double edge swap* sur des multigraphes : (a) à quatre noeuds, (b) à trois noeuds et (c) à deux noeuds et (d) à un noeud. Les liens colorés en rouge participent à l'échange. On indique la transition à l'aide d'une flèche d'une configuration à une autre avec la probabilité de proposition correspondante (à un facteur de normalisation $\binom{M}{2}$ près). Cette illustration est inspirée de [95, Figure 5].

Chapitre 3

Dynamique : La théorie des processus stochastiques

L'aspect dynamique des systèmes complexes remonte aussi loin que l'origine de la théorie des systèmes complexes elle-même, un concept au centre de l'analyse qu'en fait Herbert A. Simon dans son célèbre essai de 1962 [283]. À l'époque, Simon avait émis l'hypothèse que les systèmes biologiques et sociaux pourraient éventuellement être décrits par des modèles mathématiques à la manière des systèmes physiques pour décrire leur évolution et comprendre leur fonctionnement. Puis vint la théorie des systèmes dynamiques dans les années 1960 et 1970, avec Lorenz [186] et May [195], qui montrèrent l'émergence de comportements complexes dans des systèmes simples ; et vers la théorie des systèmes non linéaires dans les années 1980 et 1990 qui en était la suite logique [302]. C'est au début des années 2000 que l'approche réseau sera intégrée dans les systèmes dynamiques pour décrire les *processus sur graphe*, comme la propagation d'épidémies [232] et la synchronisation des oscillateurs couplés [301].

Dans ce chapitre, nous présentons les concepts importants liés au processus stochastiques qui font l'objet de cette thèse. Notamment, nous abordons les chaînes de Markov et leurs propriétés, ainsi que les modèles graphiques. Nous présentons plusieurs exemples de processus évoluant sur des graphes dont l'état des noeuds est binaire, et discuterons de l'émergence de la criticalité dans ces processus.

3.1 Chaînes de Markov

Une chaîne de Markov est un processus stochastique qu'on dit *sans mémoire*, c'est-à-dire que l'état du système dans le futur ne dépend pas de son historique, mais seulement de son état présent. Bien qu'elle représente le cas le plus simple, la chaîne de Markov permet de modéliser un grand éventail de processus, notamment ceux qui évoluent sur des graphes.

Formellement, on définit une chaîne de Markov comme suit :

Définition 3.1 (Chaîne de Markov à temps discret)**.** *Une chaîne de Markov à temps discret, notée $X = (X_1, \cdots, X_\tau)$, est une séquence de longueur τ de variables aléatoires discrètes X_t de même support $\mathcal{X}$, où X_t est l'état de X au temps t, pour tout $t \in [\tau]$. La probabilité conjointe de X, qui s'écrit*

$$P(X) = \prod_{t=1}^{\tau-1} P(X_{t+1}|X_t, \cdots, X_1)\,, \tag{3.1}$$

satisfait la propriété de Markov :

$$P(X_{t+1}|X_t, \cdots, X_1) = P(X_{t+1}|X_t)\,. \tag{3.2}$$

On appelle $P(X_{t+1}|X_t)$ la probabilité de transition de l'état X_t à l'état X_{t+1}.

Lorsque les états de X ont un support $\mathcal{X}$ fini—comme ce sera le cas ci-après—, ceci permet de représenter les probabilités de transition sous forme matricielle, où

$$\Gamma_{\mu\nu}(t) = P(X_{t+1} = x_\nu | X_t = x_\mu) \tag{3.3}$$

est l'élément (μ, ν) de la matrice de transition dépendante du temps $\mathbf{\Gamma}(t)$. Cette matrice, appelée la matrice de Markov, est de taille $|\mathcal{X}| \times |\mathcal{X}|$ et est *stochastique*—c'est-à-dire ses éléments sont non negatifs et la somme de chaque ligne est égale à 1.

On dit d'une chaîne de Markov qu'elle est homogène dans le temps si ses probabilités de transition sont indépendantes du temps, auquel cas

$$\mathbf{\Gamma}(t) = \mathbf{\Gamma}(t+h)\,, \tag{3.4}$$

pour tout t et h. Ci-après, on utilisera aussi $\mathbf{\Gamma}$ pour désigner la matrice de transition homogène.

Les chaînes de Markov peuvent être décrites comme des marcheurs aléatoires se déplaçant sur des graphes orientés et pondérés, où les noeuds sont les états possibles de la chaîne. Dans ce cas, la matrice d'adjacence du graphe est simplement donnée par la matrice de Markov $\mathbf{\Gamma}$, où chaque élément $\Gamma_{\mu\nu}$ correspond au poids du lien allant du noeud x_μ au noeud x_ν. Cette description est utile pour comprendre certaines propriétés des chaînes de Markov et pour simuler leur évolution numériquement.

3.2 Évolution temporelle

Les chaînes de Markov représentent des processus dont l'état évolue dans le temps. Ainsi, une propriété d'intérêt est la probabilité d'occupation des états du système à un temps donné t. Pour un état x_μ, on la définit comme suit :

$$\pi_\mu(t) \equiv P(X_t = x_\mu)\,. \tag{3.5}$$

L'évolution de cette probabilité est dictée par l'équation maîtresse, qui s'écrit

$$\pi_\mu(t+1) = \sum_\nu \pi_\nu(t)\Gamma_{\nu\mu}(t)\,. \tag{3.6}$$

Une autre manière d'écrire l'équation maîtresse permet d'identifier les termes entrants et sortants de chaque état. En effet, sachant que de l'Éq. (3.3) on a $1 - \sum_{\nu\neq\mu}\Gamma_{\mu\nu}(t) = \Gamma_{\mu\mu}(t)$, on peut réécrire l'équation maîtresse comme suit :

$$\pi_\mu(t+1) = \pi_\mu(t) + \sum_{\nu\neq\mu}\left[\pi_\nu(t)\Gamma_{\nu\mu}(t) - \pi_\mu(t)\Gamma_{\mu\nu}(t)\right]\,, \tag{3.7}$$

où le premier terme $\pi_\nu(t)\Gamma_{\nu\mu}(t)$ représente le flux entrant dans l'état x_μ et le dernier terme $\pi_\mu(t)\Gamma_{\mu\nu}(t)$, le flux sortant. Considérons la représentation matricielle de l'équation maîtresse dans le cas homogène :

$$\boldsymbol{\pi}(t+1) = \boldsymbol{\pi}(t)\boldsymbol{\Gamma}\,, \tag{3.8}$$

où $\boldsymbol{\pi}(t) = \big(\pi_\mu(t)\big)_\mu$ est le vecteur des probabilités d'occupation de chaque état au temps t—on réfère également à $\boldsymbol{\pi}(t)$ comme l'état du système. Sachant que le système est initialement dans l'état $\boldsymbol{\pi}(0) = \boldsymbol{\pi}_0$, on obtient son état au temps t en appliquant successivement l'Éq. (3.8) :

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}_0\boldsymbol{\Gamma}^t\,. \tag{3.9}$$

Sous certaines conditions [1], la chaîne converge vers un état stationnaire unique $\boldsymbol{\pi}^*$, qui respecte la condition suivante :

$$\boldsymbol{\pi}^* = \boldsymbol{\pi}^*\boldsymbol{\Gamma}\,. \tag{3.10}$$

La distribution stationnaire $\boldsymbol{\pi}^*$ est un vecteur propre de la matrice de Markov $\boldsymbol{\Gamma}$ associé à la valeur propre 1—sa valeur propre dominante [2].

3.3 Processus à temps continu

Bien que nous nous intéressons principalement aux cas des temps discrets, nous survolons brièvement le cas des temps continus afin de compléter la discussion. On dit d'une chaîne de Markov qu'elle est à temps continu si les intervalles de temps entre deux états successifs sont des nombres réels et aléatoires. On note une telle chaîne X comme une fonction de t, où $t \in [0,\infty)$ est le temps de sorte que $X(t)$ indique l'état du système au temps t. Contrairement au cas discret, les probabilités de transition n'interviennent pas directement dans l'évolution du processus. C'est plutôt les taux de transition, notés $Q_{\mu\nu}(t)$, qui déterminent l'évolution

1. Pour que l'Éq. (3.10) ait une solution unique, la matrice $\boldsymbol{\Gamma}$ doit être irréductible—composée d'une seule composante fortement connectée—et récurrente positive—le temps moyen de récurrence de chaque état est fini [180].

2. À titre informatif, le théorème de Gerschgorin est utilisé pour démontrer que le rayon spectral des matrices stochasiques est 1. De ce fait suit la dominance de la valeur propre 1—qui existe si l'Éq.(3.10) est vraie—pour toutes matrices de Markov.

des probabilités de transition et, ultimement, de l'état complet du système. Lorsque le temps est une variable continue, on doit définir les probabilités de transition $\Gamma_{\mu\nu}(t, t+\Delta)$ en tenant compte du temps t à partir duquel le système était dans l'état x_μ, et du temps $t+\Delta$ après lequel il a sauté à l'état x_ν.

La distribution des intervalles de temps pour les chaînes de Markov est exponentielle, puisqu'ils sont dits *sans mémoire*. Par définition, tout processus sans mémoire respecte la condition suivante [3] :

$$P(T > t+\Delta | T > t) = P(T > \Delta)\,, \tag{3.11}$$

où T représente une variable aléatoire des temps de saut. Cette condition assure, d'une part, que le temps entre deux changements d'états successifs—appelés sauts [4]—suit également une distribution exponentielle, et d'autre part, que la probabilité de sauter à un état donné ne dépend pas du temps écoulé depuis le dernier saut, mais seulement des taux de transitions. Ainsi, la fonction de répartition cumuléedes intervalles de temps entre deux sauts s'écrit comme suit :

$$P\Big(T > t+\Delta \Big| X(t) = x_\mu\Big) = e^{-Q_\mu(t)\Delta}\,, \tag{3.12}$$

où $Q_\mu(t) = \sum_{\nu\neq\mu} Q_{\mu\nu}(t)$ est le taux de saut. Qui plus est, la probabilité de sauter de x_μ à x_ν au temps t est simplement donnée par $\frac{Q_{\mu\nu}(t)}{Q_\mu(t)}$. Ainsi, on obtient la probabilité de transition de l'état x_μ au temps t à l'état x_ν après un intervalle de temps Δ en multipliant ces deux dernières probabilités ensemble :

$$\begin{aligned}\Gamma_{\mu\nu}(t, t+\Delta) &= P\Big(T \leq t+\Delta \Big| X(t) = x_\mu\Big)\frac{Q_{\mu\nu}(t)}{Q_\mu(t)}\,,\\ &= e^{-Q_\mu(t)\Delta}\frac{Q_{\mu\nu}(t)}{Q_\mu(t)}\,.\end{aligned} \tag{3.13}$$

pour $\mu \neq \nu$. Finalement, puisque la probabilité d'effectuer plus d'un saut dans un intervalle de temps Δ constitue un terme donné par $o(\Delta)$, on utilise la normalisation pour obtenir la probabilité de rester dans le même état. Ensemble, on obtient la probabilité de transition à temps continu dans le cas général :

$$\Gamma_{\mu\nu}(t, t+\Delta) = \delta(\mu,\nu) + Q_{\mu\nu}(t)\Delta + o(\Delta)\,, \tag{3.14}$$

3. On montre que seule la distribution exponentielle respecte la condition sans mémoire. Pour une distribution exponentielle $f(x) = \lambda e^{-\lambda x}$, la fonction de répartition cumulée s'écrit $F(x) = P(X \leq x) = 1 - e^{-\lambda x}$. Alors,

$$P(T > t+\Delta | T > t) = \frac{P(T > t+\Delta, T > t)}{P(T > t)} = e^{-\lambda\Delta} = P(T > \Delta)\,.$$

De plus, la condition sans mémoire peut être formulée en termes de l'équation fonctionnelle exponentielle de Cauchy, c'est-à-dire

$$P(T > t+\Delta) = P(T > t)P(T > \Delta)\,,$$

qui est satisfaite seulement par la distribution exponentielle $P(T > t) = e^{-\lambda t}$ lorsque le domaine de la fonction est les nombres réels non négatifs.

4. La description en termes de sauts des chaînes de Markov à temps continu est fort utile pour simplifier leur analyse. On réfère à la Réf. [146, Chapitre 17] pour plus de détails.

où on rappelle que $\delta(\mu, \nu)$ est le symbole de Kronecker, et on a considéré que Δ est petit.

Les processus à temps continu sont fascinants et méritent une présentation plus détaillée, qui dépasse le cadre de cette thèse. Nous renvoyons le lecteur à la Réf. [146, Chapitres 13 à 18] pour une introduction plus complète.

3.4 Réversibilité

La réversibilité est un principe qui intervient souvent en physique, où de manière générale un système est dit réversible s'il reste invariant suite à l'inversion du temps. Pour une chaîne de Markov, la réversibilité se définit en fonction de ses probabilités de transition.

Définition 3.2 (Réversibilité)**.** *Une chaîne de Markov X est réversible si sa chaîne inversée X', où $X'_t = X_{\tau - t}$, possède la même matrice de transition que X. Si une chaîne de Markov est réversible, alors elle respecte la condition du bilan détaillé :*

$$\pi_\mu \Gamma_{\mu\nu} = \pi_\nu \Gamma_{\nu\mu} \,, \tag{3.15}$$

pour tout états $x_\nu, x_\mu \in \mathcal{X}$.

On démontre aisément que la réversibilité implique la condition du bilan détaillé en montrant que les probabilités de transition de X', notées $\Gamma'_{\nu\mu}$, sont données en termes de la distribution stationnaire $\boldsymbol{\pi}$ et de la matrice de transition $\boldsymbol{\Gamma}$ de X :

$$\Gamma'_{\nu\mu} = \frac{\pi_\mu \Gamma_{\mu\nu}}{\pi_\nu} \,. \tag{3.16}$$

Ainsi, si $\Gamma'_{\nu\mu} = \Gamma_{\nu\mu}$, alors la condition du bilan détaillé est remplie et la chaîne est réversible. Notons également que la réversibilité peut être définie de manière équivalente avec le critère de Kolmogorov, qui stipule que pour tout n, μ et ν, on a

$$\Gamma_{\mu\mu_1}\Gamma_{\mu_1\mu_2}\cdots\Gamma_{\mu_n\nu} = \Gamma_{\nu\mu_n}\cdots\Gamma_{\mu_2\mu_1}T_{\mu_1\mu} \tag{3.17}$$

pour toute séquence $(\mu, \mu_1, \cdots, \mu_n, \nu)$. L'équivalence entre la réversibilité et le critère de Kolmogorov est biconditionnelle (*si et seulement si*) ; on réfère à la Réf. [151, pp. 21-25] pour la preuve complète.

Une application de premier ordre qui utilise activement la réversibilité est l'algorithme de Metropolis-Hastings [129, 199], que nous avons brièvement introduit dans la Section 2.6 et que nous utiliserons maintenant comme un exemple. Appartenant à la famille des méthodes MCMC, l'algorithme de Métropolis-Hastings permet d'échantillonner des distributions de probabilité compliquées—comme celle de plusieurs modèles de graphes—en passant par une chaîne de Markov dont la distribution stationnaire est égale à celle ciblée. Pour ce faire,

on construit une chaîne de Markov irréductible et apériodique[5] en factorisant les entrées de sa matrice de transition en une probabilité de proposition $q_{\mu\nu}$ de x_μ vers x_ν, et d'une probabilité d'acceptation $\alpha_{\mu\nu}$:

$$\Gamma_{\mu\nu} = q_{\mu\nu}\alpha_{\mu\nu} \,. \tag{3.18}$$

La probabilité d'acceptation proposée par Hastings à la Réf. [129] a la forme générale suivante :

$$\alpha_{\mu\nu} = \frac{s_{\mu\nu}}{1 + \frac{\pi_\nu q_{\nu\mu}}{\pi_\mu q_{\mu\nu}}} \,, \tag{3.19}$$

où $s_{\mu\nu}$ est une fonction symmétrique de la forme

$$s_{\mu\nu} = h\big(\Delta_{\mu\nu}\big) \,, \quad \Delta_{\mu\nu} = \min\left(\frac{\pi_\nu q_{\nu\mu}}{\pi_\mu q_{\mu\nu}}, \frac{\pi_\mu q_{\mu\nu}}{\pi_\nu q_{\nu\mu}}\right) \tag{3.20}$$

où $0 \leq h(x) \leq 1 + x$. Cette forme garantit que la chaîne est réversible et que sa distribution stationnaire est bien $\boldsymbol{\pi}$, à condition que la proposition $q_{\mu\nu}$ laisse la chaîne irréductible. Dans le cas de l'algorithme de Metropolis-Hastings, on fixe $h(x) = 1 + x$ qui revient à la probabilité d'acceptation de Metropolis-Hastings :

$$\alpha_{\mu\nu} = \min\left(1, \frac{\pi_\nu q_{\nu\mu}}{\pi_\mu q_{\mu\nu}}\right) . \tag{3.21}$$

Nous avons déjà abordé cette équation dans la Section 2.6 (Éq. (2.14)) dans le cas du modèle des configurations. Évidemment, l'algorithme de Metropolis-Hastings est tout à fait général et nous l'utiliserons au Chapitres 4, 5 et 6 également pour échantillonner des processus sur graphe et pour reconstruire des graphes à partir de données.

3.5 Dynamiques binaires sur graphe

Jusqu'à présent, nous avons discuté de processus qui, a priori, n'évoluent pas sur des graphes. Cette particularité supplémentaire apporte deux nouveautés. D'une, on associe maintenant chaque noeud à sa propre chaîne de Markov, noté $X_i = (X_{i,1}, \ldots, X_{i,\tau})$, qui décrit son état. Le processus complet est donc composé de plusieurs chaînes de Markov en interaction. De deux, la structure du graphe dicte la manière dont les noeuds interagissent entre eux. D'un point de vue mathématique, les probabilités de transition de ces processus sont sujetent à la propriété de Markov *locale*, où les noeuds déconnectés sont conditionnellement indépendants. Dans le vocabulaire des modèles graphiques, ces types de processus sont appelés réseaux bayésiens [142]—des modèles probabilistes dont l'indépendance conditionnelle des variables aléatoires est encodée dans la structure d'un graphe orienté et acyclique (DAG).[6] Par exemple, si A est dépendant de B mais pas de C et que B est dépendant de

5. Une chaîne de Markov est irréductible si elle est fortement connexe, c'est-à-dire qu'il existe un chemin entre toutes les paires de noeuds. De même, une chaîne de Markov est apériodique si le plus grand dénominateur commun des longueurs de tous ses cycles est 1. Ces deux conditions garantissent l'unicité de la distribution stationnaire et un temps de mélange fini [180].

6. En ce qui concerne les réseaux bayésiens, on réfère à la Réf. [142] pour une introduction et à la Réf. [213, Chapitre 10] pour une présentation plus détaillée.

Modèle	Activation $\alpha(n,m)$	Désactivation $\beta(n,m)$	Références
Ising-Glauber	$\frac{1}{1+\exp\left(\frac{2J}{T}(n-m)\right)}$	$\frac{\exp\left(\frac{2J}{T}(n-m)\right)}{1+\exp\left(\frac{2J}{T}(n-m)\right)}$	[112]
Ising-Metropolis	$\min\left(1, \exp\left(\frac{2J}{T}(m-n)\right)\right)$	$\min\left(1, \exp\left(\frac{2J}{T}(n-m)\right)\right)$	
SIS	λm	β	[231]
St-Onge	λm^{ν}	β	[291, 292]
Watts	$\begin{cases} 0 & \text{si } \frac{m}{n+m} < \tau \\ 1 & \text{si } \frac{m}{n+m} \geq \tau \end{cases}$	0	[322]
Cowan	$\frac{1}{1+\exp\left(\frac{1}{T}(\gamma m-\tau)\right)}$	β	[66]

TABLEAU 3.1 – Exemples de dynamiques binaires.

C, alors le DAG associé est $C \to B \to A$. Notons que si, dans cet exemple, C était dépendant de A, alors le système serait cyclique et donc ne serait plus considéré comme un réseau bayésien.

Dans le cas de processus sur graphe, le DAG connecte les noeuds à chaque temps à leurs voisins et eux-mêmes au temps précédent—la direction du lien allant dans le sens du temps. Les probabilités de transition prennent donc la forme générale suivante :

$$P(X_{i,t+1}|X_{1,t}, \cdots, X_{N,t}) = P(X_{i,t+1}|X_{i,t}, X_{\mathcal{N}_i,t}), \tag{3.22}$$

où $X_{\mathcal{N}_i,t} \equiv (X_{j,t})_{j\in\mathcal{N}_i}$ est le vecteur des états des voisins du noeud i au temps t. On définit l'homogénéité dans le voisinage d'un noeud i comme l'invariance de ses probabilités de transition sous permutation de ces voisins :

$$P(X_{i,t+1}|X_{i,t}, X_{\mathcal{N}_i,t}) = P(X_{i,t+1}|X_{i,t}, X_{\sigma(\mathcal{N}_i),t}), \tag{3.23}$$

pour toute permutation σ des voisins $\mathcal{N}_i$ de i.

Lorsque le processus est binaire, homogène dans le temps et homogène dans le voisinage, on parle alors de dynamiques binaires sur graphe. Typiquement, on dit d'un noeud qu'il est actif si son état est 1 et inactif s'il est 0. La particularité des dynamiques binaires est que les probabilités de transition sont définies en fonction du nombre de voisins actifs m et inactifs n uniquement, via les probabilités d'activation et de désactivation, notée $\alpha(n,m)$ et $\beta(n,m)$, respectivement. On donne plusieurs exemples de dynamiques binaires dans le Tableau 3.1, et des descriptions détaillées aux sous-sections qui suivent.

3.5.1 Modèles de spins

Les modèles de spins forment une classe de modèles graphiques non orientés qui décrivent la magnétisation spontanée dans les matériaux [201, 280]. Le modèle d'Ising est l'exemple le plus connu de cette classe, suivi du modèle de Potts et du modèle XY [331], qui, traditionnellement, évoluent tous sur des grilles régulières. Dans le modèle d'Ising, X représente un

vecteur de spins $X_i \in \{-1, 1\}$, où $X_i = 1$ si le spin i est orienté vers le haut et $X_i = -1$ s'il est orienté vers le bas. Ainsi, la probabilité d'une configuration de spins X à l'équilibre est donnée par une distribution de Boltzmann :

$$P(X) = \frac{e^{-\beta \mathbb{H}(X)}}{Z}, \quad \mathbb{H}(X) = -\sum_{i<j} J_{ij} X_i X_j - \sum_i h_i X_i \,. \tag{3.24}$$

où $\mathbb{H}(X)$ est l'hamiltonien, $\beta = \frac{1}{T} \in [0, \infty)$ est la température inverse d'un réservoir de chaleur auquel le réseau de spin est en contact, J_{ij} est une constante de couplage entre les spins i et j, h_i est un champ magnétique externe appliqué au spin i, et

$$Z = \sum_{X \in \{-1,1\}^N} e^{-\beta \mathbb{H}(X)} \tag{3.25}$$

est une constante de normalisation (aussi appelée fonction de partition). La matrice de couplage $\mathbf{J} = (J_{ij})_{i,j}$ représente la matrice d'adjacence d'un graphe signé et pondéré, où le couplage J_{ij} nous indique comment les noeuds i et j sont corrélés. Si $J_{ij} > 0$, alors les spins i et j ont tendance à s'aligner. Autrement, si $J_{ij} < 0$, ils se désalignent, et ils sont conditionnellement indépendants si $J_{ij} = 0$.

Sous cette forme, le modèle d'Ising décrit une distribution de probabilité à l'équilibre, non un processus en évolution. Cependant, cette distribution est réputée comme difficile à échantillonner lorsque le nombre de spins devient grand. Le calcul de la fonction de partition est à l'origine de l'*intraitabilité* du problème d'échantillonnage, puisque l'Éq (3.25) nécessite l'évaluation des 2^N différentes configurations de spins. L'algorithme de Metropolis—une variante avec probabilité de proposition uniforme de l'algorithme de Metropolis-Hastings—peut notamment être utilisé pour échantillonner des configurations de spins du modèle d'Ising. Cependant, la dynamique de Glauber est mieux adaptée que l'algorithme de Metropolis [112]. Les probabilités de transition dans les deux cas sont données au Tableau 3.1.

3.5.2 Propagation de maladies infectieuses

Lorsqu'on modélise la propagation d'une maladie infectieuse, on souhaite modéliser l'état de santé des individus dans la population. Traditionnellement, on dit d'un individu i qu'il est *susceptible* s'il est capable de contracter la maladie, et *infectieux* s'il en est porteur et peut la transmettre. Les modèles de propagation de maladies infectieuses, qu'on réfère également à des dynamiques de contagion simples, mettent en relation les individus susceptibles et infectieux via une probabilité de transmission, notée γ. De ce fait, la probabilité qu'un noeud susceptible ayant m voisins infectieux transitionne lui-même vers l'état infectieux—autrement dit, qu'il s'active—est la fonction de répartition cumulée d'une loi géométrique :

$$\alpha(n, m) = 1 - (1 - \gamma)^m \,. \tag{3.26}$$

Dans un contexte épidémiologique, on appelle aussi α la probabilité d'infection, pour laquelle l'Éq.(3.26) n'est valide qu'en temps discret. En temps continu, la probabilité qu'un voisin infectieux d'un individu susceptible transmette la maladie dans un intervalle de temps infinitésimal $\mathrm{d}t$ est $\gamma = \lambda\ \mathrm{d}t$, où λ est le taux de transmission. Ceci mène à une probabilité d'infection approximativement linéaire en m

$$\alpha(n, m) = 1 - (1 - \lambda\ \mathrm{d}t)^m \approx \lambda m\ \mathrm{d}t\,. \tag{3.27}$$

Cette probabilité d'activation est caractéristique des dynamiques de contagion simples. Sans mécanisme additionnel, on appelle cette dynamique le modèle susceptible-infectieux, ou simplement SI (voir Tableau 3.1). Si on ajoute un mécanisme de rétablissement, sous la forme d'une probabilité de rétablissement constante $\beta(n, m) = \beta$ pour les noeuds infectieux, on obtient le modèle susceptible-infectieux-susceptible—le modèle SIS.

Au-delà des modèles de contagion simple, on retrouve les modèles de *contagion complexes*, dont une infection se produit à un taux non linéaire en m. Dans les Réfs. [292] et [291], St-Onge et ses collaborateurs montrent comment une activation non linéaire peut survenir dans le contexte de propagation de maladies infectieuses. On démontre l'émergence d'un noyau prenant la forme m^ν dans le modèle SIS lorsque les temps d'expositions sont distribués selon une loi de puissance [292], ou quand la distribution des taux d'infection locaux est hétérogène [291].

3.5.3 Contagion sociale

Les processus de contagion ne sont pas seulement applicables à la propagation de maladies infectieuses. Dans certains contextes sociaux, on peut les utiliser pour modéliser la propagation d'information [79, 322], d'allégeances politiques [137, 287], de rumeurs [71, 72] et d'innovation [24, 257]. Comme les processus de contagion simples, les individus du système peuvent être susceptibles et infectieux, par contre le pathogène en question réfère à l'adoption d'un comportement social. En général, ces processus diffèrent des processus de contagion simple dans le fait que les transmissions nécessitent plusieurs expositions ; une caractéristique des processus de contagion complexe, comme nous l'avons vu plus haut.

Le modèle canonique des modèles de contagion complexe est celui de Watts [322], dit le modèle de seuil (voir Tableau 3.1). Le modèle de Watts incarne l'hypothèse de contagion complexe directement en utilisant une fonction de seuil pour la probabilité d'infection $\alpha(n, m)$, où un minimum de τ voisins infectieux est nécessaire pour déclencher une infection.

3.5.4 Processus biologique

L'exemple typique de processus binaire biologique sur graphe est celui des populations de neurones. Dans ce cas, le processus modélise l'activité calcique des neurones, connectés par des connexions synaptiques. Ici, l'activité calcique prend des valeurs binaires lorsqu'on

considère seulement les pics d'activité : si un neurone émet un pic calcique, son état vaut 1, et il vaut 0 autrement [66, 327]. Le modèle de prédilection pour les dynamiques de population neuronales est le modèle de Wilson-Cowan [327]. Dans ce modèle, on suppose que les neurones peuvent être susceptibles d'émettre un potentiel d'action, d'être en émission—ou actifs—, ou d'être dans un état de repos dit réfractaire. Alors, un neurone susceptible s'active si la somme de l'activité des neurones qui se trouvent dans son voisinage est supérieure à un certain seuil [106]—de manière analogue au modèle de Watts. On appelle *fonction de réponse* la probabilité d'activation d'un neurone qui dépend de son voisinage. Cette fonction prend généralement une forme sigmoïdale, c'est-à-dire une fonction monotone croissante, bornée inférieurement et supérieurement n'ayant qu'un seul point d'inflexion. Si le temps de repos est suffisamment petit, on se retrouve avec une dynamique binaire de la forme du modèle de Cowan [66] (voir Tableau 3.1).

3.6 Criticalité dans les processus sur graphe

La criticalité se manifeste particulièrement dans les processus sur graphe. Les phénomènes critiques sont décrits par des changements abrupts et globaux dans le comportement du système lorsque son environnement est modifié. L'exemple classique de transition de phase est celui du modèle d'Ising, décrit par l'Éq. (3.24) : au-delà d'une certaine température critique T_c, les spins s'alignent spontanément, même sans la présence d'un champs magnétique externe [200, 201]. L'alignement se manifeste à traver la magnétisation moyenne qui, pour $T > T_c$, tend vers zéro, et est non nulle lorsque $T < T_c$.

3.6.1 Transitions de phase

Il existe différents types de transitions de phase, et différentes manières de les étudier [168, 169, 200, 298]. Par exemple, dans un contexte de thermodynamique à l'équilibre, il est utile de définir et classifier les transitions de phase en termes de leur énergie libre, c'est-à-dire le logarithme de la fonction de partition, $F \propto \log Z$. Cependant, cette formulation n'est pas applicable aux processus hors d'équilibre comme ceux présentés au Tableau 3.1. Dans un contexte plus général, on utilise plutôt la formulation de Landau [168, 169] lorsqu'on fait référence à une transition de phase :

Définition 3.3 (Transition de phase)**.** *Soit un processus X de paramètre η dont la distribution stationnaire est $\boldsymbol{\pi} = \boldsymbol{\pi}(\eta)$ et une fonction $\Psi(\boldsymbol{\pi})$ appelée paramètre d'ordre. Une transition de phase survient dans la limite thermodynamique ($N \to \infty$) à un seuil critique $\eta = \eta_c$ lorsque la distribution stationnaire $\boldsymbol{\pi}$ transitionne d'une phase désordonnée ($\Psi(\boldsymbol{\pi}) = 0$) à une phase ordonnée ($\Psi(\boldsymbol{\pi}) > 0$).*

L'ordre dans le système est déterminé par les symétries dans la distribution stationnaire du système dans l'espace de η, et donc on interprète une transition de phase comme un brisure de symétrie. On mesure l'ordre à partir d'un paramètre d'ordre Ψ. Par exemple, dans le

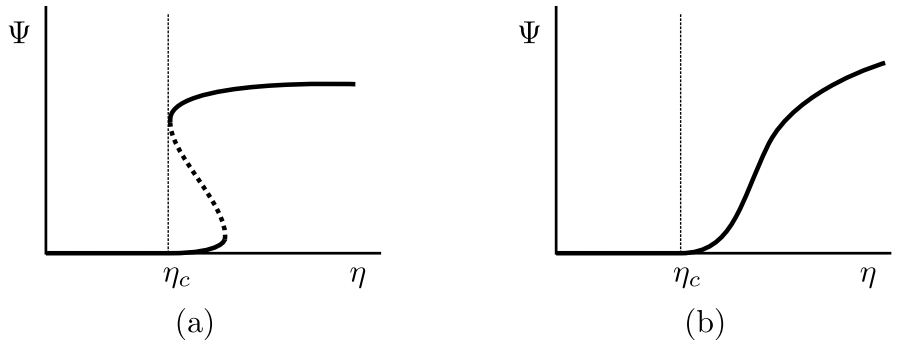


FIGURE 3.1 – Illustration de transition de phase (a) du premier ordre et (b) du deuxième ordre. Les lignes verticales représentent les seuil de transition de phase η_c. Sur la figure (a), une hystérèse est représentée par la ligne pointée—un état intermédiaire dont la présence délimite une région de bistabilité. Dans cette région, les états ordonné et désordonné sont stables, et l'état intermédiaire est instable.

modèle d'Ising, la magnétisation est un paramètre d'ordre qu'on utilise pour observer la transition de phase. Une transition de phase se produit en un point critique de l'espace des paramètres du système, en $\eta = \eta_c$ où les deux phase coexistent.

Il existe deux grandes familles de transitions de phase : les transitions de phase du premier ordre, dites *discontinues*, et les transitions de phase du deuxième ordre, dites *continues*. Le type de transition de phase dépend de la manière dont le paramètre d'ordre Ψ varie en fonction du paramètre de contrôle η.

3.6.2 Transitions du premier ordre

Les transitions de phase du premier ordre se présentent comme ayant une discontinuité dans le paramètre d'ordre Ψ (voir Fig. 3.1(a)). On dit également qu'elles sont irréversibles, puisque la transition de phase est observée différemment dépendamment des conditions initiales du système. Qui plus est, les transitions de phase du premier ordre sont parfois caractérisées par une hystérèse, c'est-à-dire une zone en η où les phases ordonnée et désordonnée sont simultanément stables. Les dynamiques sur graphe ayant un comportement explosif—comme le modèle de Cowan, une généralisation du modèle de Watts [79], et certains modèles de synchronisation et de percolation [35]—vont présenter des transitions de phase du premier ordre. Le modèle SIS non linéaire de St-Onge [292] est un autre exemple de dynamique sur graphe qui possède une telle transition de phase.

3.6.3 Transitions du deuxième ordre

Les transitions de phase du deuxième ordre sont dites continues puisqu'elles ne présentent pas de discontinuité dans Ψ, tel qu'illustré à la Fig. 3.1(b). Contrairement aux transitions du premier ordre, elles peuvent être décrites mathématiquement autour de leurs points de transition, ce qui nous permet de les caractériser et les classifier [80, 298]. En effet, un paramètre

d'ordre Ψ varie en loi de puissance autour d'un point critique η_c :

$$\Psi \sim |\eta - \eta_c|^{\beta}, \tag{3.28}$$

où β est l'exposant critique associé au paramètre d'ordre Ψ. Les exposants critiques sont des quantités universelles, indépendantes des détails microscopiques du système, qui dépendent uniquement de la topologie du système et de la classe d'universalité à laquelle le système appartient. Les modèles d'Ising et SIS sont tous les deux connus pour manifester des transitions de phase du deuxième ordre [138, 298].

Deuxième partie

Prédire et reconstruire la structure

Chapitre 4

Inférence et théorie de l'information

Reconstruire la structure des systèmes complexes à partir de données réelles est un problème qui a connu un regain d'intérêt ces dernières années non seulement dans la communauté informatique—avec l'avènement de l'apprentissage profond—, mais aussi chez les physiciens et statisticiens [236]. La raison est que ces-dites données sont de plus en plus accessibles, même nécessaires, si on souhaite pousser plus loin l'analyse de certains systèmes complexes. En effet, ce sont notamment de nouvelles approches bayésiennes d'inférence de réseaux qui sont à l'origine de cette recrudescence d'intérêt. L'article de M. E. J. Newman, publié dans *Nature Physics* en 2018, représente parfaitement cette tendance, où une technique bayésienne est utilisée pour quantifier le bruit dans les réseaux réels [220]. Plusieurs articles ont suivi, où des méthodes d'inférence ont été adaptées pour différents scénarios : un formalisme général par J.-G. Young [339], puis appliqué sur des données de relations pollinisateurs-plantes [342] et d'hypergraphes [184, 341] ; des méthodes hybrides de reconstruction et de détection de communauté par T. P. Peixoto [241, 242, 245, 246] ; et bien d'autres. Dans le contexte de reconstruction de graphes, la théorie bayésienne est devenue un outil incontournable pour développer des techniques de pointe.

Dans ce chapitre, nous survolons la théorie bayésienne dans le but de l'adapter pour la reconstruction de réseaux. Qui plus est, nous adoptons un point de vue basé sur la théorie de l'information—une théorie de la communication et de l'incertitude—pour comprendre les limites de l'approche bayésienne. Cette perspective nous permettra de développer, au Chapitre 5, une théorie de la reconstructibilité et de la prévisibilité dans les systèmes complexes, que l'on utilisera, au Chapitre 6, pour identifier les limites de la reconstruction des réseaux.

4.1 Statistique bayésienne

Supposons un ensemble de données X avec lesquelles nous souhaitons construire un modèle probabiliste pour les représenter. Ce modèle pourrait, par la suite, être utilisé pour prédire de nouvelles données et potentiellement mieux comprendre comment fonctionnent les mé-

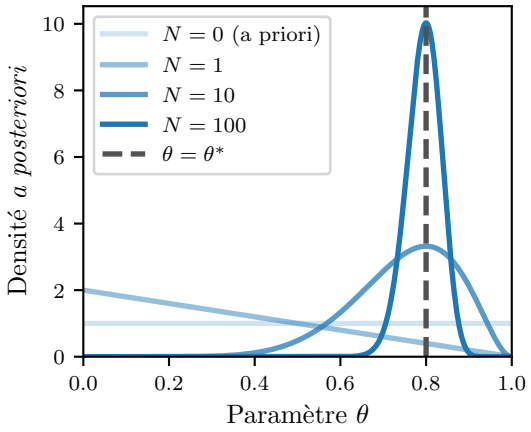


FIGURE 4.1 – Loi *a posteriori* d'une pièce de monnaie biaisée. Dans l'exemple, on fixe la probabilité réelle de la pièce à $\theta^* = 0.8$ (illustrée par la ligne pointillée noire), on fait varier la taille de l'échantillon N et on fixe le nombre de piles à $k = N\theta^*$. Les lois *a posteriori*, calculées avec l'Éq. (4.3), sont montrées en bleu (le gradient de couleur indique la valeur de N utilisée).

canismes génératifs sous-jacents. En outre, le modèle est paramétrisé par un ensemble de paramètres θ, lesquels influencent son comportement tel que mesuré par sa *vraissemblance*, notée $P(X|\theta)$. Plus cette vraissemblance est élevée, meilleurs sont les paramètres θ du modèle pour décrire les données. Ainsi, l'objectif de modélisation est d'inférer une ou plusieurs configurations de paramètres θ permettant au modèle de bien décrire les données X.

Dans l'approche bayésienne, on considère que les paramètres θ eux-mêmes sont des variables aléatoires, qui *a priori* suivent une distribution de probabilité $P(\theta)$ [103]. Ce faisant, on peut calculer la probabilité que les paramètres θ aient réellement généré les données X en utilisant le théorème de Bayes (Éq. 1.18) :

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}. \tag{4.1}$$

Cette probabilité quantifie *a posteriori*, c'est-à-dire après l'observation de X, la probabilité que les paramètres θ aient généré les données, et mesure en quelque sorte l'incertitude sur les paramètres.

Prenons un exemple simple pour illustrer le calcul de la probabilité *a posteriori*. Soit une pièce de monnaie biaisée dont la probabilité de tomber sur pile, dénotée θ^*, est utilisée pour générer une séquence $X = (X_1, X_2, \cdots X_N)$ de N lancers, où X_i vaut 1 si la pièce tombe sur pile au i-ème lancer et 0 si elle tombe sur face. Dans le formalisme bayésien, on considère que la probabilité θ^* est inconnue. Or, pour modéliser le système, on paramétrise la probabilité de tomber sur pile avec θ, une variable à inférer. On l'utilise ensuite pour calculer la probabilité de vraissemblance de générer la séquence de lancers X, qui suit une loi binomiale :

$$P(X|\theta) = \theta^k(1-\theta)^{N-k}, \tag{4.2}$$

où k est le nombre de piles observées dans la séquence X. Comme nous n'avons aucune idée de la valeur de θ, nous choisissons *a priori* une densité uniforme sur l'intervalle $[0, 1]$, c'est-à-dire $\rho(\theta) = 1$, ce qui mène à la densité *a posteriori* :

$$\rho(\theta|X) = \frac{P(X|\theta)\rho(\theta)}{P(X)} = \frac{\theta^k(1-\theta)^{N-k}}{B(k+1, N-k+1)}\,, \tag{4.3}$$

soit une distribution bêta de paramètres $(k+1, N-k+1)$, dont la constante de normalisation $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ est la fonction bêta. Sur la Fig. 4.1, on illustre comment la loi *a posteriori* de θ varie en fonction de la taille de l'échantillon N. Ainsi, plus la taille de l'échantillon est grande, plus la loi *a posteriori* se concentre autour de la vraie valeur θ^* du biais de la pièce.

4.2 Qualité des estimateurs bayésiens

En inférence bayésienne, des estimateurs ponctuels des paramètres θ peuvent être évalués à partir de la loi *a posteriori*. Ces estimateurs sont parfois plus simples à manipuler et interpréter que la loi *a posteriori* elle-même. Les plus utilisés sont l'espérance de la loi *a posteriori* (EAP) $\hat{\theta}_{\text{EAP}} = \mathbb{E}_{\theta|X}[\theta]$ et l'estimateur du mode *a posteriori* (MAP), dénoté $\hat{\theta}_{\text{MAP}} = \arg\max_\theta P(\theta|X)$. Dans l'exemple de la pièce ci-haut, ces estimateurs sont donnés par les formules suivantes :

$$\hat{\theta}_{\text{EAP}} = \frac{k+1}{N+2}\,, \quad \hat{\theta}_{\text{MAP}} = \frac{k}{N}\,. \tag{4.4}$$

Les deux estimateurs ont des valeurs relativement similaires, et tendent vers la même valeur dans la limite $N \to \infty$. Par ailleurs, l'espérance de ces deux estimateurs convergent correctement vers la vraie valeur de θ^*. Sachant que $\mathbb{E}[k] = N\theta^*$, et on obtient

$$\mathbb{E}\left[\hat{\theta}_{EAP}\right] = \frac{N\theta^*+1}{N+2} \simeq \theta^*\,, \quad \mathbb{E}\left[\hat{\theta}_{\text{MAP}}\right] = \frac{N\theta^*}{N} = \theta^*\,. \tag{4.5}$$

La capacité d'un estimateur à converger vers la vraie valeur des paramètres est appelée *cohérence*, laquelle est définie comme suit :

Définition 4.1 (Cohérence d'un estimateur)**.** *Un estimateur $\hat{\theta}_N$, qui dépend d'une séquence de N réalisations indépendantes et identiquement distribuées selon $P(X|\theta^*)$, est dit cohérent si*

$$\hat{\theta}_N \xrightarrow{\mathbb{P}} \theta^*\,. \tag{4.6}$$

La cohérence des estimateurs décrits ci-haut dépend de certaines conditions dites de régularisation (voir [178, Chapitre 7]). L'identifiabilité des paramètres est l'une de ces conditions, qui stipule que pour toutes paires de paramètres θ et θ', leur probabilité *a posteriori* doit être différente, c'est-à-dire $P(\theta|X) \neq P(\theta'|X)$ si $\theta \neq \theta'$. Autrement, on dit du modèle qu'il est non identifiable.

Partant de ces conditions, on énonce le théorème de Bernstein-Von Mises :

Theorème 4.1 (Théorème de Bernstein-Von Mises)**.** *Soit un modèle statistique décrit par une vraissemblance $p(X|\theta)$ de paramètre θ et une distribution* a priori *$p(\theta)$ de support $\Theta \subset \mathbb{R}^d$. Soit un échantillon $X = (X_1, X_2, \cdots, X_N)$ de N réalisations indépendantes et identiquement distribuées selon $p(X|\theta = \theta^*)$, pour un certain θ^*. Sous certaines conditions de régularité*[1]*, la distribution* a posteriori *$p(\theta|X)$ converge en distribution vers une loi normale $\mathcal{N}\left(\theta^*, \frac{1}{N}\mathcal{I}^{-1}(\theta^*)\right)$ centrée sur la vraie valeur du paramètre θ^*, où $\mathcal{I}(\theta^*)$ est la matrice d'information de Fisher, dont l'élément (i, j) est donné par*

$$[\mathcal{I}(\theta^*)]_{ij} = -\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log p(X|\theta)\right]_{\theta=\theta^*}. \tag{4.7}$$

Démonstration. La preuve du théorème de Bernstein-Von Mises est disponible dans l'Annexe B du livre [103]. □

Les implications du théorème de Bernstein-Von Mises sur la qualité des estimateurs bayésiens sont significatives. Notamment, il démontre la cohérence des deux estimateurs discutés ci-haut, et explique pourquoi ils convergent vers la vraie valeur du paramètre. Une conséquence importante est que la loi à posteriori converge vers une loi normale indépendamment de la loi à priori. Autrement dit, plus on observe de données, moins l'information à priori influence l'inférence.

Notons que le théorème de Bernstein-Von Mises ne s'applique pas à tous les modèles bayésiens, par exemple ceux dont l'espace de paramètres est discret comme les modèles de graphes. Cependant, des travaux recents montrent qu'il est possible, par le biais de plongements qui préservent la structure des espaces discrets, d'exprimer ces modèles dans des espaces continus. Ceci permet notamment d'appliquer le théorème de Bernstein-Von Mises à des modèles discrets, et d'adapter des méthodes d'échantillonnage spécifiques aux modèles continues pour les méthodes discrètes [346], et des familles de loi *a priori* non informatives [30].

4.3 Échantillonnage de la loi *a posteriori*

Dans le cas général, la loi *a posteriori* $P(\theta|X)$ doit être calculée numériquement. Qui plus est, dans le cas de modèles complexes dont l'espace des paramètres est hautement dimensionnel, ce calcul numérique devient rapidement fastidieux, et même intraitable. Cette limitation provient de la constante de normalisation de $P(\theta|X)$, c'est-à-dire $P(X) = \mathbb{E}_\theta\left[P(X|\theta)\right]$, qui

1. Les conditions de régularité sont les suivantes :
   a) La densité *a priori* $p(\theta)$ est continue et positive dans un voisinage de θ^* ;
   b) La vraissemblance $p(X|\theta)$ est différentiable au moins deux fois par rapport à θ ;
   c) La matrice d'information de Fisher $I(\theta)$ est continue et non singulière à $\theta = \theta^*$;
   d) Le modèle est identifiable.

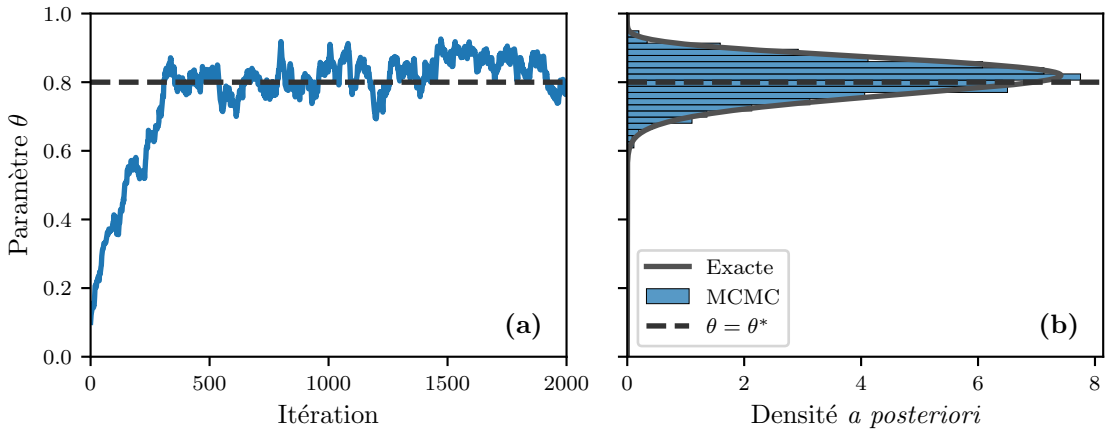


FIGURE 4.2 – Échantillonnage par MCMC de la densité *a posteriori* de θ pour une pièce de monnaie. Les données sont composées de 50 lancers d'une pièce biaisée avec une probabilité $\theta^* = 0.8$, dont 41 sont tombés sur pile. Sur la figure (a), on montre les 2000 premières itérations d'une chaîne Monte-Carlo de condition initiale $\theta_0 = 0.1$; et sur la figure (b), l'histogramme de son échantillon complet (100 000 itérations). Afin d'assurer que les échantillons soient décorrélés, on prend un échantillon sur 50 pour construire l'histogramme. Pour cette expérience, les estimateurs EAP et MAP sont $\hat{\theta}_{EAP} \simeq 0.808$ et $\hat{\theta}_{MAP} = 0.82$.

porte le nom d'*évidence*. L'évidence représente la probabilité marginale des données indépendamment de la valeur des paramètres du modèle. Comme nous le verrons plus loin, elle constitue une fonction objective fondée permettant de comparer des modèles entre eux.

Pour contourner ce problème, on repose une fois de plus sur les méthodes d'échantillonnage MCMC, décrites aux Sections 2.6 et 3.4. Dans le cas bayésien, on souhaite échantillonner la loi *a posteriori* qui, à une constante de normalisation près, est

$$P(\theta|X) \propto P(X|\theta)P(\theta)\,. \tag{4.8}$$

Ainsi, de nouveaux paramètres θ', proposés avec une probabilité $P(\theta'|\theta)$, sont acceptées avec la probabilité d'acceptation de Metropolis-Hastings :

$$\alpha = \min\left(1, \frac{P(\theta'|X)P(\theta|\theta')}{P(\theta|X)P(\theta|\theta')}\right) = \min\left(1, \frac{P(X|\theta')P(\theta')P(\theta|\theta')}{P(X|\theta)P(\theta)P(\theta'|\theta)}\right), \tag{4.9}$$

où l'évidence s'annule dans le calcul de α puisqu'elle est indépendante des paramètres θ. Le calcul de l'évidence n'est donc pas nécessaire à celui de la probabilité d'acceptation, ce qui nous permet d'échantillonner la loi *a posteriori* efficacement.

À la Fig. 4.2, on illustre le processus d'échantillonnage MCMC pour la pièce de monnaie biaisée. D'abord, on fixe la probabilité θ^* de la pièce, et on génère les données X, c'est-à-dire une séquence de lancers. Ensuite, à partir des données, on échantillonne la loi *a posteriori* $P(\theta|X)$ par MCMC en partant d'une condition initiale θ_0. Dans le cas considéré, les propositions de nouveaux paramètres θ' sont faites en utilisant une loi normale centrée sur le paramètre courant θ, dont la variance $\sigma^2 = 0.01$ est fixe. On observe que, malgré la différence entre la valeur

réelle et la condition initiale, la chaîne converge rapidement autour de la vraie valeur de θ^*. D'autre part, on montre que l'histogramme concorde parfaitement avec la densité *a posteriori* exacte prédite par l'Éq. 4.3. Le code source de l'expérience est disponible à la Réf. [209].

4.4 Sélection de modèles

La sélection de modèles est un sujet central dans le contexte de la modélisation de données, où plusieurs modèles sont comparés afin de déterminer lequel est le plus adéquat [254]. Formellement, le principe de sélection de modèles est tout à fait compatible avec la théorie bayésienne formulée ci-haut. Le modèle est ainsi considéré comme un paramètre supplémentaire M à inférer simultanément avec les autres θ. Typiquement, un ensemble fini de modèles $\{M_1, M_2, \cdots, M_k\}$ est considéré, ce qui en fait une variable discrète. La probabilité conjointe du modèle M, ses paramètres θ et les données X se factorisent comme suit :

$$P(X, \theta, M) = P(X|\theta, M)P(\theta|M)P(M)\,. \tag{4.10}$$

Pour faire l'inférence de M, on assigne une probabilité *a priori* $P(M)$ à chaque modèle, et on calcule la probabilité *a posteriori* du modèle M en utilisant le théorème de Bayes.

Dans la pratique, on n'infère pas directement le modèle M comme les autres paramètres, par exemple en utilisant un algorithme MCMC proposant des changements de modèles, puisque cette procédure serait extrêmement coûteuse. Plutôt, on utilise la méthode des facteurs de Bayes [254]. Cette méthode implique le calcul de la probabilité *a posteriori* marginale du modèle M :

$$P(M|X) = \sum_{\theta} P(M, \theta|X) \propto P(M)P(X|M)\,, \tag{4.11}$$

qui est proporitionnelle à $P(X|M)$, l'évidence du modèle M. Ainsi, pour comparer deux modèles, on calcule le rapport des évidences appelé le facteur de Bayes :

$$\mathcal{B} = \frac{P(X|M_1)}{P(X|M_2)} = \left[\frac{P(M_1|X)}{P(M_2|X)}\right]\left[\frac{P(M_1)}{P(M_2)}\right]\,. \tag{4.12}$$

Si le facteur de Bayes $\mathcal{B}$ est plus grand que 1, le modèle M_1 est plus favorable d'avoir généré les données que M_2 ; autrement, le modèle M_2 est plus adéquat. Le facteur de Bayes constitue un critère de sélection cohérent et fiable dans plusieurs cas (Réf. [56] fournit une revue de ces résultats).

Le calcul du facteur de Bayes repose sur notre capacité à évaluer l'évidence de nos modèles, qui représente une tâche généralement difficile. La raison est que l'évidence nécessite la marginalisation des paramètres du modèle, c'est-à-dire le calcul de la somme

$$P(X|M) = \sum_{\theta} P(X|\theta, M)P(\theta|M)\,, \tag{4.13}$$

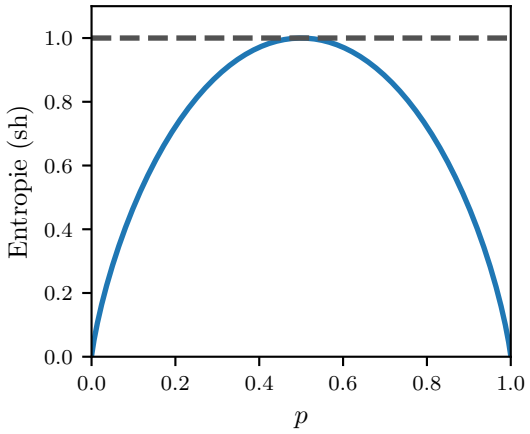


FIGURE 4.3 – Entropie d'une pièce de monnaie biaisée avec une probabilité p. La ligne pointillée horizontale indique le maximum de l'entropie binaire $\mathcal{H}(p)$, à 1 sh.

qui, pour un grand nombre de paramètres, devient rapidement intraitable. Or, des méthodes numériques existent pour évaluer l'évidence. En général, les méthodes Monte-Carlo naïves génèrent des estimateurs biaisés ou à variance élevée, c'est pourquoi on doit faire appel à des techniques plus sophistiquées qui reposent sur l'échantillonnage de la loi *a posteriori*. Quelques exemples de ces techniques sont présentés aux Réfs. [215, 225, 332].

4.5 Du point de vue de la théorie de l'information

La théorie bayésienne permet de quantifier l'incertitude sur les paramètres inférés d'un modèle, telle que mesuré par la distribution *a posteriori*. Or, l'incertitude est le langage de la théorie de l'information—une théorie mathématique de la communication développée dans un célèbre article de Claude Shannon publié en 1948 [279]. Dans ce formalisme général, l'incertitude d'une réalisation x de X est mesurée par l'auto-information :

$$i(x) = -\log P(X = x)\,, \tag{4.14}$$

qu'on associe à la surprise de l'observation de $X = x$. Typiquement, l'auto-information est mesurées en *shannon* (sh), auquel cas le logarithme est en base 2. Cette quantité est nulle si $P(X = x) \in \{0, 1\}$, auquel cas la valeur de X est certaine ; autrement, elle prend des valeurs positives.

L'espérance de l'auto-information représente une mesure bien connue en physique statistique —l'entropie de Shannon, c'est-à-dire une généralisation des formulations de Boltzmann et de Gibbs :

$$H(X) = \mathbb{E}[i(X)] = -\sum_x P(X = x) \log P(X = x)\,. \tag{4.15}$$

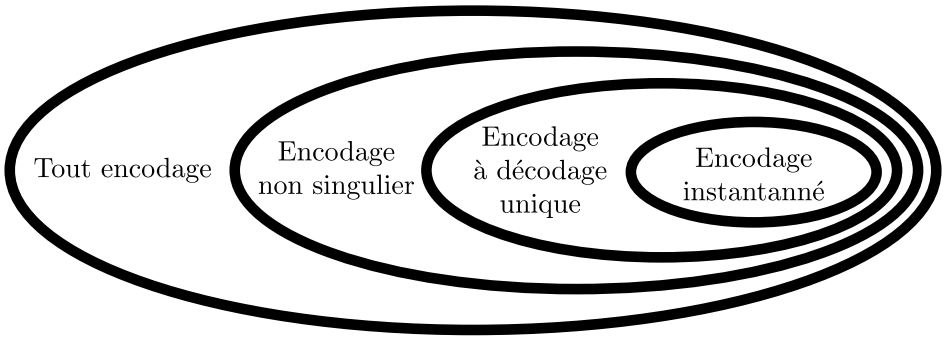


FIGURE 4.4 – Classification des types d'encodage, inspirée de la Fig. 5.1 de la Réf. [65].

Dans le cas d'une variable binaire, comme une pièce de monnaie paramétrisée par une probabilité p, l'entropie de Shannon est donnée par l'entropie binaire $\mathcal{H}(p)$:

$$\mathcal{H}(p) = -p \log p - (1-p) \log(1-p) \,. \tag{4.16}$$

L'entropie binaire est maximale pour $p = \frac{1}{2}$, c'est-à-dire lorsque la pièce est non biaisée, auquel cas pile et face sont équiprobables et l'incertitude est maximisée ; et minimale lorsque $p = 0$ ou $p = 1$ (voir Fig. 4.3). Dans le cas général, on peut montrer que l'entropie est une fonction concave, qui possède un maximum unique associé à la distribution uniforme $P(X) = |\mathcal{X}|^{-1}$ (voir la Réf. [65, Théorème 2.6.1]) :

$$H(X) \leq \log |\mathcal{X}| \,, \tag{4.17}$$

où on rappelle que $\mathcal{X}$ est l'ensemble des valeurs possibles de X.

Il est possible de formuler une interprétation de l'entropie en termes du nombre moyen de questions binaires—se répondant par oui ou non—nécessaires pour identifier la valeur de X. Dans le cas d'une variable de Bernoulli de paramètre $p = \frac{1}{2}$, comme l'entropie vaut 1 sh, il faut en moyenne une question pour identifier la valeur de X—par exemple, « la pièce est-elle tombée sur pile ?». Cette correspondance provient du fait que toute question binaire partitionne l'espace des issues possibles de X en deux, isolant ainsi des régions de plus en plus petites à chaque réponse donnée. L'algorithme optimale de questionnement est celui où, à chaque itération, une question est posée de manière à diviser l'espace en deux régions occupant le même volume de probabilité. L'entropie de Shannon est donc le nombre moyen d'itérations nécessaires pour que cet algorithme identifie la bonne valeur de X.

4.6 Encodage et longueur de description

À la base, la théorie de l'information a été proposée par C. Shannon en 1948 comme une théorie de la communication [279]. Ainsi, on imagine que de l'information peut être décrite par un message, ou de manière plus abstraite, un *encodage*. En théorie de l'information, la notion

d'encodage est importante et intervient dans l'une de ces applications les plus importantes—la compression de données—, où l'entropie joue un rôle prédominant. Formellement, tout encodage de X est généré par un *code source* $C : \mathcal{X} \to \mathcal{C}$, où $C(x) = a_1 a_2 ... a_L$ représente l'encodage de x comme une séquence de symboles. Il existe différentes classes d'encodage, certaines permettant de décoder l'information de manière unique, d'autres permettant de compresser d'avantage l'information au détriment de perte d'information (voir la Fig. 4.4). On réfère à la Réf. [65, Chapitre 5] pour une revue détaillée des encodages.

La longueur de l'encodage L représente son efficacité à compresser l'information de X. Or, on peut montrer que la longueur minimale moyenne d'un encodage sans perte est bornée inférieurement par l'entropie de Shannon $H(X)$ (voir la Réf. [65, Theorème 5.3.1]). Dans le contexte bayésien, un critère sur la longueur de l'encodage peut être utilisé pour la sélection de modèles [238, 239, 242, 245, 336]. Ce critère porte le nom de principe de longueur de description minimale, laquelle s'exprime comme suit :

$$\mathcal{L}(X, \theta) = -\log P(X|\theta) - \log P(\theta)\,. \tag{4.18}$$

Intuitivement, l'idée de minimiser la longueur de description est analogue au principe du rasoir d'Ockham, qui favorise l'explication la plus simple. Dans cette veine, la longueur de description $\mathcal{L}(X, \theta)$ est composée de deux termes : le premier représente l'encodage des données par le modèle, et le deuxième, l'encodage du modèle lui-même. Ainsi, le meilleur modèle est celui qui représente les données efficacement, sans qu'il soit lui-même trop complexe.

Le principe de longueur de description minimale est un critère fondé sur les bases de la théorie de l'information, qui peut également être relié aux estimateurs classiques de la théorie bayésienne. En effet, on peut montrer que l'estimateur $\hat{\theta}_{\text{MAP}}$ minimise également la longueur de description $\mathcal{L}(X, \theta)$:

$$\begin{aligned}\hat{\theta}_{\text{MAP}} &= \arg\max_{\theta} P(\theta|X) = \arg\max_{\theta} \left[P(\theta)P(X|\theta)\right] \\ &= \arg\max_{\theta} \left[\log P(\theta) + \log P(X|\theta)\right] = \arg\min_{\theta} \mathcal{L}(X, \theta)\,,\end{aligned}$$

où on a utilisé le fait que $P(X)$, la constante de normalisation de $P(\theta|X)$, est indépendante de θ, et que $\log(x)$ est une fonction strictement croissante.

4.7 Information mutuelle

La théorie de l'information permet de quantifier l'incertitude des paramètres θ via l'entropie de la loi *a posteriori* $H(\theta|X)$. En principe, cette quantité d'incertitude est bornée, inférieurement et supérieurement. D'une part, $H(\theta|X)$ est bornée inférieurement par zéro, lorsque les paramètres θ sont parfaitement reconstruits à partir des données—sans incertitude. D'autre

part, la valeur maximale de l'entropie *a posteriori* est bornée par l'entropie de la loi *a priori* $H(\theta)$—ce que nous allons démontrer sous peu. Lorsque $H(\theta|X) = H(\theta)$, les données n'apportent aucune information sur les paramètres.

Intuitivement, la différence entre ces deux quantités représente l'information « acquise » sur les paramètres à partir des données.[2] Cette importante quantité porte le nom d'information mutuelle, et se définit comme suit :

$$I(\theta; X) = H(\theta) - H(\theta|X)\,. \tag{4.19}$$

On peut également voir l'information mutuelle comme une mesure de comparaison entre le produit des probabilités marginales des données et des paramètres—comme s'ils étaient indépendants—et leur probabilité conjointe :

$$I(\theta; X) = \mathbb{E}_{X,\theta}\left[\log \frac{P(X,\theta)}{P(X)P(\theta)}\right]\,, \tag{4.20}$$

De ce point de vue, l'information mutuelle est une divergence de Kullback-Leibler (KL) entre la distribution jointe $P(X,\theta)$ et le produit des marginales $P(X)P(\theta)$, définie comme suit :

$$D_{KL}(P(X,\theta)||P(X)P(\theta)) = \mathbb{E}_{X,\theta}\left[\log \frac{P(X,\theta)}{P(X)P(\theta)}\right]\,, \tag{4.21}$$

qui mesure la dissimilarité entre les deux distributions. L'information mutuelle mesure donc la dépendance entre deux variables aléatoires.

L'information mutuelle possède plusieurs propriétés utiles. Comme l'entropie, $I(\theta; X)$ est bornée inférieurement et supérieurement. On obtient une borne supérieure de $I(\theta; X)$ en utilisant le fait que $H(\theta|X) \geq 0$:

$$I(\theta; X) \leq H(\theta)\,. \tag{4.22}$$

La borne inférieure est plus subtile, et se démontre grâce à l'inégalité de Jensen :

Theorème 4.2 (Inégalité de Jensen)**.** *Pour toute variable X et fonction convexe f,*

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])\,. \tag{4.23}$$

Démonstration. La preuve de l'inégalité de Jensen est donnée à la Réf. [65, Théorème 2.6.2]. □

2. Dire que l'information mutuelle représente l'information acquise sur une variable lorsque l'autre est observée n'est pas strictement correct. En réalité, l'information mutuelle représente une réduction de l'incertitude, non un gain. Nous utilisons ce raccourcis afin de facilité l'intuition.

En utilisant cette inégalité et le fait que $-\log(\cdot)$ est une fonction convexe, on obtient la non négativité de l'information mutuelle :

$$\begin{aligned} I(\theta; X) &= \mathbb{E}\left[\log \frac{P(X,\theta)}{P(X)P(\theta)}\right] \\ &\geq \log \mathbb{E}\left[\frac{P(X,\theta)}{P(X)P(\theta)}\right] \\ &= \log(1) = 0\,. \end{aligned}$$

De cette inégalité, on déduit un corollaire important sur la relation entre les entropies marginale et conditionnelle :

$$H(\theta) \geq H(\theta|X)\,, \tag{4.24}$$

qui stipule que tout conditionnement peut réduire l'incertitude—autrement dit, *information can't hurt* [65, Théorème 2.6.5]. L'information mutuelle peut également être formulée de manière complètement équivalente en termes des entropies de X :

$$I(X;\theta) = H(X) - H(X|\theta)\,. \tag{4.25}$$

Cette formulation provient du fait que $\frac{P(X,\theta)}{P(\theta)} = P(X|\theta)$, qui, remplacé dans l'Eq. 4.20, donne directement l'expression ci-haut. Conséquemment, l'information mutuelle est une mesure symétrique, où $I(X;\theta) = I(\theta;X)$, où la quantitée d'information gagnée sur les paramètres à partir des données est égale à celle sur les données à partir des paramètres.

Dans les prochains chapitres, nous montrons une application tangible de la théorie de l'information où l'entropie et l'information mutuelle servent de fondations pour développer une théorie bayésienne de la reconstructibilité des systèmes complexes. Dans ce contexte, les données X prennent la forme de séries temporelles issues d'un processus stochastique dont les composantes interagissent selon la structure d'un graphe. Ainsi, ce graphe, l'un des paramètres du modèles, peut être inféré à partir des données. Nous verrons que ces quantités, principalement l'information mutuelle, permettent de quantifier le lien entre structure et fonction, et de mettre en évidence la tension—même la *dualité*—qui existent entre les deux. Cette théorie permettra également de quantifier la limite de reconstructibilité de systèmes réels.

Chapitre 5

De la dualité entre prévisibilité et reconstructibilité dans les systèmes complexes

Article original :
Duality between predictability and reconstructibility in complex systems
Charles Murphy, Vincent Thibeault, Antoine Allard, Patrick Desrosiers
Département de Physique, de Génie Physique et d'Optique, Université Laval, Québec (Qc), Canada G1V 0A6
Référence : Nat. Commun. **15**, 4478 (2024) [212]

1. Ces sections contiennent le contenu original de l'article. Celui-ci n'a été modifié que pour se conformer au format exigé par la Faculté des études supérieures et postdoctorales de l'Université Laval.

5.1 Avant-propos

Au chapitre précédent, nous avons présenté un cadre théorique basé sur la théorie de l'information permettant d'étudier les statistiques bayésiennes d'un point de vue informationnel. De ce point de vue, l'information mutuelle entre les paramètres et les observations joue un rôle central dans la quantification de leur interdépendance. Cette idée est apparue suite à un cours sur la théorie de l'information au courant de l'année 2019. Initialement, l'idée était de quantifier la force de la relation entre la structure d'un système dynamique et son état à tout moment. Ce projet a nécessité une longue période d'idéation : comme essayer d'attraper un saumon dans une rivière à main nue, il a fallu plusieurs itérations avant de converger vers la bonne conception du problème. Celle-ci fait intervenir la prévisibilité et la reconstructibilité des systèmes complexes.

Ce travail présente donc une théorie basée sur l'information mutuelle entre la structure d'un système (représentée par un graphe aléatoire) et son état (représenté dans un processus stochastique), qui établit un lien entre prévisibilité et reconstructibilité. Par prévisibilité, on entend notre capacité à prédire l'évolution du système seulement en connaissance de sa structure, tandis que la reconstructibilité représente notre capacité à reconstruire sa structure à partir d'observations temporelles issues du système lui-même. Qui plus est, nous avons découvert que, bien que ces deux notions soient intimement liées, elles peuvent se comporter de manière duale—i.e., un système peut être difficile à prédire mais facile à reconstruire, et vice versa. Cette dualité peut exister dans plusieurs contextes, notamment elle apparaît lorsqu'on augmente le nombre d'observations du processus ; et elle semble émerger autour des zones de criticalité.

Dans ce travail, nous utilisons une notation légèrement différente de celle utilisée jusqu'à présent. En effet, X représente une matrice aléatoire et X_t, un vecteur d'état du processus à l'instant t, lesquels n'utilisent pas la notation en gras. Ce choix de notation souligne la généralité de X qui peut être n'importe quel type de variables aléatoires.

Le formalisme présenté dans cet article est général et peut être adapté pour étudier spécifiquement la reconstructibilité des réseaux complexes et des systèmes empiriques. C'est ce que nous verrons dans le Chapitre 6.

Symbole	Description
N, E	Nombre de noeuds et de liens, respectivement
a_{ij}	Multiplicité du lien entre i et j
$\boldsymbol{k}$	Séquence des degrés, l'élément k_i est le degré du noeud i

$p(k)$	Distribution des degrés, probabilité qu'un noeud aléatoirement sélectionné ait un degré k
G	Graphe aléatoire
$\mathcal{G}$	Ensemble des graphes possibles
T	Nombre d'observations temporelles
X	Processus stochastique, matrice aléatoire de dimension $N \times T$
X_t	Vecteur d'état du processus à l'instant t, de dimension N
$X_{i,t}$	État du noeud i à l'instant t
X_{past}	Historique des états du processus jusqu'à l'instant τ
X_{future}	États du processus après l'instant τ
Ω	Ensemble des états de noeuds possibles pour tout $X_{i,t}$
$H(X)$	Entropie de la variable aléatoire X
$I(X;G)$	Information mutuelle entre les variables aléatoires X et G
$U(G\|X)$	Reconstructibilité
$U(X\|G)$	Prévisibilité

TABLEAU 5.1 – Glossaire des symboles utilisés au Chapitre 5.

5.2 Résumé

Prédire l'évolution de systèmes composés d'un grand nombre de composantes à partir de la structure de leurs interactions est un problème fondamental dans la théorie des systèmes complexes. Il en va de même pour le problème de la reconstruction de la structure d'interaction à partir d'observations temporelles. Dans cet article, nous identifions une relation intriquée entre prévisibilité et reconstructibilité en utilisant la théorie de l'information. Dans notre approche, nous utilisons l'information mutuelle entre un graphe aléatoire et un processus stochastique évoluant sur ce graphe aléatoire pour quantifier leur interdépendance. Nous montrons comment les coefficients d'incertitude, intimement liés à cette information mutuelle, quantifient notre capacité à reconstruire un graphe à partir de séries temporelles, et notre capacité à prédire l'évolution d'un processus à partir de la structure de ses interactions. Nous calculons analytiquement les coefficients d'incertitude pour de nombreux systèmes différents, notamment les systèmes déterministes continus, et décrivons une procédure numérique lorsque les calculs exacts sont inaccessibles. Au coeur de ce travail se trouve dans la

découverte d'une dualité entre la prévisibilité et la reconstructibilité. Nous prouvons mathématiquement comment une telle dualité émerge universellement en changeant le nombre d'étapes dans le processus et illustrons comment les dualités entre prévisibilité et reconstruction peuvent exister dans les processus dynamiques sur des réseaux réels proches de la criticalité.

5.3 Abstract

Predicting the evolution of a large system of interacting components using the structure of their interactions is a fundamental problem in complex system theory. And so is the problem of reconstructing the structure of interaction from temporal observations. Here, we find an intricate relationship between predictability and reconstructability using an information-theoretical point of view. We use the mutual information between a random graph and a stochastic process evolving on this random graph to quantify their codependence. Then, we show how the uncertainty coefficients, which are intimately related to that mutual information, quantify our ability to reconstruct a graph from an observed time series, and our ability to predict the evolution of a process from the structure of its interactions. We provide analytical calculations of the uncertainty coefficients for many different systems, including continuous deterministic systems, and describe a numerical procedure when exact calculations are intractable. Interestingly, we find that predictability and reconstructability, even though closely connected by the mutual information, can behave differently, even in a dual manner. We prove how such duality universally emerges when changing the number of steps in the process. Finally, we provide evidence that predictability-reconstruction dualities may exist in dynamical processes on real networks close to criticality.

5.4 Introduction

The relationship between structure and function is fundamental in complex systems [18, 170, 221], and important efforts have been invested in developing network models to better understand it. In particular, models of dynamics on networks [23, 36, 139, 231] have been proposed to assess the influence of network structure over the temporal evolution of the activity in the system. In turn, data-driven models [133, 210], dimension-reduction techniques [99, 171, 249, 304, 305] and mean-field frameworks [130, 232, 293, 294, 297] have deepened our predictive capabilities. Among other things, these theoretical approaches have shed light on the relationship between dynamics criticality and many network properties such as the degree distribution [232, 297], the eigenvalue spectrum [53, 89, 230] and their group structure [131, 294, 296]. Fundamentally, these contributions justify our inclination for measuring and using real-world networks as a proxy to predict the behavior of complex systems.

Models of dynamics on networks have also been used as reverse engineering tools for network reconstruction [47], when the networks of interactions are unavailable, noisy [241, 339, 342] or faulty [172]. The network reconstruction problem has stimulated many technical contributions [196] : Thresholding matrices built from correlation [160] or other more sophisticated measures [274, 277] of time series, Bayesian inference of graphical models [1, 6, 29, 44, 267, 268] and models of dynamics on networks [242], among others. These techniques are widely used (e.g., in neuroscience [26, 43, 135], genetics [321], epidemiology [242, 250] and finance [214]) to reconstruct interaction networks on which network science tools can then be applied.

Interestingly, dynamics prediction and network reconstruction are usually considered separately, even though they are related to one another. The emergent field of the network neuroscience [25, 289] is perhaps the most actively using both notions : Network reconstruction for building brain connectomics from functional time series, then dynamics prediction for inferring various brain disorders from these connectomes [92, 310]. Recent theoretical works have also taken advantage of these notions to show that dynamics hardly depend on the structure. In Ref. [251], it was shown that time series generated by a deterministic dynamics evolving on a specific graph can be accurately predicted by a broad range of other graphs. These findings highlight how poor our intuition can be with regard to the relationship between predictability and reconstructability. Furthermore, recent breakthroughs in deep learning on graphs have benefited from proxy network substrates to enhance the predictive power of their models [344, 348], with applications in epidemiology [210], and pharmaceutics [96, 349]. However, the use of graph neural networks and those proxy network substrates is only supported by numerical evidence and lacks a rigorous theoretical justification. As a result, their enhanced predictability remains to be fully corroborated. There is therefore a need for a solid, theoretical foundation of reconstructability, predictability and their relationship in networked systems.

In this work, we establish a rigorous framework that lays such a foundation based on information theory. Information theory has been regularly applied to networks and dynamics in the past. In network science, it has been used to characterize random graph ensembles [7, 8, 31]—e.g. the configuration model [9, 145] and stochastic block models [237, 340]—, to develop network null models [57] and to perform community detection [239, 240]. In stochastic dynamical systems, information-theoretical measures have been proposed to quantify their predictability [76, 102, 158, 161, 247, 253, 273, 286], complexity [69, 88] and causal emergence [259]. In statistical mechanics, information transmission has been shown to reach a maximum value near the critical point of spin systems in equilibrium [122, 194].

Our objective is to combine these ideas into a single framework, motivated by recent works involving spin dynamics on lattices [22, 198] and deterministic dynamics [251]. Our contributions are fourfold. First, we use mutual information between structure and dynamics as

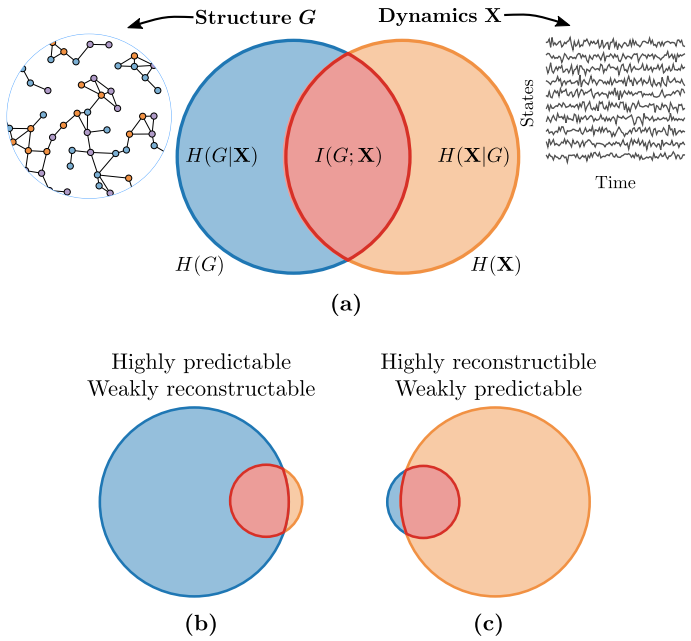


FIGURE 5.1 – **Information diagram of dynamics on random graphs**. (a) Areas represent amounts of information : The entropies related to G are shown on the left in blue and those related to X are on the right in orange. Mutual information—the red intersection of X and G—corresponds to the information shared by both G and X. (b) The highly predictable / weakly reconstructable scenario, where $H(G) \gg H(X)$ meaning that $I(X;G)$ contains most of the information related to the dynamics, but only a small fraction of the information related to the graph. (c) The reverse scenario, i.e., highly reconstructable / weakly predictable, where $H(X) \gg H(G)$ meaning that $I(X;G)$ contains most of the information related to the graph, but only a small fraction of the information related to the dynamics..

a foundation for our general framework to quantify the structure-function relationship in complex systems. Second, this codependence naturally leads to the definition of measures of predictability and reconstructability. Doing so allows us to conceptually unify prediction and reconstruction problems, i.e., two classes of problems that are usually treated separately. Third, we design efficient numerical techniques for evaluating these measures on large systems. Finally, we identify a new phenomenon—a duality—where our prediction and reconstruction capabilities can vary in opposite directions. These findings further our understanding of the complexity of modeling networked complex systems, such as the brain, where both prediction and reconstruction techniques play critical roles.

5.5 Results

5.5.1 Information theory of dynamics on random graphs

Let us consider a random graph G whose support, $\mathcal{G}$, consists in the set of all graphs of N vertices, each of which having its respective non-zero probability $P(G = g)$ with $g \in \mathcal{G}$. In our

framework, $P(G)$ can be any graph distribution and reflects, from a Bayesian perspective, our prior knowledge on the structure of the system. We also consider a general discrete-time stochastic process (also called a dynamics hereafter) with T time steps evolving on a realization of G and representing the possible states of the system. More precisely, we denote $P(X|G)$ the probability of a random and discrete-state time series $X = (X_{i,t})_{(i,t)\in[N]\times[T]}$ conditioned on G, where $X_{i,t}$ is the random state, with discrete support Ω, of vertex $i \in [N]$ at time $t \in [T]$. We stress that X is at this point any stochastic process be it Markovian or not. The initial condition of the process is $X_1 = (X_{i,1})_{i\in[N]}$. While we only exposed our framework in terms of discrete-time and discrete-state processes, it can be used for continuous-state deterministic dynamics (see Appendix 5.8.4) and in principle, it can also be generalized to continuous-state stochastic processes by considering a probability density function $\rho(X|G)$.

Together, X and G form a Bayesian chain $G \to X$, where the arrow indicates conditional dependence [83]. From this chain, we are interested in the mutual information between X and G—denoted $I(X;G)$—which is a symmetric measure that quantifies the codependence between the dynamics X and the structure G [65], where $I(X;G) = 0$ when they are independent. It is equivalently given by

$$I(X;G) = H(X) - H(X|G) \tag{5.1a}$$
$$= H(G) - H(G|X)\,, \tag{5.1b}$$

where $H(G) = -\mathbb{E}[\log P(G)]$ and $H(X) = -\mathbb{E}[\log P(X)]$ are respectively the marginal entropies of G and X, and $H(G|X) = -\mathbb{E}[\log P(G|X)]$ and $H(X|G) = -\mathbb{E}[\log P(X|G)]$ are their corresponding conditional entropies. In the previous equations, the marginal distribution for X, the evidence, is defined as $P(X) = \sum_{g\in\mathcal{G}} P(G = g)P(X|G = g)$, and the posterior is obtained from Bayes' theorem as $P(G|X) = P(G)P(X|G)/P(X)$, using the given graph prior $P(G)$ and the dynamics likelihood $P(X|G)$. $I(X;G)$ is a non-negative measure bounded by $0 \leq I(X;G) \leq \min\{H(G), H(X)\}$. Figure 5.1(a) provides an illustration of Eq. (5.1) in terms of information diagrams.

The measures presented in Eq. (5.1) and above can all be interpreted in the context of information theory. Information is generally measured in bits which in turn is interpreted as a minimal number of binary—i.e., yes/no—questions needed to convey it. While entropy measures the uncertainty of random variables like X and G, i.e., the minimal number of bits of information needed to determine their value, mutual information represents the reduction in uncertainty about one variable when the other is known. The fact that it is symmetric means that this reduction goes both ways : The reduction in the dynamics uncertainty when the structure is known is equal to that of the structure when the dynamics is known. Hence, mutual information measures the amount of information shared by both X and G.

As an illustration, let us consider the physical example of a spin system that depends on G through a coupling parameter $J \geq 0$, where the spins are more (large J) or less (small J)

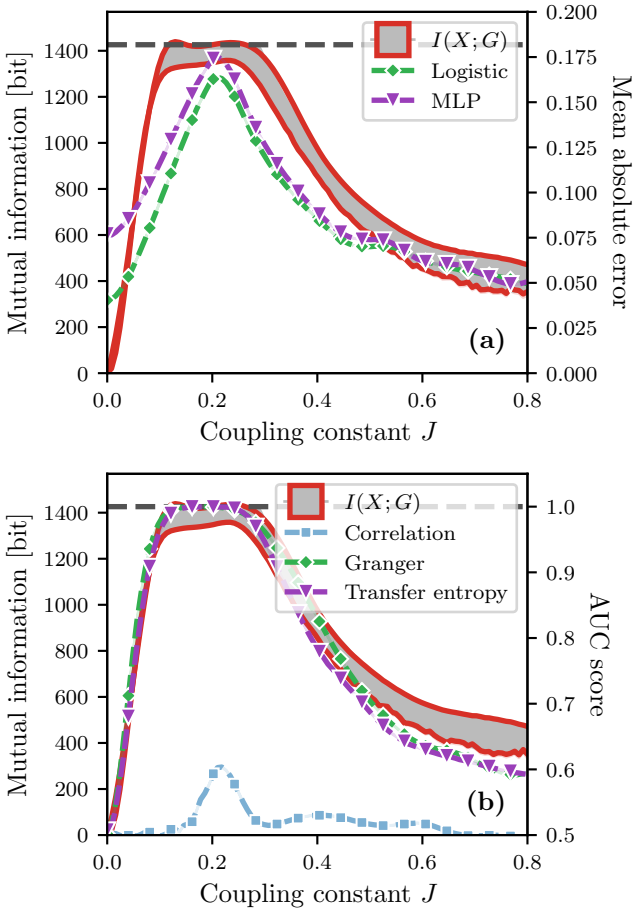


FIGURE 5.2 – **Comparison between the mutual information and algorithm performance measures** : (a) prediction algorithms and (b) reconstruction algorithms. This comparison is performed with time series of length $T = 100$ generated with the Glauber dynamics evolving on Erdős-Rényi graphs with $N = 100$ nodes and $E = 250$ edges, for different coupling constants J. Panel (a) shows the mean absolute error between the true transition probabilities used in $P(X|G)$ and the ones predicted by different graph-independent models : a logistic regression (green diamonds) and a multilayer perceptron (MLP, purple triangles). Panel (b) shows the average area under the curve (AUC) of the receiver operating characteristic (ROC) curve for different reconstruction algorithms : the correlation matrix method [160] (light blue squares), the Granger causality method [274] (green diamonds) and the transfer entropy method [277] (purple triangles). In both panels, we use two axes to represent $I(X;G)$ (left axis), denoted by the grey area bounded by the two biased estimators (red lines, see Section 5.7.5), and the performance measures (right axis) ; the maximum of $I(X;G)$ is shown with the horizontal dashed line. See Section 5.7.2 for further detail.

likely to align with their first neighbors in G. At $J = 0$, the spins are completely uncorrelated and flip with probability $\frac{1}{2}$. In this case, $H(X|G) = NT$ bits, corresponding to the maximum entropy of X : We need precisely one binary question for each spin at each time for a given structure G—e.g., is the spin of vertex i at time t up ? When $J > 0$, correlation is introduced between connected spins. As a result, a single question about the spin of vertex i at time t can provide additional information about the spins of other vertices at other times and thus, $H(X|G) < NT$. The interpretation of $H(X)$ is analogous to that of $H(X|G)$, as it measures the number of binary questions needed to determine X when the graph is unknown. From this perspective, the mutual information $I(X;G)$, as expressed by the difference between $H(X)$ and $H(X|G)$, is the reduction in the number of questions needed to predict X ensuing from the knowledge of G. Hence, $I(X;G)$ measures to which extent the knowledge of the graph G improves our ability to forecast X, i.e. its temporal predictability.

Similar observations can be made from the structural perspective. Suppose that X is the spin dynamics mentioned previously and G is a random graph, where each edge exists independently with probability p. This yields $H(G) = -\binom{N}{2}[p\log p + (1-p)\log(1-p)]$, where $\binom{N}{2}$ is the total number of possible undirected edges. When $p = \frac{1}{2}$, we have $H(G) = \binom{N}{2}$ bits, which is again the maximum entropy of G. We therefore need precisely one binary question for each of the $\binom{N}{2}$ edges in the graph—e.g., is there an edge between i and j?—to completely determine its value. When the dynamics X is known, $H(G|X)$ is interpreted similarly to $H(G)$, but also takes into account the observation of the spins X which introduces correlation between the edges of G. As a result, each bit can provide information about more than one edge, even in the case $p = \frac{1}{2}$ where we a priori need one bit per possible edge to fully reconstruct G. Consequently, the knowledge of X reduces uncertainty about G (i.e., $H(G|X) \leq H(G)$, see [65, Theorem 2.6.5]), and therefore allows for its reconstruction ; $I(X;G)$ thus measures the reconstructability of G, i.e., the extent to which information about G can be revealed from X.

In practice, $I(X;G)$ can be used to explain the performance of both prediction and reconstruction algorithms (see Section 5.7.2 for further detail). From the prediction perspective, it quantifies the sensitivity of the time series to the structure of interactions prescribed by G, i.e., the gain in predictability of including G for extrapolating of X. This can be measured by comparing the true transition probabilities of the process X as given by the conditional model $P(X|G)$, with those predicted by models that do not include G in their predictions. This experiment was performed in Ref. [251] for deterministic dynamics on graphs, to show that high prediction accuracy of time series can sometimes be achieved without the knowledge of the true graph. In Fig. 5.2(a), we use the mean absolute error—same measure as in Ref. [251]—to perform the comparison. In turn, we associate high predictive capabilities of the true conditional model where the error with the graph-independent model is high. Likewise, $I(X;G)$ provides strong insights on the reconstruction accuracy of algorithms such

as the transfer entropy method [277] [Fig. 5.2(b)]. By interpreting the reconstruction problem as a binary classification, we are allowed to quantify the reconstruction accuracy with the area under the curve (AUC) of the receiver operating characteristic (ROC) curve. In all cases, $I(X;G)$ peaks in the same coupling interval as the different reconstruction methods even if the two measures are a priori different.

The mutual information $I(X;G)$ is therefore both a measure of predictability and reconstructability, thereby unifying these two concepts. We say that a system is perfectly predictable when the mutual information contains all the information about X, that is when $I(X;G) = H(X)$ [see Fig. 5.1(b)]. Likewise, we say that it is perfectly reconstructable when $I(X;G) = H(G)$ [see Fig. 5.1(c)]. Consequently, whenever $I(X;G) > 0$, we expect the system to be predictable and reconstructable to a certain degree. Otherwise, when $I(X;G) = 0$, the system is said both unpredictable and unreconstructable. Yet, $I(X;G)$ by itself is hardly comparable from one system to another. Indeed, a specific value of $I(X;G)$ may correspond to opposing scenarios when it comes to predictability and reconstructability, as shown in Fig. 5.1(b-c). Thus, it is more convenient to use normalized quantities such as the uncertainty coefficients

$$U(X|G) = \frac{I(X;G)}{H(X)}, \tag{5.2a}$$

$$U(G|X) = \frac{I(X;G)}{H(G)}, \tag{5.2b}$$

which are bounded between 0 and 1. Contrary to $I(X;G)$, $U(X|G)$ and $U(G|X)$ represent relative amount of information. For instance, $U(G|X) = 1$ implies that $I(X;G) = H(G)$, which in principle means that perfect reconstruction can be achieved as all the information of G is contained in X. Likewise, $U(X|G) = 1$ means that $I(X;G) = H(X)$, which indicates that all the information in X is determined by G : a perfectly accurate prediction of X can be made with G alone. This maximum value is guaranteed when X is deterministic and there is only one initial condition (see Appendix 5.8.4). Having $I(X;G) = 0$ implies that $U(X|G) = U(G|X) = 0$, which again means that G and X are independent. Any value in-between of $U(X|G)$ and $U(G|X)$ represent different degrees of predictability and reconstructability, respectively.

Section 5.5.3 will present simple concrete examples to provide a better intuition about these concepts. Before we get to these examples, we investigate the influence of the knowledge of the past of X over the relationship between its future and its structure, as measured through reconstructability and predictability.

5.5.2 Past-dependent mutual information

It is often the case that predictability measures the sensitivity to the initial conditions of a process X. For instance, Refs. [76, 87, 108, 158] used different versions of the mutual information

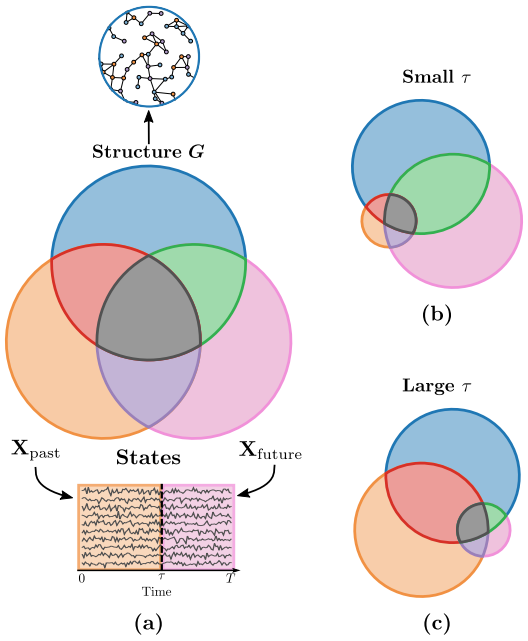


FIGURE 5.3 – **Information diagrams for the past-dependent information measures**. On panel (a), we show the information diagram of the random variable triplet (X_{past}, X_{future}, G), where X_{past} represents the past states, X_{future}, the future states and G, the structure of the system. The quantities of interest are $I(X_{\text{future}}; G|X_{\text{past}})$ indicated by the green set, $H(X_{\text{future}}|X_{\text{past}})$ shown by the union of the pink and green sets and $H(G|X_{\text{past}})$, represented by the union of the blue and green sets. Panels (b) and (c) show two extreme scenarios where the length of the past τ is small and large, which illustrates how the different information measures change with τ.

between X_1 and X as a direct measure of predictability. Then, a system is more predictable if the past allows to better predict the future. In this spirit, we generalize our framework in such a way that the mutual information between the process X and its structure G includes some information about the past of X.

We define X_{past} as the past of X and X_{future} as its future, such that $X = (X_{\text{past}}, X_{\text{future}})$, see Fig. 5.3(a). We define τ as the length of X_{past} and $T - \tau$ as the length of the future X_{future}. Our measure of interest in this case is $I(X_{\text{future}}; G|X_{\text{past}})$, and it is equal to

$$I(X_{\text{future}}; G|X_{\text{past}}) = I(X; G) - I(X_{\text{past}}; G)\,, \tag{5.3}$$

which is a conditional mutual information—the green intersection in Fig. 5.3(a). In turn, a small τ includes less contribution to the observed past, which leads to a scenario increasingly similar to that presented in Sec. 5.5.1 as shown by Fig. 5.3(b). As τ gets larger, more contribution is left to X_{past} resulting in a smaller $I(X_{\text{future}}; G|X_{\text{past}})$, even though the total mutual information $I(X; G)$—the union of the red, green and grey sets—is large (see Fig. 5.3(c)). Similarly to Sec. 5.5.1, we then define the partial uncertainty coefficients, bounded between 0

and 1 :

$$U(X_{\text{future}}|G;X_{\text{past}}) = \frac{I(X_{\text{future}};G|X_{\text{past}})}{H(X_{\text{future}}|X_{\text{past}})}\,, \tag{5.4a}$$

$$U(G|X_{\text{future}};X_{\text{past}}) = \frac{I(X_{\text{future}};G|X_{\text{past}})}{H(G|X_{\text{past}})}\,, \tag{5.4b}$$

measuring the partial predictability of X_{future} from G and partial reconstructability of G given X_{future}, respectively. The above quantities can be expressed in terms of previously visited ones. For instance, in Eq. (5.3), $I(X;G)$ and $I(X_{\text{past}};G)$ can be expressed using Eq. (5.1). Likewise, the normalizing factor $H(X_{\text{future}}|X_{\text{past}})$ is expressed in terms of state entropies using the joint entropy $H(X) = H(X_{\text{future}}X_{\text{past}})$, i.e., $H(X_{\text{future}}|X_{\text{past}}) = H(X) - H(X_{\text{past}})$. And finally, $H(G|X_{\text{past}})$ is evaluated similarly to $H(G|X)$.

Whereas the interpretation of the partial uncertainty coefficients is analogous to those presented in the previous section, they nevertheless measure conceptually different quantities. Indeed, by using $I(X_{\text{future}};G|X_{\text{past}})$, it is implied that the information about the past has been removed from the total mutual information between X and G. As a result, the partial predictability $U(X_{\text{future}}|G;X_{\text{past}})$ measures the gain in predictability over X_{future} when including G in the prediction, compared to a model which only uses the past X_{past}. Additionally, the removed information likely includes some information about G, since $I(X_{\text{past}};G) \geq 0$. Hence, the partial reconstructability, as defined by $U(G|X_{\text{future}};X_{\text{past}})$, measures the reconstructability of the remaining information about G when observing X_{future}, i.e., information which has not been unveiled from the observation of X_{past}.

In essence, for some $\xi > 0$, the case $\tau = 1$ with $T = \xi + 1$ is similar to the case $\tau = T - \xi$ with $T > \xi$ since X_{future} have the same length ξ in both cases. From a reconstruction perspective, they quantify the reconstructability of G from a process with ξ time steps. However, the reconstructed information is quite different in both cases, since with $\tau = 1$ and $T = \xi + 1$ no prior information is given—assuming that the initial conditions X_1 are independent from G—, while a lot of information has already been processed when $\tau = T - \xi$. Furthermore, increasing τ draws our attention away from the actual relationship between X and G of interest, since this relationship should exclude all information about X_{past}. For this reason, we will mostly focus on the case $\tau = 1$ in the remainder of the paper.

5.5.3 Simple example

The interpretation of reconstructability and predictability in terms of $U(G|X)$ and $U(X|G)$ can be grasp more firmly through an elementary example. We consider a system where only two graphs are possible, namely g_1 and g_2, such that $P(G = g_1) = p$ and $P(G = g_2) = 1 - p$ (see Fig. 5.4(a)). The entropy of G is therefore $H(G) = \mathcal{H}(p)$, where $\mathcal{H}(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy. These two graphs can generate together three outcomes for X, i.e., X_1, X_2 or X_3. The graph g_1 generates X_1 and X_2 with probabilities r and $1 - r$

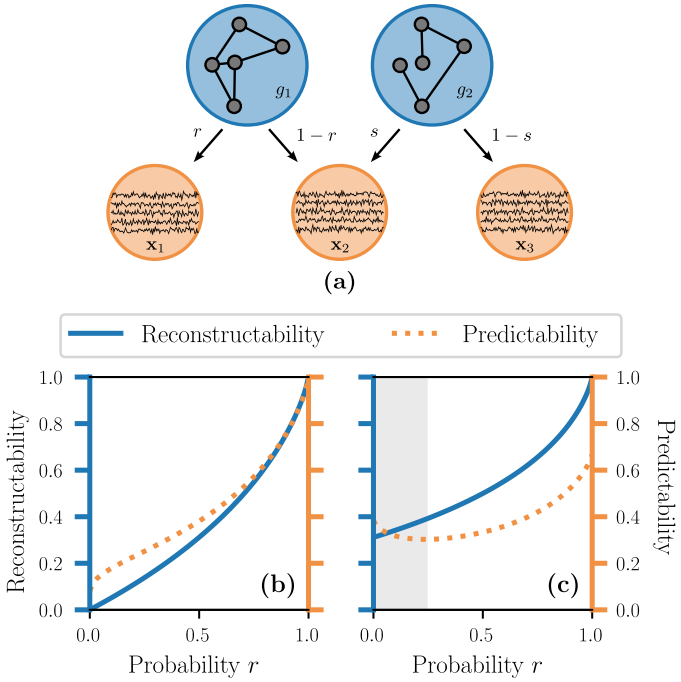


FIGURE 5.4 – **Example with two graphs and three time series,** illustrated in (a). Panels (b) and (c) show the reconstructability $U(G|X)$ (solid blue line) and predictability $U(X|G)$ (dashed orange line) when $s = 1$ and $s = \frac{1}{2}$, respectively, where we also fixed $p = \frac{1}{2}$. The shaded area in (c) shows the region where $U(G|X)$ and $U(X|G)$ vary in opposite directions.

respectively. Likewise, g_2 generates X_2 and X_3 with respective probabilities s and $1 - s$. As we can see, X_1 can only be generated by g_1 and X_3 can only be the outcome of g_2, while X_2 can be generated by both graphs.

We now focus on the scenario with $s = 0$—the general expressions for $I(X;G)$ and the other entropies are obtained in Appendix 5.8.2 of the Supplementary Information. In this case, only g_1 can generate X_1 and X_2, while g_2 can only generate X_3. Therefore, we have perfect reconstructability of either graphs, meaning $U(G|X) = 1$ for any p and r, since the outcome of X tells us immediately which graph generated it. However, X is imperfectly predictable from G since $U(X|G) = \frac{\mathcal{H}(p)}{\mathcal{H}(p)+p\mathcal{H}(r)} < 1$ when $0 < p, r < 1$, even though X_3 can be perfectly predicted from g_2 as it is its only possible outcome. The remaining entropy, i.e. the second term of the denominator, $p\mathcal{H}(r)$, corresponds to the uncertainty related to whether g_1 generates X_1 or X_2.

When $s = 1$, the system is both partially predictable and reconstructible, with $U(X|G) = 1 - \frac{p\mathcal{H}(r)}{\mathcal{H}(pr)} < 1$ and $U(G|X) = \frac{\mathcal{H}(pr)-p\mathcal{H}(r)}{\mathcal{H}(p)} < 1$ for all $0 < p, r < 1$. Both g_1 and g_2 can generate X_2, but the probability that g_1 generates X_2 decreases with r. This results in a gradual increase of predictability and reconstructability as r approaches 1, where the systems tends to a one-to-one mapping between the outcomes of G and X.

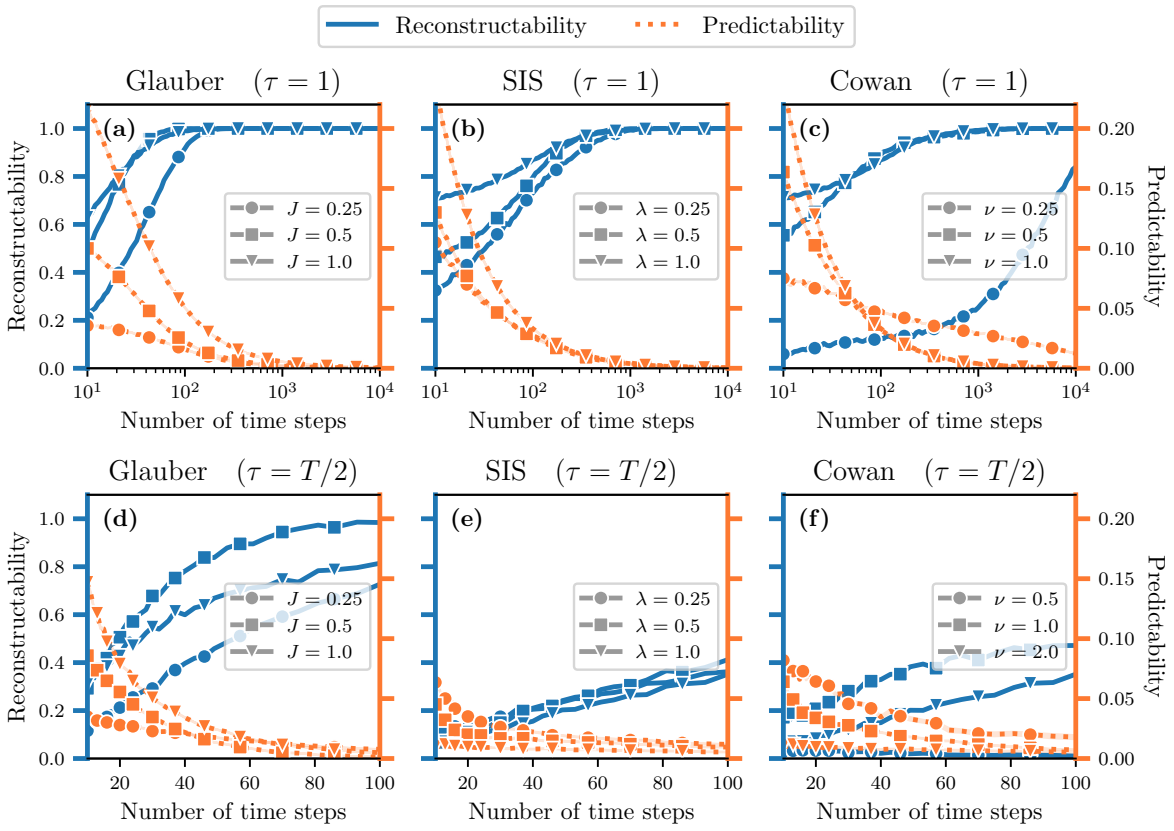


FIGURE 5.5 – **T-duality in binary dynamics evolving on small Erdős-Rényi random graphs** : (a-d) Glauber dynamics, (b-e) SIS dynamics and (c-f) Cowan dynamics. Each panel shows the reconstructability $U(G|X) \in [0,1]$ (blue) and the predictability coefficient $U(X|G) \in [0,1]$ (orange) as a function of the number of time steps T. We used graphs of $N = 5$ vertices and $E = 5$ edges, meaning an average degree of $\langle k \rangle = 2$; we fixed $\tau = 1$ in the top row, and $\tau = T/2$ in the bottom row. Each symbol corresponds to the average value measured over 1000 samples. We also show different values of the coupling parameters—normalized by the average degree—using different symbols : (a) $J\langle k \rangle \in \{\frac{1}{2}, 1, 2\}$ for Glauber, (b) $\lambda\langle k \rangle \in \{1, 2, 4\}$ for SIS and (c) $\nu\langle k \rangle \in \{1, 2, 4\}$ for Cowan.

The intermediate cases when $0 < s < 1$ are also interesting, because they give rise to an interval in r where the system becomes less predictable but more reconstructable as r increases, as highlighted by the grey area in Fig. 5.4(c). This happens because, as r increases for a fixed s, the growth of entropy of X dominates $I(X;G)$, resulting in the dual behavior of $U(X|G)$ and $U(G|X)$.

5.5.4 θ-duality between predictability and reconstructability

Predictability and reconstructability in dynamics on random graphs offer two perspectives of the same information shared by G and X—two sides of the same coin. However, as we have previously seen with simple examples, predictability and reconstructability do not ne-

cessarily go hand in hand even though they are related : An increasing $U(G|X)$ according to some parameter θ of the system does not necessarily implies an increase of $U(X|G)$ and vice-versa. Furthermore, a high value of $U(G|X)$ is not tied to a high value of $U(X|G)$, and conversely, as illustrated in Figs. 5.1(b)–(c). Indeed, $U(G|X)$ and $U(X|G)$ can take opposing values, depending on $H(G)$ and $H(X)$, for a same value of $I(X;G)$. This phenomenon can also be observed in the performance of prediction and reconstruction (see Appendix 5.8.1). In the literature, a hint of the existence of such dual behavior was recently corroborated in Ref. [251] for continuous-state deterministic dynamics. The authors showed that high prediction accuracy can be achieved with graphs reconstructed from the very time series they want to predict, even if they are different from the original graph that generated the time series. This phenomenon can be understood through our framework (see Appendix 5.8.4) and we now devote the rest of the section to precisely define and characterize the somewhat counterintuitive phenomenon of duality.

We identify a duality when $U(X|G)$ and $U(G|X)$ vary in opposite directions when a parameter, say θ, is changed. More specifically, we say that they are dual with respect to θ, or θ-dual, in an interval Θ if and only if the signs of their derivative with respect to θ are different for every $\theta^* \in \Theta$:

$$\left[\frac{\partial U(G|X)}{\partial \theta}\frac{\partial U(X|G)}{\partial \theta}\right]_{\theta=\theta^*} < 0\,. \tag{5.5}$$

This criterion formally relies on the existence of regions Θ where the variations of $U(G|X)$ and $U(X|G)$ with respect to θ are opposite, regardless of their amplitude (see also Section 5.7.3). We use this criterion to relate the existence of extrema of $U(G|X)$ and $U(X|G)$ with that of regions of θ-duality (see Lemma 5.1 in Section 5.7.3).

With our intuition being established from simple examples and with our precise definition, we are finally ready to state one of the main results of the paper. Recalling that T is the length of the process X, we prove that reconstructability and predictability are T-dual for a vast class of Markov chains.

Theorem 5.1. *Let $X = (X_1, X_2, \cdots, X_T)$ be a Markov chain of length T whose transition probabilities are conditional to some discrete random variable G that is independent of T and such that $H(X_{t+1}|X_t) > 0$ for all $t \in [T-1]$ (i.e., X is non-deterministic). Moreover, suppose that the state spaces of X and G are finite, and that X has a finite nonzero entropy rate and that G has a nonzero entropy. Then there exists a positive constant ϕ such that the uncertainty coefficients $U(G|X)$ and $U(X|G)$ are T-dual for all $T \geq \phi$.*

The proof of this theorem is in Section 5.7.3. It is a consequence of the fact that the mutual information is strictly increasing with T—and so is $U(G|X)$ since $H(G)$ is independent of T—whenever the entropy rate of X is positive. As a result, $U(G|X)$—and numerator, $I(X;G)$—stagnates at some point in T, while $U(X|G)$ keeps decreasing because its denominator in-

Dynamics	$\alpha(n,m)$	$\beta(n,m)$	Coupling
Glauber [112]	$\sigma\Big(2J(n-m)\Big)$	$\sigma\Big(2J(m-n)\Big)$	J
SIS [231]	$1-\Big(1-\frac{\lambda}{\beta}\Big)^m$	β	λ
Cowan [66]	$\sigma\Big(a(\nu m-\mu)\Big)$	β	ν

TABLE 5.2 – Activation and deactivation probability functions, $\alpha(n,m)$ and $\beta(n,m)$, respectively, for the binary dynamics considered in this study, where n corresponds to the number of inactive neighbors whose states are 0, and m corresponds to the number of active neighbors whose states are 1. We define $\sigma(x) = [\exp(-x)+1]^{-1}$ as the logistic function. Some of these parameters are fixed throughout the paper : $\beta = 0.5$ for SIS and Cowan, and $a = 7$ and $\mu = 1$ for Cowan. The coupling parameters (J for Glauber, λ for SIS and ν for Cowan) are specified in each figure. Also, to prevent the SIS dynamics from being completely inactive, we allow the inactive vertices to spontaneously activate with probability $\epsilon = 10^{-3}$ [311].

creases in an asymptotically linear manner with T. We refer to this opposing behavior as a duality between $U(G|X)$ and $U(X|G)$ with respect to T, or a T-duality for short[2]. When the entropy rate is not well-defined, like for non-stationary processes, the universality of the T-duality might not hold, while it remains possible to observe it in localized intervals of T.

Figure 5.5 illustrates the universality of the T-duality using the special case of binary Markov chains (i.e., $\Omega = \{0,1\}$, see Section 5.7.1). These systems are parametrized by their activation ($0 \to 1$) and deactivation ($1 \to 0$) probability functions, denoted $\alpha(n_{i,t}, m_{i,t})$ and $\beta(n_{i,t}, m_{i,t})$, respectively. In general, the activation and deactivation functions depends solely on $n_{i,t}$ and $m_{i,t}$, i.e., the number of active and inactive neighbors of vertex i at time t. We present multiple examples of binary Markov processes with different origins in Table 5.2 : The Glauber dynamics, the Susceptible-Infectious-Susceptible (SIS) dynamics and the Cowan dynamics.

The aforementioned Glauber dynamics [112], which have been used to describe the time-reversible evolution of magnetic spins aligning in a crystal, have been tremendously studied because of its critical behavior and its phase transition. Its stationary distribution is given by the Ising model which has found many applications in condensed-matter physics [200] and statistical machine learning [33, 83], to name a few. The Susceptible-Infectious-Susceptible (SIS) dynamics is a canonical model in network epidemiology [231] often used for modeling influenza-like disease [11], where periods of immunity after recovery are short. In this mo-

2. Not to be confused with target space duality in string theory [111].

del, susceptible (or inactive) vertices get infected by each of their infected (active) first neighbors, with a constant transmission probability, and recover from the disease with a constant recovery probability. The simplicity of the SIS model has allowed for deep mathematical analysis of its absorbing-state phase transition [89, 232, 297]. Finally, the Cowan dynamics [66] has been proposed to model the neuronal activity in the brain. In this model, quiescent neurons fire if their input current, coming from their firing neighbors, is above a given threshold. Its mean-field approximation [228] reduces to the Wilson-Cowan dynamics [328], one of the most influential models in neuroscience [78]. For each model, we can identify an inactive state—down, susceptible or quiescent—and an active one—up, infectious or firing. The corresponding activation and deactivation probabilities are given in Table 5.2.

Figure 5.5 numerically supports Theorem 5.1 and clearly illustrates the T-duality for each dynamics, with different values of their parameters and different past length τ. We used the Erdős-Rényi model as the random graph on which these dynamics evolve. The support $\mathcal{G}$ is the set of all simple graphs of N vertices with E edges, and

$$P(G) = \left(\binom{N}{2} \atop E \right)^{-1} . \tag{5.6}$$

Note that, in this example, we consider the well known Erdős-Rényi model for simplicity (Eq. (5.6)). Furthermore, we considered very small graphs of size $N = 5$, because the exact evaluation of $I(X;G)$ is computationally intractable. For larger systems, biased estimators can be designed to bound $I(X;G)$ as we show in Section 5.7.5. We demonstrate the flexibility of our framework with regards to the random graph models by using more sophisticated and data-driven graph models in the following Section 5.5.5.

The T-duality persists for the past-dependent measures presented in Section 5.5.2, as illustrated by the bottom row of Fig. 5.5, for $\tau = T/2$. However, note that for sufficiently large τ, the duality seems to disappear. We refer to Appendix 5.8.8 for further detail. One can only wonder how many different kinds of parameters can lead to θ-dualities. Maybe some may control the general behavior of the dynamics, and others some aspect of the system structure which, in turn, may also impact the dynamics. In the next section, we investigate those that are related to critical phenomena in complex systems.

5.5.5 Duality and criticality

Despite their different nature and range of applications, the models presented in Table 5.2 share several properties of interest. For instance, each model has a coupling parameter that controls the influence of first neighbors' states on the transition probabilities. They also all feature a phase transition in the infinite size limit whose position is determined by the coupling parameter (see Appendix 5.8.10). We now investigate the influence of criticality over the existence of θ-dualities, where θ is a coupling parameter.

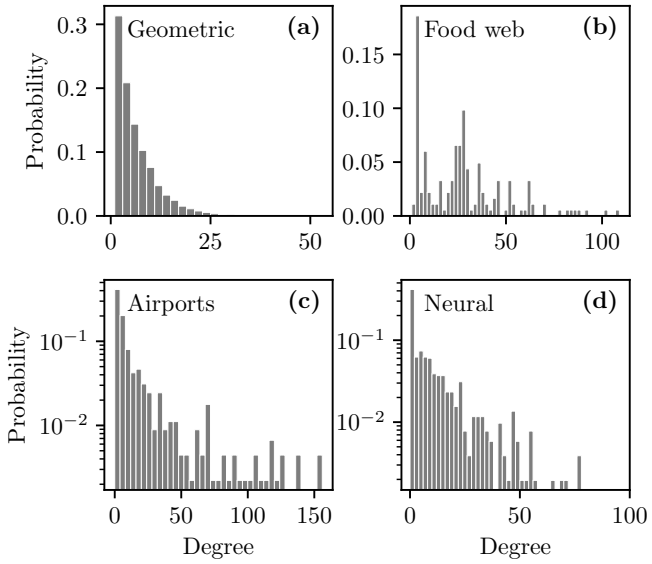


FIGURE 5.6 – **Degree distributions of the graphs used in Fig. 5.7** : (a) graphs with geometric degree distribution $p(k) = (1-p)p^k$ where $p = 5/6$, with $N = 1000$ nodes and $E = 2500$ edges, (b) Little Rock Lake food web [192], (c) European airline route network [51], (d) C. Elegans neural network [62]. See Section 5.7.7 for further details about the graphs.

For the Glauber dynamics, this parameter is the coupling constant J, which dictates the reduction (increase) in the total energy of a spin configuration when two neighboring spins are parallel (antiparallel). The Glauber dynamics features a continuous phase transition at a critical point J_c between a disordered and an ordered phase, where for $J < J_c$ the spins are disordered resulting in a vanishing magnetization, and for which this magnetization is non-zero when $J > J_c$. For the SIS dynamics, it is the transmission rate λ that acts as a coupling parameter. Like the Glauber dynamics, the SIS dynamics possesses a continuous phase transition where, when $\lambda < \lambda_c$, the system reaches an absorbing—or inactive—state from which it cannot escape, and an active state, when $\lambda > \lambda_c$, where a non-zero fraction of the vertices remain active over time [3]. The Cowan dynamics can both feature a continuous or a first-order phase transition between an inactive and an active phase depending on the value of slope a, for which the coupling parameter is ν, i.e., the potential gain for each firing neighbors. The continuous and first-order phase transitions of the Cowan dynamics are quite different in that the latter is characterized by two thresholds, namely the forward and backward thresholds $\nu_c^b < \nu_c^f$, respectively (see Appendix 5.8.10). Hence, the Cowan dynamics has a first-order phase transition that exhibits a bistable region $\nu \in (\nu_c^b, \nu_c^f)$, where both the inactive and active phases are reachable depending on the initial conditions.

3. It is not strictly accurate to say that our considered version of the SIS dynamics reaches a true absorbing state, since we allow for self-infection ϵ which allows it to escape the completely inactive state. Instead, it reaches a metastable state where most of the vertices are asymptotically inactive. However, it can be shown that the two phase transitions are quite similar for small ϵ [311].

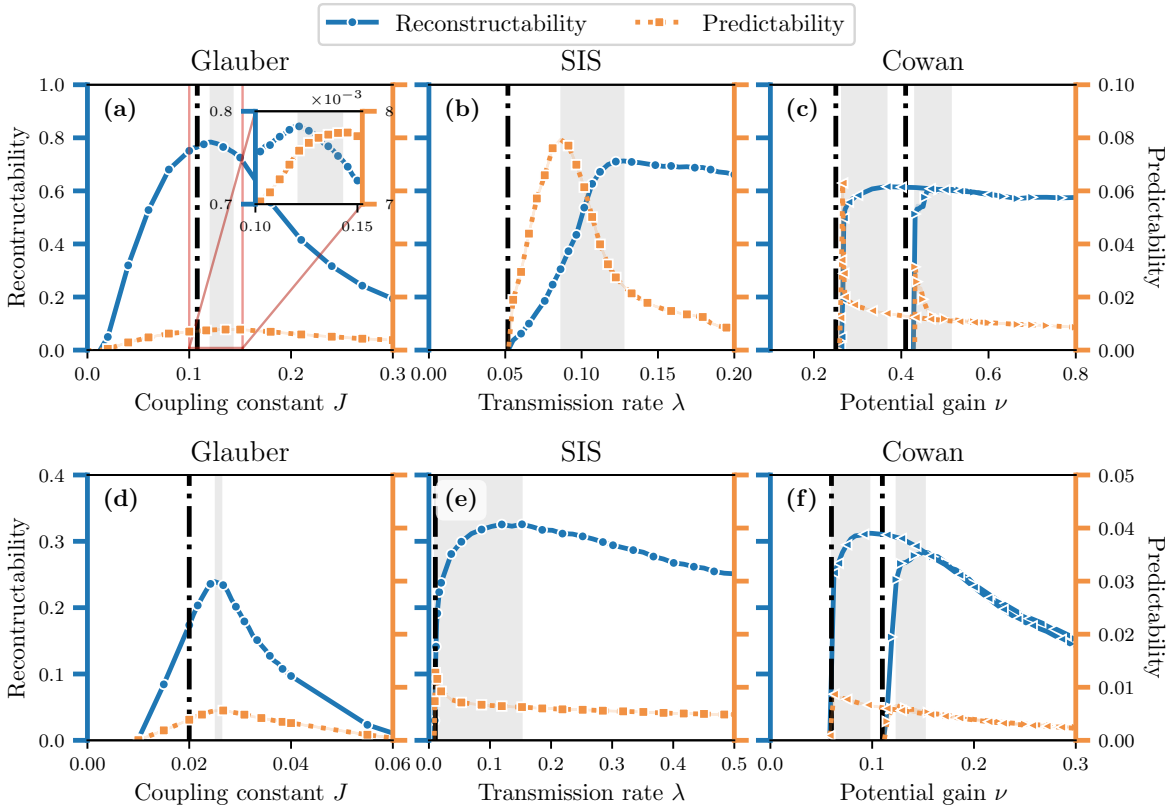


FIGURE 5.7 – **Dynamics evolving on configuration model graphs** : (a,d) Glauber dynamics, (b,e) SIS dynamics and (c,f) Cowan dynamics. We used the configuration model (see Eq. (5.7)) to generate graphs of varying sizes and degree distributions. In the top row, we generated graphs with geometric degree distribution of size $N = 1000$ and with $E = 2500$ edges (see Fig. 5.6(a)). In the bottom row, we used the degree distribution of real networks : (d) Little Rock Lake food web [192], (e) European airline route network [51], (f) C. Elegans neural network [62]. The parameters used to generate the time series are the same in the top and bottom panels (see Table 5.2), except in (f) the time series length is $T = 5000$ while in the others $T = 2000$. Similar to Fig. 5.5, $U(G|X)$ is shown in blue (left axis) and $U(X|G)$ is shown in orange (right axis). We show, for each dynamics, the uncertainty coefficients as a function of the coupling parameter : J for Glauber, λ for SIS and ν for Cowan. Each shaded area indicates a range of couplings over which duality was observed. The vertical dotted-dashed lines correspond to the phase transition thresholds of each dynamics, which are estimated from Monte Carlo simulations (see Appendix 5.8.10). For the Cowan dynamics, the forward and backward branches are shown with their corresponding thresholds and dual regions (see main text).

To account for the heterogeneous network structure observed in a wide range of complex systems [18], we simulate the dynamics on the configuration model, a random graph whose —potentially heterogeneous—degree sequence $\boldsymbol{k}$ is fixed and whose support $\mathcal{G}$ corresponds to the set of all loopy multigraphs of degree sequence $\boldsymbol{k}$. The probability of a graph g in this ensemble is

$$P(G = g) = \frac{(2E)!!}{(2E)!} \frac{\prod_i k_i!}{\prod_{i<j} a_{ij}! \prod_i a_{ii}!!}, \tag{5.7}$$

where a_{ij} counts the number of edges connecting vertices i and j in the multigraph g and $2E = \sum_i k_i$ is the number of half-edges in g. Like the Erdős-Rényi model, the configuration model fixes the number of edges, but also fixes the degree sequence $\boldsymbol{k} = (k_1, \cdots, k_N)$.

Figure 5.7 shows the predictability and reconstructability, as estimated by the MF estimator, of the three dynamics evolving on graphs drawn from the configuration model. The top row shows the results when using a synthetic degree sequence obtained from a geometric degree distribution, while for the bottom row, degree sequences from different real networks are used for each dynamics. These distributions are shown in Fig. 5.6. We used the Little Rock Lake food web [192] (as in Ref. [242]) jointly with the Glauber dynamics to simulate a simplification of the interaction between species. In the case of the SIS dynamics, we considered the European airline network [51] to mimic the spread of an epidemic. Finally, to simulate neural activity of the Cowan dynamics, we used the C. Elegans neural network [62].

First, the results of Fig. 5.7 show a meaningful comparison between the dynamics for different types of structures. For example, on the one hand, the Glauber dynamics is globally less predictable than the other two, since its predictability coefficient is overall smaller. In other words, the knowledge of a graph g provides less information about X in the Glauber dynamics in comparison with the others, relatively to the total amount of information needed to predict X. This is related to the time reversibility of the Glauber dynamics, which allows any vertex to transition from the inactive to the active state (and vice versa) with non-zero probability, at any time, effectively making the Glauber dynamics more random than the others—i.e. $H(X)$ is greater for Glauber than the other processes. On the other hand, the SIS and Cowan dynamics are shown as practically unpredictable and unreconstructable when their coupling parameter is below their respective critical point. This precisely occurs in the inactive phase, where no mutual information can be generated after a short time, when the system reaches the inactive state. By contrast, the Glauber dynamics does not reach an inactive state below its critical point, which explains the gradual increase in predictability and reconstructability in that region.

Several additional observations are worth making. All dynamics exhibit maxima for $U(X|G)$ and $U(G|X)$, which delineate a region of duality illustrated by the shaded areas (two for Cowan, that is one for each branch). These regions are close to, but systematically above, their respective phase transition thresholds, regardless of type of degree sequence. A simi-

lar phenomenon in spin dynamics on non-random lattices has been reported by previous works [22, 198], in which the information transmission rate between spins—a measure akin to $I(X;G)$—is maximized above the critical point. Our numerical results are consistent with theirs, and suggest that their findings regarding near-critical systems even apply beyond spin dynamics on fixed lattices, to other types of processes on more heterogeneous and random structures.

5.6 Discussion

In this work, we used information theory to characterize the structure-function relationship with mutual information. We showed how mutual information is a natural starting point to define both predictability and reconstructability in dynamics on networks, and even how it explains the performance accuracy of prediction and reconstruction algorithms. In turn, we demonstrated how prediction and reconstruction in complex systems are intrinsically related. Our approach is quite general, allowing the exploration of different configurations of dynamics on networks of the form $G \to X$, thus varying the nature of the process itself as well as the random graph on which it evolves. Our framework could be extended to adaptive systems [121, 153, 191, 272] where both X and G influence each other (i.e., $X \leftrightarrow G$). The relationship between X and G could also go the other way around : A system in which X generates a graph G (i.e., $X \to G$). Hyperbolic graphs [37, 163] fall into this category, where X represents a set of coordinates, and our framework could be extended to quantifying the feasibility of network geometry inference [38, 101, 229].

We exposed various examples where our measures can be computed analytically and found efficient ways to estimate them numerically when needed, thus allowing thorough investigation of large systems. More work on this front is required, however, since the evaluation of these estimators remains quite computationally costly. It would be worth investigating dimension reduction methods [171, 304, 305] and approximate master equations [113, 295], among others, for obtaining more efficient and reliable approximations of $I(X;G)$, $U(X|G)$ and $U(G|X)$.

Central to our findings is the peculiar discovery that predictability and reconstructability are not only related, but sometimes dual to one another. We found many examples of this duality in systems of increasing complexity, while we also emphasized that its universality is limited to certain circumstances. One of those circumstances occurs when we change the length of the processes, for which we mathematically proved the existence of duality. We also presented numerical evidence of duality near critical points in three different dynamics on real networks. These findings generalize and formalize—while being consistent with—previous works [22, 198] and suggest that the reconstructability-predictability duality with respect to order parameters is closely linked to the criticality in these systems.

From a practical perspective, the existence of such a θ-duality can be critical to network modeling applications, since it also suggests a predictability-reconstructability trade-off. On one hand, by choosing the parameter θ, we can minimize the uncertainty of the reconstructed structure, but this may result in a structure that is less informative regarding the dynamics. On the other hand, we can consider the reverse case, where the process is maximally influenced by the inferred structure, whose uncertainty is nevertheless not minimized. Analogous to the position-momentum duality in the Heisenberg uncertainty principle of quantum mechanics, the predictability-reconstructability duality must be accounted for in our network models if we are to disentangle complex systems.

5.7 Methods

5.7.1 Binary Markov chains on graphs

The models used throughout the paper are for the most part Markov chains $X = (X_1, X_2, ..., X_T)$, that are governed by a conditional probability $P(X|G)$ that can be factored as follows :

$$P(X|G) = P(X_1) \prod_{t=1}^{T-1} P\left(X_{t+1}|X_t, G\right) . \tag{5.8}$$

The probability $P(X_{t+1}|X_t, G)$ is the global transition probability from state X_t to state X_{t+1}, and $P(X_1)$ represents the probability distribution of the initial conditions, which is independent from G in our case. More specifically, we assume that X_i is a random binary vector of size N, and that the global transition probability can be factored in terms of local transition probabilities as follows :

$$\begin{aligned} P\left(X_{t+1}|X_t, G\right) = \prod_{i=1}^{N} \Big\{ & \left[\alpha(n_{i,t}, m_{i,t})\right]^{(1-X_{i,t})X_{i,t+1}} \left[1-\alpha(n_{i,t}, m_{i,t})\right]^{(1-X_{i,t})(1-X_{i,t+1})} \\ & \left[\beta(n_{i,t}, m_{i,t})\right]^{X_{i,t}(1-X_{i,t+1})} \left[1-\beta(n_{i,t}, m_{i,t})\right]^{X_{i,t}X_{i,t+1}} \Big\} . \end{aligned} \tag{5.9}$$

As mentioned in Section 5.5.4, the functions α and β corresponds to the activation and deactivation probabilities. In the general case, they dependent on the number of active neighbors m_i, and inactive neighbors n_i of a node i such that $m_i + n_i = k_i$ where k_i is the degree of this node.

5.7.2 Performance of prediction and reconstruction algorithms

To substanciate our claim about the interpretation of $I(X;G)$, we used different prediction and reconstruction algorithms and compared in Fig. 5.2 their performance with $I(X;G)$. In this section, we elaborate on this analysis.

Prediction algorithms

The prediction algorithms used in Fig. 5.2 correspond to Markov models that predict a transition—activation and deactivation—probability matrix P, where $P_{i,t}$ corresponds to the probability that node i at time t transitions to the active state in the next time step. To make the comparison with $I(X;G)$, we compare the transition probability matrix p^* of the true model—in the case of Fig. 5.2, the Glauber dynamics where the entries of p^* are given by the activation α and deactivation β probabilities (see Table 5.2)—with those predicted by models learned from time series generated by the Glauber dynamics. These models are trained with 100 concatenated time series, each generated using a different graph sampled from the Erdős-Rényi model. The models are then trained to predict the time series without the knowledge of the structure. The input of these models is the complete state of the system at time t, i.e., X_t, and the output is a vector $\hat{p}_t = (\hat{p}_{1,t}, \cdots, \hat{p}_{N,t})$, where $\hat{p}_{i,t}$ is the predicted probability that node i transition to the active state at time t. We use the mean absolute error (MAE) between p^* and $\hat{p}$ to compare them, i.e.,

$$\mathrm{MAE}(p^*, \hat{p}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} |p^*_{i,t} - \hat{p}_{i,t}| \,. \tag{5.10}$$

In doing so, the MAE quantifies the difference between a graph-dependent model and a graph-independent one, which highlight the importance of G over the prediction of X, which is a proxy of $I(X;G)$.

We consider two graph-independent prediction models : a logistic regression model and a multilayer perceptron (MLP). In both models, the predicted transition probabilities at time t are given by

$$\hat{p}_t = \frac{1}{e^{-f(X_t)} + 1} \,, \tag{5.11}$$

where $f(X_t)$ is a learnable function, that is linear for the logistic regression model, i.e.

$$f_{\text{logistic}}(X_t) = \mathbf{A} X_t + \boldsymbol{b} \,, \tag{5.12}$$

and non-linear for the MLP :

$$f_{\text{MLP}}(X_t) = \mathrm{ReLU}\Big[\mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 X_t + \boldsymbol{b}_1) + \boldsymbol{b}_2\Big] \,, \tag{5.13}$$

such that

$$\mathrm{ReLU}(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases} . \tag{5.14}$$

The weight matrices $\mathbf{A}$, $\mathbf{W}_1$ and $\mathbf{W}_2$, and bias vectors $\boldsymbol{b}$, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$, are learned via stochastic gradient descent using a cross-entropy loss.

Reconstruction algorithms

In Fig. 5.2, we also illustrated the relationship between the performance of reconstruction algorithms and $I(X;G)$. These algorithms are given the time series and they compute a score matrix S, such that S_{ij} for each pair of nodes (i,j) correlates with probability that an edge exists between them. For the correlation matrix method [160], this score is simply the correlation coefficient :

$$S_{ij} = \frac{C_{ij}}{\sigma_i \sigma_j}\,, \quad C_{ij} = \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - \bar{X}_i)(X_{j,t} - \bar{X}_j) \tag{5.15}$$

where $\bar{X}_i = \frac{1}{T}\sum_{t=1}^{T} X_{i,t}$ and $\sigma_i = \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - \bar{X}_i)^2$. In the Granger causality method [274], we compare via a F-test the prediction of the time series of a single node i using a linear auto-regressive model, with another auto-regressive model that includes the time series of node j. Then, the test determines if the models error are similar or different by computing the following F-statistic :

$$S_{ij} = \frac{\Sigma_{ij}}{\Sigma_i}\,, \tag{5.16}$$

where Σ_i is the error variance of the auto-regressive model of i, and Σ_{ij} is the error variance of the other model that also includes j. Finally, in the transfer entropy method [277], the score is given by the transfer entropy from the time series of j to the time series of i :

$$S_{ij} = T_{X_j \to X_i} \tag{5.17}$$

where

$$T_{X_j \to X_i} = H(X_{i,t}|X_{i,t-1}) - H(X_{i,t}|X_{i,t-1}, X_{j,t-1})\,. \tag{5.18}$$

The entropies involved in the computation of $T_{X_j \to X_i}$ are evaluated using the maximum likelihood estimators of the probabilities $P(X_{i,t}|X_{i,t-1})$ and $P(X_{i,t}|X_{i,t-1}, X_{j,t-1})$, estimated from the time series itself.

We quantify the accuracy of the reconstruction using the AUC of the ROC curve. This curve is obtained by comparing the true positive rate with the false positive rate, for different thresholds $\phi \in [\min\{S\}, \max\{S\}]$. The AUC, being the integral of that curve, therefore represents the probability that the score matrix S classifies correctly a node pair connected by an edge.

5.7.3 Formal definition of θ-duality

In what follows, we define the duality between predictability and reconstructability by taking a more general stance : Instead of considering a stochastic process X evolving on a random graph G, we let G be any discrete random variable conditioning the probability of X. First, we define the local duality of the uncertainty coefficients. The latter are considered as continuously differentiable functions with respect to a parameter θ whose domain is some non-empty interval of the real line.

Definition 5.1 (Local duality). *The uncertainty coefficients $U(X|G)$ and $U(G|X)$ are locally dual with respect to θ at $\theta = \theta^*$ if and only if*

$$\left[\frac{\partial U(X|G)}{\partial \theta}\frac{\partial U(G|X)}{\partial \theta}\right]_{\theta=\theta^*} < 0\,. \tag{5.19}$$

The definition of the θ-duality, a global property, follows that of the local duality.

Definition 5.2 (θ-Duality). *The uncertainty coefficients $U(X|G)$ and $U(G|X)$ are dual with respect to θ, or θ-dual, in the interval Θ if and only if they are locally dual for all values of θ^* in Θ.*

From these definitions, we relate the presence of extrema of $U(X|G)$ and $U(G|X)$ with the existence of a θ-duality.

Lemma 5.1 (θ-duality between extrema). *Let Θ be a non-empty subinterval of the variable θ whose one endpoint is a local extremum of $U(X|G)$ and the other, a local extremum of $U(G|X)$. Moreover, suppose that $U(X|G)$ and $U(G|X)$ do not have critical points in Θ. Then the extrema points delineate a region of θ-duality if and only if they are both maxima (or both minima).*

The proof of this lemma is available in Appendix 5.8.5.

5.7.4 Proof of the universality of the T-duality

In what follows, we prove Theorem 5.1, that shows the universality of the T-duality, where T is the number of steps in the process X. We make use of the two following lemmas, that are proved in Supplementary Information (Appendices 5.8.6 and 5.8.7), regarding the monotonicity of $I(X;G)$ with respect to T and the existence of continuous extensions of $U(X|G)$ and $U(G|X)$, that will allow us to apply the Definition 5.1 involving derivatives.

Lemma 5.2 (Monotonicity of mutual information with T). *Let $X = (X_1, X_2, \cdots, X_T)$ be a Markov chain of length T whose transition probabilities are conditional to some discrete random variable G that is independent of T and such that $H(X_{t+1}|X_t) > 0$ for all $t \in [T-1]$. Suppose moreover that the state spaces of X and G are finite. Then the mutual information $I(X;G)$ is nonzero and monotonically increasing with $T \in \mathbb{Z}_+$.*

Lemma 5.3 (Continuous extension of uncertainty coefficients with T). *Let $X = (X_1, X_2, \cdots, X_T)$ and G respectively be a Markov chain and a discrete random variable as in Lemma 5.2. Then the uncertainty coefficients $U(G|X)$ and $U(X|G)$, interpreted as functions of $T \in \mathbb{Z}_+$, can be uniquely generalized to functions, respectively $f(T)$ and $g(T)$, that are holomorphic for all $T \in \mathbb{C}$, and thus real analytic for all $T \in \mathbb{R}_+$. Moreover, $H(X)$ can be extended to a function $h(T)$ that is analytic for all $T \in \mathbb{R}_+$ except where $f(T) = 0$.*

Next, we prove Theorem 5.1.

Proof. According to Lemma 5.3, the quantities $U(X|G)$, $U(G|X)$, and $H(X)$, which were originally defined as real functions of $T \in \mathbb{Z}_+$, have unique analytic extensions on the positive real axis, i.e., $T \in \mathbb{R}_+$. This allows us to treat $U(X|G)$, $U(G|X)$, and $H(X)$ as continuously differentiable functions with respect to T, where $U(G|X) = \frac{I(X;G)}{H(G)}$ and $H(X)$ are also monotone.

Now, by hypothesis, the entropy rate of the Markov chain X, $R \equiv \lim_{T\to\infty} \frac{H(X)}{T}$, is well defined and nonzero. Hence, $H(X) \sim RT$, i.e., $H(X)$ is positive and asymptotically linearly increasing with T. Moreover, since G is independent of T and $I(X;G) > 0$, it follows that $I(X;G)$ is monotonically increasing with respect to T by Lemma 5.2. As a result, $U(G|X) = \frac{I(X;G)}{H(G)}$ is also monotonically increasing, since its denominator is independent of T, by assumption. This translates to the strict inequality $\frac{\partial U(G|X)}{\partial T} > 0$. If there exists a T-duality, i.e., there is a domain of T where Eq. (5.19) is true, then $U(X|G)$ must be monotonically decreasing with T—or $\frac{\partial U(X|G)}{\partial T} < 0$—in that domain. To prove this, note that we can relate the two uncertainty coefficients using

$$H(X) = \frac{H(G)\, U(G|X)}{U(X|G)}\,. \tag{5.20}$$

This leads to the following differential equation

$$\frac{\partial}{\partial T}[\log U(X|G)] = \frac{\partial}{\partial T}[\log U(G|X)] - \frac{\partial}{\partial T}[\log H(X)]\,, \tag{5.21}$$

where we used the fact that $\frac{\partial H(G)}{\partial T} = 0$. Hence, to show that $U(X|G)$ is monotonically decreasing with T, the following inequality must hold

$$\frac{\partial}{\partial T}[\log U(G|X)] < \frac{\partial}{\partial T}[\log H(X)]\,. \tag{5.22}$$

Suppose for a moment that $U(X|G)$ is in fact increasing, such that Eq. (5.22) is false. This will eventually give rise to a contradiction. Let $g(T) \equiv U(G|X)$ and $h(T) \equiv H(X)$ be continuous functions of T such that their derivative with respect to T are respectively given by $g'(\tau) \equiv \frac{\partial f(T)}{\partial T}\Big|_{T=\tau}$ and $h'(\tau) \equiv \frac{\partial h(T)}{\partial T}\Big|_{T=\tau}$. Note that $0 < f(\tau) \leq 1$ and $h(\tau) > 0$ for all $\tau \in \mathbb{R}_+$. If Eq. (5.22) is false, then

$$(\log g(T))' \geq (\log h(T))'\,. \tag{5.23}$$

Using Grönwall's inequality [167, Theorem 1.2.1], we get

$$\frac{g(T)}{g(a)} \geq \frac{h(T)}{h(a)}\,, \qquad 0 < a < T. \tag{5.24}$$

So far, we have established that $h(T) = H(X) \sim RT$ and that $U(G|X)$ is monotonically increasing. We have also proved that if $U(X|G)$ is not monotonically decreasing with T, then inequality (5.24) is satisfied. However, the latter inequality and $h(T) \sim RT$ readily imply

that $g(T)$ belongs to the class $\Omega(T)$, which is the set of all $\tilde{g}(T)$ such that there exist positive constants, S and T^*, for which $\tilde{g}(T) \geq ST$ for all $T \geq T^*$ (i.e., Knuth's Big Omega [159]).

Two cases must be considered. First, if $ST^* > 1$, then $\tilde{g}(T) \geq ST^* > 1$, which is in direct contradiction with $g(T) \leq 1$ whenever $T \geq T^*$. Second, if $ST^* \leq 1$, then choose $T^{**} > S^{-1} \geq T^*$, so that $\tilde{g}(T) \geq ST^{**} > 1$ for all $T \geq T^{**}$. This again contradicts the inequality $g(T) \leq 1$ whenever $T \geq T^{**}$. As a result, inequality (5.24) cannot be satisfied when $T \geq \phi$, with $\phi = \max\{T^*, T^{**}\}$. We thus conclude that $U(X|G)$ is monotonically decreasing for all $T \geq \phi$. Therefore, $U(G|X)$ and $U(X|G)$ are T-dual in the interval $[\phi, \infty)$. □

5.7.5 Estimators of the mutual information

The mutual information $I(X;G)$ is generally intractable. Its intractability stems from the evaluation of the evidence probability, which is defined by the following equation :

$$P(X = x) = \sum_{g \in \mathcal{G}} P(G = g) P(X = x | G = g) \,. \tag{5.25}$$

Indeed, this sum potentially counts a number of terms which grows exponentially with the number of vertices N in the random graph. More specifically, the evidence probability appears in two entropy terms needed to compute the mutual information, namely the marginal entropy

$$H(X) = -\mathbb{E}\left[\log P(X)\right], \tag{5.26}$$

and the reconstruction entropy

$$H(G|X) = -\mathbb{E}\left[\log \frac{P(G)P(X|G)}{P(X)}\right], \tag{5.27}$$

where $\mathbb{E}[f(Y)]$ denotes the expectation of $f(Y)$. Fortunately, the evidence probability, and in turn the mutual information, can be estimated efficiently using Monte Carlo techniques, which we present in this section.

Graph enumeration approach

For sufficiently small random graphs ($N \leq 5$), the evidence probability can be efficiently computed by enumerating all graphs of $\mathcal{G}$ and by adding explicitly each term of Eq. (5.25). Then, we can estimate the mutual information by sampling M graphs $\left\{g^{(m)} : m \in [M]\right\}$, followed by M time series $\left\{x^{(m)} : m \in [M]\right\}$—such that $x^{(m)}$ is generated with $g^{(m)}$—, and by computing the following arithmetic average :

$$\begin{aligned} I(X;G) \simeq \frac{1}{M} \sum_{m=1}^{M} & \log P\Big(X = x^{(m)} | G = g^{(m)}\Big) \\ & - \log P\Big(X = x^{(m)}\Big) \,. \end{aligned} \tag{5.28}$$

The variance of this estimator scales with the inverse of $\sqrt{M}$. In Fig. 5.5, we used this estimator to compute the mutual information, where $M = 1000$.

Variational mean-field approximation

In this approach, we estimate the posterior probability instead of the evidence probability. According to Bayes' theorem, the posterior probability is

$$P(G|X) = \frac{P(G)P(X|G)}{P(X)}\,. \tag{5.29}$$

Behind this estimator is a variational mean-field (MF) approximation that assumes the conditional independence of the edges. For simple graphs, the MF posterior is

$$P_{\text{MF}}(G|X) = \prod_{i\leq j}^{N} [\pi_{ij}(X)]^{A_{ij}} \left[1 - \pi_{ij}(X)\right]^{1-A_{ij}}\,, \tag{5.30}$$

where $\pi_{ij}(X) \equiv P(A_{ij} = 1|X)$ is the marginal conditional probability of existence of the edge (i, j) given X. For multigraphs, a similar expression can be obtained, but instead involves a probability $\pi_{ij}(m|X) \equiv P(M_{ij} = m|X)$ that there are m multi-edges between i and j. In this case, the MF posterior becomes

$$P_{\text{MF}}(G|X) = \prod_{i\leq j}^{N} \prod_{m=0}^{\infty} [\pi_{ij}(m|X)]^{\delta(m,M_{ij})}\,, \tag{5.31}$$

where $\delta(x, y)$ is the Kronecker delta. The MF approximation allows to compute a lower bound of the true posterior entropy, such that

$$H(G|X) \geq -\mathbb{E}\left[\log P_{\text{MF}}(G|X)\right], \tag{5.32}$$

as a consequence of the conditional independent between the edges [65, Theorem 2.6.5]. Using the MF approximation and a strategy similar to the exact estimator, we compute the MF estimator of the mutual information as follows :

$$\begin{aligned} I(G|X) \geq \frac{1}{M}\sum_{m=1}^{M} \Big[&\log P_{\text{MF}}\Big(G = g^{(m)}|X = x^{(m)}\Big) \\ &- \log P\Big(G = g^{(m)}\Big)\Big]\,. \end{aligned} \tag{5.33}$$

To compute $P_{\text{MF}}\Big(G = g^{(m)}|X = x^{(m)}\Big)$, we sample a set $\mathcal{Q}^{(m)} \equiv \left\{g_1^{(m)}, \cdots, g_Q^{(m)}\right\}$ of Q graphs from the posterior distribution $P(G|X = x^{(m)})$. Then, we estimate the probabilities $\pi_{ij}(X) \simeq \frac{n_{ij}^{(m)}}{Q}$ using their corresponding maximum likelihood estimate, where $n_{ij}^{(m)}$ is the number of times the edge (i, j) is seen in $\mathcal{Q}^{(m)}$. An analogous maximum likelihood estimate is made in the multigraph case, where $\pi_{ij}(\omega|X) \simeq \frac{n_{ij;\omega}^{(m)}}{K}$ and $n_{ij;\omega}^{(m)}$ counts the number of times there were ω multiedges between i and j in $\mathcal{Q}^{(m)}$. This estimator is a lower bound of the mutual information—a consequence of Eq. (5.32). Hence, it is biased, and the extent of this bias is dependent on the quality of the conditional independence assumption with respect

to the true random graph. Note that the MF estimator can yield negative estimates of the mutual information (see Appendix 5.8.9).

In Fig. 5.7, we fix the number of graphs sampled from the posterior distribution to $Q = 1000$, and propose $5N$ moves between each sample (see also Section 5.7.6 for more detail).

5.7.6 Markov chain Monte-Carlo algorithm

To sample from the posterior distribution, we use a Markov chain Monte-Carlo (MCMC) algorithm where, starting from a graph g, we propose a move to graph g', according to a proposition probability $P(G' = g'|G = g)$, and accept it with the Metropolis-Hastings probability :

$$\min\left(1, e^{-\log\Delta}\frac{P(G' = g|G = g')}{P(G' = g'|G = g)}\right), \tag{5.34}$$

where $\Delta = \frac{P(G=g')P(X=x|G=g')}{P(G=g)P(X=x|G=g)}$ is the ratio between the joint probability of the two graphs with the time series X. This ratio can be computed efficiently in $\mathcal{O}(T)$, by keeping in memory $n_{i,t}$, the number of inactive neighbors, and $m_{i,t}$, a number of active neighbors, for each vertex i at each time t (see Ref. [242]). Equation (5.34) allows to sample from the posterior distribution $P(G|X)$ without the requirement to compute the intractable normalization constant $P(X)$. We collect graph samples at every $N\delta$ moves, where we fix $\delta = 5$ in all experiments.

We consider two types of random graphs with different constraints : The Erdős-Rényi model and the configuration model. Hence, we need two different sampling propositions to apply our MCMC algorithm, that is one for each model. We assume that the support of the Erdős-Rényi model is the set of all simple graphs of N vertices with E edges. In this case, we consider a hinge flip move, where an edge (i, j) is sampled uniformly from the edge set of the graph G and a vertex k is sampled uniformly from its vertex set. Then, with probability $\frac{1}{2}$, we rewire edge (i, j) by either selecting i or j to connect with k. Note that, because we consider the support $\mathcal{G}$ of G to be a space of simple graphs, all moves resulting in the addition of a self-loop or a multiedges are rejected with probability 1. As a result, the proposition probability is the same for any move from g to g' :

$$P(G' = g'|G = g) = \frac{1}{EN} \quad \Rightarrow \quad \frac{P(G' = g|G = g')}{P(G' = g'|G = g)} = 1\,. \tag{5.35}$$

For the configuration model, we assume that the support is the set of all loopy multigraphs of N vertices whose degree sequence is $\boldsymbol{k}$. In this case, we propose double-edge swap moves according to the prescription of Ref. [95]. We refer to it for further details.

5.7.7 Real networks

In this section, we present the real networks used in the bottom panels of Fig. 5.7. The networks have been downloaded from the Netzschleuder network catalogue [244].

Little Rock Lake food web

The Little Rock Lake food web [192] is composed of $N = 183$ nodes and $E = 2\,494$ edges, where nodes represent taxa (like species) found in Little Rock Lake in Wisconsin, and edges represent feeding patterns between two taxa. As presented in Ref. [192], this network is directed, but for the purpose of our paper we reciprocated all edges. Also, note that the Glauber dynamics, which we used jointly with the Little Rock Lake food web in Fig. 5.7, was also used in Ref. [242] to simulate a simplified interaction between the taxa.

European airline route network

The European airline route network [51] is a multiplex network composed of $N = 450$ and $E = 3\,588$ edges, where nodes represent airports and edges are routes between them. These edges have different types, encoding the different airlines. In our paper, we do not make any distinction between the edge types for simplicity.

C. Elegans neural network

The C. Elegans neural network [62] used in Fig.5.7 is an undirected network of $N = 514$ and $E = 2\,363$ edges representing the neural network of male C. Elegans worms. The nodes are neurons and edges represent when there are gap junctions between neurons.

5.8 Supplementary material

5.8.1 Duality between prediction and reconstruction performance

Since $U(X|G)$ is a relative measure of information, where both its denominator and numerator vary with the coupling parameter between G and X, for instance J in the Ising-Glauber model, we need to choose an adequate performance measure in order to make a fair comparison. Note that this is not a problem for $U(G|X)$ only because its denominator remains constant with J : If we wanted to investigate the duality with respect to, say, the number of edges, a similar treatment would have to be done with reconstruction performance measures as well. That being said, we compare $U(X|G)$ with the relative MAE (RMAE), i.e., the MAE normalized by the MAE between the X itself and the probabilities of graph-independent model :

$$\mathrm{RMAE}(\hat{p}, p^*) = \frac{\mathrm{MAE}(\hat{p}, p^*)}{\mathbb{E}\left[\mathrm{MAE}(\hat{p}, X)\right]}\,, \tag{5.36}$$

where

$$\mathrm{MAE}(X, Y) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|X_{i,t} - Y_{i,t}\right|\,, \tag{5.37}$$

and where we assume that X is generated with the entries of the probability matrix p^* (see Material and Methods, Section B.1 for the details, e.g., on $\hat{p}$). The denominator measures the

absolute error of the graph-independent model, which is related with $H(X)$, while the numerator measure the distance between the graph-dependent and graph-independent models, which is analoguous to $I(X;G)$. Furthermore, the RMAE share another desirable property with $U(X|G)$.

Lemma 5.4. *If X be a NT-dimensional Bernouilli random variable with a parameter matrix p^* and the elements of $\hat{p}$ are in $(0,1)$, the RMAE between $\hat{p}$ and p^* defined in Eq. (5.36) is bounded in the unit interval $[0,1]$.*

Démonstration. First, we have that $X = (X_{i,t})_{i,t}$ with $X_{i,t} \in \{0,1\}$ for all $i \in [N]$, $t \in [T]$ and $P(X_{i,t} = 1) = p^*_{i,t}$, $P(X_{i,t} = 0) = 1 - p^*_{i,t}$. As expected for a Bernouilli random variable, we get

$$\mathbb{E}[X_{i,t}] = \sum_{x_{i,t} \in \{0,1\}} P(X_{i,t} = x_{i,t})\, x_{i,t} = p^*_{i,t} \tag{5.38}$$

and consequently, $p^* = \mathbb{E}[X]$. The RMAE becomes

$$\mathrm{RMAE}(\hat{p}, p^*) = \frac{\mathrm{MAE}(\hat{p}, \mathbb{E}[X])}{\mathbb{E}[\mathrm{MAE}(\hat{p}, X)]}\,. \tag{5.39}$$

The MAE in Eq.(5.37) is a double sum of absolute values, which are convex functions, and the MAE is therefore convex. Jensen's inequality thus implies

$$\mathrm{RMAE}(\hat{p}, p^*) = \frac{\mathrm{MAE}(\hat{p}, \mathbb{E}[X])}{\mathbb{E}[\mathrm{MAE}(\hat{p}, X)]} \leq \frac{\mathbb{E}[\mathrm{MAE}(\hat{p}, X)]}{\mathbb{E}[\mathrm{MAE}(\hat{p}, X)]} = 1\,. \tag{5.40}$$

Moreover, the MAE is obviously greater than or equal to 0 and since $\hat{p}_{i,t} \in (0,1)$ by assumption, $|\hat{p}_{i,t} - x_{i,t}| \in (0,1)$, meaning that $\mathbb{E}[\mathrm{MAE}(\hat{p}, X)] > 0$ and $\mathrm{RMAE}(\hat{p}, p^*) \geq 0$. □

Like $U(X|G)$, the case $\mathrm{RMAE}(\hat{p}, p^*) = 0$ occurs when all the elements of $\hat{p}$ and p^* are equal—the conditional and marginal models are the same—and the case $\mathrm{RMAE}(\hat{p}, p^*) = 1$ occurs when $p^* \in \{0,1\}^{N \times T}$—there is no uncertainty over X given G—and $p^* \neq \hat{p}$.

In Fig. 5.8, we compare the performance measures previously described with the uncertainty coefficients that helped us predict the existence of the coupling-duality in the context of the Glauber dynamics on small Erdős-Rényi graphs. We observe that the duality region between the prediction and reconstruction performance measures is analoguous to the one predicted by the uncertainty coefficients.

5.8.2 Analytical solutions for simple systems

Two graphs and three time series

We consider the scenario presented in Section II.C of the main paper represented by Fig. 3(a). In this system, the structure can either be given by g_1 with probability $P(G = g_1) = p$ or g_2,

with probability $P(G = g_2) = 1 - p$. For the purpose of the demonstration, the specifics of g_1 and g_2 are not needed. Then, g_1 can generate two time series, X_1 and X_2, with probabilities $P(X = X_1|G = g_1) = r$ and $P(X = X_2|G = g_1) = 1 - r$. Likewise, g_2 generates X_2 with probability $P(X = X_2|G = g_2)$ and X_3 with probability $P(X = X_3|G = g_2)$. In this example, it is assumed that X_1, X_2 and X_3 are completely different time series.

First, we need to compute the mutual information $I(X;G) = H(X) - H(X|G) = H(G) - H(G|X)$ in order to determine both uncertainty coefficients. For that, we first evaluate $H(X)$ and $H(X|G)$:

$$\begin{aligned} H(X) = &-pr\log(pr) \\ &- \Big[p(1-r) + (1-p)s\Big]\log\Big(p(1-r) + (1-p)s\Big) \\ &- (1-p)(1-s)\log\Big((1-p)(1-s)\Big) \end{aligned} \tag{5.41}$$

and

$$H(X|G) = p\,\mathcal{H}(r) + (1-p)\,\mathcal{H}(s)\,, \tag{5.42}$$

such that $\mathcal{H}(q) = -q\log q - (1-q)\log(1-q)$ is the binary entropy. The mutual information is therefore the difference between the above equations.

$$\begin{aligned} I(X;G) = &\, p(1-r)\log(1-r) + (1-p)s\log(s) \\ &- pr\log(p) - (1-p)(1-s)\log(1-p) \\ &- \Big[p(1-r) + (1-p)s\Big]\log\Big(p(1-r) + (1-p)s\Big)\,. \end{aligned} \tag{5.43}$$

Then, we evaluate the entropy of the graph, which is simply given by $H(G) = \mathcal{H}(p)$.

Now, assuming that $s = 0$ gives us $I(X;G) = \mathcal{H}(p)$, as mentioned in Section II.B, which yields $U(G|X) = 1$. Also, since $H(X) = p\mathcal{H}(r) + \mathcal{H}(p)$, then $U(X|G) = \frac{\mathcal{H}(p)}{p\mathcal{H}(r)+\mathcal{H}(p)}$. When $s = 1$, we have $H(X) = \mathcal{H}(pr)$ and $H(X|G) = p\mathcal{H}(r)$, which leads to $I(X;G) = \mathcal{H}(pr) - p\mathcal{H}(r)$. The uncertainty coefficients are, in this case, $U(X|G) = 1 - \frac{p\mathcal{H}(r)}{\mathcal{H}(pr)}$ and $U(G|X) = \frac{\mathcal{H}(pr)-p\mathcal{H}(r)}{\mathcal{H}(p)}$.

5.8.3 Markov process on a single node graph

Suppose the random graph G has only two instances : a single isolated node denoted as $a = 0$ with a probability of $(1 - p)$, and a single node connected to itself by a self-loop denoted as $a = 1$ with a probability of p. Thus, we have $H(G) = -p\log(p) - (1-p)\log(1-p)$. The state X_t of the node at time t is a Bernoulli random variable that evolves according to the following Markovian rule : if $X_{t-1} = 1$, then $X_t = 0$ with probability 1; if $X_{t-1} = 0$, then $X_t = 0$ with probability $1 - aq$, and $X_t = 1$ with probability aq. The initial condition remain fixed to $X_1 = 1$.

The parameter $q \in [0,1]$ serves as a coupling between the structure and the function of the model. When $q = 0$, both graphs generate the same time series $(1,0,0,\ldots,0)$. Therefore, observing the time series when $q = 0$ provides no information about the graph, resulting in minimal reconstructability $U(G|X) = 0$. On the contrary, when $q = 1$, the graph with no self-loop can only generate the time series $(1,0,0,\ldots,0)$ (absorbing state), while the graph with a self-loop can only generate the oscillating pattern $(1,0,1,0,1,\ldots)$. Thus, when $q = 1$ and $T > 2$, maximal reconstructability $U(G|X) = 1$ is achieved : observing only one time series allows for the unique determination of the graph. Additionally, in this case, there is also maximal predictability $U(X|G) = 1$ for $T > 1$, since observing the graph and the initial state 1 provides all subsequent values for X_t.

However, in general, when $0 < q < 1$, both reconstructability and predictability are only partial. Without a self-loop, there is still only one possible time series, namely $(1,0,0,\ldots,0)$. With a self-loop, the aforementioned time series is possible, along with others such as $(1,0,1,0,\ldots)$, $(1,0,0,1,\ldots)$, and so on, with the restriction that state 1 cannot be followed by another state 1. Thus, because the time series $(1,0,0,\ldots,0)$ has a non-zero probability of occurring in both graphs, the reconstructability satisfies $U(G|X) < 1$. However, since the likelihood of $(1,0,0,\ldots,0)$ decreases exponentially with T for the node with a self-loop, we know that $U(G|X) \to 1$ as $T \to \infty$. Consequently, the mutual information approaches its maximal value $H(G)$ as $T \to \infty$. Similarly to the previous examples, this implies that the predictability behaves *dually* to the reconstructability, satisfying $U(X|G) \to 0$ as $T \to \infty$.

5.8.4 Predictability and reconstructability of deterministic dynamics

Our framework can be applied to continuous-state processes, where $x_i^*(t;g) \in \mathbb{R}$ is the solution of the process corresponding to the i-th node's state at continuous time $t \in [0,T]$ for the graph instance g of N nodes. For a large class of these processes, the states are governed by differential equations of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}x_i^*(t;g) = F_1\Big(x_i^*(t;g)\Big) + \sum_{j=1}^{N} A_{ij}(g)F_2\Big(x_i^*(t;g), x_j^*(t;g)\Big)\,, \qquad i \in [N]\,, \tag{5.44}$$

where F_1 describes the self-dynamics of the nodes and F_2 specifies the interaction between connected nodes. A complete trajectory $X^*(g)$ is given by a set of tuples $X^*(t;g) = \big(x_1^*(t;g), \cdots, x_N^*(t;g)\big)$ for t in the interval $[0,T]$. Also, note that $A_{ij}(g)$ denotes the element (i,j) of the adjacency matrix of g, which equals one if there is an edge between i and j in graph g, and zero otherwise. The quench mean-field SIS dynamics can be formulated in this form, where $F_1(x) = \delta x$ and $F_2(x,y) = \lambda(1-x)y$. Other examples of F_1 and F_2 are provided in Refs. [171, 251, 270, 313].

This class of system was investigated in Ref. [251], where it was found that many graphs can often generate similar time series. This conclusion suggests that predicting the evolu-

tion of a time series may in fact be possible without specific knowledge of the structure of interactions. Here, we shed light on this result using our information-theoretic framework.

To connect our framework with that of Ref. [251], we work in discrete time, where $t \in \{0, dt, 2dt, \cdots, (K-1)dt\}$ for some small dt. By doing so, the functions $x_i^*(t;g)$ are replaced by the matrix elements $x_{i,k}^*(g)$, where time steps are specified by the index $k \in \{0, 1, \ldots, K-1\}$ as $t_k = kdt$ and $T = t_{K-1} = (K-1)dt$. The evolution equation (5.44) is then discretized as follows :

$$x_{i,k+1}^*(g) = x_{i,k}^*(g) + dt \left[F_1\Big(x_{i,k}^*(g)\Big) + \sum_{j=1}^{N} A_{ij}(g) F_2\Big(x_{i,k}^*(g), x_{j,k}^*(g)\Big) \right]. \tag{5.45}$$

These processes are deterministic, meaning that for a fixed graph g and initial condition $X_0^*(g) \in \mathbb{R}^N$, the probability that the trajectory $X^*(g) \in \mathbb{R}^{N\times K}$ is sampled is, by definition, exactly equal to 1. We let the initial conditions be randomly chosen in a—discrete—set of states $\mathcal{X}_0 \subset \mathbb{R}^N$ with probability distribution $P(X_0)$. Each trajectory generated by a graph g therefore becomes a $N \times K$ random matrix X whose values belong to a discrete set

$$\mathcal{X}(g) = \big\{x^*(g) \,:\, x_0^*(g) \in \mathcal{X}_0\big\}, \tag{5.46}$$

where we have omitted the explicit dependence of $\mathcal{X}(g)$ on $\mathcal{X}_0$ for brevity. The probability that the trajectory x is generated by g is

$$P\Big(X = x \Big| G = g\Big) = \begin{cases} P(X_0 = X_0) & \text{if } x \in \mathcal{X}(g)\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.47}$$

This is consistent with the equality $|\mathcal{X}(g)| = |\mathcal{X}_0|$, which is a direct consequence of the deterministic nature of the dynamics. Hence, all the stochasticity of the system is coming from the choice of initial conditions. For the remainder of the calculation, we will assume for simplicity that the initial conditions are uniformly distributed, i.e., $P(X_0 = X_0) = |\mathcal{X}_0|^{-1}$. This requires $\mathcal{X}_0$ to be finite so that the set of all trajectories generated by all graphs of N nodes in some finite set $\mathcal{G}$,

$$\mathcal{X} = \big\{X \in \mathcal{X}(g) \,:\, g \in \mathcal{G}\big\}, \tag{5.48}$$

also remains finite. This technicality allows the entropies of X to remain finite and well defined, something necessary to keep our framework unchanged. Note that the case where $\mathcal{X}_0$ is countably infinite can also be treated by properly choosing a non-uniform distribution $P(X_0)$. Further work is required if a truly continuous-state formulation of our information measures is to be considered.

This description of a deterministic system is quite restrictive, since by construction a graph is only allowed to generate a single time series once the initial conditions have been fixed. We would like to relax the notion of deterministic systems to admit a prediction error ϵ. Let

$$\mathcal{X}_\epsilon(g) = \big\{X \in \mathcal{X} \,:\, \|X - X^*(g)\| \leq \epsilon,\, X^*(g) \in \mathcal{X}(g)\big\}, \tag{5.49}$$

be the set of all trajectories X that are sufficiently close—up to a distance $\epsilon \geq 0$—to a trajectory $X^*(g)$ deterministically generated by g, where the symbol $\|\cdot\|$ denotes some matrix norm. We say that a trajectory X is ϵ-compatible with g if $X \in \mathcal{X}_\epsilon(g)$. In other words, X is ϵ-compatible with g if it is deterministically generated by g or if there is another graph $h \in \mathcal{G}$ that deterministically generate a trajectory $X^*(h)$ that satisfies $X^*(h) = X$ and $\|X^*(h) - X^*(g)\| \leq \epsilon$. Likewise, we define $\mathcal{G}_\epsilon(X) = \{g \in \mathcal{G} \,:\, X \in \mathcal{X}_\epsilon(g)\}$, i.e., the set of all graphs g that are ϵ-compatible with X. Thus, X is ϵ-compatible with g iff g is ϵ-compatible with X. In this scenario, we introduce a new $N \times K$ random matrix X_ϵ, taking values in the set of all deterministic trajectories $\mathcal{X}$, defined in Eq. (5.48). The probability of $X_\epsilon = X$ given $G = g$, interpreted as the probability that the trajectory X is generated from g within a margin of error of size ϵ, is simply defined as the uniform distribution over all trajectories in $\mathcal{X}_\epsilon(g)$:

$$P\Big(X_\epsilon = X \Big| G = g\Big) = \begin{cases} |\mathcal{X}_\epsilon(g)|^{-1} & \text{if } X \in \mathcal{X}_\epsilon(g), \\ 0 & \text{otherwise.} \end{cases} \tag{5.50}$$

As we approach the limit $\epsilon \to 0^+$, the normalization factor $|\mathcal{X}_\epsilon(g)|$ tends to $|\mathcal{X}(g)| = |\mathcal{X}_0|$ for all $g \in \mathcal{G}$, and the above conditional probability converges towards Eq. (5.47) defining X (with uniformly distributed initial conditions). Moreover, for ϵ sufficiently large, all conditional probabilities become non-zero. Increasing ϵ therefore introduces uncertainty about which graph can generate a specific trajectory, gradually blurring the deterministic nature of the process.

At this point, we have all the tools to compute the predictability and reconstructability. We begin our calculation with the mutual information. First, we evaluate the marginal probability of X_ϵ as follows :

$$P(X_\epsilon = X) = \sum_{g \in \mathcal{G}} P(G = g) P(X_\epsilon = X | G = g) = \sum_{g \in \mathcal{G}_\epsilon(X)} \frac{p(g)}{|\mathcal{X}_\epsilon(g)|},$$

where $p(g)$ denotes the probability $P(G = g)$. From there, we calculate the entropies and the mutual information :

$$H(X_\epsilon | G) = \mathbb{E}_G[\log |\mathcal{X}_\epsilon(G)|]\,, \tag{5.51a}$$

$$H(X_\epsilon) = -\mathbb{E}_{X_\epsilon}\left[\log\left(\sum_{h \in \mathcal{G}_\epsilon(X_\epsilon)} \frac{p(h)}{|\mathcal{X}_\epsilon(h)|}\right)\right], \tag{5.51b}$$

$$I(X_\epsilon; G) = -\mathbb{E}_{G, X_\epsilon}\left[\log\left(\sum_{h \in \mathcal{G}_\epsilon(X_\epsilon)} p(h) \frac{|\mathcal{X}_\epsilon(G)|}{|\mathcal{X}_\epsilon(h)|}\right)\right]. \tag{5.51c}$$

Using this expression for the mutual information leads to the following predictability :

$$U(X_\epsilon|G) = \frac{I(X_\epsilon;G)}{H(X_\epsilon)} = \frac{\mathbb{E}_{G,X_\epsilon}\left[\log\left(\sum_{g\in\mathcal{G}_\epsilon(X_\epsilon)} p(g)\frac{|\mathcal{X}_\epsilon(G)|}{|\mathcal{X}_\epsilon(g)|}\right)\right]}{\mathbb{E}_{X_\epsilon}\left[\log\left(\sum_{g\in\mathcal{G}_\epsilon(X_\epsilon)} \frac{p(g)}{|\mathcal{X}_\epsilon(g)|}\right)\right]} \tag{5.52}$$

$$= 1 - \frac{H(X_\epsilon|G)}{H(X_\epsilon)} = 1 - \frac{\mathbb{E}_G[\log|\mathcal{X}_\epsilon(G)|]}{\mathbb{E}_{X_\epsilon}\left[\log\left(\sum_{g\in\mathcal{G}_\epsilon(X_\epsilon)} \frac{p(g)}{|\mathcal{X}_\epsilon(g)|}\right)^{-1}\right]}\,. \tag{5.53}$$

Also, since $H(G) = -\mathbb{E}_G[\log P(G)]$, the reconstructability is given by

$$U(G|X_\epsilon) = \frac{I(X_\epsilon;G)}{H(G)} = \frac{\mathbb{E}_{G,X_\epsilon}\left[\log\left(\sum_{h\in\mathcal{G}_\epsilon(X_\epsilon)} p(h)\frac{|\mathcal{X}_\epsilon(G)|}{|\mathcal{X}_\epsilon(h)|}\right)\right]}{\mathbb{E}_G[\log P(G)]} \tag{5.54}$$

$$= 1 - \frac{H(G|X_\epsilon)}{H(G)} = 1 - \frac{\mathbb{E}_{G,X_\epsilon}\left[\log\left(\sum_{h\in\mathcal{G}_\epsilon(X_\epsilon)} \frac{p(h)}{P(G)}\frac{|\mathcal{X}_\epsilon(G)|}{|\mathcal{X}_\epsilon(h)|}\right)\right]}{\mathbb{E}_G[\log P(G)]}\,. \tag{5.55}$$

To build more intuition about our theoretical results, let us focus on the predictability and consider two opposite limit cases : $\epsilon \to 0^+$ (minimum error tolerance) and $\epsilon \to \infty$ (maximum error tolerance). As ϵ approaches these two limits, $|\mathcal{X}_\epsilon(g)|$ respectively tends to $|\mathcal{X}_0|$ (number of initial conditions) and $|\mathcal{X}|$ (number of trajectories generated by all possible graphs from all initial conditions). This results hold for all $g \in \mathcal{G}$. Hence,

$$H(X_\epsilon|G) \to \begin{cases} \log|\mathcal{X}_0| & \epsilon \to 0^+, \\ \log|\mathcal{X}| & \epsilon \to \infty\,. \end{cases} \tag{5.56}$$

Moreover,

$$H(X_\epsilon) \to \begin{cases} \log|\mathcal{X}_0| + \mathcal{I} & \epsilon \to 0^+, \\ \log|\mathcal{X}| & \epsilon \to \infty\,, \end{cases} \tag{5.57}$$

where $\mathcal{I} = -\mathbb{E}_X\left[\log\sum_{g\in\mathcal{G}(X)} p(g)\right] > 0$, with $\mathcal{G}(X) = \mathcal{G}_0(X)$ being the set of all graphs that deterministically generate the trajectory X starting from some initial condition in $\mathcal{X}_0$. Note that $\mathcal{I}$ is the limit of the mutual information as $\epsilon \to 0^+$ and can be written as $\mathbb{E}_X\left[\log q(X)^{-1}\right]$, where

$$q(X) = \sum_{g\in\mathcal{G}(X)} p(g) \tag{5.58}$$

is the probability to have a graph in $\mathcal{G}(X)$. Going back to the expression for the predictability, we conclude that

$$U(X_\epsilon|G) \to \begin{cases} 1 - \frac{\log|\mathcal{X}_0|}{\log|\mathcal{X}_0|+\mathcal{I}} & \epsilon \to 0^+, \\ 0 & \epsilon \to \infty\,. \end{cases} \tag{5.59}$$

The last result is easy to interpret. On the one hand, when no margin of error is allowed and the graph is known, the only factor limiting our knowledge about the states' evolution is

the uncertainty about the initial conditions. Once an initial condition is chosen, corresponding to $|\mathcal{X}_0| = 1$, no uncertainty remains and the predictability reaches its maximal value 1, thus aligning with the intuition that in a deterministic process as in Eq. (5.45), the initial condition and the graph entirely determine the future. On the other hand, when all errors are tolerated between different trajectories, a single graph is seen as capable of generating all trajectories. Consequently, knowing G provides no information about the states that will follow the initial conditions, in perfect accordance with a zero predictability. Interpolating between these extreme cases, we understand that as ϵ decreases, the number of trajectories that are ϵ-compatible with a specific graph also decreases, meaning that on average, the uncertainty about the evolution of the process that remains after the observation of a graph also decreases, resulting in an increase of predictability.

Even substantial values of ϵ can lead to high values of predictability. To understand this last claim, let us rewrite the entropy of the process as

$$H(X_\epsilon) = \mathcal{A}_\epsilon + H(G) - \mathbb{E}_{G,X_\epsilon}[\log R_\epsilon(G, X_\epsilon)], \tag{5.60}$$

where

$$R_\epsilon(g, X) = \sum_{h \in \mathcal{G}_\epsilon(X)} \frac{p(h)}{p(g)} \frac{|\mathcal{X}_\epsilon(g)|}{|\mathcal{X}_\epsilon(h)|}, \quad \mathcal{A}_\epsilon = \mathbb{E}_G[\log |\mathcal{X}_\epsilon(G)|]\,. \tag{5.61}$$

Then, assuming that the graphs are almost uniformly distributed (all graphs sharing the same a priori contribution to the process) and possess almost the same number of ϵ-compatible trajectories, we find that

$$R_\epsilon(g, X) \approx \sum_{h \in \mathcal{G}_\epsilon(X)} 1 = |\mathcal{G}_\epsilon(X)| \tag{5.62}$$

for all graphs g. Moreover, in such circumstances, $H(G) \approx \log |\mathcal{G}|$ and this leads to an approximate formula for the mutual information :

$$I(X_\epsilon; G) \approx \mathcal{I}_\epsilon = \mathbb{E}_{X_\epsilon}\left[\log \frac{|\mathcal{G}|}{|\mathcal{G}_\epsilon(X_\epsilon)|}\right], \tag{5.63}$$

which in turn leads to an approximate formula for the predictability :

$$U(X_\epsilon|G) \approx 1 - \frac{\mathcal{A}_\epsilon}{\mathcal{A}_\epsilon + \mathcal{I}_\epsilon}\,. \tag{5.64}$$

Therefore, provided $|\mathcal{G}_\epsilon(X)|$ remains substantially smaller than $|\mathcal{G}|$ for every X, a high level of predictability is retained. *This means that even in the presence of numerous graphs capable of producing a trajectory within an error margin of ϵ (though considerably fewer relative to the total graph count), high predictability is feasible. This observation reinforces the notion that an exclusive relationship between a graph and a time series isn't a prerequisite for high predictability, echoing the conclusions drawn in Ref. [251].*

We now analyze the limit cases of the reconstructability given in Eqs. (5.54)–(5.55). Since the graph distribution $P(G)$ is not affected by the error margin ϵ, leaving the corresponding entropy $H(G)$ unchanged, the only factor that influences the reconstructability is the mutual information in Eq. (5.51c). Considering $\epsilon \to 0^+, \infty$ allows us to simplify its expression quite significantly as

$$I(X_\epsilon; G) \to \begin{cases} \mathcal{I} & \epsilon \to 0^+, \\ 0 & \epsilon \to \infty, \end{cases} \tag{5.65}$$

where we recall that $\mathcal{I} = \mathbb{E}_X\left[\log q(X)^{-1}\right] > 0$ with $q(X)$ defined by Eq. (5.58). This naturally leads to the following reconstructability :

$$U(G|X_\epsilon) \to \begin{cases} 1 - \frac{H(G)-\mathcal{I}}{H(G)} & \epsilon \to 0^+, \\ 0 & \epsilon \to \infty. \end{cases} \tag{5.66}$$

Again, the interpretation of this reconstructability is simple. First, with maximal tolerance for error, any graph is ϵ-compatible with every other trajectories. Consequently, knowing X does not provide information about G whatsoever and the reconstructability is zero. Second, when we gradually decrease the margin of error, fewer graphs are ϵ-compatible with any trajectory, providing more and more information about G. Hence, like $U(X_\epsilon|G)$, $U(G|X_\epsilon)$ increases with decreasing error tolerance size ϵ. Reconstructability is maximized when $\epsilon \to 0^+$, in which case its formula is easier to interpret when G is uniformly distributed, i.e., $P(G) = |\mathcal{G}|^{-1}$. In this case, $q(X) = \frac{|\mathcal{G}(X)|}{|\mathcal{G}|}$ and $I(X;G) = \log|\mathcal{G}| - \mathbb{E}_X[\log|\mathcal{G}(X)|]$, leading to

$$\lim_{\epsilon\to 0^+} U(G|X_\epsilon) = 1 - \frac{\mathbb{E}_X[\log|\mathcal{G}(X)|]}{\log|\mathcal{G}|}. \tag{5.67}$$

When no margin of error is tolerated, the reconstructability of G given X is simply given by the ratio between the logarithms of the number of graphs that are compatible with any given trajectories (on average), and the total number of graphs. As fewer graphs become on average compatible with any trajectory, the reconstructability converges to 1. In other words, reconstructability boils down to the fraction of graphs that may have generated X : the more compatible graphs there are, the harder it gets to find the correct graph.

Using similar assumptions as before, we can evaluate an approximate formula for the reconstructability as well. Again, assuming that $P(G)$ is almost uniform and ϵ is sufficiently small so that Eq. (5.65) is valid leads to

$$U(G|X_\epsilon) \approx 1 - \frac{\log|\mathcal{G}| - \mathcal{I}_\epsilon}{\log|\mathcal{G}|}\,. \tag{5.68}$$

Using our framework allows us to reach similar conclusions to the ones in Ref. [251] described above. Indeed, we find that, for sufficiently small tolerance ϵ, it is possible to have poor reconstructability even though the predictability is almost equal to one (see Fig. 5.9). This is because of the different

scaling behaviors with respect to the mutual information of $U(X_\epsilon|G)$ and $U(G|X_\epsilon) : U(X_\epsilon|G)$ quickly saturates to one when I_ϵ increases, whereas $U(G|X_\epsilon)$ grows linearly with I_ϵ. While we corroborate the findings of Ref. [251], it is important to stress that our results are conceptually quite different. For instance, their notion of reconstructability is intrinsically related to which graphs are reconstructed by their algorithm, and how different they are from the original graph. In turn, they measure reconstructability using the AUC score of the reconstructed graph which does not incorporate the full range of graphs that can generate each specific trajectory. In this respect, our framework offers a different and complementary perspective to their work, by quantifying their observation in terms of information.

5.8.5 Proof of the θ-duality between extrema

In this section, we relate the presence of extrema of $U(X|G)$ and $U(G|X)$ with the existence of a θ-duality.

Lemma 5.5 (θ-duality between extrema)**.** *Let Θ be a non-empty subinterval of the variable θ whose one endpoint is a local extremum of $U(X|G)$ and the other, a local extremum of $U(G|X)$. Moreover, suppose that $U(X|G)$ and $U(G|X)$ do not have critical points in Θ. Then the extrema points delineate a region of θ-duality if and only if they are both maxima (or both minima).*

Démonstration. Let θ_R and θ_P be the extrema points of $U(G|X)$ and $U(X|G)$, respectively. Thus

$$\left.\frac{\partial U(G|X)}{\partial\theta}\right|_{\theta=\theta_R} = \left.\frac{\partial U(X|G)}{\partial\theta}\right|_{\theta=\theta_P} = 0\,. \tag{5.69}$$

Suppose for a moment that $\theta_R < \theta_P$ and let $\Theta = (\theta_R, \theta_P)$. This implies that $\frac{\partial U(G|X)}{\partial\theta}$ changes sign at θ_R, before $\frac{\partial U(X|G)}{\partial\theta}$, for which the sign change happens at θ_P.

On the one hand, if the extrema points θ_R and θ_P are both maxima (or minima), then $\frac{\partial U(G|X)}{\partial\theta}$ and $\frac{\partial U(X|G)}{\partial\theta}$ have different signs in Θ. Hence, the inequality

$$\left[\frac{\partial U(X|G)}{\partial\theta}\frac{\partial U(G|X)}{\partial\theta}\right]_{\theta=\theta^*} < 0 \tag{5.70}$$

is verified in this region. The uncertainty coefficients are therefore θ-dual in Θ.

On the other hand, if the uncertainty coefficients are θ-dual in Θ, then inequality (5.70) is satisfied in this interval. This in turn implies that either $U(G|X)$ decreases in Θ while $U(X|G)$ increases or $U(G|X)$ increases in Θ while $U(X|G)$ decreases. Therefore, the endpoints of Θ are either both maximum points or both minimum points.

Finally, repeating the same arguments with $\theta_R > \theta_P$ and $\Theta = (\theta_P, \theta_R)$ leads to the same conclusions about θ-duality of $U(X|G)$ and $U(G|X)$ in Θ. □

5.8.6 Proof of the monotonicity of $I(X;G)$ with T

In this section, we prove the monotonicity of $I(X;G)$ with respect to T for completing the proof of the universality of the T-duality discussed in the main paper, where T is the length of the process X. The lemma is stated as follows :

Lemma 5.6 (Monotonicity of mutual information with T)**.** *Let $X = (X_1, X_2, \cdots, X_T)$ be a Markov chain of length T whose transition probabilities are conditional to some discrete random variable G that is independent of T and such that $H(X_{t+1}|X_t) > 0$ for all $t \in [T-1]$. Suppose moreover that the state spaces of X and G are finite. Then the mutual information $I(X;G)$ is nonzero and monotonically increasing with $T \in \mathbb{Z}_+$.*

Démonstration. Let us define a Markov chain $X' = (X_1, X_2, \cdots, X_{T-1})$ of size $T-1$, such that the concatenation of X' with state variable X_T yields X. Hence, we can express the mutual information between X and G in terms of X' as $I(X;G) = I(X', X_T; G)$. Furthermore, proving the monotonicity of mutual information can be reformulated as proving the following inequality :

$$I(X', X_T; G) - I(X'; G) > 0\,, \tag{5.71}$$

for all T. By the chain rule for conditional mutual information, that is

$$I(X', X_T; G) = I(X_T; G|X') + I(X'; G)\,, \tag{5.72}$$

inequality (5.71) becomes

$$I(X_T; G|X') = H(X_T|X') - H(X_T|X', G) > 0\,. \tag{5.73}$$

The term $H(X_T|X') - H(X_T|X', G)$ is always at least non-negative, by virtue of the non-negativity of mutual information [65, Theorem 2.6.5]. Then, to prove inequality (5.73), we must verify that $H(X_T|X')$ never equals $H(X_T|X', G)$. Recalling that

$$H(X_T|X') \geq H(X_T|X', G) \geq 0\,, \tag{5.74}$$

inequality (5.73) does not hold if (*i*) $H(X_T|X') = 0$ or if (*ii*) X_T is independent of G (i.e., $I(X_T; G|X') = 0$). According to the hypothesis $H(X_{t+1}|X_t) > 0$ for all $t \in [T-1]$, condition (*i*) cannot be true. Moreover, condition (*ii*) implies that $I(X;G) = I(X_T, X'; G) = I(X'; G) = 0$. Therefore, the only instance where Eq. (5.71) is not satisfied is when the Markov chain X is independent of G, i.e., $I(X;G) = 0$ for all length T. However, this contradicts the assumption about the transition probabilities. Hence, $I(X;G) > 0$ and monotonically increases with T. □

We have to make a few remarks about the restrictions imposed in the last lemma. The condition $H(X_{t+1}|X_t) > 0$ for all $t \in [T-1]$ only asserts that the Markov chain is nondeterministic

in the sense that knowing the state of the chain at time t does not completely eliminate the uncertainty about the state at time $t+1$. This condition is satisfied for a wide variety of stochastic processes, including the irreducible Markov chains, where there is always a nonzero probability to transition from a state to any other state in a finite number of time steps.

Moreover, the finiteness of the state spaces for the chain X and the variable G is imposed to make $H(X)$, $H(G)$, and $I(X;G)$ finite. This in turn ensures that the uncertainty coefficients $U(G|X)$ and $U(X|G)$ are well defined for all $T \in \mathbb{Z}_+$, a property that is necessary to prove the next lemma.

5.8.7 Proof of the existence of continuous extensions of uncertainty coefficients with T

In this section, we prove the existence of continuous extensions of the uncertainty coefficients of $I(X;G)$ with respect to T for completing the proof of the universality of the T-duality discussed in the main paper, where T is the length of the process X. The lemma is stated as follows :

Lemma 5.7 (Continuous extension of uncertainty coefficients with T)**.** *Let $X = (X_1, X_2, \cdots, X_T)$ and G respectively be a Markov chain and a discrete random variable as in Lemma 5.6. Then the uncertainty coefficients $U(G|X)$ and $U(X|G)$, interpreted as functions of $T \in \mathbb{Z}_+$, can be uniquely generalized to functions, respectively $f(T)$ and $g(T)$, that are holomorphic for all $T \in \mathbb{C}$, and thus real analytic for all $T \in \mathbb{R}_+$. Moreover, $H(X)$ can be extended to a function $h(T)$ that is analytic for all $T \in \mathbb{R}_+$ except where $f(T) = 0$.*

Démonstration. We first consider $U(X|G)$ and $U(G|X)$. These can be interpreted as functions of $T \in \mathbb{Z}_+$ whose values belong to the interval $[0,1]$. According to Guichard's Theorem [73, Theorem 5.2.1] (see also [265, Theorem 15.13]), there exist two functions of $z \in \mathbb{C}$, denoted f and g, that are holomorphic in the whole complex plane and whose values at $z = T \in \mathbb{Z}_+$ equal those of $U(X|G)$ and $U(G|X)$, respectively.

Now, $U(X|G)$ and $U(G|X)$, and consequently $f(z)$ and $g(z)$, have bounded values for all $z = T \in \mathbb{Z}_+$. Moreover, f and g are holomorphic, so their restriction to the axis $z = T \in \mathbb{R}$ is real analytic. Hence, on that axis, f and g are Lipschitz continuous, which means that there are positive and finite constants, a and b, such that $|f(T) - f(T')| \leq a|T - T'|$ and $|g(T) - g(T')| \leq b|T - T'|$ for all $T, T' \in \mathbb{R}$. Choosing $T = T' + \epsilon$ with $T' \in \mathbb{Z}_+$ and $|\epsilon| < 1$, we conclude that $f(T)$ and $g(T)$ have finite values for all $T \in \mathbb{R}_+$.

The functions f and g are thus holomorphic in the whole complex plane and bounded on the positive real axis. This allows to use a special case of Carlson's Theorem [12, Theorem 2.8.1] according to which holomorphic functions that are bounded on the positive real axis are uniquely defined by their values on the set $\mathbb{Z}_+$. Therefore, f is the unique extension $U(X|G)$

that is holomorphic for all $T \in \mathcal{C}$. Note that the restriction of f on the positive real axis is real analytic on this domain. Thus, there is a unique extension of $U(X|G)$ that is real analytic for all $T \in \mathbb{R}_+$ and that can be further extended to a holomorphic function for all $T \in \mathbb{C}$. The same conclusion holds for g and $U(G|X)$.

To finish the proof, we need to tackle $H(X)$. We cannot use the same strategy as above because $H(X)$ is not a bounded function of $T \in \mathbb{Z}_+$. However, by definition, the identity

$$H(X) = \frac{H(G)U(G|X)}{U(X|G)} \tag{5.75}$$

is valid whenever $U(X|G) > 0$. Now, according to Lemma 5.6, $I(X;G) > 0$ and hence $U(X|G) > 0$ for all $T \in \mathbb{Z}_+$. This means that Eq. (5.75) is well defined for all $T \in \mathbb{Z}_+$. To extend the domain of validity of the identity, we use the analytic functions f and g introduced above and define a new function h as

$$h(T) = H(G)\frac{g(T)}{f(T)}\,. \tag{5.76}$$

The values of h coincide with those of $H(X)$ for all $T \in \mathbb{Z}_+$, so that Eq. (5.75) defines a unique extension of $H(X)$. Moreover, h is analytic for all $T \in \mathbb{R}_+$ except at the points T where $f(T) = 0$. □

Lemma 5.7 ensures the *existence of analytic extensions* for the uncertainty coefficients, considered as functions of the positive integer T. These extensions can thus be evaluated and derived without restriction on the whole domain $\mathbb{R}_+$, which is a desirable property that is exploited in the proof of Theorem 1. However, the same lemma does not guarantee the monotonicity of the extensions on $\mathbb{R}_+$ in the event where they are monotone on $\mathbb{Z}_+$, although we assume it when proving Theorem 1. This is a reasonable assumption since numerical methods, generalizing the well-known Fritsch-Butland algorithm [97], have been recently developed to *construct smooth* (i.e., at least continuously differentiable) *and monotone interpolating functions* from any finite monotone datasets [329, 335]. With this assumption in hand, together with Lemmas 5.6 and 5.7, we can prove our main theoretical result (Theorem 1, Section II-D of the main paper) : the universality of the T-duality in Markov chains. The proof is provided in Section III.D of the main paper.

5.8.8 Numerical analysis of the past-dependent measures

In this section, we investigate further the past-dependent measures presented in Section II.B. Specifically, we discuss the implication of the universality of the T-duality (Theorem 1, see main text) in a more general context, using the past-dependent mutual information $I(X_{\text{future}}; G|X_{\text{past}})$. We show in Fig. 5.10 the reconstructability and predictability for increasing process length T, and different values of τ.

Two scenarios are of interest : The case where τ is constant with respect to T and the case where it is not. When τ is constant with respect to T, Theorem I remains valid since the additional conditions on the Markov chain Y and the random graph G are a special case of the prior assumptions. Hence, we observe the T-duality for any value of τ in this case, as supported by Figs. 5.10(a,d,g).

The second scenario, when τ is a function of T, is more nuanced, as seen in Figs. 5.10(b-c,e-f,h-i) since Theorem 1 no longer applies. This is because both Y and G (represented by X and Y in Theorem 1, respectively) are now conditioned on X, and thus will depend on T. Consequently, we no longer can assume that the entropy rate of Y given X is constant with T and that $H(G|X)$ is independent of T. In Fig. 5.10, we break this scenario into two cases. We consider $\tau = \kappa T$ [Figs. 5.10(b,e,h) with $\kappa = \frac{1}{2}$], where the lengths of X and Y remain proportional to one another. In this case, the T-duality seems to persist for all three dynamics. However, when $\tau = T - \xi$ [Figs. 5.10(c,f,i) with $\xi = 5$] where the size of Y remains fixed and X grows linearly with T, the T-duality is no longer observed, except for the Glauber dynamics. It is important to note that, for small ξ, the partial reconstructability coefficient $U(G|X_{\text{future}}; X_{\text{past}})$ becomes numerically unstable since both $I(X_{\text{future}}; G|X_{\text{past}})$ and $H(G|X_{\text{past}})$ tend to zero. This is why the curves are much noisier in that case. Informed by these examples, we make the following conjecture :

Conjecture 5.1. *Let $X = (X_{\text{past}}, X_{\text{future}})$ be a Markov chain, composed of the two consecutive Markov chains X_{past} and X_{future} of respective length τ and $T - \tau$, both conditioned on a discrete random variable G. Then, there exists a function $g(T)$ such that, if τ is dominated by $g(T)$, the partial uncertainty coefficients $U(X_{\text{future}}|G; X_{\text{past}})$ and $U(G|X_{\text{future}}; X_{\text{past}})$ are T-dual, and they are not otherwise.*

5.8.9 Evaluation and bias of the mutual information in large systems

Effect of biased mutual information over the uncertainty coefficients

When an estimation of the mutual information is biased, it necessarily follows that an estimation of the resulting uncertainty coefficients will also be biased. Fortunately, we can show that the direction of the bias does not change either for the reconstructability $U(G|X)$ or the predictability $U(X|G)$. Suppose that $\mathcal{I}_\varepsilon = I(X;G)(1+\varepsilon)$ is an estimator of the mutual information, where $\varepsilon \in \mathbb{R}$ is a small bias which can be either positive or negative. Then, the corresponding estimators of the uncertainty coefficients, that we denote $\mathcal{P}_\varepsilon$ and $\mathcal{R}_\varepsilon$ for the predictability and the reconstructability, respectively, are

$$\mathcal{P}_\varepsilon = \frac{\mathcal{I}_\varepsilon}{H(X|G) + \mathcal{I}_\varepsilon} \tag{5.77}$$

and

$$\mathcal{R}_\varepsilon = \frac{\mathcal{I}_\varepsilon}{H(G)} = U(G|X)(1+\varepsilon)\,. \tag{5.78}$$

Note that we also suppose that $H(G)$ and $H(X|G)$ are not affected by the bias ε. For the first expression, we consider the first-order development of $\mathcal{P}_\varepsilon$ with respect to ε :

$$\mathcal{P}_\varepsilon = U(X|G)\left[1 + \Big(1 - U(X|G)\Big)\varepsilon - \mathcal{O}\left(\varepsilon^2\right)\right] . \tag{5.79}$$

Indeed, given that $U(X|G) \geq 0$, the leading biased term $\Big(1 - U(X|G)\Big)\varepsilon$ must have the same sign as ε. The second expression clearly shows that the bias of $\mathcal{R}_\varepsilon$ is exactly given by ε. Therefore, both $\mathcal{P}_\varepsilon$ and $\mathcal{R}_\varepsilon$ retain the direction of bias of $\mathcal{I}_\varepsilon$.

The variational mean-field (MF) estimator presented in Section IV.D of the main paper is biased and bounds the mutual information from below. For this reason, it is necessary to validate the MF estimator with another estimator, that is either exact or that bounds $I(X;G)$ from above. In doing so, we can estimate the gap between the MF lower bound and the upper bound to assess the magnitude of the bias. In the next section, we present one such upper bound estimator.

Upper bound : The stepping-stone algorithm

Whereas the MF estimator represents a biased estimator of the posterior probability $P(G|X)$, there exists other MCMC techniques that tackle the problem of estimating the evidence probability directly. The one we consider in this paper is obtained from an *annealed importance sampling* (AIS) procedure called the stepping-stone (SS) algorithm [332].

The procedure of the stepping-stone algorithm takes advantage of the fact that it is possible to sample efficiently from the posterior distribution $P(G|X)$ using MCMC (see main text). In order to compute an accurate estimator of the evidence probability $P(X)$, the procedure samples the space $\mathcal{G}$ according to $P_\beta(G|X)$, where $0 \leq \beta \leq 1$ is an inverse temperature parameter that dampens the influence of the likelihood such that

$$P_\beta(G|X) \propto [P(X|G)]^\beta P(G) . \tag{5.80}$$

The inverse temperature basically allows the Markov chain to navigate $\mathcal{G}$ efficiently to construct an accurate estimator of $P(X)$, that is where the graph samples are not all too close or too far from the maximum posterior. More specifically, the AIS estimator is defined by

$$P_{\text{AIS}}(X) = \prod_{k=1}^{K} \mathbb{E}\left[[P(X|G_k)]^{\beta_k - \beta_{k-1}}\right] , \tag{5.81}$$

where $0 = \beta_0 < \cdots < \beta_{K-1} = 1$ and the expectation is evaluated with respect to $G_k \sim P_{\beta_k}(G|X)$, for each k. Similarly to the mean-field estimator, we estimate this expectation by collecting a sample $\mathcal{Q}_k^{(m)}$ of Q graphs distributed according to $P_{\beta_k}(G|X = X^{(m)})$, for each k.

Taking the log of this equation gives us an estimator of the log-evidence probability, which we can use to compute the mutual information directly :

$$\log P_{\text{AIS}}(X) = \sum_{k=1}^{K} \log\left\{\mathbb{E}\left[[P(X|G = G_k)]^{\beta_k - \beta_{k-1}}\right]\right\} . \tag{5.82}$$

Although the estimator for P_{AIS} is unbiased, the one for the log-evidence probability introduces a bias :

$$\log P(X) \geq \log P_{\mathrm{AIS}}(X)\,. \tag{5.83}$$

This bias can be arbitrarily reduced by increasing K [332], although we found that doing so provides diminishing returns. Using the AIS estimator of the evidence probability, we obtain an AIS estimator of the mutual information such that

$$\begin{aligned} I(X;G) \leq \frac{1}{M}\sum_{m=1}^{M}\Big[\log P\Big(X = X^{(m)}|G = g^{(m)}\Big) \\ - \log P_{\mathrm{AIS}}\Big(X = X^{(m)}\Big)\Big]\,. \end{aligned} \tag{5.84}$$

Following Ref. [332], we use values of β_k distributed according to a beta distribution Beta(α, 1), where $\beta_k = \left(\frac{k}{K}\right)^{1/\alpha}$, such that increasing α controls how skewed around zero the sequence $\{\beta_k\}_{k=1..K}$ is. For Fig. 5.11, we fix $\alpha = 0.5$ and $K = 20$ and, for each value of β_k, we sample 1000 graphs from $P_{\beta_k}(G|X)$, proposing $5N$ moves in-between each sample (see main text).

Numerical results

Figure 5.11(a) shows the behavior of $I(X;G)$ in the Glauber dynamics on a small Erdős-Rényi random graph as approximated using the MF and AIS estimators, and compares them to an exact evaluation based on an explicit graph enumeration used in Fig. 5. As expected the two estimators provide a lower and an upper bound for $I(X;G)$, and these bounds are fairly tight.

Several caveats are in order. On the one hand, the bias of the AIS estimator can, in principle, be reduced arbitrarily by increasing the number K of temperature steps, but its evaluation becomes quickly computationally costly. On the other hand, the evaluation of MF estimator is comparatively quicker, but cannot be improved by further sampling. The AIS estimator is accordingly closer to the exact value throughout, but it can sometimes overestimate the mutual information above its upper bound since $H(X)$ is overestimated while $H(X|G)$ is not. The MF estimator can also yield negative values of $I(X;G)$ for small values of J—i.e., regimes where $H(G|X) \simeq H(G)$—due to an overestimated $H(G|X)$ becoming larger than $H(G)$.

Figure 5.11(b) shows the same experiment as in Fig. 5.11(a) but with larger graphs of $N = 100$ vertices and leads to similar observations : the AIS estimator is always greater than the MF estimator, and both estimators sometimes yields approximated values for $I(X;G)$ outside of the valid range [0,max $\{H(G), H(X)\}$]. Interestingly, these bounds are nevertheless fairly close to one another, as in the case $N = 5$.

5.8.10 Numerical estimation of the phase transition thresholds

We evaluate the phase transition thresholds of each dynamics using standard finite-size scaling techniques and Monte Carlo simulations (see Fig. 5.12). For Glauber, an adequate order parameter to visualize the phase transition is the magnetization $M \equiv \frac{1}{NT}\sum_{i,t}|2X_{i,t}-1|$, where the absolute value breaks the spin symmetry [33]. In this process, it is well known that the susceptibility of the order parameter M, given by

$$\chi_M = \frac{\mathbb{E}\left[M^2\right] - \mathbb{E}[M]^2}{\mathbb{E}[M]}, \tag{5.85}$$

diverges at the threshold $J = J_c$ of the phase transition for infinite size systems [33]. In finite systems, χ_M instead reaches a maximum at $J = J_c$. We use this fact to locate J_c and show the corresponding results in Fig. 5.12(a).

For the SIS dynamics, a similar finite-size scaling analysis can be carried out, but a suitable order parameter is rather the average state $\bar{X} \equiv \frac{1}{NT}\sum_{i,t} X_{i,t}$. We also use a definition of the susceptibility that is more convenient for spreading processes [89], given in terms of $\bar{X}$:

$$\chi_{\bar{X}} = \frac{\mathbb{E}\left[\bar{X}^2\right] - \mathbb{E}[\bar{X}]^2}{\mathbb{E}[\bar{X}]}, \tag{5.86}$$

which also diverges at the phase transition threshold $\lambda = \lambda_c$ for infinite size systems. We show the results for SIS in Fig. 5.12(b).

Finally, for the Cowan dynamics, we have a first-order phase transition characterized by a discontinuity of the order parameter $\bar{X}$ in the infinite size limit, and a bistable region bounded by two thresholds $\nu_c^b < \nu_c^f$. To find these two thresholds, we evaluate the order parameter $\bar{X}$ for varying values of the parameter ν, and find the location where the discontinuity occurs. We obtain the forward and backward branches by using different initial conditions, where the system is nearly inactive—with one active vertex—and completely active—with no inactive vertex—, respectively.

For the Cowan dynamics, it is important to mention that since we consider relatively small systems ($N = 1000$ vertices), the bistable region is not clearly defined. Hence, a system starting in the forward branch can jump on the backward branch with a non-zero probability. This is why the expected discontinuity at the threshold is, in fact, populated (see Fig. 5.12(c)). This finite-size effect should be reduced for considering larger systems, but increasing N is unfortunately too computationally costly at the moment. Hence, to get a reasonable estimation of the thresholds in this scenario, we uniformly sample the set of ν's, compute $\langle \bar{X} \rangle$ for all values of ν and find the point ν^* corresponding to the maximum gap between two points. Then, to increase the precision or this estimation, we zoom on a region centered at ν^* and do it again, until it converges. This method provides reasonably accurate thresholds for our purposes.

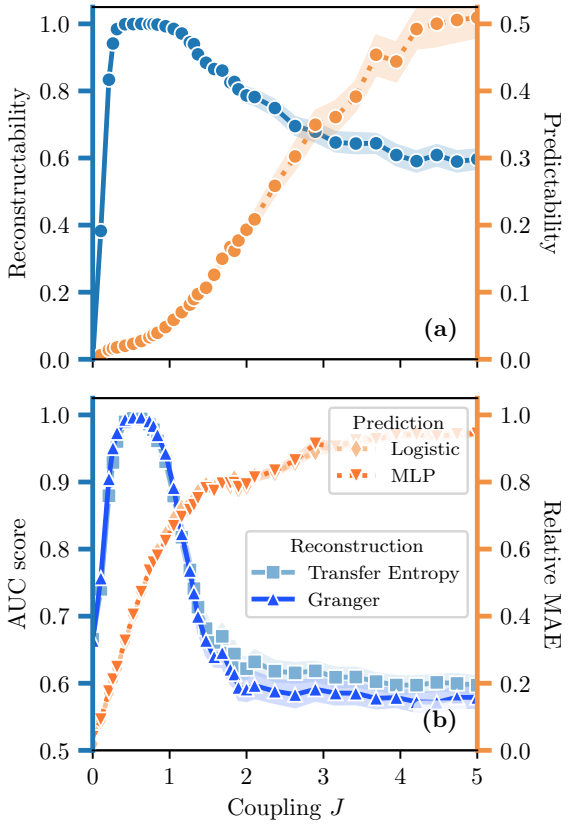


FIGURE 5.8 – **Coupling-duality versus prediction and reconstruction performance measures in the Glauber dynamics** : (a) coupling-duality between $U(G|X)$ (left axis) and $U(X|G)$ (right axis), (b) duality between reconstruction AUC score (left axis) and prediction relative mean absolute error (right axis) for different algorithms as indicated by the legend. We fixed the number of nodes to $N = 5$, the number of edges to $E = 5$ and the number of time steps to $T = 100$, and we averaged each point over 1000 simulations. See Section IV.B for further detail about the performance measures and algorithms.

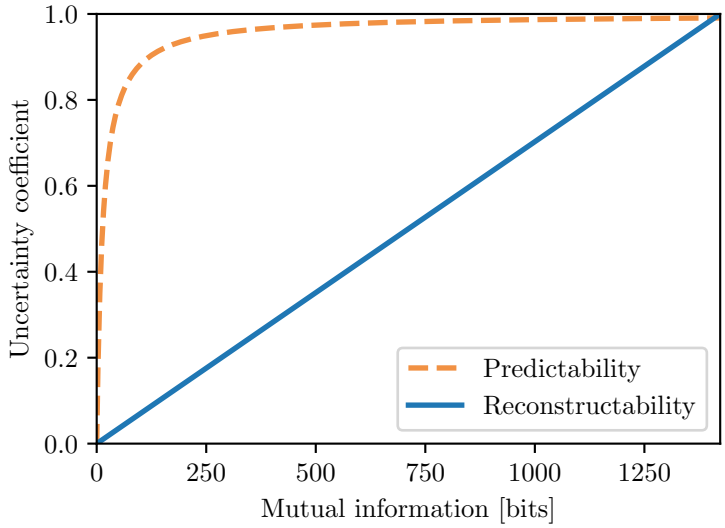


FIGURE 5.9 – **Scaling of the uncertainty coefficients with the mutual information**. We show the approximations of $U(X_\epsilon|G)$ and $U(G|X_\epsilon)$, provided by Eqs (5.66) and (5.68), respectively. For this illustration, we fixed $\log|\mathcal{G}| = \log\binom{N(N-1)/2}{E}$ and $\mathcal{A}_\epsilon = \log(\alpha|\mathcal{X}_0|)$, with $N = 100$, $E = 250$ and $\alpha = |\mathcal{X}_0| = 100$. While the choice of $\log|\mathcal{G}|$ is easy to justify—we assume that $\mathcal{G}$ is the set of all graphs with $N = 100$ nodes and $E = 250$ edges—, the choice for $\mathcal{A}_\epsilon$ needs more explanation. It can be interpreted as a system where, for each initial condition, there are on average α trajectories that are in the neighborhood of a given trajectory deterministically generated by some graph g. Here, we assume that α and $|\mathcal{X}_0|$ are equal to N for simplicity.

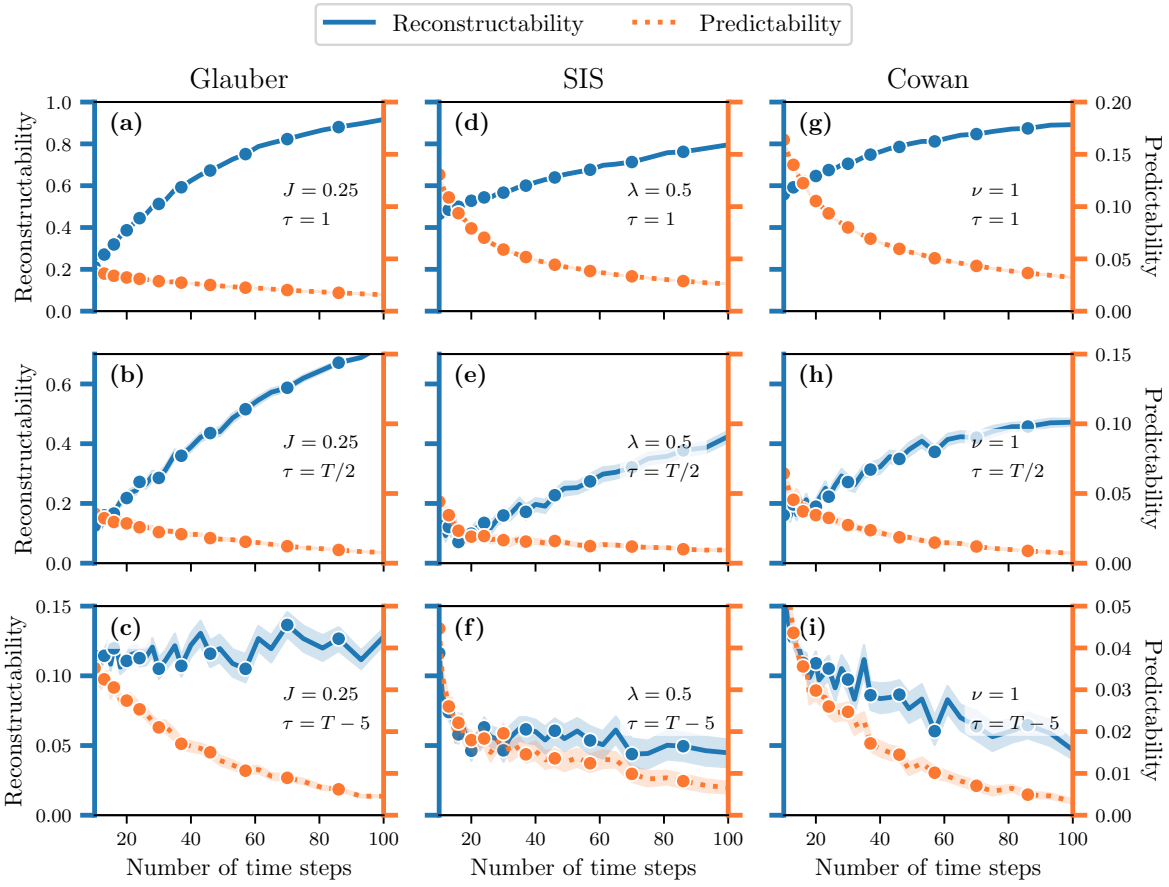


FIGURE 5.10 – **Existence of the T-duality in the past-dependent case, for binary dynamics evolving on small Erdős-Rényi random graphs** : (a-c) Glauber dynamics, (d-f) SIS dynamics and (g-i) Cowan dynamics. Like Fig 5 of the main paper, each panel shows the reconstructability coefficient $U(G|X) \in [0,1]$ (blue) and the predictability coefficient $U(X|G) \in [0,1]$ (orange) as a function of the number of time steps T. In each row, we change the value of the length τ of the past Markov chain X : (a,d,g) $\tau = 1$, (b,e,h) $\tau = T/2$ and (c,f,i) $\tau = T - 5$. We used graphs of $N = 5$ vertices and $E = 5$ edges and each symbol corresponds to the average value measured over 1000 samples. We also show different values of the coupling parameters, as indicated on each figure.

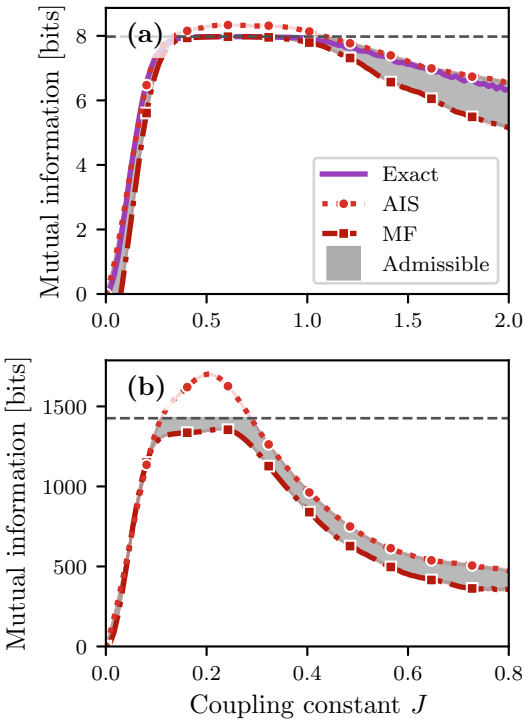


FIGURE 5.11 – **Estimators of the mutual information in the Glauber dynamics on Erdős-Rényi graphs as a function of the normalized coupling parameter** $J\langle k\rangle$: (a) $N = 5$, $E = 5$ and $T = 100$ (b) $N = 100$, $E = 250$ and $T = 1000$. The solid line in (a) corresponds to the exact evaluation of $I(X;G)$ and is the same line as the one in Fig. 5(a). The circles and square in both (a) and (b) represent the values of $I(X;G)$ computed using the AIS and the MF estimators, respectively. The dashed line indicates the upper bound of $I(X;G)$, i.e., $\max\{H(G), H(X)\}$. We also show with a gray area the admissible values of $I(X;G)$ bounded by the biased MF and AIS estimators.

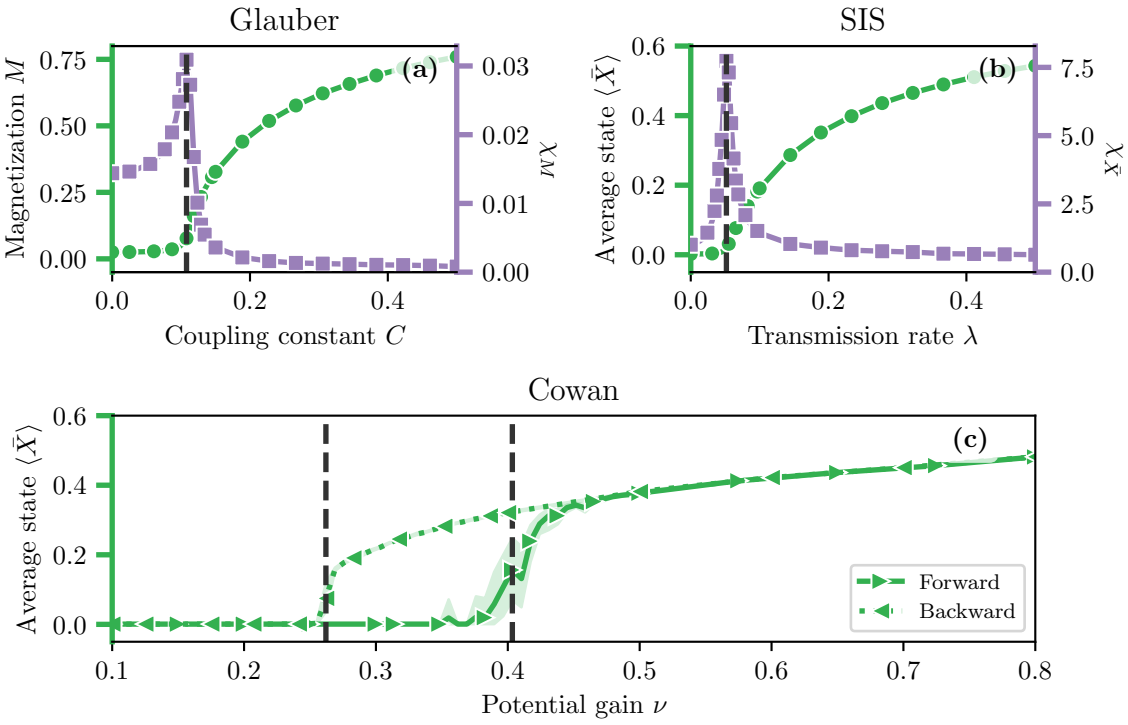


FIGURE 5.12 – Numerical evaluation of the phase transition thresholds : (a) Glauber dynamics, (b) SIS dynamics, (c) Cowan dynamics. For panels (a) and (b), the left axis (green) shows the order parameter (green circles), and the right axis (purple) shows the susceptibility (purple squares). For panel (c), only the order parameter is shown but for both the forward (right triangle) and backward (left triangle) branches. The values of the thresholds are indicated by the vertical dashed lines. We used the same parameters as those of Fig. 7 of the main paper, but increased the number of steps $T = 10^4$ to better sample from the dynamics. Each marker has been average over 48 realizations.

Chapitre 6

Limites dans la reconstruction des réseaux complexes

Article original :
On the reconstruction limits of complex networks
Charles Murphy, Vincent Thibeault, Simon Lizotte, François Thibault, Patrick Desrosiers, Antoine Allard
Département de Physique, de Génie Physique et d'Optique, Université Laval, Québec (Qc), Canada G1V 0A6
Référence : arXiv:2501.01437 [211]
(§ 6.3-6.10.16) [1]

1. Ces sections contiennent le contenu original de l'article. Celui-ci n'a été modifié que pour se conformer au format exigé par la Faculté des études supérieures et postdoctorales de l'Université Laval.

6.1 Avant-propos

Le formalisme développé au Chapitre 5 a permis de mettre en évidence la relation entre la prévisibilité et la reconstructiblité dans les systèmes complexes. Or, ce formalisme, basé sur la théorie de l'information, est général et peut être spécialisé pour étudier la reconstruction des réseaux complexes—i.e., l'inférence du graphe des interactions entre les noeuds à partir d'observations—, ce que nous faisons dans le présent chapitre. L'article ici présent est ainsi une suite naturelle du Chapitre 5. Il est également le fruit d'une retraite collaborative, principalement organisée par ses quatre premiers auteurs et ayant eu lieu à Matane en novembre 2021. Les membres de cette collaboration se sont rencontrés à plusieurs reprises pour accélérer l'avancement du projet et la réalisation numérique des algorithmes utilisés.

Au départ, l'objectif du projet était de faire la lumière sur un phénomène observé dans la Réf. [242]. Dans cet article, il est montré que le modèle stochastique par blocs—fréquemment utilisé pour la détection de communauté [148, 240]—permet une reconstruction plus précise à condition que la structure en communauté soit inférée conjointement avec le graphe. Intuitivement, on croyait que de reconstruire les deux simultanément représentait un problème plus difficile que de simplement reconstruire le graphe. D'après notre analyse, il s'est avéré que ce n'était pas nécessairement le cas. En effet, la qualité de la reconstruction avec le modèle stochastique par blocs dépend du graphe qu'on tente de reconstruire.

Ce chapitre présente un article en révision à la revue *Physical Review X*, dont l'objectif est d'étudier la nature des limites dans la reconstruction des réseaux complexes. Toujours armé de l'information mutuelle entre le graphe et les observations, nous montrons que celle-ci est une borne supérieure de la performance des algorithmes de reconstruction telle que mesurée par la similarité entre le graphe reconstruit et le graphe original. L'information mutuelle est fondamentalement liée au processus réel de génération des données. Celui-ci joue donc un rôle central dans la limite de reconstruction. Par contre, nous n'y avons généralement pas accès dans le contexte empirique, où les données sont limitées et tirées d'un processus inconnu. C'est pourquoi nous avons adapté notre formalisme de sorte à estimer l'information mutuelle à partir des données observées—une mesure que nous appelons l'*indice de reconstruction*.

L'objectif sous-jacent de cet article est fondamentalement différent de celui du Chapitre 5. Ici, nous sommes motivés à appliquer nos outils informationnels dans le contexte empirique, ce qui n'était précédemment pas le cas. Cette tendance se poursuivra dans la prochaine partie de la thèse, où nous intégrerons également des outils puissants provenant de l'apprentissage profond pour modéliser directement les dynamiques sur les réseaux à partir de données.

Acronyme	Traduction française
TDG	Processus générateur des données

TABLEAU 6.1 – Glossaire des acronymes utilisés au Chapitre 6.

Symbole	Description
N, E	Nombre de noeuds et de liens, respectivement
g^*, x^*	Graphe et données observés, respectivement
$\mathcal{G}, \mathcal{X}$	Ensembles de tous les graphes et observations possibles
G	Graphe aléatoire utilisé pour l'inférence
X	Processus stochastique binaire dépendant de G, utilisé pour l'inférence
$M = (G, X)$	Modèle bayésien pour la reconstruction de réseau
$M^* = (G^*, X^*)$	Processus génératif des données observées
ϕ, θ	Paramètres auxiliaires de X et G, respectivement
$\alpha(n, m), \beta(n, m)$	Probabilités d'activation et de désactivation, respectivement, dépendantes du nombre de voisins inactifs n et actifs m
$H(X)$	Entropie de la variable aléatoire X
$I(X; G)$	Information mutuelle entre les variables aléatoires X et G
Ψ^*	Reconstructibilité du processus $M^* = (G^*, X^*)$
p_e	Probabilité d'erreur
$h(p)$	Entropie binaire
$\boldsymbol{a}$	Matrice d'adjacence d'éléments $a_{ij} = [\boldsymbol{a}]_{ij}$
$\boldsymbol{\pi}(x)$	Matrice de probabilités *a posteriori* de l'occupation des liens sachant les données x
$\mathcal{L}(x, y)$	Fonction d'erreur *a posteriori*
$\mathcal{I}_M(x)$	Gain d'information du modèle M
$\Lambda_M(x)$	Normalisation de $\mathcal{I}_M(x)$

$\psi_M(x)$	Indice de reconstruction du modèle M
B_{M_1,M_2}	Facteur de Bayes
$\mathcal{H}_{M^*,M}$	Entropie croisée entre le processus génératif M^* et le modèle M

TABLEAU 6.2 – Glossaire des symboles utilisés dans au Chapitre 6.

6.2 Résumé

La reconstruction de réseaux consiste à retrouver la structure cachée d'interaction d'un système à partir d'observations empiriques telles que des séries temporelles. De nombreux algorithmes de reconstruction ont été proposés, bien que moins de recherches aient été consacrées à décrire leurs limitations théoriques. Dans cet article, nous adoptons un point de vue basé sur la théorie de l'information et définissons la *reconstructabilité* :la fraction d'information structurelle pouvant être extraite à partir des données. La reconstructabilité est liée au processus générateur de données véritable et définit une limite de reconstruction théorique, c'est-à-dire une borne supérieure de l'information mutuelle entre le graphe sous-jacent véritable et tout graphe reconstruit à partir d'observations. En l'occurrence, aucun algorithme ne peut outre-passer cette limite de performance. Ces concepts nous conduisent à un méthode numérique fondée sur la sélection de modèle permettant d'évaluer la validité des réseaux reconstruits empiriquement. Nous caractérisons cette méthode et la testons sur des séries temporelles et des réseaux empiriques.

6.3 Abstract

Network reconstruction consists in retrieving the hidden interaction structure of a system from observations. Many reconstruction algorithms have been proposed, although less research has been devoted to describe their theoretical limitations. In this work, we adopt a first-principles perspective to define the *reconstructability*: The fraction of structural information recoverable from data. The reconstructability is related to the true data generating process and delineates an information-theoretic reconstruction limit, i.e., the upper bound of the mutual information between the true underlying graph and any graph reconstructed from observations. In turn, no algorithm can exceed this limit. These concepts lead us to a principled numerical method to assess the validity of empirically reconstructed networks, based on model selection. We characterize this method and test it on empirical time series and networks.

6.4 Introduction

Complex systems, such as the brain, are naturally represented by complex networks that encapsulate intricate interactions between neurons or brain regions [25, 26, 190, 290]. Network representation unlocks a variety of tools with the potential to unravel not only brain functions and diseases [93, 136, 303], but also gene expressions [321], epidemics [242, 250] and the propagation of financial distress [214]. The main challenge is that such network representations are seldom measurable experimentally. For example, the collected data are often indirect observations of the interactions, taking the form of counts of interactions or times series. Moreover, these data are noisy, thereby making the network reconstruction task even more intricate [221, 236, 241, 339].

The task of reconstructing networks has been revisited many times, using different assumptions and approaches. Typically, network reconstruction is performed on multivariate time series [47], a procedure related to causal inference [85]. In this approach, we assume that the dynamics of the node activities is driven by some hidden network structure that we want to uncover. Many heuristics have been proposed to perform network reconstruction from time series—involving scores like correlation [160], Granger causality [277] or transfer entropy [274] between nodes—which are then thresholded to obtain a reconstructed network. Other approaches proposed statistical frameworks to infer network from time series using graphical models [1, 6, 29, 44], fully Bayesian models [242] and deep learning models [155].

Another promising avenue for network reconstruction involves using pairwise observations for quantifying the uncertainty of empirical graphs. In this setting, noisy pairwise observations are used to predict missing edges [58, 124, 187], estimate the edge uncertainty [220] and reconstruct the network altogether [236]. As for network reconstruction from time series, heuristics have also been considered for pairwise data (for example, in Ref. [118]). Recently, there has also been a resurgence in the interest towards Bayesian frameworks. For instance, Ref. [339] proposed a general and Bayesian procedure to infer networks leveraging the conditional independence of the edges, which was then applied to a plant-pollinator network [342]. Reference [185] extended this framework to the reconstruction of hypergraphs with noisy observations and showed the benefit of including higher-order interactions for modeling pairwise measurements. Other works used the modular structure of complex networks to improve the performance of their models [58, 124, 241]. To this date, the field of reconstruction of noisy networks remains a flourishing one.

As more technical progress is being made, more work is being dedicated to the theoretical challenges of network reconstruction. For instance, Ref. [236] proposed a unifying framework for linking network data to network science theories, in which Bayesian network reconstruction is core and where they argue the suitability of the models is essential for network reconstruction. Additionally, Ref. [251] found that network reconstruction, on the

basis of predicting the outcome of a deterministic dynamical process, can lead to a wide range of networks. This aligns with the observations of Ref. [13] and earlier computational neuroscience findings [252] of network degeneracy [67], where diverse synaptic connection patterns can yield similar neuronal activity, illustrating the non-unique relationship between network structure and function.

An information-theoretic description of random networked processes has recently led to a broader understanding of this so-called structure-function relationship, linking predictability to reconstructability in complex networks through mutual information [212]. This description revealed a duality between reconstructability and predictability, showing that in certain parameter ranges, an increase in predictability corresponds to a decrease in reconstructability, and vice versa. In this work, we aim to further this theory on the network reconstruction front, providing an information-theoretic bedrock to such applications.

In the network reconstruction problem, our task is to infer a graph likely to have generated some observed data. We first formally present this problem and the related mathematical concepts in Sec. 6.5. Then, we revisit and adapt the framework of Ref. [212] in Sec. 6.6, allowing us to interpret the reconstruction problem in information-theoretic terms. In doing so, we demonstrate the existence of an algorithm-independent limit to network reconstruction—the reconstructability Ψ^* [see Fig. 6.1(a,b)]—which bounds from above the mutual information between the true underlying network and the reconstructed one. Inspired by this limit, we present and characterize in Sec. 6.7 a principled numerical method to assess the validity of reconstructed networks in an empirical setting (i.e., hidden generative process, one or few observations). Our method is based on the reconstruction index, denoted ψ_M for some reconstruction model M, which is an approximation of the reconstructability Ψ^* that measures the dispersion of the reconstructed graph ensemble. The reconstruction index is shown to predict the reconstruction error *without* knowing the true underlying graph [see Fig. 6.1(c,d)], assuming our modeling assumptions are aligned with the true underlying process. Finally, we apply our method to real systems in Sec. 6.8.

6.5 Network reconstruction

We formulate the network reconstruction problem following the illustration in Fig. 6.1(c). Let $g^* \in \mathcal{G}$ be some graph of N nodes that represents the structure of the interactions between each pair of components in a system, where $\mathcal{G}$ is the set of all graphs of N nodes. The graph may be directed and weighted [245], but we restrict our discussion to undirected and unweighted, for simplicity. This graph structure is *a priori* unknown to us, although it is indirectly observed through some data, denoted x^*, which may take any value in the set $\mathcal{X}$. This data can take many forms—time series, pairwise measurements, etc.—and we assume it to be generated using g^*. In what follows, we will further assume that x^* is in fact a $N \times T$

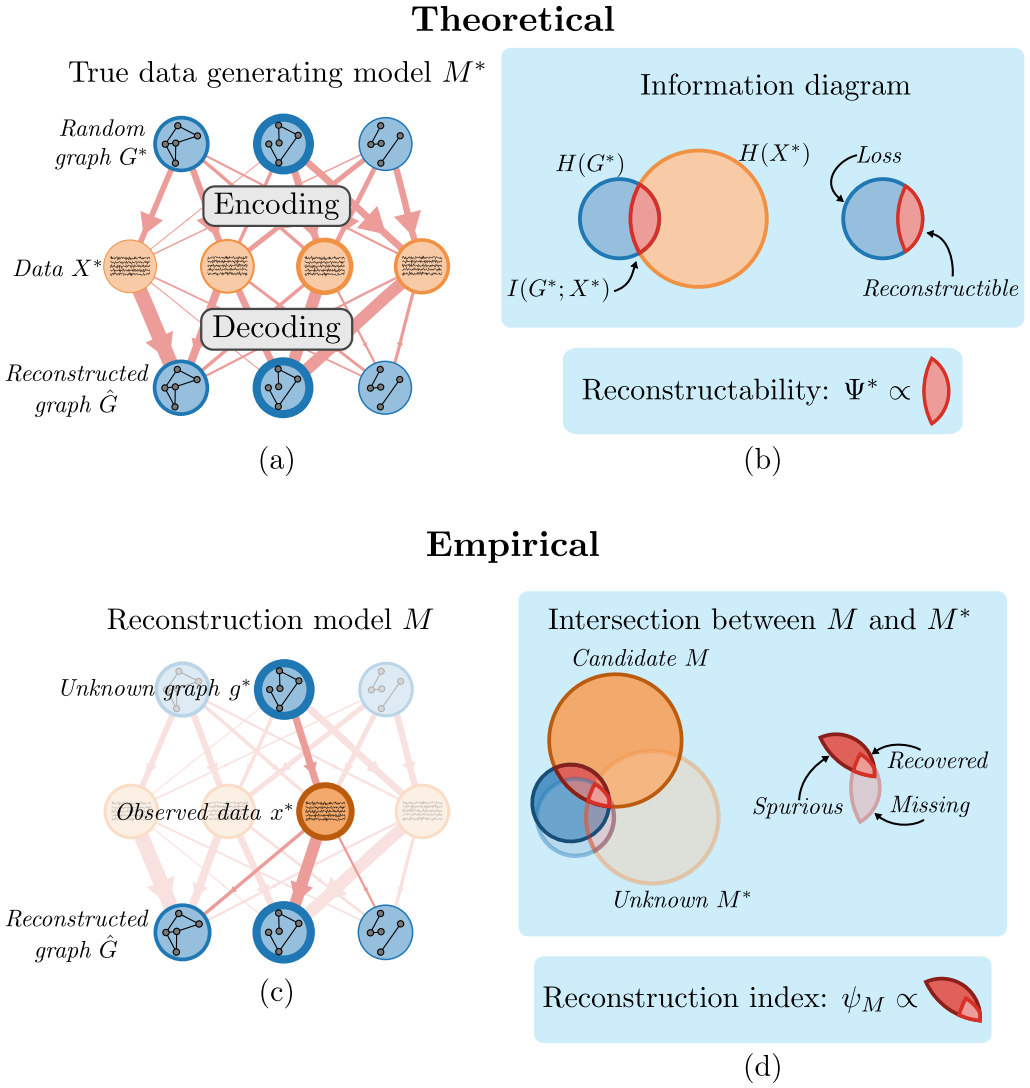


FIGURE 6.1 – Illustration of the network reconstruction context from (a, b) a theoretical perspective and (c, d) an empirical perspective. Panel (a) sketches how the true data generating model (TDG) M^* operates, first by generating a graph, then by encoding it into the observations, and finally using these to decode—or reconstruct—the graph. The thickness of the contour line around each graph and data example indicates the probabilities $P(G^*)$ (top and bottom layers) and $P(X^*)$ (middle layer). The thickness of the edges connecting the graphs to the data illustrate the likelihood of the TDG $P(X^*|G^*)$, and those connecting the data to a reconstructed graphs, some distribution $P(\hat{G}|X^*)$. In panel (b), we illustrate in red the reconstructible information, utilizing an information-theoretic perspective. This information is part of the total information of G^* and X^*—in blue and orange, respectively—and is also a fraction of the partial information of G^* needed to completely reconstruct it (blue and red). Panels (c, d) show the analog of (a, b) when the model M^* is unknown, where in panel (c) a single datum is accessible and reconstruction is done by a candidate model M, a priori different from M^*. In panel (d), we illustrate how M and M^* may overlap in the information they reconstruct—the information intersection (i.e., the correctly recovered information) and difference (i.e., the missing or spurious information). The reconstructability Ψ^* and the reconstruction index ψ_M are defined in subsection 6.6.2 and subsection 6.7.1, respectively.

matrix corresponding to N coupled time series of length T, but we stress that our analysis may apply to any type of networked data. The goal of network reconstruction is to infer the graph g^* from the data x^*.

Taking a Bayesian perspective, the plausibility of a given graph $g \in \mathcal{G}$, given the observations x^*, is described by the posterior probability $P(G = g|X = x^*)$, i.e., the output of the Bayesian inference procedure. A Bayesian reconstruction model is a generative process that consists of two discrete random variables G and X, representing the graphs and the data respectively, and thus defines their joint probability mass function $P(G, X) = P(G)P(X|G)$, where $P(G)$ is the graph prior and $P(X|G)$, the data likelihood. By virtue of Bayes' theorem, the posterior $P(G|X)$ is factored as follows :

$$P(G|X) = \frac{P(X|G)P(G)}{P(X)}\,, \tag{6.1}$$

where $P(X)$ is the normalization factor, called the evidence.

6.5.1 Data generation process

A Bayesian reconstruction model, composed of the two random variables G and X, reflects our assumptions about how the unobserved graph and observed data came to be. In other words, the model $M = (G, X)$ represents a generative process for the pairs (g^*, x^*) [see Fig. 6.1(a)]. Accordingly, there are many reconstruction models that may describe the data to various degrees of correctness. Throughout this work, we assume the existence of a unique generative process, referred to as the true data-generating (TDG) model $M^* = (G^*, X^*)$, which *truly* produced the graph g^* and the observed data x^* with probabilities $P(G^* = g^*)$ and $P(X^* = x^*|G^* = g^*)$, respectively. In turn, any reconstruction model may be described by a reconstructed random graph $\hat{G}$, that depends on X^*. The complete process consisting of the graph and data generation followed by the reconstruction of the graph is therefore described by the random variable triplet $(G^*, X^*, \hat{G})$, whose joint probability distribution is

$$P(G^*, X^*, \hat{G}) = P(G^*)P(X^*|G^*)P(\hat{G}|X^*)\,. \tag{6.2}$$

In general, the distribution $P(\hat{G}|X^*)$ may be any distribution over $\mathcal{G}$, but for Bayesian models such as M, it is precisely given by the posterior of M :

$$P(\hat{G} = g|X^* = x^*) = P(G = g|X = x^*) \tag{6.3}$$

for all $g \in \mathcal{G}$, such that $P(G|X)$ is given by Eq. (6.1). Note that the reconstructed random graph $\hat{G}$ and the random graph G of model M conceptually describe two different quantities, although they are related through Eq. (6.3). Indeed, $\hat{G}$ appears in the reconstruction process involving the TDG and G is part of a completely separate generative process. In other words, $\hat{G}$ depends explicitly on M^*, via $P(\hat{G}|X^*)$, whereas M is independent from it

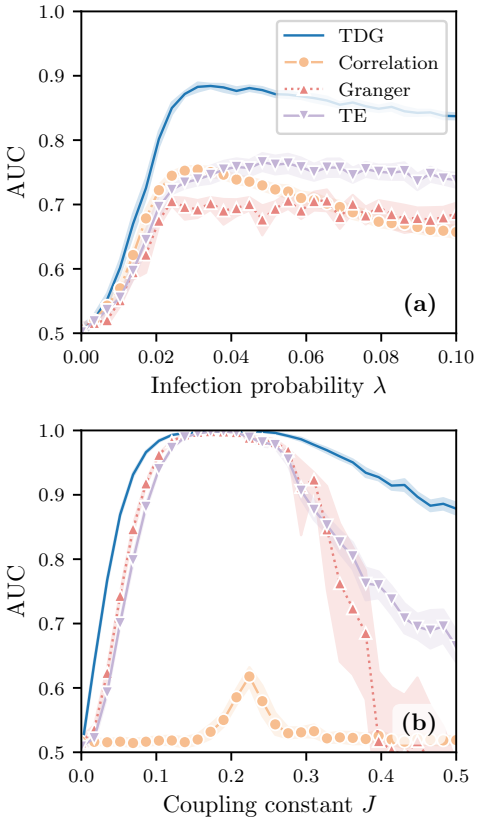


FIGURE 6.2 – Performance comparison between the TDG model and heuristic reconstruction algorithms. In both panels, we show the area under the receiver operating characteristic curve (AUC) of the reconstruction models as a function of a parameter of the model that generated the data : (a) the Susceptible-Infection-Susceptible (SIS) dynamics and (b) Glauber dynamics (see Table 6.4 for the definitions of the dynamics). We generated graphs of $N = 100$ nodes with the Erdős-Rényi model (Eq. (6.4)), where the number of edges is $E = 250$. We also generated time series of $T = 500$ time steps; the parameters other than the infection probability λ and the coupling constant J (which are fixed within the likelihood during the inference of the TDG) are specified in Table 6.4. Each data point corresponds to the AUC average over 24 reconstruction experiments, each experiment with different realizations of G^* and X^*, and the shaded regions around the points show a 90% confident interval from the mean. For further technical details, see Sec. 6.5.1.

(i.e., $P(G, X|G^*, X^*) = P(G, X)$). The consideration that $\hat{G}$ is, in fact, resulting from a Bayesian procedure through a generative model M, instead of any—potentially nongenerative—algorithm such as the inverse correlation method [160], will prove useful in the following sections.

From an information-theoretic perspective, data generation encodes information about the graph G^* into potentially noisy observations X^*, while network reconstruction decodes these observations back into a graph $\hat{G}$ as shown in Fig. 6.1(a, b). The encoding of G^* into X^* is generally lossy, meaning that only a fraction of its information can be recovered; the rest being lost in the process. In turn, any reconstruction model M different in distribution from M^* may therefore recover a fraction of the reconstructible information while potentially introducing spurious information through their inductive biases [see Fig. 6.1(c,d)], resulting in a degradation of performance.

This is well shown in Fig. 6.2 through reconstruction performance, where we used two synthetic TDG processes to compare the TDG reconstruction model performance with that of three heuristic reconstruction algorithms. In this experiment, we sample the true graph with probability

$$P(G^*) = \left(\binom{N}{2} \atop E \right)^{-1}, \tag{6.4}$$

which corresponds to the Erdős-Rényi (ER) model, with E being the (given) number of edges in the graph; and sample time series from the Susceptible-Infected-Susceptible (SIS) model in panel (a) and Glauber models in panel (b). For further details regarding the graph and data models, see Appendices 6.10.1 and 6.10.5. Then, assuming a given reconstruction model, we compare the reconstructed graph with original one using the area under the receiver operating characteristic curve (AUC) to measure reconstruction performance. We repeat this set up for many parameter values to populate the AUC performance curves in Fig. 6.2. As a comparison, we use three different well-known reconstruction algorithms : the correlation matrix method [160], Granger causality method [274] and the transfer entropy method [277] (see Appendix 6.10.6 for details). The results in Fig. 6.2 show quite unambiguously and unsurprisingly that the TDG model outperforms the reconstruction heuristics.

Yet, even the TDG reconstruction model cannot reconstruct the graph perfectly. For instance, in Fig 6.2(b), the AUC of the TDG model tends to $\frac{1}{2}$—equivalent to random guessing—when the coupling also goes to zero. In this scenario, X^* and G^* are independent and it is actually impossible to reconstruct the graph, since any graph could have generated the data with the exact same probability. The same phenomenon occurs to a lesser extent for the other coupling values as well as for the SIS dynamics, where the TDG model performance is imperfect for every infection probability. These imperfections are attributed to the lost information in the encoding of G^*; no model can extract more information than what is contained in X. In

practice, the encoding's loss stems from many sources, for example noise in the dynamics and degeneracy, where many networks lead to similar dynamics. The degeneracy phenomenon is well-established in computational neuroscience [67, 252] and has more recently appeared in network science [251] too. A reconstruction limit independent of the reconstruction algorithm clearly exists, where a perfect reconstruction is simply not attainable even in the best-case scenario. This is a key insight that we will explore in the following sections (especially Sec. 6.7).

6.5.2 Reconstructing a single edge

To gain better intuition about this reconstruction limit, we consider the reconstruction of a graph that may only contain a single edge. Let G^* be a random graph of two nodes, that may be connected by a single edge with probability p, and disconnected with probability $1-p$. This edge is observed through a noisy process $X^* = (X_1, ..., X_T)$ with T time steps, where X_i is a binary variable that takes the value 1 if the edge has been observed and 0 otherwise. We assume that the noisy process can induce true positives and false positives, each with known probabilities q and r, respectively, making the reconstruction problem more challenging. The likelihood $P(n|a)$ that the edge has been observed n times, given that it is present ($a = 1$) or not ($a = 0$), is a binomial distribution :

$$P(n|a) = \binom{T}{n} \Big[aq + (1-a)r\Big]^n \Big[1 - aq - (1-a)r\Big]^{T-n}. \tag{6.5}$$

Note that this model possesses a symmetry where interchanging q and r and mapping $a \to 1-a$ leaves the likelihood invariant. However, we avoid this non-identifiability issue by not inferring p and r.

To calculate the posterior probability of the edge being present, we find the evidence of the data

$$P(n) = \sum_{a=0}^{1} P(n|a)P(a) = \binom{T}{n} q^n (1-q)^{T-n} \Big[p + \eta^{T-n}\lambda^n(1-p)\Big], \tag{6.6}$$

where

$$\lambda = \frac{r}{q} \quad \text{and} \quad \eta = \frac{1-r}{1-q}. \tag{6.7}$$

This leads to the posterior probability of the edge being present

$$P(a=1|n) = \frac{p}{p + \eta^{T-n}\lambda^n(1-p)}. \tag{6.8}$$

Figure 6.3 shows the behavior of $P(a=1|n)$ when varying the number n of times the edge is observed and the true positive probability q. Assuming that $r < q$, observed edges are mostly true positives and thus the edge is predicted to exist if n is sufficiently large; otherwise, it is not since the expected number of true and false positives don't match the observations.

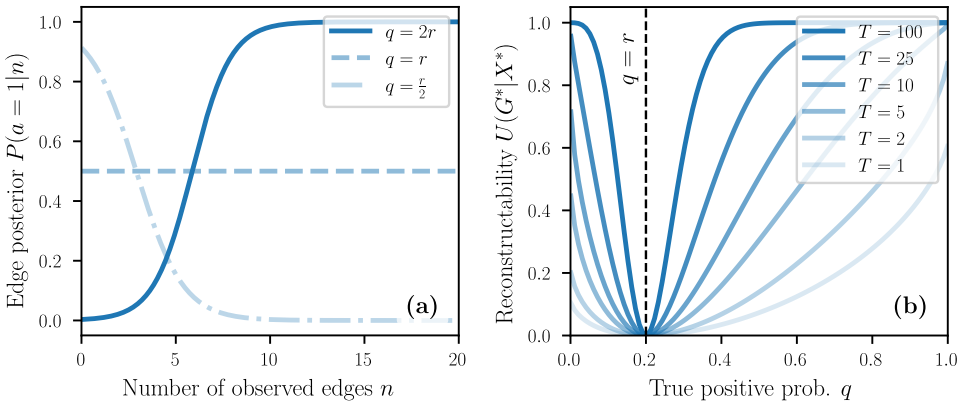


FIGURE 6.3 – Posterior probability of a reconstructed edge : (a) Posterior versus the number of times n the edge has been observed, (b) reconstructability of the edge versus q. In panel (a), we fixed the number of observations $T = 20$, the prior edge occupancy probability $p = \frac{1}{2}$ and the false positive probability $r = 0.2$. We varied the true positive probability such as $q \in \{2r, r, \frac{r}{2}\}$ (solid, dashed and dotted lines, respectively). In panel (b), we show the reconstructability curves for different numbers of observations T as indicated in the legend. The vertical dashed line indicates the value of q for which the edge is not reconstructable, i.e., when the true positive and false positive probabilities are the same—i.e., $q = r$.

Conversely, if $r > q$, then most observed edges are false positives, meaning that g^* is more likely to contain an edge when n is small. Interestingly, the edge becomes more challenging to reconstruct as q gets closer to r, where the probability to reconstruct the edge approaches $\frac{1}{2}$ (see Fig. 6.3(a)). In this regime, a and $1 - a$ are interchangeable and it becomes impossible to tell if the edge exists or not—any attempt at reconstructing this graph would be unfruitful. This is precisely the intuition we want to capture with the reconstruction limit : When is there enough information to properly reconstruct the structure, or to what extent is a system's structure reconstructible ? In the next section, we present an information-theoretic framework that quantifies this limit.

6.6 Information-theoretic reconstruction limits

As discussed above, we can think of the TDG process X^* as a noisy encoding of the true graph G^*. This amount of encoded information is fundamentally limiting our ability to reconstruct G^* accurately ; it is impossible to recover more information than what is contained in the data. This also means that the limit is independent of the reconstruction models or algorithms. Any reconstruction algorithm therefore aims to extract as much of the encoded information as possible, some being more efficient than others.

6.6.1 Entropy

Our goal is to formalize this intuition in information-theoretic terms. In information theory, information is related to the concept of entropy, which measures the uncertainty of a random

variable. For a random variable G, the entropy $H(G)$ is expressed as

$$H(G) = -\mathbb{E}_G[\log P(G)] = -\sum_{g\in\mathcal{G}} P(G=g)\log P(G=g)\,. \tag{6.9}$$

The entropy $H(G)$ measured in bits (assuming $\log(x) \equiv \log_2(x)$, which will henceforth be the case) quantifies the minimal number of binary questions one needs to answer, on average, to perfectly identify the graph generated by G. When $H(G) = 0$, the random variable G can only yield one graph, meaning that $P(G = g) = 1$ for some g. One can also measure the conditional entropy of a random variable G, given another random variable X, as

$$H(G|X) = -\mathbb{E}_{X,G}[\log P(G|X)] = -\sum_{g\in\mathcal{G}}\sum_{x\in\mathcal{X}} P(G=g, X=x)\log P(G=g|X=x)\,. \tag{6.10}$$

Like $H(G)$, $H(G|X)$ also measures uncertainty, but this time assuming that X is known. In Bayesian terms, $H(G)$ is the entropy of the prior $P(G)$, while $H(G|X)$ is the entropy of the posterior $P(G|X)$.

6.6.2 Network reconstructability

Those information-theoretic tools can be used to define the reconstruction limit. Consider the mutual information between the true and the reconstructed random graphs

$$I(G^*;\hat{G}) = H(G^*) - H(G^*|\hat{G})\,, \tag{6.11}$$

where

$$H(G^*|\hat{G}) = -\mathbb{E}_{G^*,\hat{G}}\big[\log P(G^*|\hat{G})\big] \tag{6.12}$$

is the entropy of the true graph given the reconstructed one. The conditional probability $P(G^*|\hat{G}) = P(G^*,\hat{G})/P(\hat{G})$ is such that both $P(G^*,\hat{G})$ and $P(\hat{G})$ are marginal distributions of $P(G^*, X^*, \hat{G})$ [Eq. (6.2)]. Three observations regarding this performance measure are in order. First, the quantity $I(G^*;\hat{G})$ may be interpreted as measuring the similarity between the information contents of G^* and $\hat{G}$. The higher it is, the more similar G^* and $\hat{G}$ are and the better is the reconstruction. Conversely, when $I(G^*;\hat{G}) = 0$, it is minimized and both graphs are independent from one another. Note that similar mutual information measures have been used as a performance measure in the context of community detection for comparing pairs of partitions [143, 222].

Second, $I(G^*;\hat{G})$ is related to the probability of error, defined as

$$p_e = P(\epsilon)\,, \tag{6.13}$$

where $\epsilon = \mathbb{I}[G^* \neq \hat{G}]$, with $\mathbb{I}[\cdots]$ being the indicator function, denotes a binary random variable that takes the value 1 when $G^* \neq \hat{G}$ and 0 otherwise. This relationship can be shown through Fano's inequality [65] :

$$H(G^*|\hat{G}) \leq h(p_e) + H(G^*)p_e\,, \tag{6.14}$$

where $h(p) \equiv -p \log p - (1-p)\log(1-p)$ is the binary entropy. Indeed, given that $h(p_e) \leq 1$, modifying Fano's inequality yields

$$p_e \geq 1 - \frac{I(G^*; \hat{G}) + 1}{H(G^*)} \,. \tag{6.15}$$

This lower bound on the probability of error is minimized when $I(G^*; \hat{G})$ is maximized.

Third, using the data processing inequality [65], it is also related to the mutual information between G^* and X^* as follows :

$$I(G^*; \hat{G}) \leq I(G^*; X^*) \,, \tag{6.16}$$

where the mutual information upper bound is expressed as

$$I(G^*; X^*) = H(G^*) - H(G^*|X^*) \tag{6.17}$$

is the mutual information between the true graph G^* and the data process X^*. Intuitively, $I(G^*; X^*)$ quantifies the amount of reconstructible information that both X^* and G^* share—i.e., the amount of information that X^* contains about G^* [see Fig. 6.1(b)]. The mutual information $I(G^*; X^*)$ also sets the maximum in reconstruction performance as measured by $I(G^*; \hat{G})$: It is the reconstruction limit.

The mutual information $I(G^*; X^*)$ is itself bounded between 0 and $H(G^*)$ [65]. When $I(G^*; X^*) = 0$, X^* and G^* are independent and thus the data X^* contains no information about the graph G^*. In turn, it is impossible for any reconstruction model M to extract information from the data, regardless of its specification. When $I(G^*; X^*) = H(G^*)$, the data X^* contains all the information about the graph G^*. In this case, it is in principle possible to perfectly reconstruct the graph without any error, assuming the model M is optimal, i.e., it can extract all the available information. We will further explore this notion of optimality in Sec. 6.6.3.

Since the value of $I(G^*; X^*)$ depends on the amount of information $H(G^*)$ that needs to be extracted, it is easier to reason about it in terms of proportions. Thus, we define the *reconstructability* Ψ^* of G^* from X^* as the uncertainty coefficient

$$\Psi^* = \frac{I(G^*; X^*)}{H(G^*)} \,. \tag{6.18}$$

The reconstructability has been described thoroughly in Ref. [212] and helped unveiling a special duality between our ability to predict the time evolution of a system and our ability to reconstruct the interactions between its constituents. As it is a normalized version of the mutual information upper bound $I(G^*; X^*)$, the reconstructability is bounded between 0 and 1. When $\Psi^* = 0$, any attempt at reconstruction is futile, whereas it is theoretically possible to decode all the information when $\Psi^* = 1$. As such, the reconstructability is a measure of the average proportion of information that can be extracted from the data about the graph.

For instance, when it is equal to $\frac{1}{2}$, it precisely means that half of the graph information is, on average, contained in the data and that in turn half of it can possibly be reconstructed. We stress that $\Psi^* = \frac{1}{2}$ may not be directly interpreted as half of the graph's edges being reconstructible. Rather, information may generally be distributed in a heterogeneous way over the graph's structure, as a single bit of information may reconstruct more than one edge in the graph depending on how correlated they are.

Going back to the earlier example of a single edge of Sec. 6.5.2, we can perform the complete calculation analytically. First, we can calculate the entropy of the prior,

$$H(G^*) = h(p)\,, \tag{6.19}$$

recalling that $h(p)$ is the binary entropy defined below Eq. (6.14), and the entropy of the posterior is

$$\begin{aligned} H(G^*|X^*) &= -\sum_{n=0}^{T}\sum_{a=0}^{1} P(a|n)P(n)\log P(a|n) \\ &= \sum_{n=0}^{T}\binom{T}{n} q^n(1-q)^{T-n}\left[p+\eta^{T-n}\lambda^n(1-p)\right] \\ &\qquad \times h\left(\frac{p}{p+\eta^{T-n}\lambda^n(1-p)}\right). \end{aligned} \tag{6.20}$$

This leads to the reconstructability of the edge using Eq. (6.18), which is plotted in Fig. 6.3(b). As expected, the reconstructability typically increases as the number of observations T increases, even reaching 1 in some cases (e.g., when $q \to 1$). Also, notice how the reconstructability is zero for every value of T when the true positive probability q is equal to false positive probability r. This shows that as true positives and false positives become indistinguishable, the edge becomes impossible to reconstruct.

6.6.3 Optimal reconstruction performance

The reconstruction limit corresponds to the maximum performance, as measured by $I(G^*;\hat{G})$, achievable by any algorithm. Hence, any reconstruction model that is capable of reaching this limit, i.e., $I(G^*;\hat{G}) = I(G^*;X^*) = \Psi^* H(G^*)$, is said optimal in its reconstruction abilities. It is not surprising that the TDG model is optimal according to this definition. As a result, the reconstructability Ψ^* can also be interpreted as a reconstruction performance measure of the TDG model M^*.

In fact, the reconstructability relates to standard performance measures. One such example is the *posterior loss*—also known as the log loss and the cross-entropy loss in the machine learning community. This measure is defined as

$$\mathcal{L}\Big(\boldsymbol{a}^*, \boldsymbol{\pi}^*(x)\Big) = -\textstyle\sum_{i<j}\left[a^*_{ij}\log\pi_{ij}(x) + (1-a^*_{ij})\log\big(1-\pi_{ij}(x)\big)\right], \tag{6.21}$$

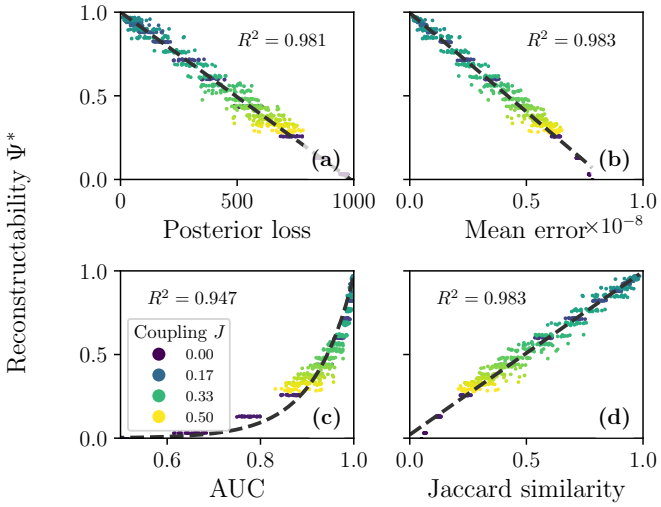


FIGURE 6.4 – Comparison between reconstructability and different performance metrics : (a) posterior loss (Eq. (6.21)), (b) mean error $\binom{N}{2}^{-1} \sum_{i<j} |a_{ij} - \pi_{ij}(x)|$, (c) area under the receiver operating characteristic curve (AUC) and (d) Jaccard similarity (see Ref. [242, Eq. 11]). Each point shows a different realization of the Glauber dynamics whose graphs are generated from the Erdős-Rényi model with $N = 100$ nodes and $E = 250$ edges, and whose initial conditions are random. Reconstructions are performed with the same model, whose parameters are fixed to those used for generating the data. We used time series of $T = 500$ time steps (as in Fig. 6.2, the parameters other than the coupling constant J are specified in Table 6.4). We generated 24 realizations of the process for each value of J and used 30 different coupling values uniformly spaced between 0 and 0.5. These coupling values are fixed during inference. The colors indicated in the legend show the value of J associated with the point (only 6 colors are shown for conciseness). Finally, we show the determination coefficients R^2 relating the performance metrics to Ψ^* in each plot. For panel (a), we used Eq. (6.22) directly to evaluate the determination coefficient, and for panels (b) and (d), we used standard linear regression to find the slope and estimate R^2. For panel (c), because the scaling is not linear like the other cases, we used instead log-linear regression to estimate R^2.

where $\boldsymbol{a}^*$ denotes the adjacency of the true graph g^*, such that a^*_{ij} counts the number of edges connecting the nodes i and j (we use the convention that a^*_{ii} is always a multiple of 2) and $\boldsymbol{\pi}(x) = [\pi_{ij}(x)]_{ij}$ is the predicted matrix of the posterior marginal probabilities of the edge occupancy for some model M. We show in Appendix 6.10.7 that, provided that the data is generated with M^* and the reconstruction is performed with M, the posterior entropy and the expected posterior loss are equal if M is equal to M^* in distribution. Consequently, the reconstructability is linearly related to the posterior loss as follows :

$$\Psi^* \approx 1 - \frac{\mathbb{E}_{G^*,X^*}\left[\mathcal{L}\left(\boldsymbol{A}^*, \boldsymbol{\pi}(X^*)\right)\right]}{H(G^*)}, \tag{6.22}$$

where $\boldsymbol{A}^*$ is the random adjacency matrix of G^*.

Figure 6.4 shows further numerical evidence of the relationship between the reconstructabi-

lity of the Glauber model and reconstruction performance measures, including the posterior loss. In Figs. 6.4(a) and (b), we show how Ψ^* is well correlated with metrics quantifying the recontruction error, such as the posterior loss and the mean error. Similarly, Figs. 6.4(c) and (d) show that the reconstructability is positively correlated with the area under the receiver operating characteristic curve (AUC) and the Jaccard similarity [242], both measuring the similarity between the true and reconstructed graphs.

6.6.4 Reconstructability of hierarchical Bayesian models

Hierarchical models may be used for network reconstruction where additional parameters, namely the random variables θ and ϕ, are included to parametrize the prior and likelihood respectively. In this case, the likelihood $P(X|G,\phi)$ of the model M depends on some unknown parameters ϕ with prior $P(\phi)$ and the graph prior $P(G|\theta)$ depends on other unknown hyperparameters θ with hyperprior $P(\theta)$. During network reconstruction, hyperparameters θ are inferred jointly with G, as they are included in the posterior distribution of the model, while the parameters ϕ are marginalized as follows :

$$P(G,\theta|X) = \sum_{\varphi\in\Phi} \frac{P(X|G,\phi=\varphi)P(\phi=\varphi)P(G|\theta)P(\theta)}{P(X)}\,, \tag{6.23}$$

where ϕ and θ are assumed independent. Note that the sum becomes an integral over the corresponding probability density functions where ϕ is continuous, such that $\rho(\phi)$ is its prior density. In this section, we show how our framework can used on such hierarchical models, without any modification.

Consider the case where a TDG model with variables (G^*, X^*) also includes hyperparameters, denoted θ^* with probability distribution $P(\theta^*)$, such that G^* is conditioned on θ^*—i.e., $P(\theta^*, G^*) = P(\theta^*)P(G^*|\theta^*)$. Let $\hat{\theta}$ and $\hat{G}$ be the reconstructed random parameters and graph, respectively, which are reconstructed from X^* via some distribution $P(\hat{\theta},\hat{G}|X^*)$. The random variables (θ^*, G^*) are related to those of the reconstruction model $(\hat{\theta},\hat{G})$ via X^* as follows :

$$P(\theta^*, G^*, X^*, \hat{\theta}, \hat{G}) = P(\theta^*, G^*)P(X^*|G^*)P(\hat{\theta},\hat{G}|X^*)\,, \tag{6.24}$$

where, again assuming that we use a Bayesian reconstruction model M, we let $P(\hat{\theta}=\vartheta, \hat{G} = g|X^* = x^*) = P(\theta=\vartheta, G = g|X = x^*)$, which is given by Eq. (6.23). In this case, the mutual information between (θ^*, G^*) and $(\hat{\theta},\hat{G})$ can be bounded using the following data processing inequality :

$$I(\theta^*, G^*; \hat{\theta},\hat{G}) \leq I(\theta^*, G^*; X^*)\,. \tag{6.25}$$

In the hierarchical context, $I(\theta^*, G^*; X^*)$ sets the reconstruction limit. By the chain rule, we have

$$I(\theta^*, G^*; X^*) = I(G^*; X^*) - I(\theta^*; X^*|G^*)\,, \tag{6.26}$$

for which the second term of the RHS is zero, by the conditional independence of X^* and θ^* given G^*. We are left with the mutual information upper bound $I(\theta^*, G^*; X^*) = I(G^*; X^*)$, which is equal to the non-hierarchical case. This means that the reconstruction limit is always set by $I(G^*; X^*)$, even if the hyperparameters θ^* are not marginalized over.

6.7 Data-driven reconstructability and model selection

Until now, we have assumed that the TDG model M^* was known to compute the mutual information $I(G^*; X^*)$. Outside of theoretical settings however, the TDG process is typically unknown. Hence, we generally cannot evaluate the true reconstruction limit, although we may have access to many realizations of X^* which should help get closer to it. Three remarks are in order. First, the reconstructability Ψ^* is independent of the observations ; it strictly depends on M^*. Second, any generative model M has a reconstructability, which is calculated identically to Eq. (6.18). In other words, the condition that M is capable of generating new data is crucial to our ability to calculate a reconstructability value. However and thirdly, the reconstructability of M differ in two ways from Ψ^* related to the actual reconstruction limit of the data : *(i)* their values are potentially different and *(ii)* the data generation process is M^*, not M. Consequently, we can leverage the reconstructability of M, with these considerations in mind, to get a data-driven proxy of the true upper bound Ψ^*.

6.7.1 Reconstruction index based on information gain

To bring back the dependency of the reconstructability on the observations, we take a similar approach as before and start with an information measure. For a model M and any instance $x \in \mathcal{X}$, the data-driven version of mutual information is called the *information gain* [204], and it is defined as

$$\mathcal{I}_M(x) = -\mathbb{E}_{G|X=x}\left[\log\left(\frac{P(G|X)}{P(G)}\right)\right] = \sum_{g\in\mathcal{G}} P(G=g|X=x)\log\left(\frac{P(G=g|X=x)}{P(G=g)}\right). \tag{6.27}$$

Note that the expectation of the information gain yields back the mutual information between G and X, i.e., $\mathbb{E}_X[\mathcal{I}_M(X)] = I(G;X)$. The information gain measures the reduction in the entropy of a variable G achieved by learning the state x of another variable X. It is primarily used in feature selection, especially decision tree training, where it is used as a criterion for how to best split the data [204, Chapter 3]. Like the mutual information, the information gain can be shown to be non-negative (see Appendix 6.10.8) and upper-bounded :

$$0 \leq \mathcal{I}_M \leq \Lambda_M\,,$$

where

$$\Lambda_M(x) = -\mathbb{E}_{G|X=x}[\log P(G)] \tag{6.28}$$

is the maximum value of the information gain, and can be interpreted as the cross-entropy between the reconstruction posterior and the prior probabilities of M. It is therefore convenient to define a normalized version of the information gain, which we refer to as the *reconstruction index* :

$$\psi_M = \frac{\mathcal{I}_M}{\Lambda_M} \,. \tag{6.29}$$

Like the reconstructability, the reconstruction index ψ_M is bounded between 0 and 1. However its interpretation is more subtle, as we will see in the following sections. Indeed, the information gain, on which the reconstruction index is based, is the Kullback-Leibler (KL) divergence between the posterior and the prior of the reconstruction model. As a result, it quantifies how different the posterior of the reconstruction model is from the prior. When $\psi_M = 0$, the posterior and the prior are identical—no information is gained from knowing the data. On the other hand, when $\psi_M = 1$, the posterior probability mass is entirely located on a single graph, which is reflected in the fact that the KL divergence is maximized.

6.7.2 Interpretation of the reconstruction index under incorrect assumptions

We must be careful in our interpretation of the reconstruction index, as its value can be misleading if not used in the correct way. Figure 6.5 shows how the reconstruction index behaves when the reconstruction model is incorrect to different extents. In this example, we generated time series of the Glauber dynamics with a given coupling constant J^*, and reconstructed the graphs using the same Glauber model, but typically with an erroneous coupling constant $J \neq J^*$. As we can see in Fig. 6.5(a), the reconstruction index keeps increasing as J gets larger, even when it gets larger than J^*. While the reconstruction index is larger for $J > J^*$, the posterior loss actually shows, as expected, that the reconstruction becomes worse. Figure 6.5(a) also illustrates how the reconstruction index correlates with the posterior loss, depending on J. Indeed, as J gets closer to J^*, the reconstruction index converges to the true reconstructability of the reconstruction model, which increasingly becomes linearly related to the posterior loss as previously shown (see the Appendix 6.10.7) In this regime, the reconstruction index is a good proxy of the true reconstructability because M properly approximates the behavior of M^*.

The behavior of the reconstruction index when the reconstruction model is incorrect raises some important remarks. Recall that, fundamentally, the reconstruction index is a normalized version of the KL divergence between the posterior and the prior of the reconstruction model. Therefore, it is perhaps not surprising that we lose the correspondence between reconstruction index and performance established in Sec. 6.6.3 when the model is incorrect. Indeed, having a posterior that is very different from the prior implies a high reconstruction index even though the posterior distribution is actually wrong.

Maximizing the reconstruction index can also lead to inadequate modeling of the observed

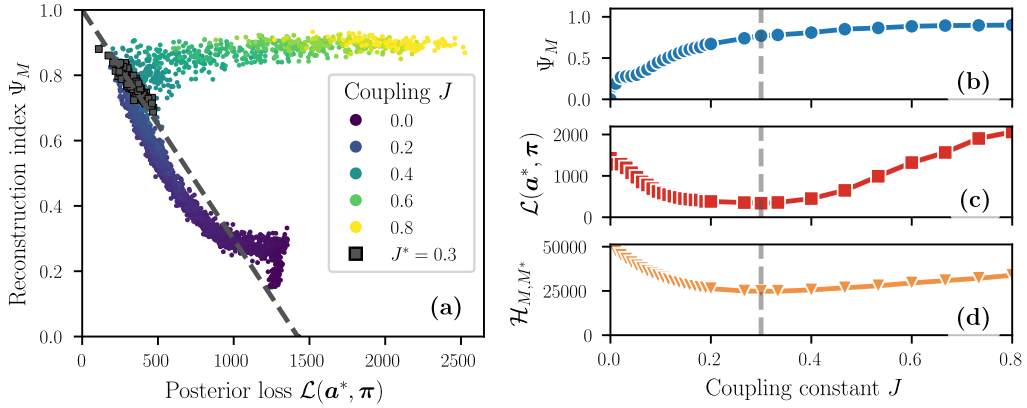


FIGURE 6.5 – Effect of varying the coupling constant on the validity of the reconstruction index. We generated time series of the Glauber dynamics with fixed $J^* = 0.3$ on Erdős-Rényi graphs with $N = 100$ nodes and $E = 250$ edges, then reconstructed the graphs using the same Glauber model with other coupling constants J, used during the inference. Panel (a) shows the relationship between the reconstruction index ψ_M and the posterior loss $\mathcal{L}(\boldsymbol{a}^*, \boldsymbol{\pi})$ between the true graphs and the posterior—each point corresponding to a different realization of the TDG process (graph and observations) from which we reconstructed the graph. Panels (b–d) respectively show the reconstruction index ψ_M, posterior loss $\mathcal{L}(\boldsymbol{a}^*, \boldsymbol{\pi})$, and evidence cross-entropy $\mathcal{H}_{M,M^*}$ (Eq. (6.32)) as functions of J. The dashed vertical line shows where $J = J^*$. We color-coded the points according to J, as shown in the legend, including the true value J^* (grey squares). As in Fig. 6.4, we show the linear relationship between the reconstructability and the posterior loss (Eq. (6.22)) with the dashed line in (d). Glauber time series were generated with $T = 500$ time steps, and we generated 24 realizations with random initial conditions for each value of J between 0 and 0.8 (like in Fig. 6.4, we show only a few values in the legend of (d)). In panels (b–d), we show the 90% confident intervals around the mean (displayed by the markers), although they are too small to be visible.

data. Consider the following alternative but equivalent form of the information gain :

$$\mathcal{I}_M(x) = \mathbb{E}_{G|X=x}[\log P(X|G)] - \log P(X = x)\,. \tag{6.30}$$

In this formulation, the two terms—i.e., the expected log-predictive and the log-evidence, respectively—are in opposition. The first term is maximized when the model is good at describing the data using the posterior graphs, while the second is maximized when the model describes the data correctly altogether. The log-evidence is even used as a measure of goodness-of-fit for model selection, as we will see in the next section. Yet, maximizing the information gain is equivalent to maximizing the expected log-predictive and minimizing the log-evidence, which is why incorrect models may be selected by this criterion. Following these remarks, we devote the next section to describing a principled approach to adequately interpret the reconstruction index and use it in the context of data-driven reconstruction.

6.7.3 Role of evidence-based model selection

Maximizing evidence as a criterion for model selection is a well-known practice in Bayesian modeling [150]. In particular, Bayes factors are ratios between the evidence of two models, say $M_1 = (G_1, X_1)$ and $M_2 = (G_2, X_2)$:

$$B_{M_1,M_2}(x) = \frac{\zeta_1(x)}{\zeta_2(x)}\,, \tag{6.31}$$

where $\zeta_{M_i}(x) = P(X_i = x)$ is the evidence of model M_i for x. If $B_{M_1,M_2}(x) > 1$, M_1 is better supported by the data x than M_2. From an information-theoretic perspective, the minimization of the *evidence cross-entropy* (CE)—which is equivalent to maximizing the evidence—can be shown to be a necessary condition for finding the TDG model, whose evidence function is $\zeta_{M^*}(x)$. Indeed, for a reconstruction model M with evidence function $\zeta_M(x)$, the evidence CE is expressed as

$$\mathcal{H}_{M^*,M} = -\mathbb{E}_{X^*}[\log \zeta_M(X^*)] = -\sum_{x^* \in \mathcal{X}} \zeta_{M^*}(x^*) \log \zeta_M(x^*)\,. \tag{6.32}$$

Equation (6.32) is minimized when X^* and X are equal in distribution [65]. Note that it is a necessary condition to find the correct TDG model, but may not be a sufficient one, as it is easy to show that, in the general case, many reconstruction model may have the same evidence distribution, but different posterior distributions.

The problem of evidence-based model selection is the computation of the evidence itself which is often intractable in practice. In fact, this is the case for most graph models, where the evaluation of the evidence requires graph enumeration. This problem also arises in the evaluation of both the information gain and the mutual information. Fortunately, the same numerical techniques can be used to evaluate the evidence and the information gain simultaneously, as the two are related to each other. In Ref. [212], we showed that variational mean-field methods provide efficient approximations for both the mutual information and the evidence. The same techniques are used here (see Appendix 6.10.9).

Model selection is crucial for the validity of the reconstruction index as a proxy of reconstruction performance. Suppose we have many observations $(x_1, x_2, ...)$ and used the reconstruction model M^*. The empirical average of the information gain of some model M converges to $\mathbb{E}_{X^*}[\mathcal{I}_M(X^*)]$. Now, assume for a moment that M in fact maximizes the expected log-evidence. This implies, as we mentioned before, that X^* and X are equal in distribution. In turn, the empirical information gain becomes equal to the mutual information for model M, i.e., $I(X;G)$, which we recall is the reconstruction limit of M. This means that empirical information gain converges to the reconstructability and that ultimately the reconstruction index naturally extends the concept of reconstruction limit to real systems observed only through data.

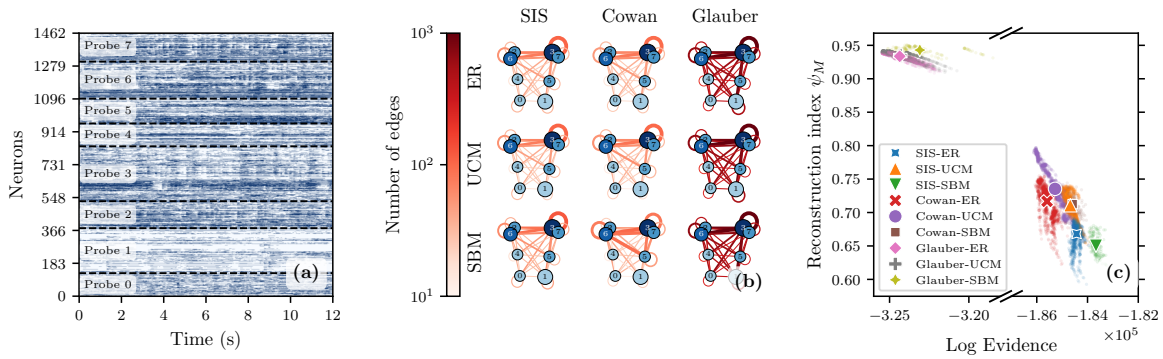


FIGURE 6.6 – Reconstruction from spontaneous neuronal activity in the mouse brain [299, 300] : (a) Raster plot of the 1462 monitored neurons, (b) reconstruction of the probe network using different reconstruction models and (c) reconstructability diagram. In panel (a), the neurons are ordered by the probe they were measured from. Each spike is represented in blue. Panel (b) shows the posterior average network projected onto the probes, as predicted by each reconstruction model where rows correspond to different graph models (see Appendix 6.10.1), and columns to different dynamics models (see Table 6.4). The color of an edge connecting two probes shows the absolute number of edges and the thickness indicates the average proportion among all the edges. The size of the probe nodes is proportional to the number of neurons monitored by the probe, and the color indicates the measured number of spikes. The node locations correspond to the actual probe locations in the mouse brain obtained from [299]. The reconstruction index as a function of the model log evidence is shown in panel (c), comparing the different models. Small markers are estimated by a single Markov chain and large markers are the average of these estimations. In these experiments, the parameters of the graph prior and likelihood are inferred jointly with the graph. For additional details about the inference procedure, we refer to Appendix 6.10.16.

When the reconstruction model does not minimize the evidence CE, the picture becomes more nuanced, as shown in Fig. 6.5. As the evidence CE decreases, the correlation between the reconstruction index and the posterior loss increases. Our ability to identify the reconstruction limit without knowing the true model or network structure is therefore as good as the reconstruction model's ability to describe the data. This key conceptual observation leads us to conclude that we can indeed leverage the reconstruction index as a proxy for assessing the reconstructability of real networks, provided it is interpreted in conjunction with the posterior loss, as in Fig. 6.5(a).

6.8 Network reconstructability in empirical networks

Reconstructing empirical graphs represents a technical and conceptual challenge. The true network structure being unknown, it is hard to quantify how close the predicted graph is to the true one, let alone calculate its actual reconstructability. In light of our exploration in Sec. 6.7, however, we have shown that the reconstruction index ψ_M can offer a means to approximate the reconstructability, if certain conditions are met. Additionally, our analysis

shows that a correctly calibrated reconstruction index ψ_M predicts the error in a reconstructed network in comparison with the true one. Under these considerations, we present a principled method based on the reconstruction index to assess the validity of network reconstruction.

Let x be some time series generated by a hidden process M^*, on which we wish to infer a network. As shown in Sec. 6.7.3, the validity of the reconstruction index heavily relies on the model's aptitude to represent the data. Hence, we may consider a set of d reconstruction model candidates $\mathcal{M} = \{M_1, M_2, ..., M_d\}$, on which we will perform model selection later. These models may differ in their underlying assumptions, via their prior, hyper prior, and/or likelihood functions, as previously stated. Next, we perform reconstruction with each candidate model M and determine its corresponding evidence $\zeta_M(x)$ and reconstruction index $\psi_M(x)$. Finally, we choose the reconstruction index of the model $\hat{M}$ with the highest value of $\zeta_M(x)$:

$$\hat{M} = \underset{M \in \mathcal{M}}{\arg\max}\, \zeta_M(x)\,. \tag{6.33}$$

If $\mathcal{M}$ contains a model that resembles the TDG process, this procedure will generate a reconstruction index $\psi_{\hat{M}}(x)$ that closely approximates the true reconstructability of the process Ψ^*. Of course, determining if such a model is in $\mathcal{M}$ is hardly feasible experimentally. We work around this issue by performing a posterior predictive check of $\hat{M}$, i.e.,

1. by generating a sample $\{\hat{x}_1, ..., \hat{x}_K\}$ of synthetic data with $\hat{M}$, assuming its parameters (θ, ϕ) are sampled from the model's posterior given x;
2. then, by calculating some test quantities $\tau(\hat{x}_k)$, i.e., statistics used for comparison, for each sample $\hat{x}_k$, generating a set of samples $\mathcal{T} = \{\tau(\hat{x}_1), ..., \tau(\hat{x}_K)\}$;
3. finally, by comparing $\tau(x)$ with $\mathcal{T}$.

If $\tau(x)$ is typical in $\mathcal{T}$, then we can be sure $\hat{M}$ is statistically similar to M^*.

In what follows, we will demonstrate this method in two different empirical use cases. The first use case reflects a realistic modeling scenario, where only the time series on which reconstruction is performed are known. From a theoretical perspective, this use case presents challenges that falls outside of the scope of this paper (we will discuss those in the following section). The second use case lift those limitations by reconstructing real networks from synthetic time series.

6.8.1 Reconstruction from empirical neuronal spiking data

We consider the spontaneous activity of 1462 neurons from a mouse brain recorded over 20 minutes using eight neopixel probes [Fig. 6.6(a)] [300]. Starting from the recorded spike times of the neurons, indicating when they fire, we create a binary time series of the activity of each neuron. In these binary time series, a '1' marks the moments when a neuron is fired, and a '0' when it is not. For more detail regarding our data processing procedure,

see Appendix 6.10.16. Following our method described at the beginning of Sec. 6.8, we infer the network using a variety of Bayesian reconstruction models. As a result, we get the reconstructed networks shown in Fig. 6.6(b), where we aggregated the edges connecting the neurons of all pairs of probes, thus illustrating how they interact. Finally, we compute the log evidence for each model and select the model with the highest one (see Appendix 6.10.9 for details on the evidence estimation). In doing so, we extract the reconstruction index that is, in principle, closest to the reconstruction limit, given the set of considered models. Figure 6.6(c) shows a diagram of both measures for each model.

In our example, the analyzed reconstruction models are combinations of time series likelihoods —SIS, Cowan and Glauber—and graph priors—the ER model, the configuration model with a uniform degree sequence hyperprior (UCM) and the stochastic block model (SBM). The ER model, having a uniform distribution, is the most entropic model, followed by the SBM and the CM. Additional details about the graph prior and about these reconstruction models are given in Appendices 6.10.1 and 6.10.5, respectively. Of course, these models oversimplify the observed neuronal activity. Moreover, certain critical factors are not captured in the current dataset, such as the latency and deactivation rates of neuronal activity, as well as the substantial number of neurons undetected by the probes, which could contribute as input currents to the modeled neurons. In addition, the lack of detailed connectomic data for such small brain regions prevents a rigorous assessment of the accuracy reconstructed probe network. The purpose of this analysis is therefore not to perform the most accurate reconstruction but to illustrate the complete procedure as well as the results it generates.

Among the considered models, the SIS model with a SBM prior is the one achieving the highest log evidence. As detailed in Appendix 6.10.16, the posterior predictive checks of the inferred SIS model suggests that its reconstruction index of approximately 67% is reasonable. Additionally, given that the network contains 1462 neurons and that the estimated number of edges for this model is approximately 1722 (see Table 6.5), the inferred network is sparser than expected (e.g., see Table 2 in [181]) with an average degree of 2.35. This suggests that, although our estimation of reconstructability is not close to zero, the inferred network does not seem to account for most of the neural activity. Of course, considering the decreasing tendency observed in Fig. 6.6(c), more detailed neuronal models—better suited for these data and with potentially higher log evidence—could yield reconstruction indices even lower than 67%, with possibly denser inferred networks. In the following section, we circumvent the limitations of the neuronal activity data by transitioning to a controlled setting, where synthetic activity data is used to reconstruct empirical networks.

6.8.2 Reconstruction of empirical graphs from synthetic activity data

While reconstructing empirical graphs, we assume that all observations come from the same graph g^*. Hence, the graph prior, whose associated random variable is G, plays an impor-

Graph	ER	UCM	CM	SBM
Karate club	343.69	316.52	200.82	328.61
Political books	2267.99	2177.04	1756.70	2158.49

TABLE 6.3 – Negative log probability—i.e., $-\log P(G = g^*)$—of the graphs considered in Fig. 6.7 using the Erdős-Rényi (ER) model, the configuration model with uniform degree sequence prior (UCM), the configuration model with given degree sequence (CM) and the stochastic block model (SBM).

tant role : Injecting prior information about g^*. Depending on the value of $P(G = g^*)$, the graph prior may either improve the reconstruction or impede it. Consequently, the amount of information that one may need to achieve a desired level of reconstruction accuracy may change depending on the choice of $P(G)$.

Our method can also be used to assess the role of the prior, when using empirical graphs. However, since we consider here synthetic data generation process, we can omit the posterior predictive checks. Figures 6.7(a) and (f) show the reconstruction index of two empirical graphs, namely Zachary' karate club [343] and the Political books network [162], as a function of the posterior loss for different graph prior models. Here, the reconstruction indices ψ_M are calculated using Eq. (6.29) like before, where all x^* are generated using the same graph g^*. For this analysis, in addition to the previous graph priors, we consider the standard CM where the degree sequence is given. The standard CM is less entropic than the UCM since it is given all the information of the degree sequence—it does not need to be inferred from X as for the UCM. The graph support of the CM is also quite smaller than that of the other priors, which should improve the reconstruction. We generated many synthetic observations on these two graphs g^* to perform the reconstruction : A spreading epidemics on the karate club social network using the SIS model, and the Voter model for the Political books network, as to simulate the propagation of political opinions. For both examples, we chose many values of parameters for the dynamics to present the complete range of reconstructability scenarios. To highlight the role of the graph prior, we reconstruct all graphs with the TDG data models. This means that, in this case, ψ_M is indeed a point-wise measure of reconstructability.

The reconstruction index is shown to scale linearly with the error, as measured by the posterior loss, identically to Fig. 6.5(a). This shows that, even though only one graph is used to generate the data, the relationship between ψ_M and performance still holds. This is also observed in Figs. 6.7(b–e) and (g–j), where examples of reconstructed graphs with increasing reconstruction indices are shown with their corresponding—and increasingly accurate—posterior.

Moreover, notice how the reconstruction index scaling changes as a function of the prior,

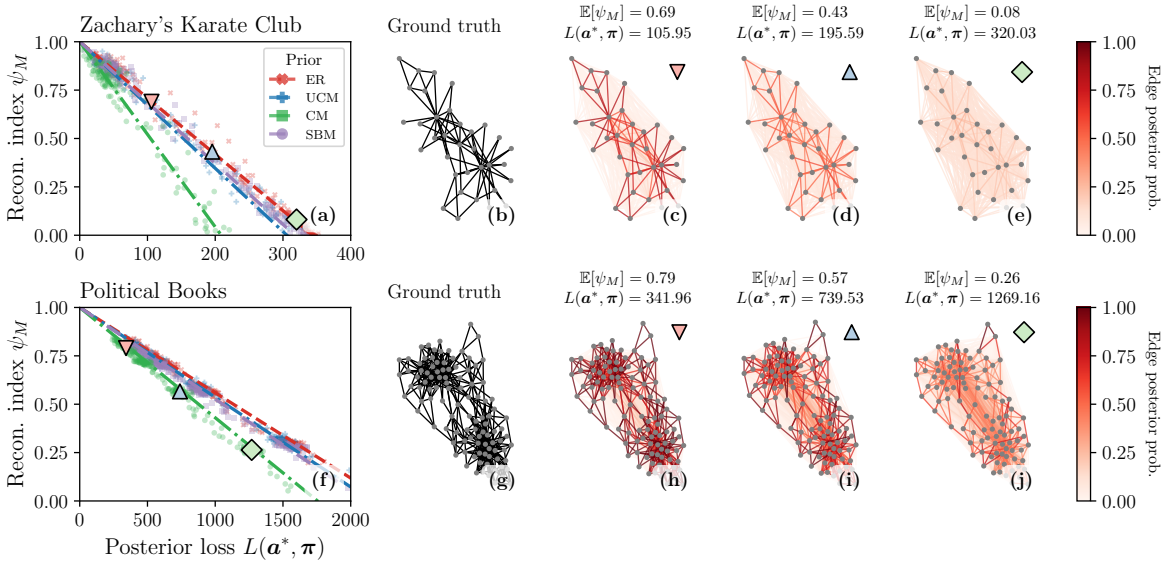


FIGURE 6.7 – Reconstruction indices of empirical graphs with different graph prior models : (top) SIS dynamics on the Zachary's karate club [343] and (bottom) Voter dynamics on the Political books network. We show in panels (a) and (f) the reconstruction indices ψ_M as a function of the posterior loss. We consider different values of dynamics parameters to populate the diagrams : for Zachary's karate club we fixed the infection probability to $\lambda \in \{0.1, 0.12, 0.15, 0.2, 0.3\}$, and for the Political books network, we let $\alpha_0 \in \{0.001, 0.01, 0.1, 0.25, 0.5\}$—we omit illustrating their values in the plots for simplicity. We use and fix these parameter values within the model during the inference. For each combination of graph model and dynamics parameters, we generated 48 time series of $T = 300$ steps and performed reconstruction of each of them individually. Each point in (a) and (f) corresponds the reconstruction index and posterior loss of one of these time series. In each plot, the different symbols and colors indicates the graph prior model used for the reconstruction : The Erdős-Rényi model (ER, blue diagonal crosses), the configuration model with uniform degree sequence prior (UCM, orange crosses) and with the correct degree sequence (CM, red circles), and the stochastic block model (SBM, green squares). The lines correspond to the scaling of ψ_M with respect to the posterior loss [Eq. (6.29)]. In panels (b–e) and (g–j), we show the true network g^* (far left) followed by the reconstructed graphs, as illustrated by their respective posteriors, of three different models. We indicate on top of each example the corresponding expected reconstruction index and posterior loss, and we highlight their location in the diagrams of (a) and (f) using the symbols (inverted triangle, triangle and diamond). For panels (c–e), we choose the posteriors of the ER model such that (c) $\lambda = 0.1$, (d) $\lambda = 0.15$ and (e) $\lambda = 0.2$. For panels (h–j), we choose the posteriors of the CM where (h) $\alpha_0 = 0.5$, (i) $\alpha_0 = 0.25$ and (j) $\alpha_0 = 0.1$.

where the slope is precisely given by Λ_M^{-1} [Eq. (6.28)]. In both cases, the graph model with the steepest slope is the CM, which also has the most prior information about g^* as shown by the prior negative log probability in Table 6.3. This happens because Λ_M and the prior probability $P(G = g^*)$ are intrinsically related to one another, since Λ_M is the posterior average of the prior probability. Hence, as the posterior becomes more concentrated around g^*, Λ_M converges to $P(G = g^*)$. The relationship between the ψ_M scaling and the prior implies that ψ_M tends to diminish faster as a function of the posterior loss, as the prior gets more informative. In other words, for two models with identical reconstruction indices, the one with a graph prior more concentrated around g^* generates a more accurate reconstruction.

The reconstruction limit, as measured by ψ_M, changes as a function of the graph prior : If information about g^* is *a priori* given, this same information cannot be reconstructed from realizations of X^*. Thus, it is no longer taken into account in ψ_M which by construction factors out the contributions of the prior. This intuition can be mathematically studied if we let the prior put more and more weight on g^*. In Appendix 6.10.15, we prove that as the graph generative model converges to a Kronecker delta distribution, i.e.,

$$P(G^* = g) = \begin{cases} 1 & \text{if } g = g^* \\ 0 & \text{otherwise} \end{cases}, \tag{6.34}$$

the reconstructability Ψ^* converges to zero, except if the mutual information is maximized in which case it is always equal to 1.

6.9 Conclusion

To what extent is a complex network reconstructible? Our ability to reconstruct is strongly constrained by the information content of the underlying structure within the data, making perfect reconstruction generally infeasible. The best reconstruction therefore amounts to finding a network that reaches a reconstruction limit, extracting all available information in such a way that no further improvement can be achieved on average.

Our information-theoretic framework characterizes this network reconstruction limit, which is closely tied to the true generating process of the observed data. The reconstruction limit is analogous to the detectability limit in community detection [75, 107, 340] in that it is algorithm independent. We find that this limit is expressed in terms of the reconstructability—a normalized version of the mutual information between the graph and the data of the true data generating process. While a small reconstructability implies a bad performance regardless of the reconstruction model, a high reconstructability implies that good performance can be achieved with the appropriate model.

Our approach is general and can be extended to real modeling settings, where the data is limited and the reconstruction model is unknown. Using the same principles, we defined the

reconstruction index, analogous to the reconstructability, that is also data-dependent and can be used as a proxy of the reconstruction performance. When coupled with evidence-based model selection, the reconstruction index is an appropriate performance measure, even when the graph is unknown. This further emphasizes the importance of model quality in network reconstruction [236].

Finally, we presented different applications of our framework using real networks and real time series data. We showed how to use the reconstruction index on spiking neural networks. Our analysis suggests that the reconstructability of the network formed by the recorded neurons in the mouse brain is approximately 67%, which is consistent with experimental studies of brain networks reporting structure-function couplings of about 40% in humans [27] and up to 68% in smaller animal models, such as zebrafish [176]. We believe that more work is needed in the inference process, primarily because of the simplicity of the reconstruction models used in this analysis and the incompleteness of the data. A more thorough analysis of such a case study could reveal that certain neuronal activity datasets are insufficient to build a comprehensive picture of the functional activity of the brain. Additionally, although the reconstructability is based on random variables and ensembles, we demonstrate that our framework can be reliably used on single instances of graphs. In this context, the reconstruction index can be used to determine the reconstruction limit, even if the true graph is unknown, as it is shown to correlate strongly with the error between the inferred graphs and the true one.

We envision a future where network reconstruction applications incorporate a reconstructability analysis in their pipeline, such as the one presented in Sec. 6.8. By doing so, the reconstruction index would indicate how informative the reconstructed networks are and perhaps inform us on how they should be used within the said applications. Of course, there is still plenty of work to be done on this front, such as improving the computational methods required to compute the reconstruction index as they do not scale well to large networks, and improving the reconstruction models themselves as we have alluded to earlier. Some of these models might also require modifying our framework, for example in the case of weighted and directed networks. These specific models could prove considerably valuable for the neuroscience community and, more broadly, for complex systems research.

6.10 Appendix

6.10.1 Graph priors

In the paper, we use different random graph models as graph priors for Bayesian network reconstruction. These models are undirected and unweighted and may include self-loops and multiedges, although our general framework is not restricted to these assumptions. Indeed, one could consider directed or weighted graphs as well ; as long as the set of possible graphs

remains countable. We use the adjacency matrix, denoted $\boldsymbol{a}$, in order to define the probability distribution of some of these models, where a_{ij} counts the number of edges connecting nodes i and j. To simplify the notation, we will sometimes express a graph g directly with its adjacency matrix $g = \boldsymbol{a}$. We use the convention that a_{ii} is always a multiple of 2. Below, we describe these priors in more detail.

6.10.2 Erdős-Rényi model

The Erdős-Rényi (ER) model corresponds to the maximum entropy random graph model, i.e., the uniform distribution over all simple graphs with N nodes and E edges, such that

$$P(G|E = e) = \left(\binom{\frac{N(N-1)}{2}}{e} \right)^{-1}, \tag{6.35}$$

where we recall that $\binom{n}{k}$ is the binomial coefficient. The ER model is also generalizable to loopy multigraphs, where

$$P(G|E = e) = \left(\left(\binom{\frac{N(N+1)}{2}}{e} \right) \right)^{-1}, \tag{6.36}$$

such that $\left(\!\binom{n}{k}\!\right) = \binom{n+k-1}{k}$ counts the number of possible multisets of size k composed of n different objects—i.e., multiedges.

Note that the number of edges E must be provided to the ER model, and in the other graph models described below. This means that $\theta = E$ is the hyperparameter of ER graph prior and that E should be inferred. We use the prior $P(E)$ to weigh in the number of edges. In most of our experiments, the number of edges is fixed to a specific value e^*, meaning that $P(E = e) = \delta(e, e^*)$, where $\delta(m, n)$ is the Kronecker delta function. However, in Sec. 6.8.1, as E is unknown in this case, we use a geometric prior of the form

$$P(E = e) = \frac{\bar{\lambda}^e}{(\bar{\lambda} + 1)^{e+1}}, \tag{6.37}$$

where $\bar{\lambda}$ is a parameter that fixes the expected number of edges. See also Appendix 6.10.16 for further detail about the complete inference procedure.

6.10.3 Configuration model

The configuration model (CM) describes an ensemble of loopy multigraphs where the degree sequence is given [95]. From a network reconstruction perspective, the CM can also be used as a prior, assuming that its probability factors as follows

$$P(G, \boldsymbol{k}, E) = P(G|\boldsymbol{k})P(\boldsymbol{k}|E)P(E), \tag{6.38}$$

where $P(G|\boldsymbol{k})$ is the graph likelihood given the degree sequence $\boldsymbol{k}$, $P(\boldsymbol{k}|E)$ is the prior over the degree sequence and $P(E)$ is again the prior over the number of edges (same as in the ER model). In the CM, half-edges (or stubs) are considered distinguishable and a realization of the model is generated by randomly pairing all available half-edges. Hence, the probability of generating pairings leading to a graph g, whose adjacency matrix is $\boldsymbol{a}$, given its degree sequence $\boldsymbol{\kappa}$ is

$$P(G = \boldsymbol{a}|\boldsymbol{k} = \boldsymbol{\kappa}) = \frac{\prod_{i=1}^{N} \kappa_i!}{\prod_{i<j} a_{ij}! \prod_{i=1}^{N} a_{ii}!!}\,, \tag{6.39}$$

where $(2n)!! = 2^n n!$ is the double factorial of $2n$, i.e., the product of all even numbers up to $2n$.

Furthermore, only one degree sequence, denoted $\boldsymbol{\kappa}^*$, is considered in the standard formulation of the CM. This results in a delta degree sequence of the form

$$P(\boldsymbol{k} = \boldsymbol{\kappa}) = \delta(\boldsymbol{\kappa}, \boldsymbol{\kappa}^*)\,. \tag{6.40}$$

When the degree sequence is unknown, we use the uniform non-informative prior

$$P(\boldsymbol{k}|E = e) = \left(\!\!\binom{N}{2e}\!\!\right)^{-1}, \tag{6.41}$$

where $\left(\!\!\binom{N}{2e}\!\!\right)$ counts the number of possible degree sequences for a graph of N nodes and E edges.

6.10.4 Stochastic block model

The stochastic block model (SBM), in its microcanonical version [240], closely resembles the ER model, where edges are picked uniformly at random. However, unlike the ER model, each node i is associated with a random block $b_i \in \{1, 2, \ldots, B\}$ and instead the number of edges $e_{rs} = \sum_{ij} a_{ij}\delta(b_i, r)\delta(b_j, s)$ connecting two blocks r and s is fixed such that there are E edges in total. We summarize the node partition as a tuple $\boldsymbol{b} = (b_i)_{i=1..N}$ and the number of edges between blocks by the edge matrix $\boldsymbol{e} = (e_{rs})_{r,s=1..B}$. The blocks are required to be non-empty. Since B is the number of non-empty blocks in $\boldsymbol{b}$ and $\boldsymbol{e}$ are completely determined by the graph and $\boldsymbol{b}$, $\theta = (\boldsymbol{b}, E)$ are the hyperparameters of the SBM, then $\boldsymbol{b}$ and E must be inferred jointly with the graph. In theory, one could factor the joint prior probability $P(G, E, \boldsymbol{b})$ as $P(G|E, \boldsymbol{b})P(E)P(\boldsymbol{b})$ assuming E and $\boldsymbol{b}$ are independent. However, it is more convenient to factor the prior also using $\boldsymbol{e}$ and B as intermediate random variables, following

$$\begin{aligned} P(G, E, \boldsymbol{b}) &= P(G, E, \boldsymbol{e}, \boldsymbol{b}, B) \\ &= P(G|\boldsymbol{e}, \boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b}, E)P(E)P(\boldsymbol{b}|B)P(B)\,, \end{aligned} \tag{6.42}$$

where

$$P(G|\boldsymbol{e} = \boldsymbol{\epsilon}, \boldsymbol{b} = \boldsymbol{\beta}) = \prod_{r<s} \binom{n_r n_s}{\epsilon_{rs}}^{-1} \prod_r \binom{\frac{n_r(n_r+1)}{2}}{\frac{\epsilon_{rr}}{2}}^{-1} \tag{6.43}$$

and $n_r = \sum_{i=1}^{N} \delta(\beta_i, r)$ counts the number of nodes in block r for the partition $\boldsymbol{\beta}$. Next, we choose the edge matrix hyperprior. This matrix can be seen as the adjacency matrix of the multigraph connecting the blocks together. In Ref. [240], a hierarchical SBM was used as a prior for the edge matrix where each level came with its own node partition and edge matrix that, in turn, can also be modeled by a SBM, and so on until only one block remains. Here, we focus on the simpler version of this scheme, where the edge matrix prior is simply given by a multigraph ER model with b nodes :

$$P(\boldsymbol{e}|E = e, \boldsymbol{b} = \boldsymbol{\beta}) = \left(\left(\begin{array}{c} \frac{b(b+1)}{2} \\ e \end{array}\right)\right)^{-1}, \tag{6.44}$$

where, again, b is the number of blocks in $\boldsymbol{\beta}$. For the node partition hyperprior, we choose a non-informative uniform distribution on all partitions with B non-empty blocks :

$$P(\boldsymbol{b}|B = b) = \binom{N-1}{b-1}^{-1}, \tag{6.45}$$

which counts the number of possible arrangements of N nodes into b non-empty groups. Likewise, we choose a non-informative uniform hyperprior over the number of non-empty blocks B :

$$P(B) = N^{-1}. \tag{6.46}$$

6.10.5 Markov chain likelihoods

Throughout the paper, we consider likelihoods where the observations are time series of binary variables for each node, denoted $X = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_T)$, $\boldsymbol{X}_t \in \{0,1\}^N$ for every t. These data models are based on Markov chains, where the state X_{t+1} at time $t+1$ is conditioned on every previous state except the previous one $\boldsymbol{X}_t$ at time t, that is

$$P(X|G) = P(\boldsymbol{X}_1) \prod_{t=1}^{T} P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t, G), \tag{6.47}$$

where $P(\boldsymbol{X}_1)$ is the probability distribution of the initial state, and $P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t, G)$ is the transition probability from $\boldsymbol{X}_t$ to $\boldsymbol{X}_{t+1}$. The type of Markov chains we are interested in are graphical models [83], meaning that the transition probability for a single node i only depends on the previous state of its neighbors including itself :

$$P(\boldsymbol{X}_{t+1}|\boldsymbol{X}_t, G = g) = \prod_{i=1}^{N} P(X_{i,t+1}|X_{i,t}, X_{\mathcal{N}_i,t}), \tag{6.48}$$

where $X_{\mathcal{N}_i,t} \equiv (X_{j,t})_{j\in\mathcal{N}_i}$ contains the state of the neighbors $\mathcal{N}_i$ of node i in the graph g. For time homogeneous Markov chains, the transition probability of a node i is expressed in terms of its number of active neighbors $n_{i,t}$, inactive neighbors $m_{i,t}$ and some set of parameters φ. We denote the activation and deactivation probabilities $\alpha(n_{i,t}, m_{i,t}, \varphi)$ and $\beta(n_{i,t}, m_{i,t}, \varphi)$,

Dynamics	ϕ	$\tilde{\alpha}(n,m)$	$\tilde{\beta}(n,m)$
Glauber [112]	J	$\sigma(2J(n-m))$	$\sigma(2J(m-n))$
SIS [**?**]	(β,λ)	$1-\left(1-\frac{\lambda}{\beta}\right)^m$	β
Voter [59]	$\varnothing$	$\frac{m}{n+m}$	$\frac{n}{n+m}$
Cowan [66]	(a,β,μ,ν)	$\sigma(a(\nu m-\mu))$	β

TABLE 6.4 – Activation and deactivation probability functions for the likelihoods used in this paper, where n corresponds to the number of inactive neighbors whose states are 0, and m corresponds to the number of active neighbors whose states are 1. We define $\sigma(x) = [\exp(-x)+1]^{-1}$ as the sigmoid function.

respectively. Putting everything together, the transition probability of the Markov chain is given by

$$P\left(\boldsymbol{X}_{t+1}=\boldsymbol{y}|\boldsymbol{X}_t=\boldsymbol{x},G,\phi=\varphi\right)=\prod_{i=1}^{N}\Big\{\left[\alpha(n_{i,t},m_{i,t},\varphi)\right]^{(1-x_i)y_i}\left[1-\alpha(n_{i,t},m_{i,t},\varphi)\right]^{(1-x_i)(1-y_i)}\left[\beta(n_{i,t},m_{i,t},\varphi)\right]^{x_i(1-y_i)}\left[1-\beta(n_{i,t},m_{i,t},\varphi)\right]^{x_iy_i}\Big\}. \tag{6.49}$$

We allow the inactive nodes to spontaneously activate with probability α_0, and spontaneously deactivate with probability β_0. Denoting $\tilde{\alpha}$ and $\tilde{\beta}$ the activation and deactivation probabilities without spontaneous activation, respectively, we obtain

$$\alpha(n,m,\varphi)=(1-\alpha_0)\tilde{\alpha}(n,m,\varphi)+\alpha_0\,, \tag{6.50}$$

$$\beta(n,m,\varphi)=(1-\beta_0)\tilde{\beta}(n,m,\varphi)+\beta_0\,. \tag{6.51}$$

In general, we fix $\alpha_0=\beta_0=0$ for the synthetic experiments, and infer them in Sec. 6.8.1.

Table 6.4 presents the activation and deactivation probability functions for four different processes used in various contexts. The Glauber dynamics is a spin model that describe the time-reversible evolution of magnetic spins (0 or 1) aligning in a crystal. In this model, the nodes are connected through their neighbors via a coupling constant J, that modulates the probability of a node to align with its neighbors. The susceptible-infected-susceptible (SIS) dynamics is a canonical model of epidemic spreading, where the nodes are either susceptible

(0) or infected (1), and has often been used to model disease with short immunity after recovery, similar to influenza-like disease [11]. Susceptible (or inactive) nodes get infected by each of their infected (active) first neighbors, with a constant transmission probability, and recover from the disease with a constant recovery probability. The Voter dynamics model the adoption of opinions; A node randomly selects the opinion (two opinions, 0 or 1, are considered) of one of its neighbors. The Cowan dynamics is a model of neural activity of biological neural networks, where the nodes—referred to as neurons—are either active (1) or inactive (0), and has been used to model the dynamics of single neurons or neuronal populations [66, 228]. Inactive neurons fire—i.e., become active—if their input current, coming from their firing neighbors, is above a given threshold.

The parameters ϕ of these models are fixed except in Sec. 6.8.1 where they are inferred by sampling from the joint posterior $P(G, \phi, \theta|X)$. In these experiments, we use non-informative uniform prior densities for all parameters in ϕ, and we constrain their value in finite intervals. For probability parameters, such as α_0 and β for the SIS and Cowan models, the prior density is $\rho(\phi) = 1$. For positive unbounded parameters, such as J for the Glauber model, and μ and ν for the Cowan model, we set the maximum value to 10 such that $\rho(\phi) = \frac{1}{10}$. Note that we fix $a = 1$ in the case of the Cowan model and $\beta_0 = 0$ for the SIS and the Cowan models in Sec. 6.8.1, without loss of generality since they are redundant parameters that may lead to non-identifiability issues.

6.10.6 Heuristic reconstruction algorithms

We consider three heuristic reconstruction approaches in this paper : the correlation matrix method [160], the Granger causality method [274], and the transfer entropy method [277]. The technical details can be found in Ref. [77], and we used the implementations of the `netrd` package [196].

These techniques compute a score matrix S, such that S_{ij} for each pair of nodes (i, j) correlates with probability that an edge exists between them. For the correlation matrix method, this score is the autocorrelation coefficient of the Markov chain :

$$S_{ij} = \frac{C_{ij}}{\sigma_i \sigma_j}, \quad C_{ij} = \frac{1}{T-1} \sum_{t=1}^{T} (X_{i,t} - \bar{X}_i)(X_{j,t} - \bar{X}_j), \tag{6.52}$$

where $\bar{X}_i = \frac{1}{T} \sum_{t=1}^{T} X_{i,t}$ and $\sigma_i^2 = \frac{1}{T-1} \sum_{t=1}^{T} (X_{i,t} - \bar{X}_i)^2$. The Granger causality method tests the hypothesis that the prediction of the time series of a single node i using a linear autoregressive model is improved by including the time series of node j. Specifically, it evaluates the statistical significance of error variances to determine if including node j's time series reduces prediction error of i's time series. This statistical tests is performed using the F-statistic :

$$S_{ij} = \frac{\Sigma_{ij}}{\Sigma_i}, \tag{6.53}$$

where Σ_i is the error variance of the auto-regressive model of i, and Σ_{ij} is the error variance of the other model that also includes j. In the transfer entropy method, the score is given by the transfer entropy from the time series of j to the time series of i :

$$S_{ij} = T_{X_j \to X_i} \,, \tag{6.54}$$

where

$$T_{X_j \to X_i} = H(X_{i,t+1}|X_{i,t}) - H(X_{i,t+1}|X_{i,t}, X_{j,t}) \,. \tag{6.55}$$

The entropies involved in the computation of $T_{X_j \to X_i}$ are evaluated by estimated the probabilities $P(X_{i,t}|X_{i,t-1})$ and $P(X_{i,t}|X_{i,t-1}, X_{j,t-1})$ with the corresponding frequency observed in the time series itself.

6.10.7 Relationship between the posterior loss and the reconstructability

In this section we show that the posterior loss is related to the reconstructability, under certain conditions. Let (g^*, x^*) be generated by the TDG model $M^* = (G^*, X^*)$, and $M = (G, X)$ be the reconstruction model. We define $p_i(x) = P(G = g_i|X = x)$ be the posterior probability of the graph g_i given some observation x. We also denote $\boldsymbol{p}(x) = \big(p_1(x), p_2(x), ..., p_{|\mathcal{G}|}(x)\big)$ the vector of the posterior probabilities of all graphs in $\mathcal{G}$. The posterior loss $L\big(\boldsymbol{y}, \boldsymbol{p}(x)\big)$ measures the accuracy of the posterior probabilities $\boldsymbol{p}(x)$ at predicting the correct labels $\boldsymbol{y}$, where $y_i = \delta(g^*, g_i)$ is a one-hot encoding of the true graph g^* using a Kronecker delta. It is defined as

$$L\big(\boldsymbol{y}, \boldsymbol{p}\big) = -\sum_{i=1}^{|\mathcal{G}|} y_i \log p_i(x) \,. \tag{6.56}$$

We also write $L\big(\boldsymbol{y}, \boldsymbol{p}^*\big)$ the posterior loss of the TDG model M^*, such that $\boldsymbol{p}^*$ is its corresponding posterior probability vector. Rewriting the posterior loss in terms of the posterior probability, we simply get

$$L(\boldsymbol{y}, \boldsymbol{p}) = -\log P(G = g^*|X = x^*) \,. \tag{6.57}$$

When the posterior probability factors with respect to the edges and the graphs do not contain multiedges, i.e.,

$$P(G = \boldsymbol{a}|X = x) = \prod_{i<j} \pi_{ij}(x)^{a_{ij}} \Big(1 - \pi_{ij}(x)\Big)^{1-a_{ij}} \,, \tag{6.58}$$

the posterior loss is given by Eq. (6.21).

The posterior loss averaged over the graph and data generated by M^* is

$$\mathbb{E}_{X^*,G^*}[L(\boldsymbol{y}, \boldsymbol{p})] \approx -\mathbb{E}_{X^*,G^*}[\log P(G = G^*|X = X^*)] \,, \tag{6.59}$$

where equality is achieved when the posterior distribution $P(G^*|X^*)$ truly factors as in Eq. (6.58). Hence, when M and M^* are equal in distribution, $\mathbb{E}_{X^*,G^*}[L(\boldsymbol{y}, \boldsymbol{p}^*)] \approx H(G^*|X^*)$.

Furthermore, the expected posterior loss is linearly related to the reconstructability, with a proportionality factor given by the entropy of G^* :

$$\mathbb{E}_{X^*,G^*}[L(\boldsymbol{y}, \boldsymbol{p}^*)] \approx H(G^*)\Big[1 - \Psi^*\Big]\,. \tag{6.60}$$

6.10.8 Bounds of the information gain

In this section, we show that the information gain is non-negative and bounded by the CE between the posterior and the prior. Recall that the information gain is given by Eq. (6.27) (equivalently Eq. (6.30)) :

$$\mathcal{I}_M(x) = \mathbb{E}_{G|X=x}\left[\log \frac{P(G|X)}{P(G)}\right].$$

Jensen's inequality states that for any random variable Y and any convex function ξ,

$$\xi\left(\mathbb{E}[Y]\right) \leq \mathbb{E}[\xi(Y)]\,. \tag{6.61}$$

Given that $\xi = -\log$ is a convex function, the information gain can be bounded using Jensen's inequality :

$$\mathcal{I}_M(x) \geq -\log \mathbb{E}_{G|X=x}\left[\frac{P(G)}{P(G|X=x)}\right]. \tag{6.62}$$

Simplifying the right-hand side yields

$$\begin{aligned}\mathcal{I}_M(x) &\geq -\log\left(\sum_{g\in\mathcal{G}} P(G|X=x)\frac{P(G)}{P(G|X=x)}\right)\\ &= -\log(1) = 0,\end{aligned}$$

the information gain lower bound $\mathcal{I}_M(x) \geq 0$ for all M and $x \in \mathcal{X}$.

The information gain can also be written as

$$\mathcal{I}_M(x) = \mathcal{H}\Big(P(G|X=x), P(G)\Big) - H(G|X=x)\,, \tag{6.63}$$

where $\mathcal{H}(p,q) = -\sum_x p(x)\log q(x)$ is the CE between two distributions p and q, and

$$H(G|X=x) = -\mathbb{E}_{G|X=x}[\log P(G|X=x)] \tag{6.64}$$

is the point-wise entropy of the graph posterior distribution for the observation x. The information gain is maximized when $H(G|X=x)$ is minimized, i.e., zero. The remaining term—the cross-entropy—is thus the upper bound of the information gain :

$$\mathcal{I}_M(x) \leq -\mathbb{E}_{G|X=x}[\log P(G)] = \Lambda_M(x)\,. \tag{6.65}$$

6.10.9 Numerical approximations of the mutual information

The mutual information $I(X;G)$ and information gain $\mathcal{I}_M(x)$ are generally intractable. Their intractability stems from the evaluation of the posterior, which requires computing of the evidence, denoted by $\zeta_M(x) = P(X = x)$:

$$\zeta_M(x) = \sum_{g \in \mathcal{G}} P(G = g) P(X = x | G = g) \,. \tag{6.66}$$

Indeed, there are potentially an exponential number of terms in this sum that need to be evaluated. Moreover, if M involves hyperparameters θ or parameters ϕ, they must also be marginalized to find the evidence. Fortunately, the evidence probability can be estimated efficiently using Monte Carlo techniques as described in this section. Note that we focus on the mutual information computation, but the same techniques can be applied to the information gain.

6.10.10 Graph enumeration approach

For sufficiently small random graphs ($N \approx 5$), the evidence probability can be computed by enumerating all graphs of $\mathcal{G}$ and by adding explicitly each term of Eq. (6.66). Using the law of large numbers, we can estimate the mutual information

$$I(X;G) \simeq \frac{1}{K} \sum_{k=1}^{K} \left[\log P\Big(X = x^{(k)} | G = g^{(k)}\Big) - \log P\Big(X = x^{(k)}\Big) \right], \tag{6.67}$$

where $(x^{(k)}, g^{(k)})_{k=1..K}$ are pairs of time series and graph sampled from (X, G) for M, the Bayesian generative model. The variance of this estimator scales with $K^{-1/2}$.

6.10.11 Variational mean-field approximation

This approach is based on Ref. [212] which uses a variational mean-field approximation to estimate the posterior probability instead of the evidence probability. The variational mean-field (MF) approximation assumes the conditional independence of the edges. For simple graphs, the MF posterior is

$$P_{\text{MF}}(G = \boldsymbol{a} | X = x) = \prod_{i \leq j} [\pi_{ij}(x)]^{a_{ij}} \left[1 - \pi_{ij}(x)\right]^{1 - a_{ij}} , \tag{6.68}$$

where $\pi_{ij}(x) \equiv P(A_{ij} = 1 | X = x)$ is the marginal conditional probability of existence of the edge (i, j) given x. For multigraphs, we obtain a similar expression involving a probability $\pi_{ij}(m \mid x) = P(A_{ij} = m | X = x)$ that there are m multiedges between i and j. In this case, the MF posterior becomes

$$P_{\text{MF}}(G = \boldsymbol{a} | X = x) = \prod_{i < j} \pi_{ij}(a_{ij} \mid x) \,. \tag{6.69}$$

By the conditional independent between the edges [65, Theorem 2.6.5], the MF approximation is a lower bound of the posterior entropy

$$H(G|X) \leq -\mathbb{E}_{X,G}[\log P_{\text{MF}}(G|X)]\,. \tag{6.70}$$

As for the graph enumeration approach, we compute the MF estimator of the mutual information with the Monte Carlo estimator

$$I(G;X) \gtrsim \frac{1}{K}\sum_{k=1}^{K}\left[\log P_{\text{MF}}\Big(G = g^{(k)}|X = x^{(k)}\Big) - \log P\Big(G = g^{(k)}\Big)\right]. \tag{6.71}$$

The posterior probability $P_{\text{MF}}\Big(G = g^{(k)}|X = x^{(k)}\Big)$ is also found using the law of large numbers : $\pi_{ij}(x)$ is estimated as the proportion of graphs that contain the edge (i,j) in a sample of the posterior. An analogous estimation is made in the multigraph case, where $\pi_{ij}(a_{ij} \mid X)$ is the proportion of graphs that contain a_{ij} edges between i and j in the sample. Although Eq. (6.71) is a biased estimator of the mutual information, it was shown in Ref. [212] that the bias is generally small, especially for large networks.

6.10.12 Graph evidence estimation for the stochastic block model

Using the stochastic block model (SBM) as the prior for our reconstruction model and for estimating the mutual information is challenging. Indeed, computing the graph entropy $H(G)$ requires that we marginalize the partition out of the prior probability

$$P(G) = \sum_{\boldsymbol{b}} P(G,\boldsymbol{b})\,, \tag{6.72}$$

which is intractable, but can be estimated. In Ref. [243], the author proposes a way to estimate the probability $P(\boldsymbol{b}|G)$ of partition given G—i.e., the posterior of a Bayesian model for community detection—by sampling a set of M partitions from it using Markov chain Monte Carlo (MCMC). The complete procedure is complex and involves aligning the sampled partitions, identifying aligned partition clusters and estimating the node marginal partition distribution $P(b_i = r|G) \equiv \pi_{i,r}(G)$ that node i is in group r—we refer to the original paper for technical details. To evaluate the graph marginal log probability, we first notice that

$$\log P(G) = \mathbb{E}_{\boldsymbol{b}|G}[\log P(G)] = \mathbb{E}_{\boldsymbol{b}|G}[\log P(G,\boldsymbol{b})] - H(\boldsymbol{b}|G)\,, \tag{6.73}$$

Given that we know the joint probability $P(G,\boldsymbol{b})$, the goal is then to estimate the partition entropy $H(\boldsymbol{b}|G)$. In Ref. [243], they propose a standard mean-field estimator :

$$P_{\text{MF}}(\boldsymbol{b}|G) = \prod_{i=1}^{N}\pi_{i,b_i}(G)\,, \tag{6.74}$$

where the marginal probabilities $\pi_{i,r}(G)$ can be estimated by the fraction of sampled relabeled partitions where node i is in block r. The mean-field estimator of the partition entropy is then

$$H_{\text{MF}}(\boldsymbol{b}|G) = -\sum_{i=1}^{N}\sum_{r=1}^{B_{\max}}\pi_{i,r}(G)\log\pi_{i,r}(G)\,, \tag{6.75}$$

where $B_{\max}$ is typically chosen to be equal to N, as there can be at most N non-empty groups. Also, note that $H_{\mathrm{MF}}(\boldsymbol{b}|G) \geq H(\boldsymbol{b}|G)$, since by factoring as in Eq. (6.74) we assume that the node memberships are conditionally independent, which has the effect of increasing the entropy [65]. Finally, the graph evidence entropy can be estimated using that mean-field estimator as follows :

$$H(G) \geq H(G, \boldsymbol{b}) - H_{\mathrm{MF}}(\boldsymbol{b}|G)\,, \tag{6.76}$$

which in turn constitutes a lower bound of $H(G)$.

6.10.13 Evidence estimation for model selection

The estimation of the evidence log probability relies on the previously discussed techniques for evaluating the posterior probability. Using the same approach as for Eq. (6.73), we obtain

$$\log \zeta(x) = \mathbb{E}_{G|X=x}[\log P(X, G)] - H(G|X = x)\,, \tag{6.77}$$

which follows from the fact that $\log P(X) = \log P(X, G) - \log P(G|X)$. Hence, we build an estimator by sampling from the posterior K graphs $g^{(k)}$ given x

$$\begin{aligned} \log \zeta(x) \simeq & \frac{1}{K} \sum_{k=1}^{K} \Big[\log P(G = g^{(k)}, X = x) \\ & - H(G|X = x) \Big]\,. \end{aligned} \tag{6.78}$$

By replacing $H(G|X = x)$ with a MF estimator of the posterior entropy, e.g. using Eq. (6.68) for simple graphs, we asymptotically get a lower bound of the evidence log probability :

$$\begin{aligned} \log \zeta(x) \gtrsim & \frac{1}{K} \sum_{k=1}^{K} \Big[\log P(G = g^{(k)}, X = x) \\ & - \sum_{i<i} h\big(\pi_{ij}(x)\big) \Big]\,, \end{aligned}$$

where we recall that $h(p) = -p \log p - (1 - p) \log(1 - p)$ is the binary entropy.

When there are parameters θ and ϕ for the graph G and data X, respectively, to infer alongside G, they must also be marginalized in the calculation of the evidence. Using a similar strategy as in the case where only G is inferred, we start from

$$\begin{aligned} \log \zeta(x) = & \mathbb{E}_{\theta,\phi,G|X=x}[\log P(X, \phi, G, \theta)] \\ & - H(\phi, G, \theta|X = x)\,, \end{aligned} \tag{6.79}$$

where we note that the expectation is taken over the complete joint posterior distribution $P(\phi, G, \theta|X = x)$. While the estimation of the first term is performed as previously, that of the second term, i.e., the posterior joint entropy, is more tricky. To build an estimator, we take advantage of the fact that the variables ϕ, G and θ are conditionally dependent in a specific

way $\theta \to G \to X \leftarrow \phi$, as we previously have pointed out. This means that the posterior joint entropy can be factored in the following way

$$\begin{aligned} H(\phi, G, \theta|X) &= H(\phi|G, \theta, X) + H(\theta|G, X) + H(G|X) \\ &= H(\phi|X) + H(\theta|G) + H(G|X)\,, \end{aligned} \tag{6.80}$$

where $H(\phi|X) = H(\phi|G, \theta, X)$ and $H(\theta|G, X) = H(\theta|G)$, by virtue of the facts that ϕ is conditionally independent of G and θ, and that θ is conditionally independent of X. For evaluating $H(\phi|X)$, since ϕ are continuous random variables, we estimate the posterior density with kernel density estimation (KDE) with a Gaussian kernel and estimate the differential entropy from the estimated density. In the case of $H(\theta|G)$, this term only concerns the SBM prior in our experiments, where θ are discrete variables $\boldsymbol{b}$ where b_i denotes the membership of node i to a group. We use the procedure described in Sec. 6.10.12 to estimate $H(\boldsymbol{b}|G)$.

6.10.14 Markov chain Monte-Carlo algorithm

To sample from the posterior distribution, we use a Markov chain Monte Carlo (MCMC) algorithm. Starting from a graph g, we propose a move to graph g', according to a proposition probability $P(G' = g'|G = g)$, and accept it with the Metropolis-Hastings probability :

$$\min\left(1, e^{-\log\Delta}\frac{P(G' = g|G = g')}{P(G' = g'|G = g)}\right)\,, \tag{6.81}$$

where $\Delta = \frac{P(G=g')P(X=x|G=g')}{P(G=g)P(X=x|G=g)}$ is the ratio between the posterior probabilities of g and g'. This ratio can be computed efficiently in $\mathcal{O}(T)$ by keeping in memory, for each node i and time t, the number of inactive neighbors $n_{i,t}$ and the number of active neighbors $m_{i,t}$ (see Refs. [212, 242]). Equation (6.81) allows to sample from the posterior distribution $P(G \mid X)$ without the requirement to compute the intractable normalization constant $P(X)$.

We use two types of graph move propositions : double-edge swaps and hinge flips [63]. Double-edge swaps consists in selecting two edges at random, breaking them into two pairs of stubs and reconnecting the stubs to create two new edges. This type of move leaves the degree sequence and total edge count unchanged. Hinge flips consist in selecting an edge and a node at random, and reconnecting the edge to this node by detaching it from one of its end. Unlike double-edge swaps, hinge flips do not preserve the degree sequence. There are many considerations to take when implementing these moves and computing their proposal probabilities ; we refer to Refs. [63, 95, 212] for technical details.

For most of our numerical experiments, the total number of edges is fixed. At each proposition, we randomly select to perform a double-edge swap or a hinge flip with equal probability. We found that by doing this, the mixing time was significantly improved.

This sampling scheme can be generalized when additional hyperparameters θ of the graph prior or parameters ϕ from the likelihood must be inferred as in Sec. 6.8. We consider a

Gibbs sampling scheme where each random variable G, θ and ϕ is sampled sequentially and conditioned on the others. In all cases, the acceptance probability follows Eq. (6.81), where G is replaced by either θ or ϕ when these parameters are sampled. In this paper, only the SBM among the considered graph models contains parameters of the type of θ. To sample from these, we use the same procedure as in Ref. [240, Sec. VI]—we refer to it for further detail. The data models considered contains many parameters that we would like to infer, for example, the infection and recovery probabilities, λ and β, respectively, in the SIS. These parameters are real number constrained within an interval (for instance, $[0, 1]$ for the recovery probability β); Hence, any proposed move where φ falls outside of this interval is rejected. We propose moves drawn from a Gaussian distribution with density

$$p(\varphi'|\varphi) \propto \exp\left[-\frac{(\varphi' - \varphi)^2}{2\sigma^2}\right] . \tag{6.82}$$

In Sec. 6.8, we fix $\sigma = 0.1$.

6.10.15 Reconstructability of graph models with delta distribution

Suppose X is generated using a single graph g^*. If we were to observe many realizations of X with the single graph g^*, the graph prior of the TDG would be $P(G = g) = \delta(g, g^*)$ and the evidence of this process would be exactly equal to the likelihood of the TDG process, denoted $p^*(X) \equiv P(X|G = g^*)$. As a result, the mutual information and the entropy of G would both be zero, and so the reconstructability would be undefined.

To bypass this problem, suppose that the graph generating model is instead parametrized by a probability ϵ such that G yields g^* with probability $1 - \epsilon$ and the others uniformly, that is,

$$P(G = g) = \begin{cases} (1 - \epsilon) & \text{if } g = g^*, \\ \frac{\epsilon}{Z} & \text{otherwise,} \end{cases} \tag{6.83}$$

where $Z = |\mathcal{Z}|$ such that $\mathcal{Z} = \{g \in \mathcal{G} : g \neq g^*\}$ is the set of graphs different from g^*. Then, by taking the limit when $\epsilon \to 0$, we recover the scenario where the graph generating model is a Kronecker delta distribution.

Let us investigate the scaling of $H(G)$ and $I(X; G)$. First, we have

$$H(G) = -(1 - \epsilon)\log(1 - \epsilon) - \epsilon \sum_{g \in \mathcal{Z}} \frac{1}{Z} \log \frac{\epsilon}{Z} = h(\epsilon) + \epsilon \log Z . \tag{6.84}$$

Second, we have the evidence of this joint model :

$$P(X = x) = (1 - \epsilon)p^*(x) + \epsilon \sum_{g \in \mathcal{Z}} \frac{P(X = x|G = g)}{Z} = (1 - \epsilon)p^*(x) + \epsilon q(x) ,$$

where $q(x) = \sum_{g\in\mathcal{Z}} \frac{P(X=x|G=g)}{Z}$ is the evidence of x in the complementary model for which the only possible graphs are those in $\mathcal{Z}$. Then, the reconstruction entropy $H(G|X)$ is evaluated as follows :

$$
\begin{aligned}
H(G|X) = &-\sum_x \Bigg[P(X=x, G=g^*)\log P(G=g^*|X=x) \\
&\qquad + \sum_{g\in\mathcal{Z}} P(X=x, G=g)\log P(G=g|X=x)\Bigg] \\
= &-\sum_x \Bigg[P(X=x|G=g^*)P(G=g^*)\log\frac{P(X=x|G=g^*)P(G=g^*)}{P(X=x)} \\
&\qquad + \sum_{g\in\mathcal{Z}} P(X=x|G=g)P(G=g)\log\frac{P(X=x|G=g)P(G=g)}{P(X=x)}\Bigg] \\
= &-\sum_x \Bigg[(1-\epsilon)p^*(x)\log\left[\frac{(1-\epsilon)p^*(x)}{(1-\epsilon)p^*(x)+\epsilon q(x)}\right] \\
&\qquad + \epsilon\sum_{g\in\mathcal{Z}} Z^{-1}P(X=x|G=g)\log\left[\frac{\epsilon Z^{-1}P(X=x|G=g)}{(1-\epsilon)p^*(x)+\epsilon q(x)}\right]\Bigg] \\
= &-\sum_x \Bigg[(1-\epsilon)p^*(x)\log(1-\epsilon) + (1-\epsilon)p^*(x)\log\left[\frac{p^*(x)}{(1-\epsilon)p^*(x)+\epsilon q(x)}\right] \\
&\qquad + \epsilon\sum_{g\in\mathcal{Z}} Z^{-1}P(X=x|G=g)\log\frac{\epsilon}{Z} \\
&\qquad + \epsilon\sum_{g\in\mathcal{Z}} Z^{-1}P(X=x|G=g)\log\left[\frac{P(X=x|G=g)}{(1-\epsilon)p^*(x)+\epsilon q(x)}\right]\Bigg] \\
= &\, h(\epsilon) + \epsilon\log Z - (1-\epsilon)A - \epsilon B + \epsilon H(X|\bar{G}),
\end{aligned}
$$

where

$$
\begin{aligned}
A &= -\sum_x p^*(x)\log\left[1+\epsilon\left(\frac{q(x)}{p^*(x)}-1\right)\right], \\
B &= -\sum_x q(x)\log\left[(1-\epsilon)p^*(x)+\epsilon q(x)\right],
\end{aligned}
$$

and $\bar{G}$ denotes the random graph uniformly distributed over the complementary set $\mathcal{Z}$. Recalling Eqs. (6.17) and (6.84), we conclude that

$$
I(X;G) = (1-\epsilon)A + \epsilon B - \epsilon H(X|\bar{G}).
$$

By developing the logarithms as $\log(1+x) = x + \mathcal{O}(x^2)$, we can show easily that the leading term of A is of second order in ϵ (no constant or linear terms) and the leading terms of B is

$$
B = -\sum_x q(x)\left[\log p^*(x) + \epsilon\left(\frac{q(x)}{p^*(x)}-1\right) + \mathcal{O}(\epsilon^2)\right].
$$

This leaves us with

$$
I(X;G) = \epsilon\Big(-H(X|\bar{G}) - \sum_x q(x)\log p^*(x)\Big) + \mathcal{O}(\epsilon^2).
$$

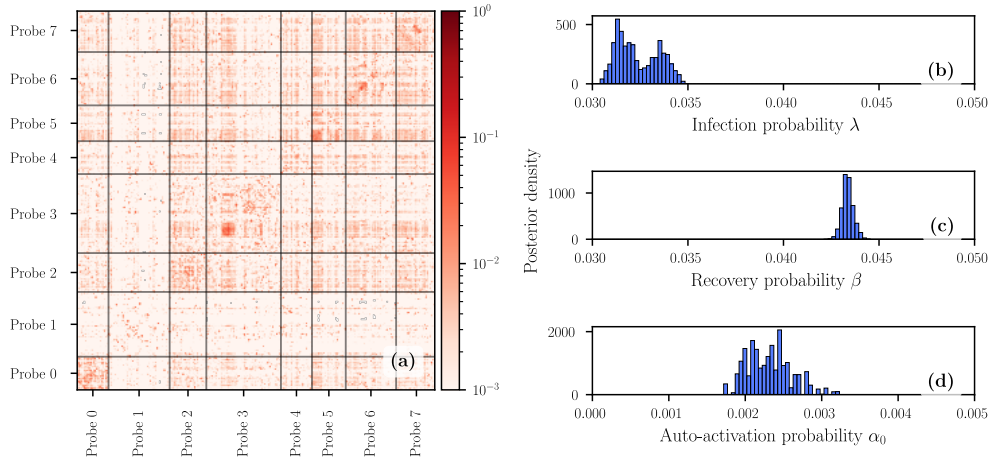


FIGURE 6.8 – Posterior of the maximum evidence model (SIS model with SBM prior) : (a) posterior probability matrix of the edge occupancy, (b) histogram of the infection probability, (c) the recovery probability and (d) the auto-activation probability. In (a), each entry of the matrix represents the number of times the edge has been sampled, among the 8000 posterior samples. Also, we highlight the probe partition of the graph using deemed black separation lines.

Also, from the above equation for $H(G)$, we have that the leading terms are $H(G) = \epsilon(\log \epsilon^{-1} + 1 + \log Z) + \mathcal{O}(\epsilon^2)$. Consequently, the reconstructability, being the ratio of $I(X;G)$ and $H(G)$, approaches zero as $\epsilon \to 0$ with leading term $\mathcal{O}\left(\frac{1}{\log \epsilon^{-1}}\right)$.

6.10.16 Inference of brain networks

In this section, we describe the procedure we used to reconstruct the mouse brain network from Sec. 6.8.1. The raw data is available in [299], which was originally presented in Ref. [300]. We refer to their paper for any technical detail regarding the data collection.

6.10.17 Data preprocessing

This dataset is composed of the spontaneous activity of the brains of three mice (Krebs, Robbins and Waksman) monitored via eight neuropixel probes each. These probes record the time stamps of each spike of individual neurons in different regions of the brain for a duration of 20 minutes.

For the purpose of the experiment, we choose the Krebs recoding which count 1462 monitored neurons. First, we discretize time into 10^5 steps and map each spike time stamp to the correct discrete time interval. Then, since the time duration of the spikes are not available in the original dataset, we artificially extend the spikes for a random duration, which is exponentially distributed with mean 0.012 seconds—this value corresponds to an approximate activation duration of 10 time steps. Finally, we partition the complete time series into 100 segments of equal size (1000 steps). Figure 6.6(a) corresponds to the first among the 100

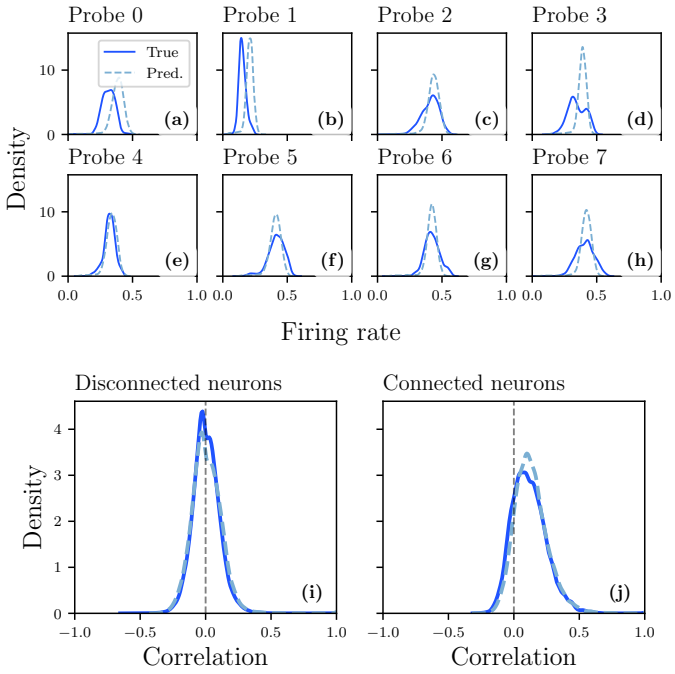


FIGURE 6.9 – Posterior predictive checks of the maximum evidence model (SIS model with SBM prior), showing Gaussian kernel density estimations of the distributions of (a–h) firing rates (i–j) correlation. Panels (a–h) show the firing rate probability density for each probe. In panels (i–j), we show the probability density of the correlation coefficients (Eq. (6.52)) between neurons that are connected [panel (i)] and disconnected [panel (j)] in the posterior graph. In all panels, the statistics corresponding to the observed time series [Fig. 6.6(a), labeled "True"] are shown using the solid dark blue lines, while those of the posterior predictions are shown using the dashed light blue line (labeled "Pred."). Also, the predictions are gathered from 100 samples of the model, where each used different parameters and graph jointly sampled from the posterior.

segments of the discretized time series.

6.10.18 Inference procedure

The inference procedure is very similar to that presented in Appendix 6.10.14. We consider a model parametrized by X, G and their parameters ϕ and θ, respectively. However, we have an additional limitation : We do not know the number of edges in the graph. We tried extending our MCMC algorithm by including moves that do not preserve the number of edges—i.e., adding or removing a single edge—, but we found that these attempts suffered from poor mixing time.

To alleviate this problem, we propose to search for the number of edges first, by minimizing the description length $\log P(X, \phi, G, \theta)$ [245]. We solve this optimization problem using a semi-greedy algorithm where we propose K move candidates, and select the one that locally minimizes the objective. Like for our MCMC algorithm, we iterate over G, θ and ϕ sequen-

		Average	Std. Dev.
Model	Graph prior		
Cowan	ER	1 296.75	77.00
	UCM	1 302.75	70.38
	SBM	1 462.00	126.03
Glauber	ER	18 098.88	53.94
	UCM	18 612.38	93.50
	SBM	18 895.63	79.47
SIS	ER	1 388.50	28.92
	UCM	1 296.63	50.15
	SBM	**1 722.38**	**89.98**

TABLE 6.5 – Statistics for the number of edges determined from the semi-greedy algorithm for each reconstruction model considered in Sec. 6.8.1. The highlighted row (SIS with SBM) corresponds to the maximum evidence model associated with Figs. 6.8 and 6.9. The average and standard deviations (std. dev.) are obtained from the 8 parallel chains used for the inference.

tially to locally perform the optimization on each of them independently. At each step, we sample 10000 candidates for G and θ, and 10 for ϕ. Once the number of edges has converged, we stop the semi-greedy algorithm and freeze the number of edges. The MCMC algorithm then proceeds to sample from the posterior with a fixed number of edges.

In Table 6.5, we summarize the results of the semi-greedy search for the number of edges. Given that the number of nodes is 1462, our results show that the inferred networks are surprisingly sparse, except for the Glauber model which inferred one order of magnitude more edges than the Cowan and SIS models.

6.10.19 Posterior inspection

The graph and parameter marginal posteriors of the maximum evidence model are illustrated in Fig. 6.8. We also include a validation of the posterior on the inference data. Figure 6.9 shows the posterior predictive check validation, which includes a prediction of the firing rates and the correlation coefficients between connected and disconnected neurons as test quantities. The inferred graph also allows to reproduce the shape of the cross-correlation density distribution.

Troisième partie

Reconstruire la dynamique

Chapitre 7

Théorie de l'apprentissage profond sur graphes

Actuellement, une révolution technologique est à l'oeuvre, propulsée par des avancées scientifiques rapides, par la collecte de données massives et par l'accroissement de la puissance de calcul. L'apprentissage de réseaux profonds (*deep learning* en anglais) est le moteur de cette révolution et permet aujourd'hui un développement de l'intelligence artificielle sans précédent. En outre, le traitement d'images et du langage naturel ont été résolument transformés par l'apprentissage profond [81, 312], bien que la frontière de cette nouvelle technologie ne cesse de s'étendre. Le prix Nobel de Physique en 2024 a d'ailleurs été octroyé à Geoffrey Hinton et John Hopfield pour leur contribution en apprentissage profond, témoignant de l'impact de ces travaux sur la communauté physicienne. De récents travaux ont également démontré l'applicabilité des réseaux profonds pour étudier les systèmes chaotiques [234, 235], la mécanique des fluides [166] et les systèmes dynamiques au sens large [74, 189]. De plus, les réseaux profonds ont été adaptés pour les graphes [126, 156], menant notamment vers la synthèse de nouvelles molécules [144] utilisées en pharmacologie [96, 349]. La promesse des réseaux profonds est donc immense, et trouve certainement un domaine d'applications dans les systèmes complexes.

Dans ce chapitre, on présente les bases de l'apprentissage profond, lesquelles sont utilisées au Chapitre 8 pour reconstruire les dynamiques sur graphes à partir de données. Notre discussion commence par une brève histoire du réseau de neurones artificiels à la Sec. 7.1—le modèle de base de l'apprentissage profond. Ensuite, on présente l'apprentissage profond qui consiste à entraîner ces modèles à effectuer des tâches (§ 7.2 et § 7.3). On termine la section en discutant des biais inductifs qui guident l'apprentissage de certaines architectures de modèles (Sec. 7.4), notamment les réseaux de neurones sur graphes (Sec. 7.5) que nous utilisons dans le Chapitre 8.

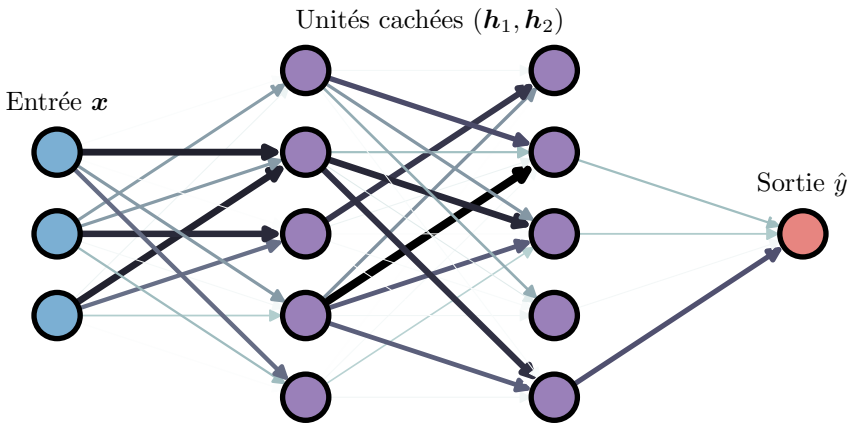


FIGURE 7.1 – Illustration d'un perceptron multicouche (MLP) composé de deux couches cachées de neurones ($\boldsymbol{h}_1$ et $\boldsymbol{h}_2$). Les poids du MLP sont représentés par les liens reliant les neurones d'une couche à ceux de la couche suivante : l'épaisseur des liens et leur couleur représentent l'amplitude du poid. Le signal d'entrée $\boldsymbol{x}$ est transformé en un signal de sortie $\hat{y}$, en traversant les couches cachées $\boldsymbol{h}_1$ et $\boldsymbol{h}_2$ de gauche à droite.

7.1 Brève histoire des réseaux de neurones artificiels

Historiquement, les réseaux de neurones artificiels ont été introduits comme modèles computationnels du cerveau. Le perceptron, un modèle de neurones artificiels introduit par W. McCulloch et W. Pitts en 1943 [197]—puis réalisé pour la première fois en 1957 par F. Rosenblatt [260, 261]—, est le précurseur du réseau de neurones moderne. La motivation de leurs travaux était de relier des principes connus de l'époque en neurosciences (tels que le principe *all-or-none*[1]) à des opérations logiques simples pouvant être utilisées pour d'autres applications, notamment la reconnaissance d'images. Dans un perceptron, les neurones forment des unités de calcul élémentaires qui transforment un signal d'entrée $\boldsymbol{x} \in \mathbb{R}^N$ en un signal de sortie binaire $\hat{y} \in \{0, 1\}$. À la manière d'un cerveau biologique, ces neurones sont connectés par des synapses, représentées par des poids $\boldsymbol{w} \in \mathbb{R}^N$.

Dans l'implémentation originale de Rosenblatt (*Perceptron Mark I* [260]), la transformation se faisait effectivement via une fonction d'activation Heaviside σ de seuil τ :

$$\hat{y} = \sigma(\boldsymbol{w} \cdot \boldsymbol{x} - \tau)\,. \tag{7.1}$$

Les paramètres $\boldsymbol{w}$ et τ étaient alors appris à partir d'une règle d'ajustement simple :

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \varepsilon(y - \hat{y})\boldsymbol{x}\,, \tag{7.2}$$

1. En neurosciences, le principe *all-or-none* stipule que l'intensité d'une impulsion neuronale sollicitée par un stimulus, c'est-à-dire l'activation d'une connexion synaptique, est maximale et invariante à l'intensité du stimulus.

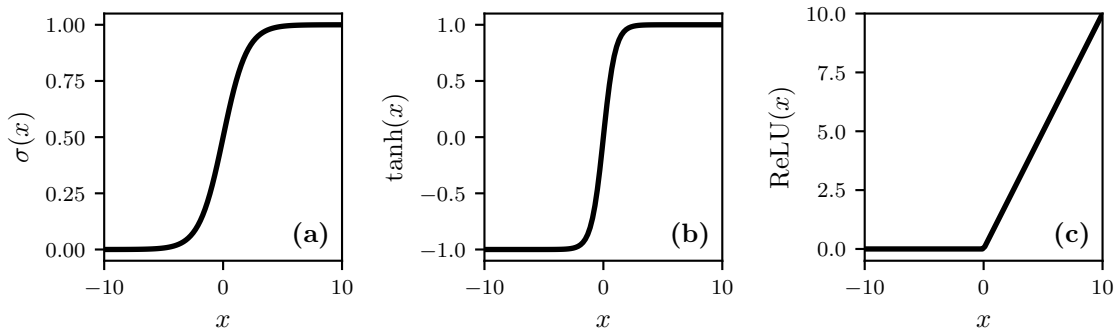


FIGURE 7.2 – Exemples de fonction d'activation : (a) fonction sigmoidale $\sigma(x) = (1 + e^{-x})^{-1}$, (b) fonction tangeante hyperbolique $\tanh(x) = \frac{1-e^{-x}}{1+e^{-x}}$, (c) fontion (ReLU) $\text{ReLU}(x) = \max(0, x)$.

où ε est le taux d'apprentissage et y est la valeur désirée. Cette règle permet de mettre à jour les poids $\boldsymbol{w}$ de sorte que la prédiction $\hat{y}$ minimise l'erreur quadratique moyenne :

$$\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2 . \tag{7.3}$$

Nous verrons à la Section 7.2 comment obtenir systématiquement une règle d'ajustement pour un modèle et une fonction d'erreur arbitraires. Alors que Rosenblatt voyait un énorme potentiel d'applications pour le perceptron, plusieurs travaux ont suivi pour établir les limites fondamentales de son modèle (par exemple, ceux de Minsky et Papert [202]). Notamment, le fait que le perceptron soit linéaire en $\boldsymbol{x}$ lui confère de faibles performances sur des tâches concrètes. C'est avec le perceptron multicouche (MLP, pour *multilayer perceptron*) qu'apparaît le potentiel réel des réseaux de neurones artificiels, bien qu'ils nécessitent encore plusieurs innovations théoriques et demeurent à l'époque difficiles à entraîner. À la différence du perceptron simple, le MLP possède des couches de neurones dites *cachées*, typiquement dénotées par $\boldsymbol{h} = (h_j)_{j \in [n]}$, lesquels encodent des représentations abstraites (et non linéaires) du signal d'entrée (voir Fig. 7.1). Par exemple, un MLP à deux couches est une composition de deux fonctions de la forme suivante :

$$h_j = \sigma\left(\boldsymbol{w}_j^{(1)} \cdot \boldsymbol{x} + b_j^{(1)}\right) , \tag{7.4}$$

$$\hat{y} = \boldsymbol{w}^{(2)} \cdot \boldsymbol{h} + b^{(2)} . \tag{7.5}$$

Dans cet exemple, le MLP contient un ensemble de poids $\boldsymbol{w}_1^{(1)}, \boldsymbol{w}_2^{(1)}, ..., \boldsymbol{w}_n^{(1)}$ et $\boldsymbol{w}^{(2)}$, où $\boldsymbol{w}_j^{(1)} \in \mathbb{R}^N$ et $\boldsymbol{w}^{(2)} \in \mathbb{R}^n$ (on rappelle que N est le nombre de dimensions du vecteur d'entrée, $\boldsymbol{x}$) ; et de biais $b_1^{(1)}, ..., b_n^{(1)}$ et $b^{(2)}$ avec $b_j^{(1)}, b^{(2)} \in \mathbb{R}$, lesquels sont appris à partir d'un jeu de données d'entraînement. On dénote l'ensemble complet des paramètres par $\boldsymbol{\theta}$, qui contient tous les poids et biais du modèle. Dans la mesure où la fonction d'activation σ est non linéaire, le signal de sortie $\hat{y}$ est lui aussi non linéaire en $\boldsymbol{x}$.

La suite d'innovations liées au perceptron dans les années 60 est de courte durée, mais elle reprend au courant des années 80. En effet, les premiers théorèmes d'approximation univer-

selle (TAU) font leur apparition et démontrent que les MLPs peuvent approximer n'importe quelle fonction continue à une précision arbitraire. Le plus célèbre est celui de G. Cybenko en 1989 [70] :

Theorème 7.1 (Théorème d'approximation universel (Cybenko, 1989))**.** *Soit σ une fonction sigmoïdale continue (voir Fig. 7.2(a)). Pour toute fonction réelle et continue $f : \mathcal{I}_N \to \mathbb{R}$ définie sur l'hypercube $\mathcal{I}_N = [0,1]^N$ et pour tout $n \in \mathbb{N}$, il existe un perceptron ayant une couche cachée de n neurones, noté $\hat{f}_n(\boldsymbol{x})$ avec $\boldsymbol{x} \in \mathcal{I}_N$, telle que*

$$\hat{f}_n(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \sigma\Big(\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i\Big)\,, \tag{7.6}$$

où α_i et $\boldsymbol{w}_i = (w_{ij})_{j\in[n]}$ sont des paramètres de $\hat{f}_n$, de sorte que, pour un ensemble compact $D \subset \mathcal{I}_n$ de mesure $\mu(D) \geq 1 - \epsilon$ avec $\epsilon > 0$,

$$\left|\hat{f}_n(\boldsymbol{x}) - f(\boldsymbol{x})\right| < \epsilon\,, \quad \forall \boldsymbol{x} \in D\,. \tag{7.7}$$

Démonstration. La preuve est disponible à la Réf. [70, Théorème 3]. □

Intuitivement, le TAU de Cybenko stipule qu'il existe toujours une configuration de paramètres α_i et $\boldsymbol{w}_i$ pour laquelle $\hat{f}_n$ peut approximer une fonction continue f donnée, et ce, à une précision ϵ. En outre, la précision ϵ de l'approximation dépend du nombre de neurones n dans la couche cachée ; plus n est grand, plus ϵ est petit. Plusieurs variantes du TAU de Cybenko ont été démontrées depuis, dans lesquelles des architectures différentes sont considérées : par exemple, des MLPs ayant un nombre arbitraire de couches [188], des différentes fonctions d'activation [174] (notamment pour les fonctions tangeante hyperbolique et ReLU sur la Fig. 7.2) et des architectures plus complexes [46]. Encore aujourd'hui, les TAUs, qui constituent une fondation mathématique solide pour l'apprentissage profond, représentent un domaine de recherche actif.

7.2 Apprentissage profond

Le propre de l'apprentissage profond implique l'entraînement de réseaux de neurones profonds, noté $\hat{f}_{\boldsymbol{\theta}}$, dont l'architecture s'inspire des mêmes principes que ceux sur lequel le MLP repose. Or, la profondeur fait ici référence au nombre de couches cachées qu'un modèle contient. Ainsi, les modèles profonds composés de plusieurs couches possèdent un grand nombre de paramètres qui, couplés à leur non-linéarité, leur confèrent une énorme capacité d'approximation. C'est pourquoi les réseaux de neurones profonds sont aujourd'hui utilisés pour résoudre des problèmes complexes—par exemple, la génération de texte [312] et d'images [258], et la reconnaissance d'objets [164]—, pour lesquels historiquement les méthodes classiques peinaient à offrir des solutions satisfaisantes et efficaces.

L'entraînement représente un problème d'optimisation qui consiste à trouver un ensemble de paramètres $\boldsymbol{\theta}^*$ minimisant une fonction dite *objectif* $\mathcal{L}$, c'est-à-dire qu'on cherche à optimiser. Celle-ci encode formellement l'objectif de l'Éq. (7.7), et mesure le niveau de précision du modèle $\hat{f}_{\boldsymbol{\theta}}$ par rapport à la fonction cible f. Or, f est généralement inconnue ou trop complexe pour être évaluée directement. On doit donc se contenter de jeux de données composés de paires d'exemples d'entrée $\boldsymbol{x}$ et de leur valeur cible associée, c'est-à-dire $y = f(\boldsymbol{x})$. L'objectif de l'entraînement est alors donné par

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{X},Y} \left[\mathcal{L}\Big(Y, \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{X})\Big) \right] , \tag{7.8}$$

où l'espérance est calculée sur l'ensemble de données d'entraînement issu des variables aléatoires $(\boldsymbol{X}, Y)$.

L'optimisation des réseaux de neurones profonds est en soit un problème complexe. En effet, la fonction objectif $\mathcal{L}$ est généralement non convexe dans l'espace des paramètres $\boldsymbol{\theta}$, ce qui signifie que $\mathcal{L}$ contient plusieurs minima locaux. De plus, on peut montrer que la recherche du minimum global constitue un problème NP-complet [34]. Ces limites théoriques ont longtemps freiné le développement de l'apprentissage profond. Les progrès du calcul graphique (GPU), l'avènement des optimisateurs modernes [154] et des plateformes de calculs de différentiation automatique ont permis de surmonter ces limites théoriques au courant des années 2010.

Les optimisateurs des plus récentes générations fonctionnent par descente de gradient stochastique (consulter la Réf. [264] pour une revue détaillée de ces algorithmes). En bref, ils optimisent la fonction objectif en ajustant les paramètres $\boldsymbol{\theta}$ comme suit :

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \varepsilon g_{\boldsymbol{\theta}} \tag{7.9}$$

où

$$g_{\boldsymbol{\theta}} \equiv \mathbb{E}_{\boldsymbol{X},Y} \left[\nabla_{\boldsymbol{\theta}} \mathcal{L}(Y, \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{X})) \right] \tag{7.10}$$

$$= \mathbb{E}_{\boldsymbol{X},Y} \left[\mathcal{L}'(Y, \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{X}))\ \hat{g}_{\boldsymbol{\theta}}(\boldsymbol{X}) \right] , \tag{7.11}$$

est le gradient de la fonction objectif par rapport aux paramètres $\boldsymbol{\theta}$, avec

$$\mathcal{L}'(y, z) \equiv \frac{\partial}{\partial x} \mathcal{L}(y, x)\Big|_{x=z} \quad \text{et} \quad \hat{g}_{\theta}(\boldsymbol{x}) \equiv \nabla_{\boldsymbol{\theta}} \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \, .$$

Reprenant l'exemple précédent du *Perceptron Mark I*, on montre directement que la règle d'ajustement de l'Éq. (7.2) est en fait un cas de l'Éq. (7.9), considérant que $\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$ et $\hat{f}_{\boldsymbol{\theta}}$ est donnée par l'Éq. (7.1). Intuitivement, ces optimisateurs effectuent une recherche dans l'espace des paramètres en suivant la direction du gradient. Le gradient est alors évalué par l'algorithme de rétropropagation, qui pour les réseaux de neurones profonds s'effectue

efficacement en un temps linéaire avec le nombre de paramètres du modèle. La rétropropagation applique successivement la règle de dérivation en chaîne, couche après couche. Par exemple, pour le MLP à deux couches, on calcule pour la dernière couche

$$\hat{g}_{b^{(2)}}(\boldsymbol{x}) = 1\,, \quad \hat{g}_{w_j^{(2)}}(\boldsymbol{x}) = h_j\,, \quad \hat{g}_{h_j}(\boldsymbol{x}) = w_j^{(2)}\,, \tag{7.12}$$

et pour la couche qui précède, la première couche cachée, on a

$$\hat{g}_{b_j^{(1)}}(\boldsymbol{x}) = \hat{g}_{h_j}(\boldsymbol{x}_j)\sigma'(\tilde{h}_j) = w_j^{(2)}\sigma'(\tilde{h}_j)\,, \tag{7.13}$$

$$\hat{g}_{w_{i,j}^{(1)}}(\boldsymbol{x}) = \hat{g}_{h_j}(\boldsymbol{x})\sigma'(\tilde{h}_j)x_i = w_j^{(2)}\sigma'(\tilde{h}_j)x_i\,, \tag{7.14}$$

où $\sigma'(x)$ est la dérivée de la fonction d'activation évaluée en x, et $\tilde{h}_j = \boldsymbol{w}_j^{(1)} \cdot \boldsymbol{x} + b_j^{(1)}$. Comme on le constate, les dérivées partielles des paramètres proches de la sortie interviennent directement dans le calcul des dérivées des paramètres qui les précèdent. Pour des architectures plus profondes, la rétropropagation s'applique tout à fait similairement, couche après couche ; le gradient est en pratique gardé en mémoire pour chaque paramètre précisément pour cette raison. En ajustant les paramètres $\boldsymbol{\theta}$ de cette manière, le modèle $\hat{f}_{\boldsymbol{\theta}}$ converge vers un minimum local de la fonction objectif $\mathcal{L}$, qui, si le modèle est correctement entraîné, correspond à une bonne approximation de la fonction cible f.

7.3 Entraînement des modèles supervisés

Le choix de la fonction objectif dépend du type d'apprentissage que l'on souhaite effectuer. Il y existe plusieurs types d'apprentissage que l'on catégorise en deux grandes familles : supervisé et non supervisé. Lorsqu'un modèle est supervisé, on l'entraîne à prédire des valeurs cibles y, comme nous l'avons vu précédemment avec le perceptron. Chacune de ces valeurs est ainsi associée à une valeur d'entrée $\boldsymbol{x}$. Les modèles non supervisés, quant à eux, apprennent à partir de données $\boldsymbol{x}$ sans annotations (c'est-à-dire sans valeur cible), pour en générer de nouvelles ou pour construire des représentations compressées. Nous nous intéresserons particulièrement aux modèles supervisés, qui sont utilisés au Chapitre 8. Pour plus d'information à propos des modèles non supervisés, on réfère le lecteur à la Réf. [119, Chapitres 5, 14 et 15] pour une introduction. Également, voir les Réfs. [50, 334] pour une revue des récentes avancées dans les approches génératives, et la Réf. [123] pour une autre concernant les méthodes de pointe en apprentissage non supervisé. [2]

Dans la famille des modèles supervisés, on distingue plusieurs types de tâches, comme la classification, la regression, la détection d'object, etc. La classification est le cas qui nous intéressera particulièrement : voir la Réf. [119, Section 5.1] pour plus de détails sur les autres

2. Notons que les domaines de l'apprentissage non supervisé et des méthodes génératives connaissent présentement une croissance fulgurante, et que les références citées ici ne seront possiblement plus à jour dans un futur plus ou moins loin.

types de tâches. En classification, la cible y prend des valeurs discrètes $\{1, 2, \cdots, C\}$, que l'on associe à des classes. Qui plus est, le modèle profond est entraîné pour prédire un vecteur $\hat{\boldsymbol{y}} = (\hat{y}_1, \cdots, \hat{y}_C)$, où $\hat{y}_c$ prédit la probabilité que $\boldsymbol{x}$ appartienne à la classe c. On entraîne typiquement un tel modèle avec une fonction objectif appelée l'*entropie croisée* :[3]

$$\mathcal{L}(y, \hat{\boldsymbol{y}}) = -\sum_{c=1}^{C} \delta(y, c) \log \hat{y}_c \,, \tag{7.15}$$

où $\delta(y, c)$ est le delta de Kronecker qui vaut 1 si $y = c$ et 0 autrement. La minimisation de l'entropie croisée réalise deux objectifs :

1. Maximiser la probabilité que le modèle prédise correctement les classes de chacun des exemples du jeu de données ;
2. Minimiser la divergence de KL (Éq. (4.21))—c'est-à-dire le degré de dissimilarité—entre la distribution de probabilité prédite et la distribution cible, si cette dernière contient de l'incertitude.

Le premier point s'explique par le fait que l'entropie croisée n'est rien d'autre que la log-vraisemblance négative de la distribution prédite. Le deuxième point est plus subtil. La divergence de KL s'écrit

$$\mathcal{D}_{KL}(p \| q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \mathcal{H}(p, q) - \mathcal{H}(p) \,, \tag{7.16}$$

pour deux distributions de probabilité discrètes p et q, de support $\mathcal{X}$, où

$$\mathcal{H}(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x) \,, \quad \text{et} \quad \mathcal{H}(p) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \,, \tag{7.17}$$

sont respectivement l'entropie croisée et l'entropie de p. Or, si $p(y|x)$ est la distribution cible sachant les données d'entrée x, et q, la distribution prédite par le modèle, c'est-à-dire

3. Il existe des fonctions objectif de classification autre que l'entropie croisée, comme la perte focale [182], l'entropie croisée de Rényi [306], ainsi que des variantes pondérées de chacune de ces fonctions. Elles ont chacunes leurs avantages et inconvénients, mais elles servent généralement à contrôler le déséquilibre des classes dans les jeux de données. Par exemple, la perte focale,

$$\mathcal{L}(y, \hat{\boldsymbol{y}}) = -\sum_{c=1}^{C} \delta(y, c)(1 - \hat{y}_c)^{\gamma} \log \hat{y}_c \,,$$

où γ est un hyperparamètre, pénalise davantage les exemples mal classifiés. L'inconvénient principal de ces autres fonctions objectif est qu'elles distorsionnent la distribution a posteriori que le modèle apprend. On réfère à la Réf. [307] qui en fait une analyse détaillée et propose une manière de recalibrer la sortie du modèle pour corriger cette distorsion.

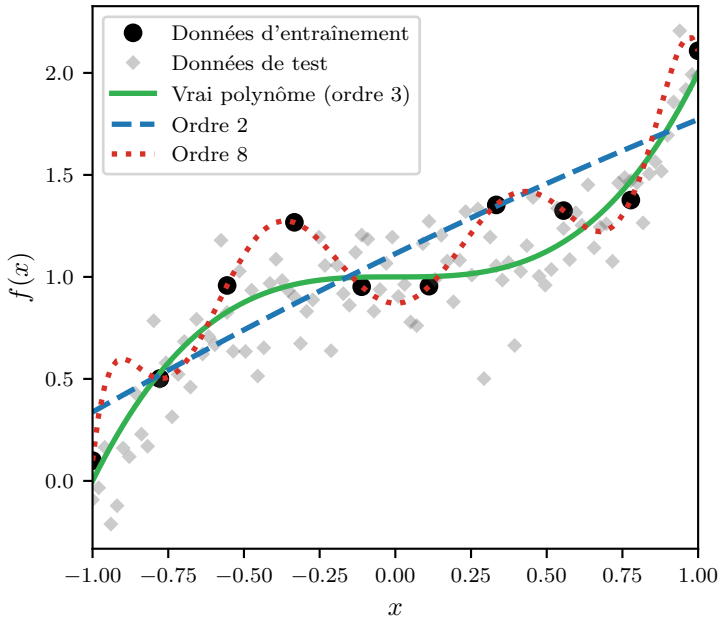


FIGURE 7.3 – Illustration des phénomènes de sous- et surapprentissage via l'interpolation polynômiale. Les symboles représentent les données d'entraînement (cercles noirs), et de test (carrés gris). Les courbes représentent des polynômes de différents ordres ajustés sur les données d'entraînement. Les données sont générés à partir d'un polynôme d'ordre 3, c'est-à-dire $f(x) = x^3 + 1$ (courbe verte), auxquels on a ajouté un bruit gaussien de variance 0.2. Les polynômes d'ordre 2 (courbe pointillée bleue) et d'ordre 8 (courbe pointillée rouge), ajustés sur les données d'entraînement, illustrent respectivement les phénomènes de sous- et surapprentissage. L'erreur quadratique moyenne (MSE, Eq. (7.3)) pour le polynôme d'ordre 2 est 5.57×10^{-2} sur les données d'entraînement et 7.40×10^{-2} sur les données test. Pour le polynôme d'ordre 8, l'erreur sur les données d'entraînement est 5.67×10^{-7} et, sur les données test, 8.42×10^{-2}.

$q(c|x) = [\hat{f}(x;\boldsymbol{\theta})]_c = \hat{y}_c$, alors

$$\begin{aligned}
\mathcal{D}_{KL}(p(\cdot|x)\|q(\cdot|x)) &= \mathcal{H}\Big(p(\cdot|x), q(\cdot|x)\Big) - \mathcal{H}\Big(p(\cdot|x)\Big)\,, \\
&= -\sum_{y=1}^{C} p(y|x)\log[\hat{f}(x;\boldsymbol{\theta})]_y + \sum_{y=1}^{C} p(y|x)\log p(y|x)\,. \\
&= \mathbb{E}_{Y|x}\left[-\sum_{c=1}^{C}\delta(c,Y)\log[\hat{f}(x;\boldsymbol{\theta})]_c\right] + \sum_{c=1}^{C} p(c|x)\log p(c|x)\,, \\
&= \mathbb{E}_{Y|x}\Big[\mathcal{L}(Y,\hat{f}(x;\boldsymbol{\theta}))\Big] + \mathcal{H}(p(\cdot|x))\,.
\end{aligned}$$

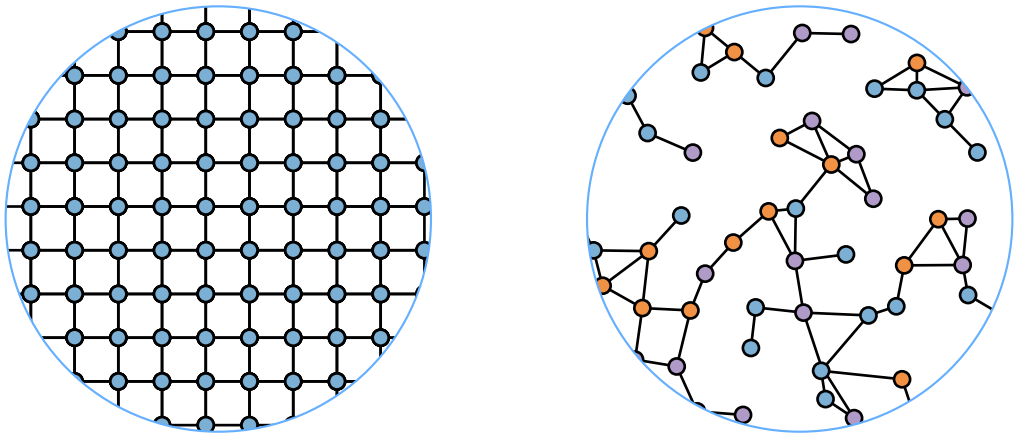
On obtient l'avant-dernière égalité en utilisant le fait que $p(c|x) = \sum_{y=1}^{C}\delta(c,y)p(y|x) = \mathbb{E}_{Y|x}[\delta(c,Y)]$. Ici, seul le premier terme—soit, l'espérance de l'entropie croisée—dépend des paramètres $\boldsymbol{\theta}$. Ainsi, l'entropie croisée et la divergence de Kullback-Leibler sont des objectifs d'entraînement équivalents, à une constante près.

Les modèles profonds supervisés sont souvent sujets au surapprentissage. Ce phénomène survient lorsqu'un modèle apprend à mémoriser les exemples du jeu de données d'entraînement. En outre, les modèles surappris ne parviennent pas à prédire correctement des exemples n'appartenant pas au jeu de données d'entraînement. Un exemple qui illustre bien le surapprentissage est celui de la régression polynomiale illustré à la Fig. 7.3. Bien que cet exemple ne décrit pas un problème de classification mais de régression, le surapprentissage est tout à fait analogue dans les deux cas. Supposons un ensemble de points générés par un polynôme d'ordre 3, c'est-à-dire $f(x) = x^3 + 1$, auxquels on ajoute un bruit gaussien. On essaie alors d'entraîner un modèle—dans ce cas un autre polynôme—pour reconstruire la fonction cible $f(x)$. Si l'ordre du polynôme est supérieur à 3, celui-ci a la capacité de reproduire la fonction cible, mais également de passer exactement par tous les points de l'ensemble d'entraînement, modélisant le bruit dans l'élan. Conséquemment, le modèle n'apprend pas la fonction cible correctement. Ceci est illustré à la Fig. 7.3 par le polynôme d'ordre 8, qui reproduit parfaitement les données d'entraînement, mais qui ne généralise pas bien sur les données de test. En effet, la performance en test du modèle d'ordre 8 est même inférieure à celle d'un modèle d'ordre 2, qui en principe n'a pas la capacité de reproduire $f(x)$. Le modèle d'ordre 8 est donc surappris.

Essentiellement, la capacité d'un modèle à surapprendre est une conséquence directe de sa capacité d'apprentissage reliée à son nombre de paramètres. On mesure le niveau de surapprentissage par la performance du modèle sur un jeu de données de validation, composé d'exemples qui n'ont pas été utilisés pour l'optimisation des paramètres. Si la performance en validation diminue alors que celle en entraînement augmente, alors on peut se douter que le modèle est en train de surapprendre. Il existe plusieurs techniques pour contrer le surapprentissage, notamment la *régularisation* qui consiste d'une manière ou d'une autre à restreindre l'espace des paramètres et la capacité du modèle. On réfère à la Réf. [119, Section 7.1] pour une revue plus détaillée des techniques de régularisation.

7.4 Biais inductif dans les réseaux profonds

Depuis les premières implémentations de réseaux de neurones artificiels, les architectures ont beaucoup évolué. À l'origine, le MLP était utilisé comme une architecture générique et applicable à virtuellement n'importe quel type de données. Cependant, la propension du MLP à surapprendre a mené vers un nouveau paradigme en apprentissage profond, celui des architectures spécialisées guidées par des biais inductifs. Les réseaux de neurones convolutifs (CNN, pour *convolutional neural network* en anglais), spécialisés dans le traitement d'images, incarnent parfaitement ce paradigme. Leur biais inductif présuppose une structure spatialement locale et régulière dans les données d'entrée [119, 175] [Fig. 7.4(a)], ce qui facilite la reconnaissance de motifs localisés sur les images. En retour, ce biais stabilise l'entraînement des CNNs et bonifie leur capacité à généraliser sur de nouvelles images.



(a) (b)

FIGURE 7.4 – Illustration de la structure des données à l'origine du biais inductif (a) des CNNs et (b) des GNNs. Alors que les CNNs se spécialisent dans le traitement de données structurées en grille régulière, les GNNs sont conçus pour traiter des données structurées selon des graphes arbitraires.

Les réseaux de neurones sur graphes (GNN, pour *graph neural network* en anglais) possèdent, comme les CNNs, un biais inductif qui facilite l'apprentissage de données structurées. Cependant, la structure des données dans le cas des GNNs est, contrairement aux images, non régulière et spécifiée par un graphe arbitraire, tel qu'illustrée à la Fig. 7.4(b). Il existe deux familles de GNNs, qu'on appelle *transductive* et *inductive* [126]. Dans la famille transductive, les modèles sont entraînés sur des graphes fixes et ne permettent pas d'étendre l'analyse à de nouveaux noeuds. Les modèles transductifs sont principalement utilisés dans le cadre d'analyse de graphes statiques, pour la classification ou le partitionnement non supervisé de noeuds (par exemple, la Réf. [248]). À l'inverse, les GNNs inductifs sont applicables sur des graphes arbitraires. Les modèles inductifs ont cette capacité de généralisation puisque, d'une part, ils prennent en argument la structure du graphe lui-même, et d'autre part, ils apprennent une représentation des noeuds basée sur leur voisinage. L'approche inductive a démontré sa supériorité par rapport à celle transductive dans plusieurs applications, dont en biologie [91], en chimie [109] et en recommandation [337].

7.5 Mécanisme d'agrégation du voisinage

L'architecture des GNNs inductifs est guidée par un mécanisme d'aggrégation de l'information du voisinage des noeuds. Dans ces dernières, plusieurs approches d'aggrégation ont été proposées, notamment des méthodes spectrales [48, 128, 156, 227, 281, 347] et géométriques [45, 205]. Aujourd'hui, les méthodes de pointe en apprentissage profond sur graphes favorisent des mécanismes géométriques de type *message-passing* [91, 109], dans lesquels l'information se propage de proche en proche à travers la structure du graphe. Spécifiquement,

considérons un graphe g composé de N noeuds, chaque noeud i étant associé à une représentation $x_i \in \mathbb{R}^d$ (à une couche donnée du modèle, laquelle est omise pour simplifier la discussion) qui le caractérise. La représentation d'un noeud i est mise à jour avec celle de ses voisins via une fonction d'aggrégation de la forme suivante :

$$x_i' = \gamma \left(x_i, \underset{j \in \mathcal{N}_i}{\square} \mu_{ij}(x_i, x_j) \right) , \tag{7.18}$$

où x_i' est la nouvelle représentation du noeud i, $\mathcal{N}_i$ est l'ensemble de ses voisins, γ est une fonction non linéaire, $\square$ est un opérateur d'aggrégation et μ_{ij} est le message envoyé de j vers i. Généralement, γ et μ_{ij} sont des fonctions entraînables (par exemple, des MLPs), et l'opérateur $\square$, qui est invariant sous permutation des voisins, est typiquement une somme pondérée ou une fonction maximum.

La capacité des GNNs à apprendre des représentations efficaces dépend du mécanisme d'aggrégation, spécifiquement l'opérateur $\square$. Autrement dit, tous les mécanismes ne sont pas équivalents en termes de leur expressivité. Xu et al. investiguent ce phénomène dans la Réf. [333], où ils démontrent que certaines architectures GNN ne permettent pas de distinguer des paires de graphes non isomorphes. Spécifiquement, ils utilisent l'heuristique de Weisfeiler-Lehman (WL) [326], un test de l'isomorphisme de deux graphes, comme critère de comparaison. Ils identifient une classe d'opérateur $\square$ dont l'expressivité est équivalente au critère WL. La somme pondérée fait partie de cette classe d'opérateurs, contrairement à la moyenne arithmétique et à la fonction maximum. Ces considérations théoriques seront importantes pour la suite de notre travail.

Chapitre 8

Apprentissage profond de dynamiques de contagion sur réseaux complexes

Article original :
Deep learning of contagion dynamics in complex networks
Charles Murphy, Edward Laurence, Antoine Allard
Département de Physique, de Génie Physique et d'Optique, Université Laval, Québec (Qc), Canada G1V 0A6
Référence : Nat. Commun. **12**, 4720 (2021) [210]

1. Ces sections contiennent le contenu original de l'article. Celui-ci n'a été modifié que pour se conformer au format exigé par la Faculté des études supérieures et postdoctorales de l'Université Laval.

8.1 Avant-propos

À l'époque où l'article présenté dans ce chapitre a été rédigé, les applications de l'apprentissage profond étaient plutôt rares en physique, et commençaient à émerger en science des réseaux. L'objectif initial de ce projet était ainsi profondément exploratoire : comment utiliser de telles approches pour apprendre nos modèles de réseau à partir de données empiriques ? La pandémie de COVID-19 était également en cours au moment de l'idéation du projet. L'idée était donc de combiner les deux : développer des approches génératives pour modéliser les dynamiques de contagion sur réseaux à partir de données temporelles.

Dans ce chapitre, nous proposons une architecture générative basée sur les GNNs, qui s'apparentent aux *transformers* utilisés abondamment en traitement du langage naturel [312]. Notre architecture est précisément conçue pour apprendre les mécanismes locaux d'évolution d'une dynamique sur réseaux générale, ce qui diffère de l'utilisation typique qu'on fait des GNNs—i.e., les plongements de noeuds. Nous démontrons la capacité de notre architecture à apprendre des processus de contagion simples, complexes, des processus en co-évolution, des processus de réaction-diffusion et même des processus de contagion réels de la COVID-19 en Espagne. Néanmoins, la puissance de notre approche réside dans notre capacité à extraire des mécanismes locaux effectifs. Après l'entraînement, notre modèle peut alors être utilisé comme un laboratoire numérique pour simuler des scénarios de contagion, explorer les propriétés critiques du processus inconnu et prédire son évolution.

Symbole	Description
$\mathcal{V}$	Ensemble des noeuds
$\mathcal{E}$	Ensemble des liens
$\mathbf{\Phi}$	Métadonnées des noeuds
$\mathbf{\Omega}$	Métadonnées des liens
G	Graphe et ses métadonnées, i.e. $(\mathcal{V}, \mathcal{E}, \mathbf{\Phi}, \mathbf{\Omega})$
D	Jeu de données d'entraînement $D = (x, y)$
$\boldsymbol{x}$	Tuple des valeurs d'entrée des noeuds $x = (x_1, \cdots, x_T)$
$x_i(t)$	Valeur d'entrée du noeud i au temps t
$\boldsymbol{y}$	Tuple des valeur de sortie des noeuds $\boldsymbol{y} = (y_1, \cdots, y_T)$
$y_i(t)$	Valeur de sortie du noeud i au temps t
$\mathcal{S}$	Ensemble des états possibles des noeuds

$\mathcal{R}$	Ensemble des sorties possibles des noeuds
$\mathcal{T}$	Ensemble des indices temporels
F	Fonction d'évolution de la dynamique complète
f	Fonction d'évolution de la dynamique locale
$\alpha(\ell)$	Fonction d'infection
$\hat{F}$	Fonction apprise d'évolution de la dynamique complète
$\hat{f}$	Fonction apprise d'évolution de la dynamique locale
$\boldsymbol{\Theta}$	Paramètres du modèle
$\mathcal{L}$	Fonction objective complète
$L(x, y)$	Fonction objective locale
ρ	Distribution des exemples d'entraînement
$w_i(t)$	Poids du noeud i au temps t dans la fonction objective
Z'	Constante de normalisation des poids $w_i(t)$
λ	Coefficient de rebalancement de la fonction objective

TABLEAU 8.1 – Glossaire des symboles utilisés au Chapitre 8

8.2 Résumé

Prédire l'évolution des processus de contagion est encore un problème ouvert dans lequel les modèles mécaniques peinent à offrir une réponse. En effet, afin de rester mathématiquement ou computationnellement traitables, ces modèles doivent s'appuyer sur des hypothèses simplificatrices, limitant ainsi la précision quantitative de leurs prédictions et la complexité des dynamiques qu'ils peuvent modéliser. Dans cet article, nous proposons une approche complémentaire basée sur l'apprentissage profond, où les mécanismes locaux effectifs régissant une dynamique sur un réseau sont appris à partir de données de séries temporelles. Notre architecture de réseau de neurones sur graphe ne fait que très peu d'hypothèses sur la dynamique, et nous démontrons sa capacité en utilisant différentes dynamiques de contagion de complexité croissante. Capable de générer des simulations sur des structures de réseau arbitraires, notre architecture permet d'explorer les propriétés de la dynamique apprise au-delà des données d'entraînement. Enfin, nous illustrons l'applicabilité de notre approche en utilisant des données réelles de l'épidémie de COVID-19 en Espagne. Nos résultats

montrent comment l'apprentissage profond offre une nouvelle perspective complémentaire pour construire des modèles efficaces de dynamiques de contagion sur des réseaux.

8.3 Abstract

Forecasting the evolution of contagion dynamics is still an open problem to which mechanistic models only offer a partial answer. To remain mathematically or computationally tractable, these models must rely on simplifying assumptions, thereby limiting the quantitative accuracy of their predictions and the complexity of the dynamics they can model. Here, we propose a complementary approach based on deep learning where the effective local mechanisms governing a dynamic on a network are learned from time series data. Our graph neural network architecture makes very few assumptions about the dynamics, and we demonstrate its accuracy using different contagion dynamics of increasing complexity. By allowing simulations on arbitrary network structures, our approach makes it possible to explore the properties of the learned dynamics beyond the training data. Finally, we illustrate the applicability of our approach using real data of the COVID-19 outbreak in Spain. Our results demonstrate how deep learning offers a new and complementary perspective to build effective models of contagion dynamics on networks.

8.4 Introduction

Our capacity to prevent or contain outbreaks of infectious diseases is directly linked to our ability to accurately model contagion dynamics. Since the seminal work of Kermack and McKendrick almost a century ago [152], a variety of models incorporating ever more sophisticated contagion mechanisms has been proposed [42, 134, 157, 282]. These mechanistic models have provided invaluable insights about how infectious diseases spread, and have thereby contributed to the design of better public health policies. However, several challenges remain unresolved, which call for contributions from new modeling approaches [120, 233, 317].

For instance, many complex contagion processes involve the nontrivial interaction of several pathogens [132, 206, 226, 271], and some social contagion phenomena, like the spread of misinformation, require to go beyond pairwise interactions between individuals [54, 139, 179]. Also, while qualitatively informative, the forecasts of most mechanistic models lack quantitative accuracy [32]. Indeed, most models are constructed from a handful of mechanisms which can hardly reproduce the intricacies of real complex contagion dynamics. One approach to these challenges is to complexify the models by adding more detailed and sophisticated mechanisms. However, mechanistic models become rapidly intractable as new mechanisms are added. Moreover, models with higher complexity require the specification of a large number of parameters whose values can be difficult to infer from limited data.

There has been a recent gain of interest towards using machine learning to address the issue of the often-limiting complexity of mechanistic models [49, 55, 74, 132, 166, 189, 234, 235]. This new kind of approach aims at training predictive models directly from observational time series data. These data-driven models are then used for various tasks such as making accurate predictions [234, 235], gaining useful intuitions about complex phenomena [132] and discovering new patterns from which better mechanisms can be designed [49, 166]. Although these approaches were originally designed for regularly structured data, this new paradigm is now being applied to epidemics spreading on networked systems [82, 278], and more generally to dynamical systems [172, 255, 269]. Meanwhile, the machine learning community has dedicated a considerable amount of attention on deep learning on networks, structure learning and graph neural networks (GNN) [333, 344, 348]. Recent works showed great promise for GNN in the context of community detection [248], link prediction [127], network inference [345], as well as for the discovery of new materials and drugs [96, 349]. Yet, others have pointed out the inherent limitations of a majority of GNN architectures in distinguishing certain network structures [333], in turn limiting their learning abilities. Hence, while recent advances and results [98, 100, 147, 285] suggest that GNNs could be prime candidates for building effective data-driven dynamical models on networks, it remains to be shown if, how and when GNNs can be applied to dynamics learning problems.

In this paper, we show how GNN, usually used for structure learning, can also be used to model contagion dynamics on complex networks. Our contribution is threefold. First, we design a training procedure and an appropriate GNN architecture capable of representing a wide range of dynamics, with very few assumptions. Second, we demonstrate the validity of our approach using various contagion dynamics on networks of different natures and of increasing complexity, as well as on real epidemiological data. Finally, we show how our approach can provide predictions for previously unseen network structures, therefore allowing the exploration of the properties of the learned dynamics beyond the training data. Our work generalizes the idea of constructing dynamical models from regularly structured data to arbitrary network structures, and suggests that our approach could be accurately extended to many other classes of dynamical processes.

8.5 Results

In our approach, we assume that an unknown dynamical process, denoted F, takes place on a known network structure—or ensemble of networks—, denoted $g = (\mathcal{V}, \mathcal{E}, \mathbf{\Phi}, \mathbf{\Omega})$, where $\mathcal{V} = \{v_1, \cdots, v_N\}$ is the node set and $\mathcal{E} = \left\{e_{ij} : j \text{ is connected to } i \wedge (i,j) \in \mathcal{V}^2\right\}$ is the edge set. We also assume that the network(s) can have some metadata, taking the form of node and edge attributes denoted $\Phi_i = (\phi_1(i), \cdots, \phi_Q(i))$ for node i and $\Omega_{ij} = (\omega_1(e_{ij}), \cdots, \omega_P(e_{ij}))$ for edge e_{ij}, respectively, where $\phi_q : \mathcal{V} \to \mathbb{R}$ and $\omega_p : \mathcal{E} \to \mathbb{R}$. These metadata can take various forms like node characteristics or edge weights. We also denote the node and edge

attribute matrices $\mathbf{\Phi} = (\Phi_i)_{i\in\mathcal{V}}$ and $\mathbf{\Omega} = (\Omega_{ij})_{(i,j)\in\mathcal{E}}$, respectively.

Next, we assume that the dynamics F has generated a time series D on the network g. This time series takes the form of a pair of consecutive snapshots $D = (x, y)$ with $x = (x_1, \cdots, x_T)$ and $y = (y_1, \cdots, y_T)$, where $x_t \in \mathcal{S}^{|\mathcal{V}|}$ is the state of the nodes at time t, $y_t \in \mathcal{R}^{|\mathcal{V}|}$ is the outcome of F defined as

$$y_t = F(x_t, g)\,, \tag{8.1}$$

$\mathcal{S}$ is the set of possible node states, and $\mathcal{R}$ is the set of possible node outcomes. This way of defining D allows us to formally concatenate multiple realizations of the dynamics in a single dataset. Additionally, the elements $x_i(t) \equiv [x_t]_i$ and $y_i(t) \equiv [y_t]_i$ correspond to the state of node i at time t and its outcome, respectively. Typically, we consider that the outcome $y_i(t)$ is simply the state of node i after transitioning from state $x_i(t)$. In this case, we have $\mathcal{S} = \mathcal{R}$ and $x_i(t + \Delta t) = y_i(t)$ where Δt is the length of the time steps. However, if $\mathcal{S}$ is a discrete set—i.e. finite and countable—$y_i(t)$ is a transition probability vector conditioned on $x_i(t)$ from which the following state, $x_i(t + \Delta t)$, will be sampled. The element $[y_i(t)]_m$ corresponds to the probability that node i evolves to state $m \in \mathcal{S}$ given that it was previously in state $x_i(t)$—i.e. $\mathcal{R} = [0, 1]^{|\mathcal{S}|}$. When F is a stochastic dynamics, we do not typically have access to the transition probabilities $y_i(t)$ directly, but rather to the observed outcome state—e.g. $x_i(t + \Delta t)$ in the event where $\boldsymbol{X}$ is temporally ordered—, we therefore define the observed outcome $\tilde{y}_i(t)$ as

$$[\tilde{y}_i(t)]_m = \delta\big(x_i(t + \Delta t), m\big)\,,\ \forall m \in \mathcal{S} \tag{8.2}$$

where $\delta(x, y)$ is the Kronecker delta. Finally, we assume that F acts on X_t locally and identically at all times, according to the structure of G. In other words, knowing the state x_i as well as the states of all the neighbors of i, the outcome y_i is computed using a time independent function f identical for all nodes

$$y_i \equiv f\big(x_i, \Phi_i, x_{\mathcal{N}_i}, \Phi_{\mathcal{N}_i}, \Omega_{i\mathcal{N}_i}\big)\,, \tag{8.3}$$

where $x_{\mathcal{N}_i} = (x_j)_{j\in\mathcal{N}_i}$ denotes the states of the neighbors, $\mathcal{N}_i \equiv \big\{j : e_{ij} \in \mathcal{E}\big\}$ is the set of the neighbors, $\Phi_{\mathcal{N}_i} \equiv \big\{\Phi_j : v_j \in \mathcal{N}_i\big\}$ and $\Omega_{i\mathcal{N}_i} \equiv \big\{\Omega_{ij} : v_j \in \mathcal{N}_i\big\}$. As a result, we impose a notion of locality where the underlying dynamics is time invariant and invariant under the permutation of the node labels in G, under the assumption that the node and edge attributes are left invariant.

Our objective is to build a model $\hat{F}$, parametrized by a GNN with a set of tunable parameters $\mathbf{\Theta}$ and trained on the observed dataset D to mimic F given G, such that

$$\hat{F}(x'_t, g'; \mathbf{\Theta}) \approx F(x'_t, g')\,, \tag{8.4}$$

for all states x'_t and all networks g'. The architecture of $\hat{F}$, detailed in Sec. 8.7.1, is designed to act locally similarly to F. In this case, the locality is imposed by a modified attention mechanism inspired by Ref. [312]. The advantage of imposing locality allows our architecture

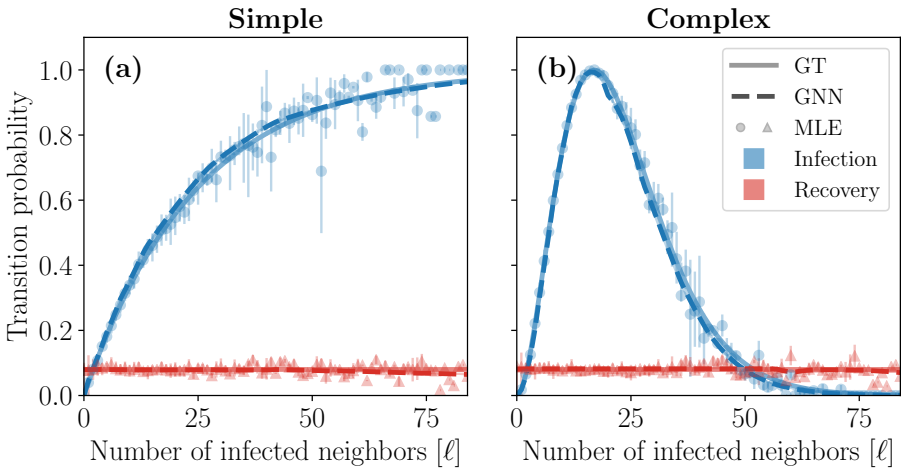


FIGURE 8.1 – **Predictions of GNN trained on a Barabási-Albert random network [19] for the (a) simple and (a) complex contagion dynamics.** The solid and dashed lines correspond to the transition probabilities of the dynamics used to generate the training data (labeled GT for "ground truth"), and predicted by the GNN, respectively. Symbols correspond to the maximum likelihood estimation (MLE) of the transition probabilities computed from the dataset D. The colors indicate the type of transition : infection ($S \to I$) in blue and recovery ($S \to I$) in red. The standard deviations, as a result of averaging the outcomes given ℓ, are shown using a colored area around the lines (typically narrower than the width of the lines) and using vertical bars for the symbols.

to be *inductive* : If the GNN is trained on a wide range of local structures—i.e. nodes with different neighborhood sizes (or degrees) and states—it can then be used on any other networks within that range. This suggests that the topology of g will have a strong impact on the quality of the trained models, an intuition that is confirmed below. Similarly to Eq. (8.3), we can write each individual node outcome computed by the GNN using a function $\hat{f}$ such that

$$\hat{y}_i \equiv \hat{f}\left(x_i, \Phi_i, x_{\mathcal{N}_i}, \Phi_{\mathcal{N}_i}, \Omega_{i\mathcal{N}_i}; \boldsymbol{\Theta}\right) \tag{8.5}$$

where $\hat{y}_i$ is the outcome of node i predicted by $\hat{F}$.

The objective described by Eq. (8.4) must be encoded into a global loss function, denoted $\mathcal{L}(\boldsymbol{\Theta})$. Like the outcome functions, $\mathcal{L}(\boldsymbol{\Theta})$ can be decomposed locally, where the local losses of each node $L(y_i, \hat{y}_i)$ are arithmetically averaged over all possible node inputs $(x_i, \Phi_i, x_{\mathcal{N}_i}, \Phi_{\mathcal{N}_i}, \Omega_{i\mathcal{N}_i})$, where y_i and $\hat{y}_i$ are given by Eqs. (8.3) and (8.5), respectively. By using an arithmetic mean to the evaluation of $\mathcal{L}(\boldsymbol{\Theta})$, we assume that the node inputs are distributed uniformly. Consequently, the model should be trained equally well on all of them. This is important because in practice we only have access to a finite number of inputs in D and g, for which the node input distribution is typically far from being uniform. Hence, in order to train effective models, we recalibrate the inputs using the following global loss

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{t \in \mathcal{T}'} \sum_{i \in \mathcal{V}'(t)} \frac{w_i(t)}{Z'} L\big(y_i(t), \hat{y}_i(t)\big) \tag{8.6}$$

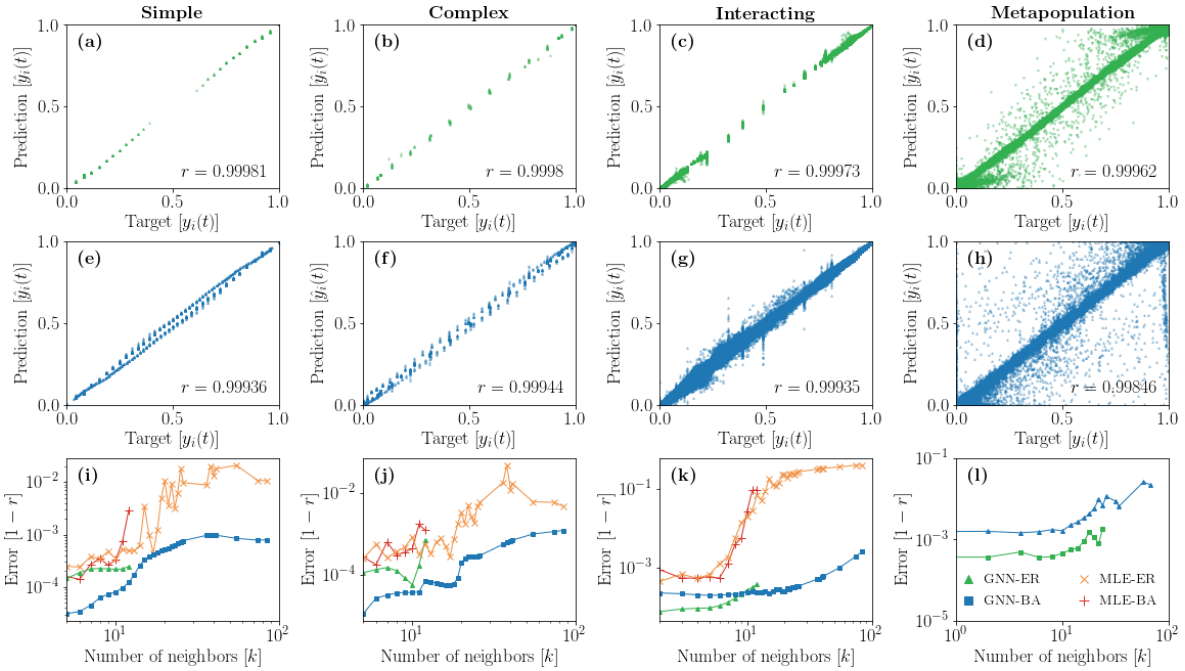


FIGURE 8.2 – **Comparison between the targets and the predictions of GNN trained on Erdős-Rényi networks (ER, top row) and on Barabási-Albert networks [19] (BA, middle row) for the (a, e, i) simple, (b, f, j) complex, (c, g, k) interacting and (d, h, l) metapopulation dynamics.** Each point shown on the panels (a–h) corresponds to a different pair $(y_i(t), \hat{y}_i(t))$ in the complete dataset D. We also indicate the Pearson coefficient r on each panel to measure the correlation between the predictions and the targets and use it as a global performance measure. The panels (i–l) show the errors $(1 - r)$ as a function of the number of neighbors for GNN trained on ER and BA networks, and those of the corresponding MLE. These errors are obtained from the Pearson coefficients computed from subsets of the prediction-target pairs where all nodes have degree k.

where $w_i(t)$ is a weight assigned to node i at time t, and $Z' = \sum_{t\in\mathcal{T}'}\sum_{i\in\mathcal{V}'(t)} w_i(t)$ is a normalization factor. Here, the training node set $\mathcal{V}'(t) \subseteq \mathcal{V}$ and the training time set $\mathcal{T}' \subseteq [T]$ allow us to partition the training dataset for validation and testing when required.

The choice of weights needs to reflect the importance of each node at each time. Because we wish to lower the influence of overrepresented inputs and increase that of rare inputs, a sound choice of weights is

$$w_i(t) \propto \rho\Big(k_i, x_i, \Phi_i, x_{\mathcal{N}_i}, \Phi_{\mathcal{N}_i}, \Omega_{i\mathcal{N}_i}\Big)^{-\lambda} \tag{8.7}$$

where k_i is the degree of node i in g, and $0 \leq \lambda \leq 1$ is an hyperparameter. Equation (8.7) is an ideal choice, because it corresponds to a principled importance sampling approximation of Eq. (8.6) [262], which is relaxed via the exponent λ. We obtain a pure importance sampling scheme when $\lambda = 1$. Note that the weights can rarely be exactly computed using Eq. (8.7), because the distribution ρ is typically computationally intensive to obtain from data, espe-

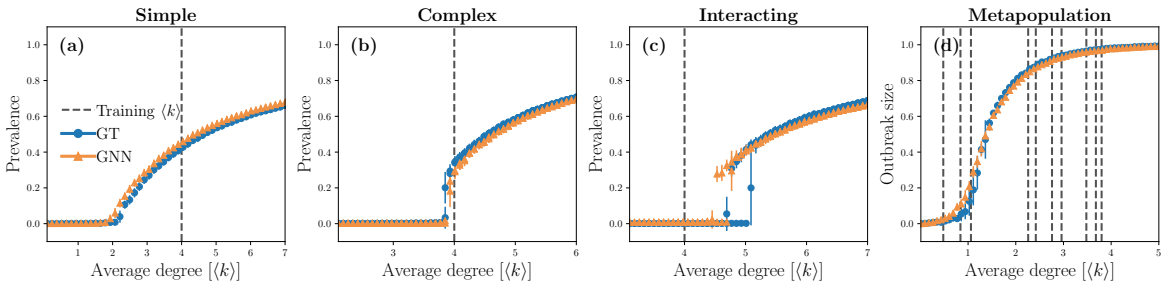


FIGURE 8.3 – **Bifurcation diagrams of the (a) simple, (b) complex, (c) interacting and (d) metapopulation dynamics on Poisson networks [19] composed of $|\mathcal{V}| = 2000$ nodes with different average degrees $\langle k \rangle$.** The prevalence is defined as the average fraction of nodes that are asymptotically infected by at least one disease and the outbreak size corresponds to the average fraction of nodes that have recovered. These quantities are obtained from numerical simulations using the "ground truth" (GT) dynamics (blue circles) and the GNN trained on Barabási-Albert networks (orange triangles). The error bars correspond to the standard deviations of these numerical simulations. The trained GNN used are the same ones as those used for Fig. 8.2. As a reference, we also indicate with dashed lines the value(s) of average degree $\langle k \rangle$ corresponding to the network(s) on which the GNN were trained. On panel (d), more than one value of $\langle k \rangle$ appear as multiple networks with different average degrees were used to train the GNN.

cially for continuous $\mathcal{S}$ with metadata. We illustrate various ways to evaluate the weights in Sec. 8.7.2 and in Supplementary Information.

We now illustrate the accuracy of our approach by applying it to four types of synthetic dynamics of various natures (see Sec. 8.7.3 for details on the dynamics). We first consider a *simple contagion* dynamics : The discrete-time susceptible-infected-susceptible (SIS) dynamics. In this dynamics, nodes are either susceptible (S) or infected (I) by some disease, i.e. $\mathcal{S} = \{S, I\} = \{0, 1\}$, and transition between each state stochastically according to an infection probability function $\alpha(\ell)$, where ℓ is the number of infected neighbors of a node, and a constant recovery probability β. A notable feature of simple contagion dynamics is that susceptible nodes get infected by the disease through their infected neighbors independently. This reflects the assumption that disease transmission behaves identically whether a person has a large number of infected neighbors or not.

Second, we relax this assumption by considering a *complex contagion* dynamics with a non-monotonic infection function $\alpha(\ell)$ where the aforementioned transmission events are no longer independent [179]. This contagion dynamics has an interesting interpretation in the context of the propagation of a social behavior, where the local popularity of a behavior (large ℓ) hinders its adoption. The independent transmission assumption can also be lifted when multiple diseases are interacting [271]. Thus, we also consider an asymmetric *interacting contagion* dynamics with two diseases. In this case, $\mathcal{S} = \{S_1S_2, I_1S_2, S_1I_2, I_1I_2\} = \{0, 1, 2, 3\}$ where U_1V_2 corresponds to a state where a node is in state U with respect to the

first disease and in state V with respect to the second disease. The interaction between the diseases happens via a coupling that is active only when a node is infected by at least one disease, otherwise it behaves identically to the simple contagion dynamics. This coupling may increase or decrease the virulence of the other disease.

Whereas the previously presented dynamics capture various features of contagion phenomena, real datasets containing this level of detail about the interactions among individuals are rare [104, 139, 193]. A class of dynamics for which dataset are easier to find is that of mass-action *metapopulation* dynamics [2, 16, 60, 288], where the status of the individuals are gathered by geographical regions. These dynamics typically evolve on the weighted networks of the individuals' mobility between regions and the state of a region consists in the number of people that are in each individual health state. As a fourth case study, we consider a type of deterministic metapopulation dynamics where the population size is constant and where people can either be susceptible (S), infected (I) or recovered from the disease (R). As a result, we define the state of the node as three-dimensional vectors specifying the fraction of people in each state—i.e. $\mathcal{S} = \mathcal{R} = [0, 1]^3$.

Figure 8.1 shows the GNN predictions for the infection and recovery probabilities of the simple and complex contagion dynamics as a function of the number of infected neighbors ℓ. We then compare them with their ground truths, i.e. Eq. (8.20) using Eqs. (8.18)–(8.22) for the infection functions. We also show the maximum likelihood estimators (MLE) of the transition probabilities computed from the fraction of nodes in state x and with ℓ infected neighbors that transitioned to state y in the complete dataset D. The MLE, which are typically used in this kind of inference problem [84], stands as a reference to benchmark the performance of our approach.

We find that the GNN learns remarkably well the transition probabilities of the simple and complex contagion dynamics. In fact, the predictions of the GNN seem to be systematically smoother than the ones provided by the MLE. This is because the MLE is computed for each individual pair (x, ℓ) from disjoint subsets of the training dataset. This implies that a large number of samples of each pair (x, ℓ) is needed for the MLE to be accurate; a condition rarely met in realistic settings, especially for high degree nodes. This also means that the MLE cannot be used directly to interpolate beyond the pairs (x, ℓ) present in the training dataset, in sharp contrast with the GNN which, by definition, can interpolate within the dataset D. Furthermore, all of its parameters are hierarchically involved during training, meaning that the GNN benefits from any sample to improve all of its predictions, which are then smoother and more consistent. Further still, we found that not all GNN architectures can reproduce the level of accuracy obtained in the Fig. 8.1 (see Sec. III F of the Supplementary Information). In fact, we showed that many standard GNN aggregation mechanisms are ineffective at learning the simple and complex contagion dynamics, most likely because they were specifically designed with structure learning in mind rather than dynamics learning.

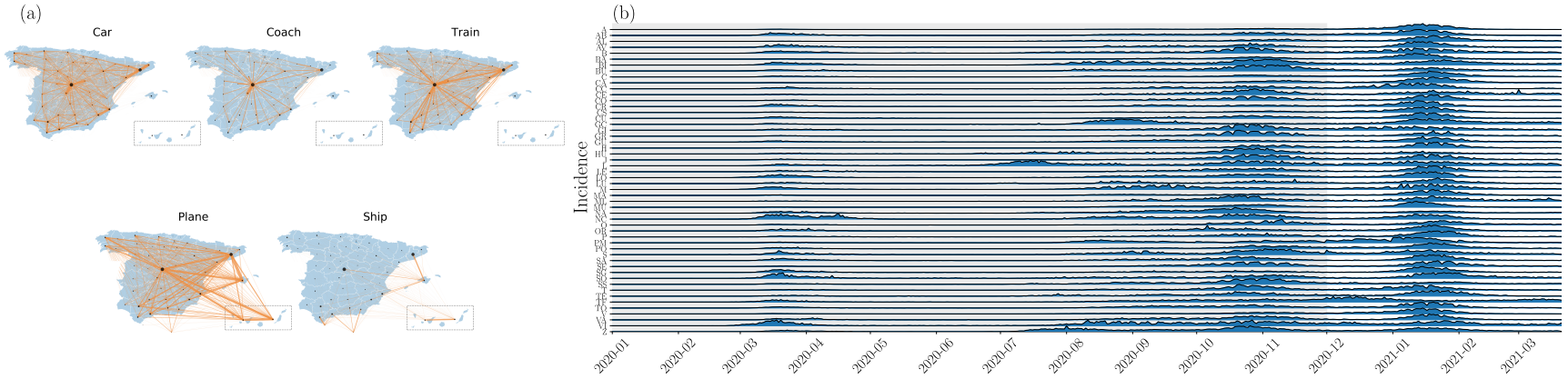


FIGURE 8.4 – **Spain COVID-19 dataset.** (a) Spain mobility multiplex network [116]. The thickness of the edges is proportional to the average number of people transitioning between all connected node pairs. The size of the nodes is proportional to the population N_i living in the province. (b) Time series of the incidence for the 52 provinces of Spain between January 2020 and March 2021 [117]. Each province is identified by its corresponding ISO code. Each incidence time series has been rescaled by its maximum value for the purpose of visualization. The shaded area indicates the training and validation datasets (in-sample) from January 1st 2020 to December 1st 2021. The remaining of the dataset is used for testing.

It is worth mentioning that the GNN is not specifically designed nor trained to compute the transition probabilities as a function of a single variable, namely the number of infected ℓ. In reality, the GNN computes its outcome from the complete multivariate state of the neighbors of a node. The interacting contagion and the metapopulation dynamics, unlike the simple and complex contagions, are examples of such multivariate cases. Their outcome is thus harder to visualize in a representation similar to Fig. 8.1. Figures 8.2(a–h) address this issue by comparing each of the GNN predictions $\hat{y}_i(t)$ with its corresponding target $y_i(t)$ in the dataset D. We quantify the global performance of the models in different scenarios, for the different dynamics and underlying network structures, using the Pearson correlation coefficient r between the predictions and targets (see Sec. 8.7.1). We also compute the error, defined from the Pearson coefficient as $1 - r$ for each degree class k (i.e. between the predictions and targets of only the nodes of degree k). This allows us to quantify the GNN performance for every local structure.

Figures 8.2(i–k) confirm that the GNN provides more accurate predictions than the MLE in general and across all degrees. This is especially true in the case of the interacting contagion, where the accuracy of the MLE seems to deteriorate rapidly for large degree nodes. This is a consequence of how scarce the inputs are for this dynamics compared to both the simple and complex contagion dynamics for training datasets of the same size, and of how fast the size of the set of possible inputs scales, thereby quickly rendering MLE completely ineffective for small training datasets. The GNN, on the other hand, is less affected by the scarcity of the data, since any sample improves its global performance, as discussed above.

Figure 8.2 also exposes the crucial role of the network G on which the dynamics evolves in the global performance of the GNN. Namely, the heterogeneous degree distributions of Barabási-Albert networks (BA)—or any heterogeneous degree distribution—offer a wider range of degrees than those of homogeneous Erdős-Rényi networks (ER). We can take advantage of this heterogeneity to train GNN models that generalize well across a larger range of local structures, as seen in Fig. 8.2(i–l) (see also Supplementary Information). However, the predictions on BA networks are not systematically always better for low degrees than those on ER networks, as seen in the interacting and metapopulation cases. This nonetheless suggests a wide applicability of our approach for real complex systems, whose underlying network structures recurrently exhibit a heterogeneous degree distribution [319].

We now test the trained GNN on unseen network structures by recovering the bifurcation diagrams of the four dynamics. In the infinite-size limit $|\mathcal{V}| \to \infty$, these dynamics have two possible long-term outcomes : the absorbing state where the diseases quickly die out, and the endemic/epidemic state in which a macroscopic fraction of nodes remains (endemic) or has been infected over time (epidemic) [157, 231, 271]. These possible long-term outcomes exchange stability during a phase transition which is continuous for the simple contagion and metapopulation dynamics, and discontinuous for the complex and interacting contagion

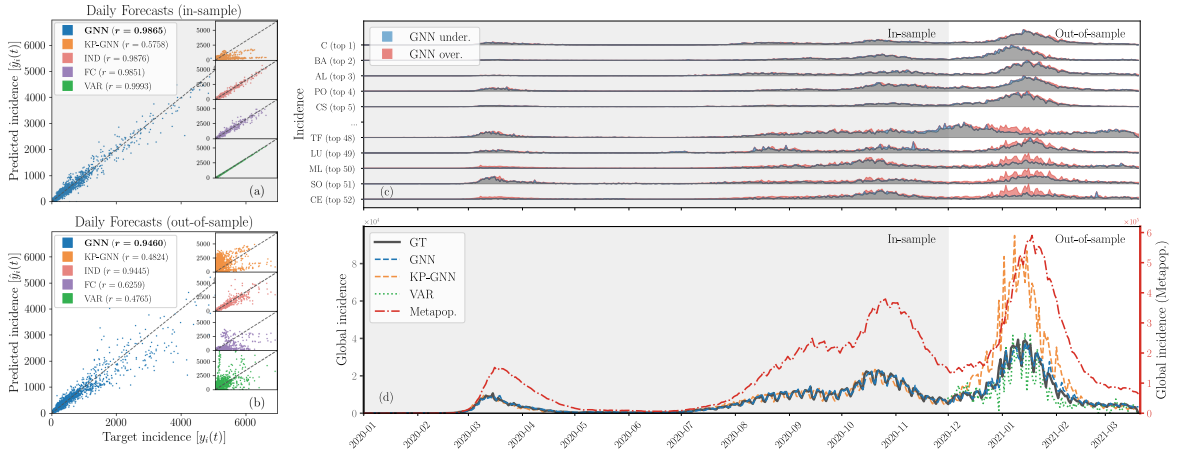


FIGURE 8.5 – **Learning the Spain COVID-19 dataset.** (a-b) Comparison between the targets and the predictions in the in-sample and the out-of-sample datasets for our GNN model (blue) and for other models (KP-CNN in orange, IND in pink, FC in purple and VAR in green; see main text). The accuracy of the predictions is quantified by the Pearson correlation coefficient provided in the legend. (c) Forecasts by our GNN model for individual time series of the provincial daily incidence compared with the ground truth. Underestimation and overestimation are respectively shown in blue and red. Each time series has been rescaled as in Fig. 8.4(b) and are ordered according to mean square error of the GNN's predictions. (d) Forecasts for the global incidence (sum of the daily incidence in every province). The solid grey line indicates the ground truth (GT); the dashed blue line, the dashed orange line and dotted green line show the forecast of our GNN model, of KP-CNN and of VAR, respectively. We also show the forecast of an equivalent metapopulation model (red dash-dotted line) which has its own scale (red axis on the right) to improve the visualization; the other lines share the same axis on the left. Similarly to Fig. 8.4, we differentiate the in-sample from the out-of-sample forecasts using a shaded background.

dynamics. The position of the phase transition depends on the parameters of the dynamics as well as on the topology of the network. Note that for the interacting contagion dynamics, the stability of absorbing and endemic states do not change at the same point, giving rise to a bistable regime where both states are stable.

Figure 8.3 shows the different bifurcation diagrams obtained by performing numerical simulations with the trained GNN models [using Eq. (8.5)] while varying the average degree of networks, on which the GNN has not been trained. Quantitatively, the predictions are again strikingly accurate—essentially perfect for the simple and complex contagion dynamics—which is remarkable given that the bifurcation diagrams were obtained on networks the GNN had never seen before. These results illustrate how insights can be gained about the underlying process concerning the existence of phase transitions and their order, among other things. They also suggest how the GNN can be used for diverse applications, such as predicting the dynamics under various network structures (e.g. designing intervention strategies that affect the way individuals interact and are therefore connected).

Finally, we illustrate the applicability of our approach by training our GNN model using the evolution of COVID-19 in Spain between January 1st 2020 and March 27th 2021 (see Fig. 8.4). This dataset consists in the daily number of new cases (i.e. incidence) for each of the 50 provinces of Spain as well as Ceuta and Melilla [117]. We also use a network of the mobility flow recorded in 2018 [116] as a proxy to model the interaction network between these 52 regions. This network is multiplex—each layer corresponds to a different mode of transportation—, directed and weighted (average daily mobility flow).

We compare the performance of our approach with that of different baselines : Four data-driven techniques—three competing neural network architectures [147, 308] and a linear vector autoregressive model (VAR) [266, 325]—, and an equivalent mechanistic metapopulation model (Metapop.) driven by a simple contagion mechanism [60]. Among the three neural network architectures, we used the model of Ref. [147] (KP-GNN) that has been used to predict the evolution of COVID-19 in the US. In a way of an ablation study, the other two GNN architectures embody the assumptions that the nodes of the networks are mutually independent (IND), or that the nodes are arbitrarily codependent (FC) in a way that is learned by the neural network. Note that there exists a wide variety of GNN architectures designed to handle dynamics *of* networks [285]—networks whose topology evolves over time—but that these architectures are not typically adapted for learning dynamics *on* networks (see Sec. III F of the Supplementary Information). Finally, we used the parameters of Ref. [4] for the metapopulation model. Section 8.7.4 provides more details on the baselines.

Figure 8.5 shows that all data-driven models can generate highly accurate in-sample predictions, with the exception of the KP-GNN model which appears to have a hard time learning the dynamics, possibly because of its aggregation mechanism (see Sec. III F of the Supplementary Information). This further substantiates the idea that many GNN architectures designed for structure learning, like the graph convolutional network [156] at the core the KP-GNN model, are suboptimal for dynamics learning problems. However, the other architectures do not appear to have the same capability to generalize the dynamics out-of-sample : The FC and the VAR models, especially, tend to overfit more than the GNN and the IND models. While this was expected for the linear VAR model, the FC model overfits because it is granted too much freedom in the way it learns how the nodes interact with one another. Interestingly, the IND and the GNN models seem to perform similarly, which hints to the possibility that the specifics of the mobility network might not have contributed significantly to the dynamics. This is perhaps not surprising since social distancing and confinement measures were in place during the period covered by the dataset. Indeed, our results indicate that the global effective dynamics was mostly driven by internal processes within each individual province, rather than driven by the interaction between them. This last observation suggests that our GNN model is robust to spurious connections in the interaction network.

Finally, Fig. 8.5(d) shows that the metapopulation model is systematically overestimating

the incidence by over an order of magnitude. Again, this is likely due to the confinement measures in place during that period which were not reflected in the original parameters of the dynamics [4]. Additional mechanisms accounting for this interplay between social restrictions and the prevalence of the disease—e.g. complex contagion mechanisms [132] or time-dependent parameters [320]—would therefore be in order to extend the validity of the metapopulation model to the full length of the dataset. Interestingly, a signature of this interplay is encoded in the daily incidence data and our GNN model appears to be able to capture it to some extent.

8.6 Discussion

We introduced a data-driven approach that learns effective mechanisms governing the propagation of diverse dynamics on complex networks. We proposed a reliable training protocol, and we validated the projections of our GNN architecture on simple, complex, interacting contagion and metapopulation dynamics using synthetic networks. Interestingly, we found that many standard GNN architectures do not handle correctly the problem of learning contagion dynamics from time series. Also, we found that our approach performs better when trained on data whose underlying network structure is heterogeneous, which could prove useful in real-world applications of our method given the ubiquitousness of scale-free networks [319].

By recovering the bifurcation diagram of various dynamics, we illustrated how our approach can leverage time series from an unknown dynamical process to gain insights about its properties—e.g. the existence of a phase transition and its order. We have also shown how to use this framework on real datasets, which in turn could then be used to help build better effective models. In a way, we see this approach as the equivalent of a numerical Petri dish—offering a new way to experiment and gain insights about an unknown dynamics—that is complementary to traditional mechanistic modeling to design better intervention procedures, containment countermeasures and to perform model selection.

Although we focused the presentation of our method on contagion dynamics, its potential applicability reaches many other realms of complex systems modeling where intricate mechanisms are at play. We believe this work establishes solid foundations for the use of deep learning in the design of realistic effective models of complex systems.

Gathering detailed epidemiological datasets is a complex and labor-intensive process, meaning that datasets suitable for our approach are currently the exception rather than the norm. The current COVID-19 pandemic has, however, shown how an adequate international reaction to an emerging infectious pathogen critically depends on the free flow of information. New initiatives like Golbal.health [115] are good examples on how the international epidemiological community is coming together to share data more openly and to make available

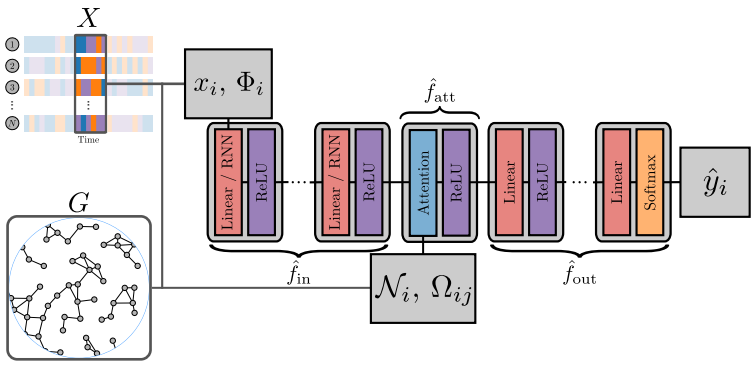


FIGURE 8.6 – **Visualization of the GNN architecture**. The blocks of different colors represent mathematical operations. The red blocks correspond to trainable affine transformation parametrized by weights and biases. The purple blocks represent activation functions between each layer. The core of the model is the attention module [314], which is represented in blue. The orange block at the end is an exponential Softmax activation that transforms the output into properly normalized outcomes.

comprehensive datasets to all researchers. Thanks to such initiatives, it is likely that future pandemics will see larger amount of data available to the scientific community in real time. It is therefore crucial for the community to start developing tools, such as the one presented here, to leverage these datasets so that we are ready for the next pandemic.

8.7 Methods

8.7.1 Graph neural network and training details

In this section, we briefly present our GNN architecture, the training settings, the synthetic data generation procedure and the hyperparameters used in our experiments.

Architecture

We use the GNN architecture shown in Fig. 8.6 and detailed in Table 8.2. First, we transform the state x_i of every node with a shared multilayer perceptron (MLP), denoted $\hat{f}_{\text{in}} : \mathcal{S} \to \mathbb{R}^d$ where d is the resulting number of node features, such that

$$\xi_i = \hat{f}_{\text{in}}(x_i)\,. \tag{8.8}$$

We concatenate the node attributes Φ_i to x_i, when these attributes are available, in which case $\hat{f}_{\text{in}} : \mathcal{S} \times \mathbb{R}^Q \to \mathbb{R}^d$. At this point, ξ_i is a vector of features representing the state (and attributes) of node i. Then, we aggregate the features of the first neighbors using a modified attention mechanism $\hat{f}_{\text{att}}$, inspired by Ref. [314] (see Sec. 8.7.1),

$$\nu_i = \hat{f}_{\text{att}}(\xi_i, \xi_{\mathcal{N}_i})\,, \tag{8.9}$$

where we recall that $\mathcal{N}_i = \{j : e_{ij} \in \mathcal{E}\}$ is the set of nodes connected to node i. We also include the edge attributes Ω_{ij} into the attention mechanism, when they are available. To do

so, we transform the edge attributes Ω_{ij} into abstract edge features, such that $\psi_{ij} = \hat{f}_{\text{edge}}(\Omega_{ij})$ where $\hat{f}_{\text{edge}} : \mathbb{R}^P \to \mathbb{R}^{d_{\text{edge}}}$ is also a MLP, before they are used in the aggregation. Finally, we compute the outcome $\hat{y}_i$ of each node i with another MLP $\hat{f}_{\text{out}} : \mathbb{R}^d \to \mathcal{R}$ such that

$$\hat{y}_i = \hat{f}_{\text{out}}(\nu_i) \,. \tag{8.10}$$

Attention Mechanism

We use an attention mechanism inspired by the graph attention network architecture (GAT) [314]. The attention mechanism consists of three trainable functions $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}$, $\mathcal{B} : \mathbb{R}^d \to \mathbb{R}$ and $\mathcal{C} : \mathbb{R}^{d_{\text{edge}}} \to \mathbb{R}$, that combine the feature vectors ξ_i, ξ_j and ψ_{ij} of a connected pair of nodes i and j, where we recall that d and d_{edge} are the number of node and edge features, respectively. Then, the attention coefficient a_{ij} is computed as follows

$$a_{ij} = \sigma\Big(\mathcal{A}(\xi_i) + \mathcal{B}(\xi_j) + \mathcal{C}(\psi_{ij})\Big) \tag{8.11}$$

where $\sigma(x) = [1 + e^{-x}]^{-1}$ is the logistic function. Notice that, by using this logistic function, the value of the attention coefficients is constrained to the open interval $(0, 1)$, where $a_{ij} = 0$ implies that the feature ξ_j do not change the value of ν_i, and $a_{ij} = 1$ implies that it maximally changes the value of ν_i. In principle, a_{ij} quantifies the influence of the state of node j over the outcome of node i. In reality, the representation learned by the GNN can be non-sparse, meaning that the neighbor features $\xi_{\mathcal{N}_i}$ can be combined in such a way that their noncontributing parts are canceled out without having a_{ij} being necessarily zero. This can result in the failure of this interpretation of this attention coefficients (see the Supplementary Information for further details). Nevertheless, the attention coefficients can be used to assess how connected nodes interact together.

We compute the aggregated feature vectors ν_i of node i as

$$\nu_i = \hat{f}_{\text{att}}(\xi_i, \xi_{\mathcal{N}_i}) = \xi_i + \sum_{j \in \mathcal{N}_i} a_{ij} \xi_j \,. \tag{8.12}$$

It is important to stress that, at this point, ν_i contains some information about i and all of its neighbors in a pairwise manner. In all our experiments, we fix $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ to be affine transformations with trainable weight matrix and bias vector. Also, we use multiple attention modules in parallel to increase the expressive power of the GNN architecture, as suggested by Ref. [314].

The attention mechanism described by Eq. (8.11) is slightly different from the vanilla version of Ref. [314]. Similarly to other well-known GNN architectures [127, 156, 207], the aggregation scheme of the vanilla GAT is designed as an average of the feature vectors of the neighbors—where, by definition, $\sum_{j \in \mathcal{N}_i} a_{ij} = 1$ for all i—rather than as a general weighted sum like for Eq. (8.12). This is often reasonable in the context of structure learning, where

Dynamics	Simple	Complex	Interacting	Metapopulation	COVID-19
Input layers	$\mathrm{Linear}(1,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU	$\mathrm{Linear}(1,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU	$\mathrm{Linear}(1,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU	$\mathrm{Linear}(4,32)^*$ ReLU $\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU	$\mathrm{RNN}(2,4;L)^*$ ReLU $\mathrm{RNN}(4,8;L)$ ReLU $\mathrm{RNN}(8,16;L)^{**}$ ReLU $\mathrm{Linear}(16,32)$ ReLU
Number of attention layers	2	2	4	$8^\dagger$	$5^{\dagger\dagger}$
Output layers	$\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,2)$ Softmax	$\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,2)$ Softmax	$\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,4)$ Softmax	$\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,32)$ ReLU $\mathrm{Linear}(32,3)$ Softmax	$\mathrm{Linear}(32,16)^*$ ReLU $\mathrm{Linear}(16,8)$ ReLU $\mathrm{Linear}(8,4)$ ReLU $\mathrm{Linear}(4,1)$
Number of parameters	6 698	6 698	11 188	99 883	7 190

TABLE 8.2 – **Layer by layer description of the GNN models for each dynamics.** For each sequence, the operations are applied from top to bottom. The operations represented by $\mathrm{Linear}(m,n)$ correspond to linear (or affine) transformations of the form $f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$, where $\boldsymbol{x} \in \mathbb{R}^m$ is the input, $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ and $\boldsymbol{b} \in \mathbb{R}^n$ are trainable parameters. The operation $\mathrm{RNN}(m,n;L)$ corresponds to an Elman recurrent neural network module [119] with m input features and n output features applied on sequences of length L [119]. The operations ReLU and Softmax are activation functions given by $\mathrm{ReLU}(x) = \max\{x,0\}$ and $\mathrm{Softmax}(\boldsymbol{x}) = \frac{\exp(\boldsymbol{x})}{\sum_i \exp(x_i)}$. (*) Here, the dimension of the input is increased by one, because we aggregated to the state of the nodes x_i their rescaled and centered population size N_i. (**) Here, only the features of the last element of the sequence—those corresponding to the state X_t—are kept to proceed further into the architecture. (†) Because the networks are weighted for the metapopulation dynamics, we initially transform the edge weights into abstract feature representations using a sequence of layers, i.e. $\big(\mathrm{Linear}(1,4), \mathrm{ReLU}, \mathrm{Linear}(4,4)\big)$ applied from left to right, before using them in the attention modules. These layers are trained alongside all the other layers. (††) The network is also weighted in this case, hence we used the same set up as for the metapopulation GNN model to transform the edge weights. Also, note that the five attention modules are each associated to a different layer in the multiplex network.

the node features represent some coordinates in a metric space where connected nodes are likely to be close [127]. Yet, in the general case, this type of constraint was shown to lessen dramatically the expressive power of the GNN architecture [333]. We also reached the same conclusion while using average-like GNN architectures (see the Supplementary Information). By contrast, the aggregation scheme described by Eq. (8.12) allows our architecture to represent various dynamic processes on networks accurately.

Training settings

In all experiments on synthetic data, we use the cross entropy loss as the local loss function,

$$L(y_i, \hat{y}_i) = -\sum_m y_{i,m} \log \hat{y}_{i,m} \,, \tag{8.13}$$

where $y_{i,m}$ corresponds to the m-th element of the outcome vector of node i, which either is a transition probability for the stochastic contagion dynamics or a fraction of people for the metapopulation dynamics. For the simple, complex and interacting contagion dynamics, we used the observed outcomes $\tilde{y}_i$, corresponding to the stochastic state of node i at the next time step, as the target in the loss function. While we noticed a diminished performance when using the observed outcomes as opposed to the true transition probabilities (see Supplementary Information), this setting is more realistic and shows what happens when the targets are noisy. The effect of noise can be tempered by increasing the size of the dataset (see the Supplementary Information). For the metapopulation dynamics, since this model is deterministic, we used the true targets without adding noise.

Performance measures

We use the Pearson correlation coefficient r as a global performance measure defined on a set of targets Y and predictions $\hat{Y}$ as

$$r = \frac{\mathbb{E}\left[(Y - \mathbb{E}[Y])(\hat{Y} - \mathbb{E}[\hat{Y}])\right]}{\sqrt{\mathbb{E}\left[(Y - \mathbb{E}[Y])^2\right] \mathbb{E}\left[(\hat{Y} - \mathbb{E}[\hat{Y}])^2\right]}} \tag{8.14}$$

where $\mathbb{E}[W]$ denotes the expectation of W. Also, because the maximum correlation occurs at $r = 1$, we also define $1 - r$ as the global error on the set of target-prediction pairs.

Synthetic data generation

We generate data from each dynamics using the following algorithm :

1. Sample a network G from a given generative model (e.g. the Erdős-Rényi $G(N, M)$ or the Barabási-Albert network models).
2. Initialize the state of the system $x(0) = \left(x_i(0)\right)_{i \in [N]}$. For the simple, complex and interacting contagion dynamics, sample uniformly the number of nodes in each state.

For the metapopulation dynamics, sample the population size for each node from a Poisson distribution of average 10^4 and then sample the number of infected people within each node from a binomial distribution of parameter 10^{-5}. For instance, a network of $|\mathcal{V}| = 10^3$ nodes will be initialized with a total of 100 infected people, on average, distributed among the nodes.

3. At time t, compute the observed outcome—y_t for the metapopulation dynamics, and $\tilde{y}_t$ for the three stochastic dynamics. Then, record the states x_t and y_t (or $\tilde{y}_t$).
4. Repeat step 3 until $(t \mod t_s) = 0$, where t_s is a resampling time. At this moment, apply step 2 to reinitialize the states x_t and repeat step 3.
5. Stop when $t = T$, where T is the targeted number of samples.

The resampling step parametrized by t_s indirectly controls the diversity of the training dataset. We allow t_s to be small for the contagion dynamics ($t_s = 2$) and larger for the metapopulation dynamics ($t_s = 100$) to emphasize on the performance of the GNN rather than the quality of the training dataset, while acknowledging that different values of t_s could lead to poor training (see Supplementary Information).

We trained the simple, complex and interacting contagion GNN models on networks of size $|\mathcal{V}| = 10^3$ nodes and on time series of length $T = 10^4$. To generate the networks, we either used Erdős-Rényi (ER) random networks $G(N, M)$ or BA random networks. In both cases, the parameters of the generative network models are chosen such that the average degree is fixed to $\langle k \rangle = 4$.

To train our models on the metapopulation dynamics, we generated 10 networks of $|\mathcal{V}| = 100$ nodes and generated for each of them time series of $t_s = 100$ time steps. This number of time steps roughly corresponds to the moment where the epidemic dies out. Similarly to the previous experiments, we used the ER and the BA models to generate the networks, where the parameters were chosen such that $\langle k \rangle = 4$. However, because this dynamics is not stochastic, we varied the average degree of the networks to increase the variability in the time series. This was done by randomly removing a fraction $p = 1 - \ln\left(1 - \mu + e\mu\right)$ of their edges, where μ was sampled for each network uniformly between 0 and 1. In this scenario, the networks were directed and weighted, with each edge weight e_{ij} being uniformly distributed between 0 and 1.

Hyperparameters

The optimization of the parameters was performed using the rectified Adam algorithm [183], which is hyperparameterized by $b_1 = 0.9$ and $b_2 = 0.999$, as suggested in Ref. [183].

To build a validation dataset, we selected a fraction of the node states randomly for each time step. More specifically, we chose node i at time t proportionally to its importance weight $w_i(t)$. For all experiments on synthetic dynamics, we randomly selected 10 nodes to be part

of the validation set, on average. For all experiments, the learning rate ϵ was reduced by a factor 2 every 10 epochs with initial value $\epsilon_0 = 0.001$. A weight decay of 10^{-4} was used as well to help regularize the training. We trained all models for 30 epochs, and selected the GNN model with the lowest loss on validation datasets. We fixed the importance sampling bias exponents for the training to $\lambda = 0.5$ in the simple, complex and interacting contagion cases, and fixed it to $\lambda = 1$ in the metapopulation case.

8.7.2 Importance weights

In this section, we show how to implement the importance weights in the different cases. Other versions of the importance weights are also available in the Supplementary Information.

Discrete state stochastic dynamics

When $\mathcal{S}$ is a finite countable set, the importance weights can be computed exactly using Eq. (8.7),

$$w_i(t) \propto \Big[\rho\big(k_i, x_i(t), x_{\mathcal{N}_i}(t)\big)\Big]^{-\lambda} \tag{8.15}$$

where $\rho\big(k, x, x_{\mathcal{N}}\big)$ is the probability to observe a node of degree k in state x with a neighborhood in state $x_{\mathcal{N}}$ in the complete dataset D. The inputs can be simplified from $(k, x, x_{\mathcal{N}})$ to $(k, x, \boldsymbol{\ell})$ without loss of generality, where $\boldsymbol{\ell}$ is a vector whose entries are the number of neighbors in each state. The distribution is then estimated from the complete dataset D by computing the fraction of inputs that are in every configuration

$$\rho(k, x, \boldsymbol{\ell}) = \frac{1}{|\mathcal{V}|T} \sum_{i=1}^{|\mathcal{V}|} \mathbb{I}\Big[k_i = k\Big] \sum_{t=1}^{T} \mathbb{I}\Big[x_i(t) = x\Big] \; \mathbb{I}\Big[\boldsymbol{\ell}_i(t) = \boldsymbol{\ell}\Big] \,, \tag{8.16}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

Continuous state deterministic dynamics

The case of continuous states—e.g. for metapopulation dynamics—is more challenging than its discrete counterpart, especially if the node and edge attributes, Φ_i and Ω_{ij}, need to be accounted for. One of the challenges is that we cannot count the inputs like in the previous case. As a result, the estimated distribution ρ cannot be estimated directly using Eq. (8.16), and we use instead

$$w_i(t) = \big[p(k_i)\, \Sigma(\Phi_i, \Omega_i | k_i)\, \Pi\big(\bar{x}(t)\big)\big]^{-\lambda} \tag{8.17}$$

where $p(k_i)$ is the fraction of nodes with degree k_i, $\Sigma(\Phi_i, \Omega_i | k_i)$ is the joint probability density function (pdf) conditioned on the degree k_i for the node attributes Φ_i and the sum of the edge attributes $\Omega_i \equiv \sum_{j \in \mathcal{N}_i} \Omega_{ij}$, and where $\Pi\big(\bar{x}(t)\big)$ is the pdf for the average of node states

at time t $\bar{x}(t) = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} x_i(t)$. The pdf are obtained using nonparametric Gaussian kernel density estimators (KDE) [61]. Provided that the density values of the KDE are unbounded above, we normalize the pdf such that the density of each sample used to construct the KDE sum to one. Further details on how we designed the importance weights are provided in the Supplementary Information.

8.7.3 Dynamics

In what follows, we describe in detail the contagion dynamics used for our experiments. We specify the node outcome function f introduced in Eq. (8.3) and the parameters of the dynamics.

Simple contagion

We consider the simple contagion dynamics called the susceptible-infected-susceptible (SIS) dynamics for which $\mathcal{S} = \{S, I\} = \{0, 1\}$—we use these two representations of $\mathcal{S}$ interchangeably. Because this dynamics is stochastic, we let $\mathcal{R} = [0, 1]^2$. We define the infection function $\alpha(\ell)$ as the probability that a susceptible node becomes infected given its number of infected neighbors ℓ

$$P(S \to I|\ell) = \alpha(\ell) = 1 - (1-\gamma)^{\ell}\,, \tag{8.18}$$

where $\gamma \in [0, 1]$ is the disease transmission probability. In other words, a node can be infected by any of its infected neighbors independently with probability γ. We also define the constant recovery probability as

$$P(I \to S) = \beta\,. \tag{8.19}$$

The node outcome function for the SIS dynamics is therefore

$$f(x_i, x_{\mathcal{N}_i}) = \begin{cases} \Big(1-\alpha(\ell_i),\, \alpha(\ell_i)\Big) & \text{if } x_i = 0, \\ (\beta,\, 1-\beta) & \text{if } x_i = 1, \end{cases} \tag{8.20}$$

where

$$\ell_i = \sum_{j\in\mathcal{N}_i} \delta(x_j, 1) \tag{8.21}$$

is the number of infected neighbors of i and $\delta(x, y)$ is the Kronecker delta. Note that for each case in Eq. (8.20), the outcome is a two-dimensional probability vector, where the first entry is the probability that node i becomes/remains susceptible at the following time step, and the second entry is the probability that it becomes/remains infected. We used $(\gamma, \beta) = (0.04, 0.08)$ in all experiments involving this simple contagion dynamics.

Complex contagion

To lift the independent transmission assumption of the SIS dynamics, we consider a complex contagion dynamics for which the node outcome function has a similar form as Eq. (8.20),

but where the infection function $\alpha(\ell)$ has the nonmonotonic form

$$\alpha(\ell) = \frac{1}{z(\eta)} \frac{\ell^3}{e^{\ell/\eta} - 1} \tag{8.22}$$

where $z(\eta)$ normalizes the infection function such that $\alpha(\ell^*) = 1$ at its global maximum ℓ^* and $\eta > 0$ is a parameter controlling the position of ℓ^*. This function is inspired by the Planck distribution for the black-body radiation, although it was chosen for its general shape rather than for any physical meaning whatsoever. We used $(\eta, \beta) = (8, 0.06)$ in all experiments involving this complex contagion dynamics.

Interacting contagion

We define the interacting contagion as two SIS dynamics that are interacting and denote it as the SIS-SIS dynamics. In this case, we have $\mathcal{S} = \{S_1S_2, I_1S_2, S_1I_2, I_1I_2\} = \{0, 1, 2, 3\}$. Similarly to the SIS dynamics, we have $\mathcal{R} = [0, 1]^4$ and we define the infection probability functions

$$\alpha_r(\ell_r) = 1 - (1 - \gamma_r)^{\ell_r} \qquad \text{if } x = 0 \tag{8.23a}$$

$$\alpha_r^*(\ell_r) = 1 - (1 - \zeta\gamma_r)^{\ell_r} \qquad \text{if } x = 1, 2\,, \tag{8.23b}$$

where $\zeta \geq 0$ is a coupling constant and ℓ_r is the number of neighbors infected by disease g, and also define the recovery probabilities β_r for each disease ($r = 1, 2$). The case where $\zeta > 1$ corresponds to the situation in which the diseases are synergistic (i.e. being infected by one increases the probability of getting infected by the other), whereas competition is introduced if $\zeta < 1$ (being already infected by one decreases the probability of getting infected by the other). The case $\zeta = 1$ falls back on two independent SIS dynamics that evolve simultaneously on the network. The outcome function is composed of 16 entries that are expressed as follows :

$$\begin{aligned} f(0, x_{\mathcal{N}_i}) = \Big(&\big[1 - \alpha_1(\ell_{i,1})\big]\big[1 - \alpha_2(\ell_{i,2})\big],\ \alpha_1(\ell_{i,1})\big[1 - \alpha_2(\ell_{i,2})\big], \\ &\big[1 - \alpha_1(\ell_{i,1})\big]\,\alpha_2(\ell_{i,2}),\ \alpha_1(\ell_{i,1})\alpha_2(\ell_{i,2})\Big) \end{aligned} \tag{8.24a}$$

$$f(1, x_{\mathcal{N}_i}) = \Big(\beta_1\big[1 - \alpha_2^*(\ell_{i,2})\big],\ \big[1 - \beta_1\big]\big[1 - \alpha_2^*(\ell_{i,2})\big],\ \beta_1\alpha_2^*(\ell_{i,2}),\ \big[1 - \beta_1\big]\alpha_2^*(\ell_{i,2})\Big) \tag{8.24b}$$

$$f(2, x_{\mathcal{N}_i}) = \Big(\big[1 - \alpha_1^*(\ell_{i,1})\big]\,\beta_2,\ \alpha_1^*(\ell_{i,1})\beta_2,\ \big[1 - \alpha_1^*(\ell_{i,1})\big]\,\big[1 - \beta_2\big],\ \alpha_1^*(\ell_{i,1})\big[1 - \beta_2\big]\Big) \tag{8.24c}$$

$$f(3, x_{\mathcal{N}_i}) = \Big(\beta_1\beta_2,\ [1 - \beta_1]\beta_2,\ \beta_1[1 - \beta_2],\ [1 - \beta_1][1 - \beta_2]\Big) \tag{8.24d}$$

where we define $\ell_{i,g}$ as the number of neighbors of i that are infected by disease g. We used $(\gamma_1, \gamma_2, \beta_1, \beta_2, \zeta) = (0.01, 0.012, 0.19, 0.22, 50)$ in all experiments involving this interacting contagion dynamics.

Metapopulation

The metapopulation dynamics considered is a deterministic version of the susceptible-infection-recovered (SIR) metapopulation model [2, 16, 60, 288]. We consider that the nodes are populated by a fixed number of people N_i, which can be in three states—susceptible (S), infected (I) or recovered (R). We therefore track the number of people in every state at each time. Furthermore, we let the network g be weighted, with the weights describing the mobility flow of people between regions. In this case, $\Omega_{ij} \in \mathbb{R}$ is the average number of people that are traveling from node j to node i. Finally, because we assume that the population size is on average steady, we let $\Phi_i = N_i$ be a node attribute and work with the fraction of people in every epidemiological state. More precisely, we define the state of node j by $x_j = (s_j, i_j, r_j)$, where s_j, i_j and r_j are the fractions of susceptible, infected and recovered people, respectively. From these definitions, we define the node outcome function of this dynamics as

$$f(x_j, x_{\mathcal{N}_j}, g) = \begin{pmatrix} s_j - s_j\tilde{\alpha}_j \\ i_j - \frac{i_j}{\tau_r} + s_j\tilde{\alpha}_j \\ r_j + \frac{i_j}{\tau_r} \end{pmatrix} \tag{8.25}$$

where

$$\tilde{\alpha}_j = \alpha(i_j, N_j) + \sum_{v_l \in \mathcal{N}_j} \frac{k_j \Omega_{jl} \alpha(i_l, N_l)}{\sum_{v_n \in \mathcal{N}_j} \Omega_{jn}}, \tag{8.26}$$

and k_j is the degree of node j. The function $\alpha(i, N)$ corresponds to the infection rate, per day, at which an individual is infected by someone visiting from a neighboring region with iN infected people in it, and is equal to

$$\alpha(i, N) = 1 - \left(1 - \frac{R_0}{\tau_r N}\right)^{iN} \approx 1 - e^{-\frac{R_0}{\tau_r} i}, \tag{8.27}$$

where R_0 corresponds to the reproduction number and, τ_r is the average recovery time in days. In all experiments with this metapopulation dynamics, we used $(R_0, \tau_r) = (8.31, 7.5)$.

8.7.4 COVID-19 outbreak in Spain

Dataset

The dataset is composed of the daily incidence of the 52 Spanish provinces (including Ceuta and Melilla) monitored for 450 days between January 1$^{\text{st}}$ 2020 and March 27$^{\text{th}}$ 2021 [117]. The dataset is augmented with the origin-destination (OD) network of individual mobility [116]. This mobility network is multiplex, directed and weighted, where the weight of each edge e_{ij}^{ν} represents mobility flow from province i and to province j using transportation ν. The metadata associated to each node is the population of province i [140], noted $\Phi_i = N_i$. The metadata associated to each edge, Ω_{ij}^{ν}, corresponds to the average number of people that moved from j to i using ν as the main means of transportation.

Models

The GNN model used in Fig. 8.5 is very similar to the metapopulation GNN model—with node and edge attributes—, with the exception that different attention modules are used to model the different OD edge types (plane, car, coach, train and boat, see Table 8.2). To combine the features of each layer of the multiplex network, we used average pooling on the output features of the attention modules. We also generalize our model to take as input a sequence of L states of the system, that is

$$\hat{y}_t = \hat{F}(x_{t:t-L+1}, g; \mathbf{\Theta}) \tag{8.28}$$

where $x_{t:t-L+1} = (x_t, x_{t-1}, \cdots, x_{t-L+1})$ and L is a lag. At the local level, it reads

$$\hat{y}_i(t) = \hat{f}\Big(x_i(t : t-L+1), \Phi_i, x_{\mathcal{N}_i}(t : t-L+1), \Phi_{\mathcal{N}_i}, \Omega_{i\mathcal{N}_i}; \mathbf{\Theta}\Big) \tag{8.29}$$

where $x_i(t : t-L+1)$ corresponds to the L previous state of node i from time t to time $t-L+1$. As we now feed sequences of node states to the GNN, we use Elman recurrent neural networks [119] to transform these sequences of states before aggregating them instead of linear layers, as shown in Fig. 8.6 and Table 8.2. Additionally, because the outputs of the models are not probability vectors, like for the dynamics of Sec. 8.7.3, but real numbers, we use the mean square error (MSE) loss to train the model :

$$L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2 . \tag{8.30}$$

We use five different baseline models to compare with the performance of our GNN : Three additional neural network architectures, a vector autoregressive model (VAR) [325] and an equivalent metapopulation model driven by a simple contagion mechanism. The first neural network architecture, denoted the KP-GNN model, was used in Ref. [147] to forecast the evolution COVID-19 in the US using a similar strategy as ours with respect to the mobility network. As described in Ref. [147], we used a single-layered MLP with 64 hidden units to transform the input, and then we used two graph convolutional networks (GCN) in series, each with 32 hidden units, to perform the feature aggregation. Finally, we computed the output of the model using another single-layered MLP with 32 hidden units. The layers of this model are separated by ReLU activation functions and are sampled with a dropout rate of 0.5. Because this model is not directly adapted to multiplex networks, we merged all layers together into a single network and summed the weights of the edges. Then, as prescribed in Ref. [147], we thresholded the merged network by keeping at most 32 neighbors with the highest edge weight for each node. We did not use our importance sampling procedure to train the KP-GNN model—letting $\lambda = 0$—to remain as close as possible to the original model.

The other two neural network architectures are very similar to the GNN model we presented in Table 8.2 : The only different component is their aggregation mechanism. The IND model,

where the nodes are assumed to be mutually independent, does not aggregate the features of the neighbors. It therefore acts like a univariate model, where the time series of each node are processed like different elements of a minibatch. In the FC model, the nodes interact via a single-layered MLP connecting all nodes together. The parameters of this MLP are learnable, which effectively allows the model to express any interaction patterns. Because the number of parameters of this MLP scales with $d|\mathcal{V}|^2$, where d is the number of node features after the input layers, we introduce another layer of 8 hidden units to compress the input features before aggregating.

The VAR model is a linear generative model adapted for multivariate time series forecasting

$$\hat{y}_t = \hat{F}(x_t, x_{t-1}, \cdots, x_{t-L+1}) = \sum_{l=0}^{L-1} \boldsymbol{A}_l x_{t-l} + \boldsymbol{b} + \boldsymbol{\epsilon}_t \tag{8.31}$$

where $\boldsymbol{A}_l \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ are weight matrices, $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{V}|}$ is a trend vector and $\boldsymbol{\epsilon}_t$ is an error term with $\mathbb{E}[\boldsymbol{\epsilon}_t] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_s] = \delta(t,s)\boldsymbol{\Sigma}$, with $\boldsymbol{\Sigma}$ being a positive-semidefinite covariance matrix. While autoregressive models are often used to predict stock markets [284], they have also been used recently to forecast diverse COVID-19 outbreaks [266]. This model is fitted to the COVID-19 time series dataset also by minimizing the MSE.

The metapopulation model is essentially identical to the model presented in Sec. 8.7.3. However, because we track the incidence, i.e. the number of newly infectious cases $\chi_i(t)$ in each province i, instead of the complete state (S_i, I_i, R_i) representing the number of individuals in each state, we allow the model to internally track the complete state based on the ground truth. At first, the whole population is susceptible, i.e. $S_i(1) = N_i$. Then, at each time step, we subtract the number of newly infectious cases in each node from $S_i(t)$, and add to I_i. Finally, the model allows a fraction $\frac{1}{\tau_r}$ of its infected people to recover. The evolution equations of this model are as follows

$$S_i(t+1) = S_i(t) - \chi_i(t)\,, \tag{8.32a}$$

$$I_i(t+1) = I_i(t) + \chi_i(t) - \frac{1}{\tau_r} I_i(t)\,, \tag{8.32b}$$

$$R_i(t+1) = R_i(t) + \frac{1}{\tau_r} I_i(t)\,. \tag{8.32c}$$

Finally, we computed the incidence $\hat{\chi}_i(t)$ predicted by the metapopulation model using the current internal state as follows :

$$\hat{\chi}_i(t) = S_i \tilde{\alpha}_i\,, \tag{8.33}$$

where $\tilde{\alpha}_i$ is given by Eq. (8.26), using the mobility network G, Eq. (8.27) for $\alpha(i,N)$ and $i_j = \frac{I_j}{N_j}$. Since the mobility weights Ω_{ij}^{ν} represent the average number of people traveling from province j to province i, we assumed all layers to be equivalent and aggregated each layer into a single typeless network where $\Omega_{ij} = \sum_{\nu} \Omega_{ij}^{\nu}$. We fixed the parameters of the model to $R_0 = 2.5$ and $\tau_r = 7.5$, as these values were used in other contexts for modeling the propagation of COVID-19 in Spain [4].

Training

We trained the GNN and other neural networks for 200 epochs, while decreasing the learning rate by a factor of 2 every 20 epochs with an initial value of 10^{-3}. For our GNN, the IND and the FC models, we fixed the importance sampling bias exponent to $\lambda = 0.5$ and, like the models trained on synthetic data, we used a weight decay of 10^{-4} (see Sec. 8.7.1). We fixed the lag of these models, including the VAR model, to $L = 5$. The KP-GNN model was trained using a weight decay of 10^{-5} following Ref. [147], and we chose a lag of $L = 7$. For all models, we constructed the validation dataset by randomly selecting a partition of the nodes at each time step proportionally to their importance weights $w_i(t)$: 20% of the nodes are used for validation in this case. The test dataset was constructed by selecting the last 100 time steps of the time series of all nodes, which roughly corresponds to the third wave of the outbreak in Spain.

8.8 Supplementary material

8.8.1 Derivation of the importance weights

Preliminaries

The importance weight $w_i(t)$ quantifies the extent to which the configuration of a node i at a time t weighs in the loss function. In turn, this affects how the parameters are optimized by correcting further for more important configurations. Yet, it is necessary to correctly define what "importance" means in this context, otherwise it can lead to badly trained models.

In the main paper, we considered the idea that the configurations should be weighted by an importance sampling (IS) scheme [263] where the target distribution is assumed uniform over all possible configurations. By doing so, we enforce the assumption that all configurations are equally important. Thus, the weights must be inversely proportional to the distribution of these configurations as they are observed in the training dataset. In the context of dynamical processes with a discrete and finite state set $\mathcal{S}$ on simple networks, this observed distribution is simply $\rho(k, x, x_{\mathcal{N}})$, where k is the degree of a node, $x \in \mathcal{S}$ is its state and $x_{\mathcal{N}} \in \mathcal{S}^k$ is the vector state of its neighbors. The importance weight of a node i at time t is then

$$w_i(t) \propto \left[\rho(k_i, x_i, x_{\mathcal{N}_i})\right]^{-\lambda} \tag{8.34}$$

where we allow λ to vary between 0 and 1, corresponding to no IS and pure IS, respectively.

Generalizing the importance weights to continuous states

For the metapopulation dynamics, we need to generalize Eq. (8.34) because the probability distribution ρ in this form can only be evaluated where $\mathcal{S}$ is a finite and countable set : ρ can

be computed by counting. When $\mathcal{S}$ is a subset of $\mathbb{R}$, counting cannot really be done directly and efficiently, which in turn prevents us from evaluating ρ. Instead, we must rely on some assumptions in order to evaluate ρ efficiently.

Neighbor-Dependent Weights We consider the direct generalization of $\rho(k_i, x_i, x_{\mathcal{N}_i})$ to real numbers. First, we factor $\rho(k_i, x_i, x_{\mathcal{N}_i}) = p(k_i)p(x_i, x_{\mathcal{N}_i}|k_i)$. By doing so, the dependence of the importance weight with the degree is more conspicuous. Then, we break apart $p(x_i, x_{\mathcal{N}_i}|k_i)$, because $p(x, x_{\mathcal{N}}|k)$ must be permutation invariant under the neighbors states. This can be done in various ways, but the simplest one is probably

$$p(x_i, x_{\mathcal{N}_i}|k_i) = \prod_{j \in \mathcal{N}_i} [p(x_i, x_j|k_i)]^{1/k_i}, \tag{8.35}$$

where $p(x, x'|k)$ is the pairwise state probability conditioned on the degree k. Here, the geometric mean ensures that $p(x, x_{\mathcal{N}}|k)$ does not have an artificially small value for nodes of high degree, as the values of $p(x, x_{\mathcal{N}}|k)$ should roughly be of the same magnitude, for any degrees. In this context, we interpret $p(x, x'|k)$ as being the probability to observe in the training dataset a node of degree k that is in state x and connected to node in state x'. Therefore, its values must be normalized and bounded by the interval $[0, 1]$.

We make use of kernel density estimators (KDE) [61] with a Gaussian kernel to represent $p(x, x'|k)$. For each value of k, we simply build a different KDE, denoted $\hat{p}(x, x'|k)$. The function $\hat{p}(x, x'|k)$ returns density values, which can have any positive value. Thereby, we normalize it to obtain a probability value such that

$$p(x, x'|k) = \frac{\hat{p}(x, x'|k)}{z_k} \tag{8.36}$$

where

$$z_k = \sum_{t=1}^{T} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \mathbb{I}[k_i = k] \hat{q}\Big(x_i(t), x_j(t)\Big|k_i\Big) \tag{8.37}$$

where $I(\cdot)$ is the indicator function.

Furthermore, the configurations must also be weighted with all additional information, that is with the node and edge attributes, i.e. Φ_i and Ω_{ij}, respectively. This is easily achieved with KDE, where we simply concatenate these attributes to the state pairs, i.e. $p(x, x', \phi, \omega|k)$.

Reducing the Complexity For continuous state dynamics such as the metapopulation one, one would like to compute the importance weights using Eq. (8.36). Now, the problem with using Eq. (8.36) is that, for a given KDE function $p(x, x', \phi, \omega|k)$, a lot of samples are used to build it. The evaluation of standard KDE is known to scale like $\mathcal{O}(nm)$, where n is the number of samples used to build the KDE and m is the number of samples on which we wish to evaluate the KDE. Consequently, we need $\mathcal{O}\left(Nk_{\max}T)^2\right)$ steps in order to evaluate all the normalization constant z_k and all the importance weights. For reasonably lengthy

time series with not too large networks, it renders the evaluation of the importance weights very inefficient. This is especially intensive for scale-free networks whose maximum degree is $k_{\max} = \mathcal{O}\left(N^{\frac{1}{\nu-1}}\right)$ [39], where ν is the exponent of the degree distribution.

To reduce the computational burden of evaluating the importance weights, we consider including additional assumptions. First, we assume that the node and edge attributes are conditionally independent from the state pair. This is equivalent to assuming that the degree encodes all the information needed to describe the state pairs. This allows us to factor them out of the pairwise state probability such that $p(x, x', \Phi, \Omega|k) = \Sigma(\Phi, \Omega|k)p(x, x'|k)$. We additionally assume that the edge attributes are correctly described by their respective mean taken over the neighbors of the node. This is equivalent to using the strength of the node $\Omega_i = \sum_{j\in\mathcal{N}_i} \Omega_{ij}$. Finally, we assume that, at a given time t, the state of the nodes are also correctly described by the average taken over all the nodes, i.e. $\bar{x}(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} x_i(t)$. Thus, we obtained the form we used in the main paper,

$$w_i(t) = \Big[p(k_i)\Sigma\big(\Phi_i, \Omega_i|k_i\big)\ \Pi\big(\bar{x}(t)\big)\Big]^{-\lambda} \tag{8.38}$$

where Σ and Π are KDE functions trained in a similar way to Eq. (8.36). Those simplifications reduce the complexity of evaluating the importance weights to $\mathcal{O}\left(N^2 + T^2\right)$, a considerable improvement to $\mathcal{O}\left((Nk_{\max}T)^2\right)$.

8.8.2 Loss descent patterns

In the case of the simple, complex and interacting contagion dynamics, we address a problem similar to a classification problem : For a given input, the model learns to assign it the correct label, i.e. the discrete state to which the node transitions to. However, contrary to more standard classification problems, the label that the graph neural network (GNN) model learns to assign is not deterministic. Instead, it is assigned stochastically with a transition probability distribution provided by the dynamical process. This dramatically changes how the cross entropy loss decreases as the training goes on, because it is no longer expected to descend to zero (see Fig. 8.7(a–c)). What is expected to descend to zero is hopefully the difference between the ground truth transition probabilities and those of the GNN. Hence, choosing an objective function such as the Kullback-Liebler divergence (KL), denoted D_{KL}, which is intimately related to the cross entropy loss and that quantifies the difference between two probability distributions, should shed some light as to why the cross entropy loss drops to a non-zero constant value. Consider the KL divergence between two discrete probability distributions p and q,

$$D_{\mathrm{KL}}(p||q) = \mathcal{H}(p, q) - \mathcal{H}(p) \tag{8.39}$$

where $\mathcal{H}(p, q) = -\sum_i p_i \log q_i$ is the cross entropy of p and q, and $\mathcal{H}(p) = -\sum_i p_i \log p_i$ is the entropy of p. It is well known that minimizing $D_{\mathrm{KL}}(p||q)$ with respect to q yields

$D_{\mathrm{KL}}(p||q) = 0$ when $q = p$, which in turn leads to

$$\min_q \left[D_{\mathrm{KL}}(p||q)\right] = \min_q \left[\mathcal{H}(p,q) - \mathcal{H}(p)\right] = \min_q \left[\mathcal{H}(p,q)\right] - \mathcal{H}(p) = 0.$$

From this expression, we readily obtain the minimum expected value of the cross entropy loss,

$$\min_q \left[\mathcal{H}(p,q)\right] = \mathcal{H}(p)\,. \tag{8.40}$$

In a realistic scenario, where the ground truth probabilities are not accessible directly, the entropy $\mathcal{H}(p)$ is also not accessible directly, which prevents the use of the KL divergence as the objective function altogether. For this reason, we use the cross entropy loss in our experiments involving stochastic dynamics. It is also worth monitoring the average entropy of the GNN outcome, which should, in the event that it is close to the ground truth, be of the same magnitude as the minimum of cross entropy loss.

In Fig. 8.7, we show an example of loss descent for each stochastic dynamics case. We also show the entropy of the GNN outcome averaged over the training dataset, and show the average Jensen-Shannon distance [65] (JSD), a symmetric version of the KL divergence, between the ground truth transition probabilities and the GNN predictions.

8.8.3 Impact of hyperparameters

Performance measures and other metrics

In this section, we investigate the impact of several hyperparameters. To quantify the performance of our models, we used two kinds of metrics. The first one is similar to the one we used in the main paper, which is the Pearson error $1 - r$ computed from the Pearson correlation coefficient r between all target-prediction pairs in the dataset. For completeness, the exact definition of r is provided in the Material and Methods section of the main paper. The second one corresponds to the log Jensen-Shannon distance [65] averaged over all target-prediction pairs. The two metrics provide a similar picture of the global performance of the GNN model. Also, because we use discrete state dynamics in this context, we are allowed to evaluate the effective sample size (ESS) in the following way :

$$n_{\mathrm{eff}} = \frac{\left[\sum_{x\in\mathcal{S}} \sum_{\boldsymbol{\ell}} n(x,\boldsymbol{\ell})\right]^2}{\sum_{x\in\mathcal{S}} \sum_{\boldsymbol{\ell}} \left[n(x,\boldsymbol{\ell})\right]^2} \tag{8.41}$$

where

$$n(x,\boldsymbol{\ell}) = \sum_{i\in\mathcal{V}} \sum_{t=1}^{T} \mathbb{I}\Big[x_i(t) = x \,\wedge\, \boldsymbol{\ell}_i(t) = \boldsymbol{\ell}\Big] \tag{8.42}$$

is the number of nodes at any times in the dataset that were in state x and that had a neighborhood state vector $\boldsymbol{\ell}$. To better appreciate the relationship between the performance metrics and the ESS, we center and rescale the ESSs with the mean and standard deviation ESS across the experiments which varies the same hyperparameter.

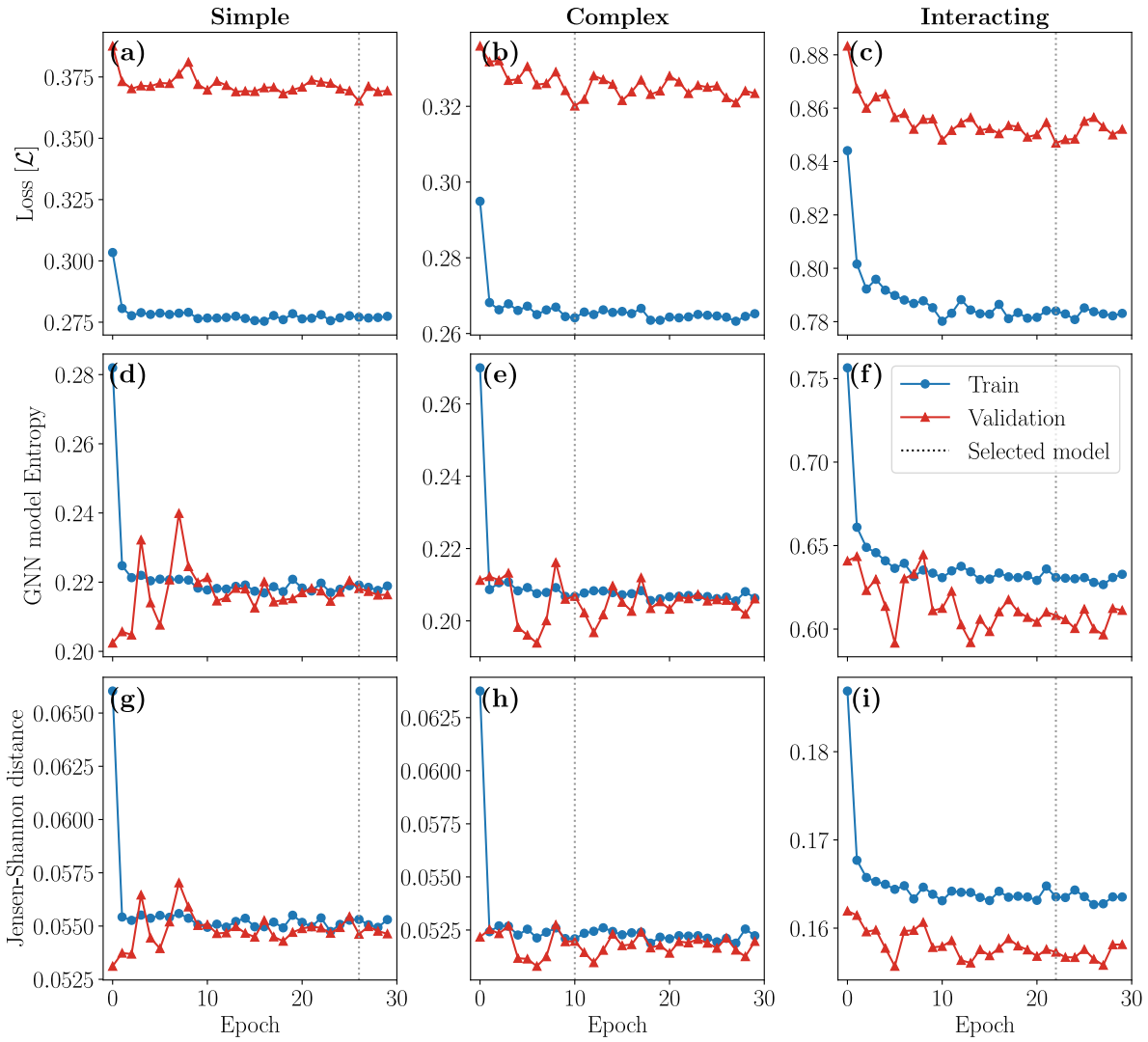


FIGURE 8.7 – **Loss optimization patterns during training.** (a–c) Loss as expressed by Eq. (3) in the main text, (d–f) average entropy of the GNN model predictions, (g–i) average Jensen-Shannon distance (JSD) between the GNN predicted LTPs and the ones given by the MLE. We show the results obtained when using Barabási-Albert networks to generate the data; similar conclusions are obtained when using data generated with Erdős-Rényi networks. All measures shown by these plots are approximated using the importance sampling scheme used to compute the loss. The vertical dotted lines show the minimum value of the validation loss, corresponding to our criterion for the model selection.

Time Series Length

The time series length, denoted by T, corresponds to the number of time steps in the training dataset. It also affects the length of an epoch.

Figure 8.8 shows the accuracy diagrams of GNN models trained using different time series lengths, for $T \in \{100, 500, 1000, 5000, 10000\}$. As we can expect, longer time series tend to yield better models. This is unsurprising for two reasons. First, because the targets with

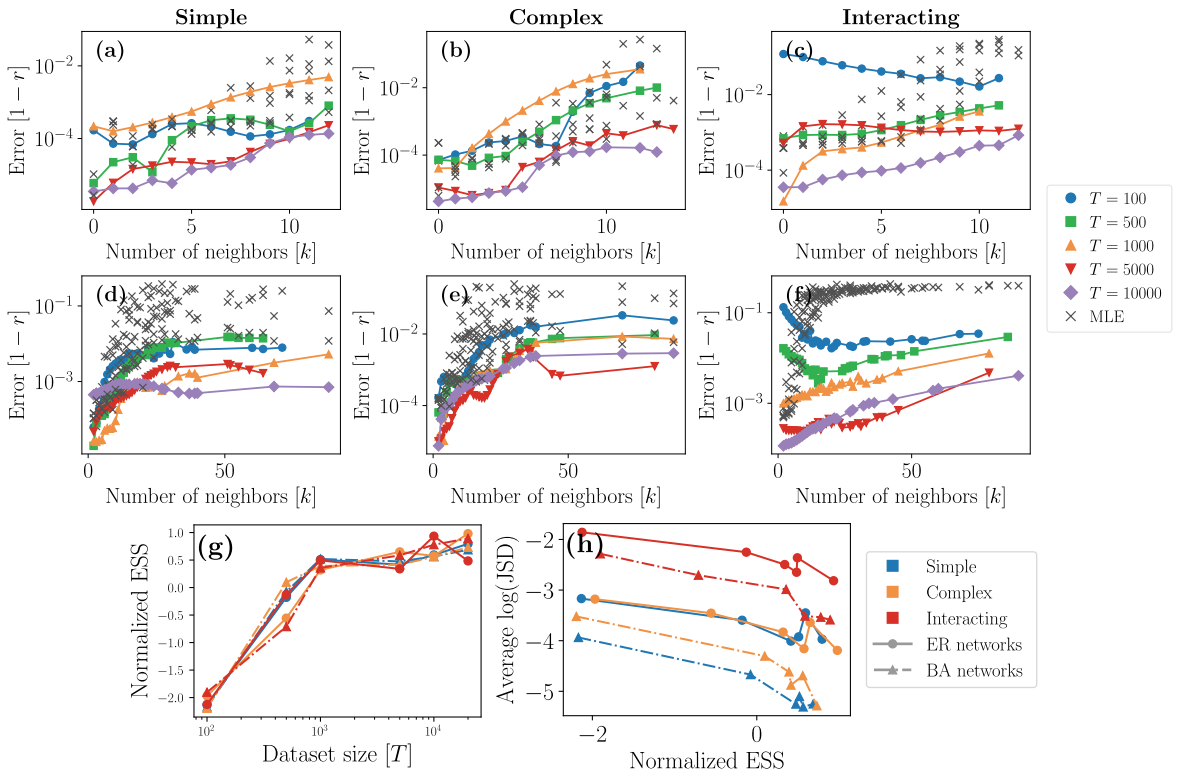


FIGURE 8.8 – **Accuracy diagrams for different time series lengths T :** We show the accuracy diagrams, that is the error as a function of the degree of the nodes, of GNNs trained on the simple (left column), complex (middle column) and interacting contagion dynamics evolving on Erdős-Rényi (ER, top row) and Barabási-Albert (BA, bottom row) networks. In every panel, we indicate the value of the changing hyperparameter, namely the time series length, with the symbols and the colors according to the legend. The maximum likelihood estimators (MLE), computed from the procedure specified in the main paper, is indicated as a reference. Panel (g) shows the normalized effective sample size (ESS) as a function of the hyperparameter. Finally, panel (h) shows the relationship between the error—the average log-JSD error to be more precise—as a function of the ESS. In panels (g, h), the symbols and line style encode the type of networks used to generate the training dataset and the colors indicate the dynamics.

which the models are trained are noisy, it generally helps to have a larger training dataset. Using noisy targets yields a noisy objective function as well, for which the noise can be reduced by increasing the number of samples. Second, using larger datasets means that we train the model for a longer period of time. We also note that, because the gradient descent is performed using a stochastic technique, the results can be a bit inconsistent with our previous observations. This is likely to also affect our next results, hence we need to keep it in mind. A time-consuming way of addressing this issue would be to train multiple GNNs in the same configurations, and to then average their errors together.

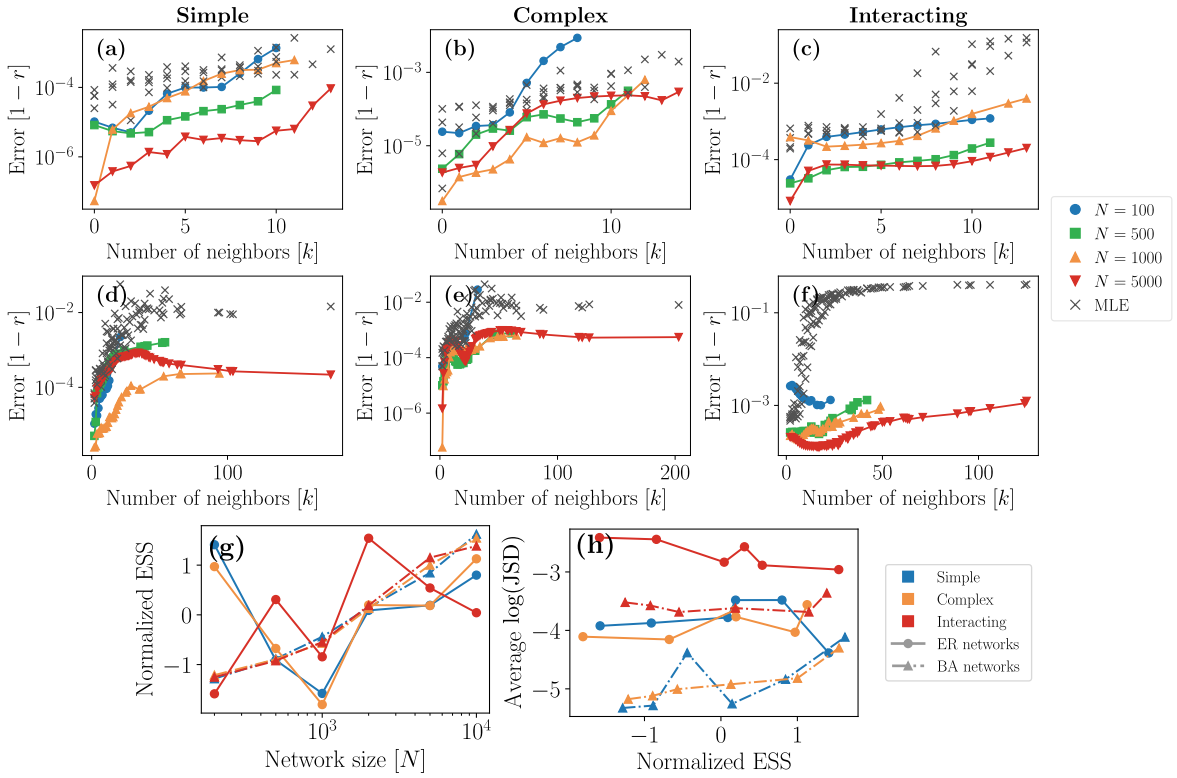


FIGURE 8.9 – **Accuracy diagrams for different network sizes N :** We refer to Fig. 8.8 for the organization of the panels.

Network size

The network size, denoted by N, is the number of nodes in the networks on which the dynamics evolved to generate the training dataset.

Similarly, Fig. 8.9 shows the accuracy diagrams when changing the network size, i.e., for $N \in \{100, 500, 1000, 5000\}$. At first, increasing N seems to affect the performance of the models differently depending on the type of networks used. First, for Erdős-Rényi (ER) networks, increasing N does not tend to increase the ESS. This is expected because the maximum degree only slightly increases when the number of nodes is increased, for fixed average degree $\langle k \rangle$. Hence, we do not observe additional degree classes when N is marginally increased and the training dataset variety remains similar. For BA networks, we observe something different : While the increase in N leads to higher ESS, there is still no substantial gain in performance. This can be explained by looking at the degree distribution. As more nodes are added to the network, the degree classes get more populated, resulting in increased ESS. However, because the degree distribution is scale-free (with exponent -3), these are not populated evenly and more degree classes are created as N increases.

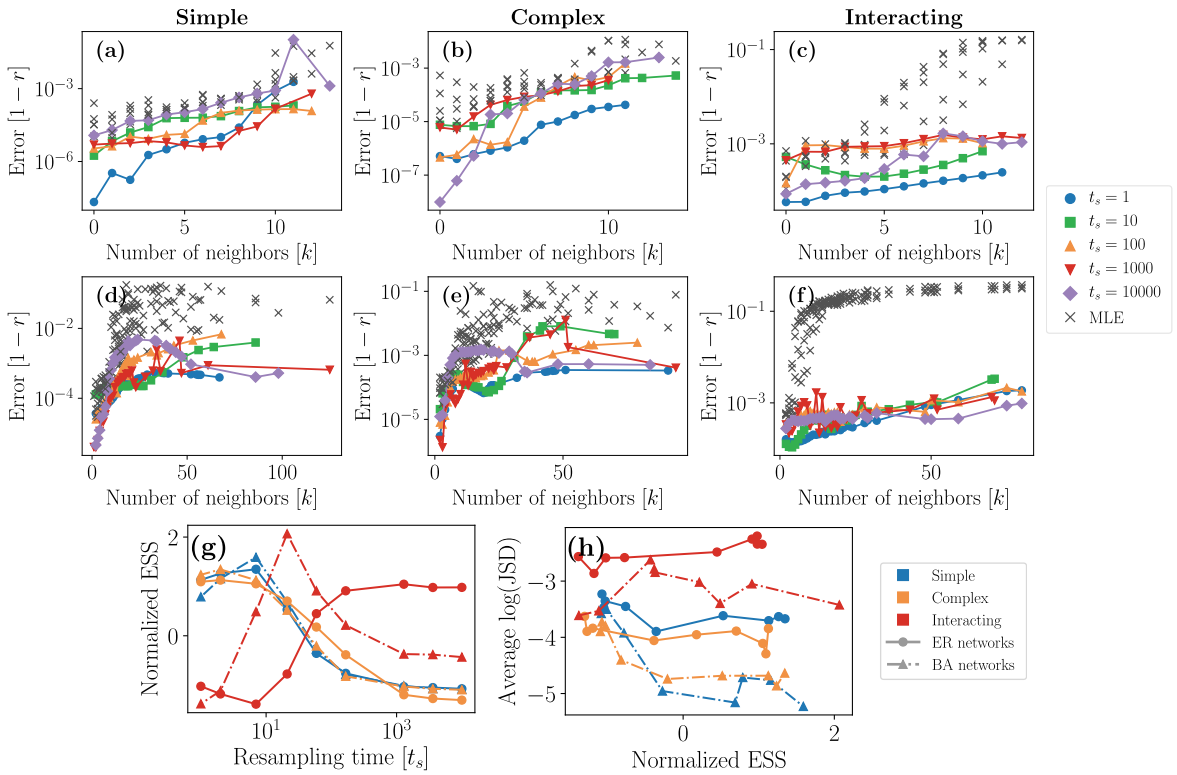


FIGURE 8.10 – **Accuracy diagrams for different resampling times t_s :** We refer to Fig. 8.8 for the organization of the panels.

Resampling time

The resampling time, denoted by t_s, corresponds to the number of time steps before the states of the nodes are reinitialized when generating the training dataset. Recall that this hyperparameter was introduced in the main paper to improve the variability in the dataset, where small values of t_s is expected to increase the ESS. We investigate the values $t_s \in \{1, 10, 100, 1000, 10000\}$.

In Fig. 8.10, we show the accuracy diagrams when the resampling time is changed. It is clear from Fig. 8.10 that decreasing the resampling time increases the ESS, thus we tend to train better models. However, in most cases, the gain seems to be marginal.

Importance Sampling Bias

The role of the importance sampling bias, denoted λ, is to modulate the influence of the importance weights, where $\lambda = 1$ corresponds to the ideal case, which is a standard IS scheme, and $\lambda = 0$ correspond to a uniform sampling scheme, that is without IS. There are multiple reasons why it would be preferable to use an exponent $\lambda < 1$. First, it is possible to poorly define the importance weights $w_i(t)$ by a bad choice of assumptions (see Sec. 8.8.1). This is in part due to the fact that we rely on some statistics to represent the training dataset, which can

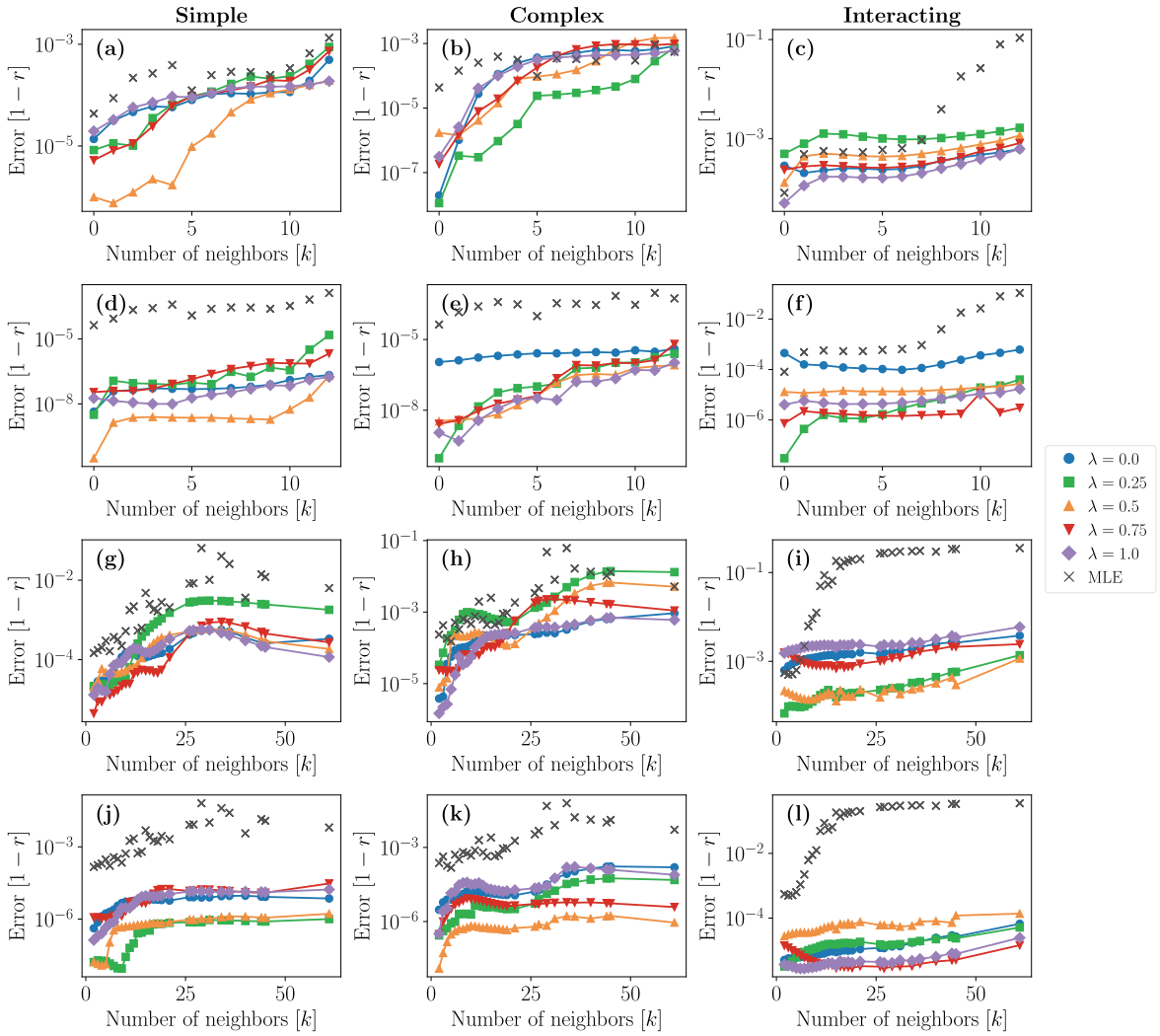


FIGURE 8.11 – **Accuracy diagrams for different important sampling bias exponents λ :** Similarly to Fig. 8.8, we show the accuracy diagrams of GNN models trained on (left column) simple, (middle column) complex and (right column) interacting contagion dynamics propagating on (a–f) Erdős-Rényi (ER) and (g–l) BA networks. We also show the maximum likelihood estimators (MLE) for comparison. Additionally, the panels (a–c) and (g–i) correspond to GNN models trained using the observed outcome, denoted $\tilde{y}_i(t)$ in the main paper, which corresponds to the state of the node at the next time step : the labels are noisy in this case. Conversely, the GNNs corresponding to panels (d–f) and (j–l) used the true transition probabilities, denoted $y_i(t)$ in the main paper : the labels are deterministic in this case. On all panels, the symbols and colors indicate the value of λ as specified by the legend.

either contain false assumptions or be poorly estimated due to a small sample size. Second, in the case of stochastic dynamics, letting $\lambda = 1$ is analogous to putting a strong emphasis on rare configurations, which in turn are likely to suffer from a small sample size. This will lead to a poor estimation of the objective function, which is likely to reduce the overall performance of the model. That being said, we investigate the values $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$.

In Fig. 8.11, we show the accuracy diagrams when the IS bias exponent λ is changed. To better appreciate the comparison between these different training settings, we used the same dataset, networks and training settings. We also trained our models in two different scenarios : we considered using the observed outcome $\tilde{y}_i(t)$, defined as the state of node i at the next time step, to evaluate the objective function. We also used the true transition probabilities, that is the outcome $y_i(t)$, to evaluate the loss. The difference between these two scenarios is that, in the first case, the targets $\tilde{y}_i(t)$ are noisy, and in the other, the targets $y_i(t)$ are deterministic. As it was mentioned earlier, we argue that the choice of λ will be dependent on the stochastic nature of the underlying dynamics.

First, from Fig. 8.11, we can see that choosing $\lambda = 1$ rarely leads to the most proficient models, for models trained on both the ER and the BA networks. Only the case of the interacting contagion dynamics does it seem to improve the performance, but as we discussed before, this is conditioned on the fact that either the sample size is large enough or the dynamic is deterministic. Interestingly, the case $\lambda = 0$ often has similar performance as the case $\lambda = 1$ in the noisy scenarios. In general, the best models seem to be obtained when λ is somewhat in the middle, as if the pure IS and the no IS cases are both too strong assumptions.

Graph Neural Network Architecture

We investigate the accuracy diagrams of the models when we use different GNN architectures. To be more specific, we consider six additional GNN architectures that has been shown to perform well in the context of structure learning [127, 156, 207, 314].

Models We label the 6 models as follows : We call our architecture the *Att-GNN*, which is described in detail in the main paper. We also consider the architecture from Ref. [207] with multiple aggregation scheme. This class of architectures aggregate the neighbors' features as follows :

$$v_i = \mathcal{A}(\xi_i) + f_{\text{AGG}}(\{\!\{\mathcal{B}(\xi_j) : j \in \mathcal{N}_i\}\!\}) \tag{8.43}$$

where $\{\!\{\cdot\}\!\}$ denotes a multiset and we recall that $\mathcal{A}$ and $\mathcal{B}$ are linear transformations with a trainable weight matrix and bias vector. Also, we need to specify the f_{AGG} function, which is a differentiable and permutation-invariant function that aggregates the neighbors' features. We consider three cases for the f_{AGG} function : the mean pooling case, denoted *Mean-GNN*, where

$$f_{\text{AGG}}(\{\!\{x_1, \cdots, x_k\}\!\}) = \sum_{i=1}^{k} \frac{x_i}{k}\,, \tag{8.44}$$

the max pooling case, denoted *Max-GNN*, where the μ^{th} feature is aggregated such that

$$[f_{\text{AGG}}(\{\!\{x_1, \cdots, x_k\}\!\})]_\mu = \max\{x_{\mu,1}, \cdots, x_{\mu,k}\}\,, \tag{8.45}$$

and the sum pooling case, denoted *Sum-GNN*, where similar to the mean pooling case,

$$f_{\text{AGG}}(\{\!\{x_1, \cdots, x_k\}\!\}) = \sum_{i=1}^{k} x_i \,, \tag{8.46}$$

Then, we consider four additional standard architectures : the *GraphSage* architecture from Ref. [126], the graph convolution network (denoted *GCN*) from Ref. [156], the original graph attention network (denoted *GAT*) from Ref. [314] and a GNN architecture used to forecast COVID-19 [147] (denoted *Kapoor-GNN*). The GraphSage aggregates the neighbors' features similarly to the Mean-GNN :

$$\nu_i = \boldsymbol{W}_1 \xi_i + \boldsymbol{W}_2 \sum_{j \in \mathcal{N}_i} \xi_j \,, \tag{8.47}$$

which, in turn is similar to the GCN,

$$\nu_i = \boldsymbol{W} \sum_{j \in \mathcal{N}_i \vee \{i\}} \frac{\xi_j}{(k_i + 1)(k_j + 1)} \,. \tag{8.48}$$

Here, $\boldsymbol{W}$ and $\boldsymbol{W}_i$ are trainable weight matrices. The GAT architecture aggregates the neighbors' features as follows :

$$\nu_i = \boldsymbol{W} \sum_{j \in \mathcal{N}_i \vee \{i\}} a_{ij} \xi_j \tag{8.49}$$

where

$$a_{ij} = \frac{e^{\theta_{ij}}}{\sum_{j \in \mathcal{N}_i \vee \{i\}} e^{\theta_{ij}}} \tag{8.50}$$

and

$$\theta_{ij} = \text{LeakyReLU}_\alpha \left(\boldsymbol{a}^T \boldsymbol{W} \xi_i + \boldsymbol{b}^T \boldsymbol{W} \xi_j \right) . \tag{8.51}$$

In this case, $\boldsymbol{a}$ and $\boldsymbol{b}$ are weight vectors and LeakyReLU_α is an activation function such that

$$\text{LeakyReLU}_\alpha(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases} , \tag{8.52}$$

and α, the negative slope, is generally fixed to 0.2. Finally, we consider the Kapoor-GNN architecture which was used to forecast COVID-19 in this US at the county level based on the mobility flow [147] between counties. This architecture is composed of a sequence of two GCN layers in series. As the first layer aggregates the features of the first neighbor, the second adds that of the second neighbors as well.

Results

In Fig. 8.12, we show the predicted transition probabilities for the simple and complex contagion dynamics. We see that, in general, the standard GNN architectures yield poor performance in predicting the infection probabilities, even though they have been trained using the same dataset, networks and hyperparameters. We believe this is due to the fact that they

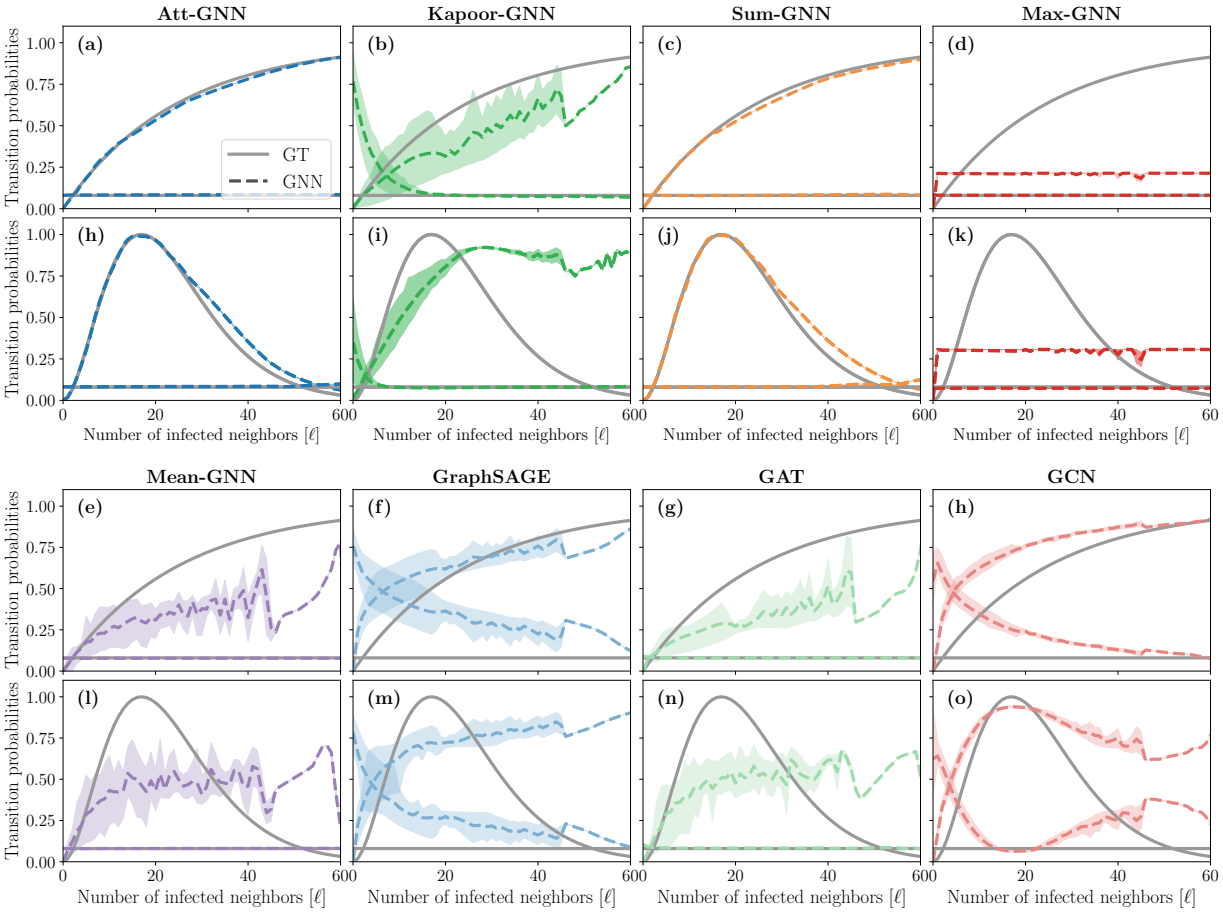


FIGURE 8.12 – **Prediction of different GNN architectures on (a–h) simple and (i–o) complex contagion dynamics on Barabási-Albert Networks :** We show the infection and recovery probabilities as predicted by the trained GNNs (dashed lines), and given by the ground truth (GT, solid lines). Each column corresponds to a different architecture. In the top and bottom rows, all models have been trained on the same training dataset and networks. The training settings and parameters of the dynamics are the same as described in the main paper. Also, we used the same training dataset and networks to train each GNN architecture.

internally use a non-extensive aggregation operator—for instance mean-pooling and max-pooling, whose output does not scale with the size of the input. To be clearer, let us assume a node of degree k of which we wish to aggregate the features of its k neighbors. By using a non-extensive aggregator, the output is expected to be of a similar scale as that of any other node of degree k'. Hence, the GNN model is likely to have a hard time distinguishing the vector features of nodes of different degrees—a structural feature that we know has a huge impact on most of the dynamical processes on networks. This is in part why almost all GNN architectures described above fail at learning and representing contagion dynamics. The only ones that perform similarly are the Att-GNN and Sum-GNN, which both use extensive aggregators.

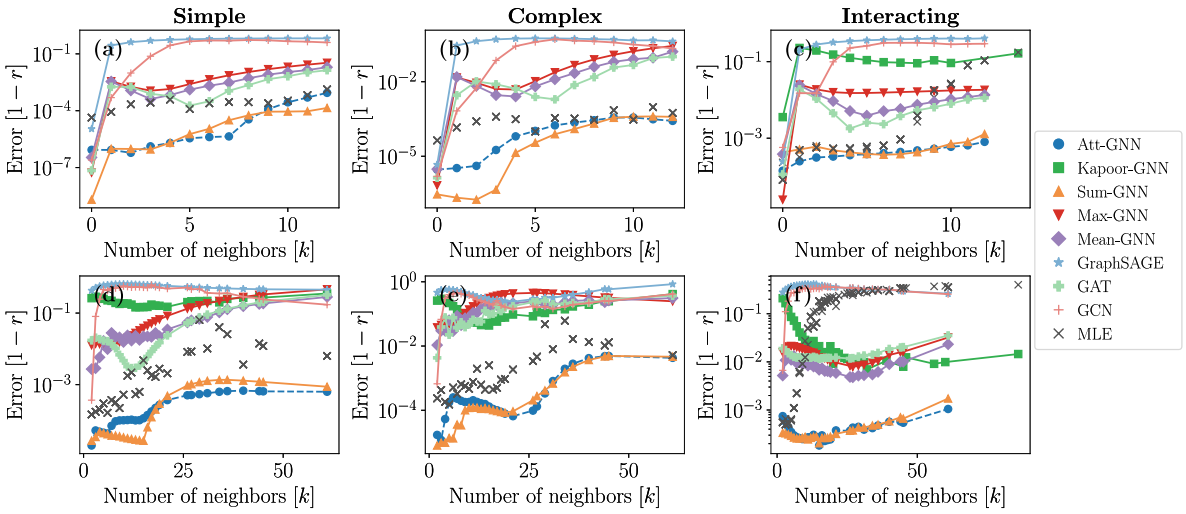


FIGURE 8.13 – **Accuracy diagrams for different GNN architectures :** On each panel, the different GNN architectures were trained on the same dataset with the same training settings and hyperparameters. For further details, we refer to Fig. 8.8.

From Figs. 8.12 and 8.13, we can also appreciate how some GNN architectures are better than others at predicting the recovery probability, which is independent from the neighbors' states unlike the infection probability. Specifically, the GraphSage and GCN architectures have a hard time predicting the recovery probabilities. This may be due to the fact that their aggregator does not distinguish the different values of neighbor features and accepts all contributions equally, as opposed to, for instance, the GAT which is expected to weigh the neighbors' features before aggregating them. The same principle applies to the other architectures that predict correctly the recovery probability. Also, while the Kapoor-GNN, which we recall is composed of two GCN layers in series, seem to perform better than the single GCN architecture, it still struggles overall in comparison with the Att-GNN. Hence, it suggests that increasing the depth of the GNN aggregation scheme may be insufficient to improve the accuracy of the models.

In summary, not all GNN architectures are capable of learning a dynamical process on networks, which also supports the idea, presented in Ref. [333], that most GNN architectures in fact do not have a high expressive power. Then, we can ask if having an extensive aggregator will always be sufficient in the context of dynamical process learning. From our work, it seems to be the case, but the few examples we provide in this paper are far from conclusive in that regard. However, in the case where extensive aggregator would be insufficient, one could consider new strategies such that which is presented in Ref. [64], where multiple aggregators are used in parallel.

Graph neural networks on dynamic networks

It is worth mentioning that there exists a wide variety of GNN architectures designed spe-

cifically to handle dynamic networks [285], i.e. networks whose topology evolves over time. These architectures are typically composed of a sequence of GNN layers each of which being applied to one element of a sequence of temporally ordered networks $(\cdots, g_{t-1}, g_t, g_{t+1}, \cdots)$. These learned representations are then combined to obtain some network and/or node embeddings containing temporal information. While useful for temporal networks, this class of models could hardly be adapted to the task of dynamics learning on *static* networks where it is solely the states of nodes that evolve over time—such as in Figs. 8.12 and 8.13. They could, nevertheless, be useful in the context of adaptive systems, where the topology of the network changes over time according to the dynamics of the nodes [121]. However, we suspect that the GNN layers at their core should also be chosen carefully to avoid the complications caused by standard GNN architectures in the context of dynamics learning.

8.8.4 Interpretability of the models

Like most deep learning models, ours suffer from an interpretability problem in that the parameters learned during training can hardly be associated with specific mechanisms guiding the dynamics. This is in contrast with mechanistic models where the parameters are chosen beforehand to emulate a specific, interpretable behavior. The interpretability problem of our models surely is a drawback of the method, but it can be slightly alleviated. First, as we have demonstrated in the main paper (see Fig. 3), our models can be used to recover the phase transition bifurcation diagrams. As a result, even though the learned parameters of the model are non-interpretable, the behavior of the model, and more specifically the influence of the structure on the dynamics, can still be investigated.

Second, we argue in the Material and Methods section of the main paper that the parameters of our attention mechanism are partially interpretable [276, 312]. Recall that the attention mechanism computes attention coefficients a_{ij} that weigh the interaction between some node features ξ_i and that of one of its neighbors, ξ_j. Hence, we should expect a_{ij} to be high—close to 1—when two nodes are expected to interact, and it should be low—close to 0—otherwise. For example, in simple contagion dynamics, two nodes interact strictly when the target node is susceptible and the source node is infected. Otherwise, the transition probabilities of a target node i are invariant with respect to the states of its neighbors, $x_{\mathcal{N}_i}$. We say that the state of a node is neighbor invariant when the transition probability of this node is independent of the state of its neighbors.

In reality, it is not exactly what happens, as we can see in Fig. 8.14. In fact, when the target nodes are in a neighbor-invariant state (I for the simple and complex contagion dynamics, and $I_1 I_2$ for the interacting contagion dynamics), the attention coefficients are correctly close to zero. However, when they are not (e.g. state S for the simple and complex contagion dynamics), the attention mechanism can be non-zero regardless of the state of the neighbors. We think this is due to the possibility that the representations combined by the attention

mechanism are not sparse : the features of the neighbors of a node are combined in such a way that they cancel out the contributions of the features of the noncontributing neighbors. This way, the attention coefficients are not necessarily constrained to be zero even though the nodes are effectively not interacting together. This degeneracy seems to be amplified with a greater number of parallel attention modules, i.e. the number of different available representations learned by the model. This phenomenon is analogous to the sparsity problem in under-determined linear regression models, where multiple parameters can fit the same training dataset depending on the loss function [309]. The addition to the loss function of a L1-norm regularization on the attention coefficients is therefore a promising avenue to increase the interpretability of our approach.

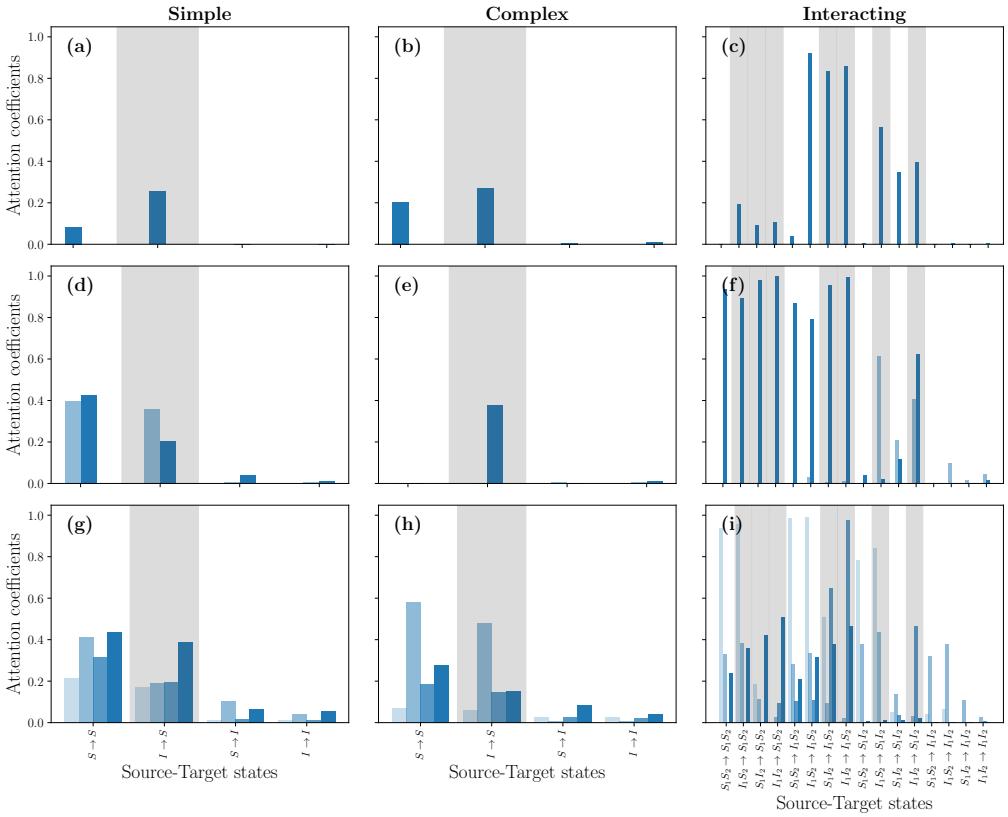


FIGURE 8.14 – **Attention coefficients as a function of the source-target states for (a-d-g) simple contagion, (b-e-h) complex contagion and (c-f-i) interacting contagion dynamics**. We show the attention coefficients of different models : (a–c) are models with one attention layer, (d–f) have two attention layers and (g–i) have four. The values of the different attention layers are shown by the increasingly lighter colored bars. For the source-target states, we indicate the type of node using the directionality of the arrows : for $X \to Y$, X is state of the source node and Y is the state of the target node. We also highlight the source-target states where we expect the transition probability of the target node to be non neighbor invariant. The other hyperparameters are given in Tab. 1 of the main paper and in Sec. A6.

Épilogue

Conclusion et perspectives

La science des systèmes complexes est un sujet pour le moins atypique. Comment serait-il possible d'expliquer et même contrôler l'évolution des sociétés humaines et les fonctions cognitives du cerveau humain, simultanément à partir des mêmes principes fondamentaux ? C'est pourtant l'ambitieux objectif de cette entreprise. Néanmoins, les avancées des six dernières décennies suggèrent que cet objectif est réaliste, et la science des réseaux semble être l'une des avenues les plus prometteuses pour y parvenir. L'histoire de la science nous a montré que les grandes transitions de paradigmes—guidées par des bonds conceptuels spontanés [165]—se font graduellement, partant d'idées intuitives se transformant en théories bien définies, sophistiquées et supportées empiriquement. La science des réseaux vit actuellement une telle révolution, où les fondations théoriques des modèles proposés au début des années 2000 sont activement revisitées et de nouvelles approches statistiques plus expressives sont développées pour tirer profit des bases de données grandissantes.

Au centre de cette révolution se trouvent les problèmes de reconstruction des modèles de réseaux. Leur reconstruction—au sens employé dans cette thèse—encapsule les problèmes d'inférence statistique à partir de données empiriques, par exemple l'inférence de la structure du réseau ou des paramètres du modèle lui-même. Cette approche est vitale pour la prochaine génération de la science des réseaux puisqu'elle fait le pont entre les modèles théoriques et les données empiriques [236]. Dans cette thèse, nous avons étudié ces problèmes de reconstruction en empruntant plusieurs perspectives différentes, autant au niveau des cas d'étude considérés que des méthodes d'analyse employées. L'objectif général reste de contribuer à l'ambitieuse entreprise de comprendre les systèmes complexes ; ce pourquoi nous avons opté pour une telle approche généraliste.

Spécifiquement, au Chapitre 5, nous avons introduit un formalisme basé sur la théorie de l'information pour étudier la relation entre la structure d'un système et son état dynamique. De ce formalisme, nous avons découvert un lien profond entre la prévisibilité du système et sa reconstructibilité, lequel n'avait jamais été établi auparavant. Ce lien prend la forme d'une dualité dans des cas particuliers, où prévisibilité et reconstructibilité varient de manière opposée l'un par rapport à l'autre, et nos résultats suggèrent que la criticalité des systèmes pourrait être un facteur déterminant dans cette dualité.

Le formalisme du Chapitre 5 nous a également permis d'identifier les limites de la reconstruction des réseaux complexes au Chapitre 6. Dans ce chapitre, nous établissons rigoureusement la reconstructibilité comme une mesure comprise entre 0 et 1, reliée mathématiquement à la borne supérieure de la performance des algorithmes de reconstruction. Motivés par des considérations empiriques, nous avons montré comment la reconstructibilité peut être adaptée pour estimer la limite de reconstruction dans les réseaux réels. Nos résultats révèlent l'importance du choix de modèles utilisés pour la reconstruction pour estimer la qualité de ses prédictions.

Au Chapitre 8, nous avons étudié une perspective différente du problème de reconstruction : celle dans laquelle la dynamique du système est reconstruite directement à partir des données. L'une des limitations des approches réseaux conventionnelles est celle des modèles de dynamiques eux-mêmes, dans lesquels les mécanismes d'évolution sont basés sur des hypothèses simplificatrices utiles pour leur interprétabilité mais limités par leur expressivité. De surcroît, nous étions motivés à repousser ces limitations en utilisant des outils puissants tirés de l'apprentissage profond—les GNNs. Notre approche, capable de décrire plusieurs types de processus de contagion de différente complexité, s'est avérée prometteuse comme outil *in vitro* pour étudier leur criticalité empiriquement.

Reconstructabilité et prévisibilité

De par sa nature interdisciplinaire et généraliste, notre travail soulève plusieurs questions et applications potentielles que nous n'avons pu explorer en profondeur dans la durée du doctorat. Dans plusieurs cas, nous avons eu l'occasion d'ébaucher des pistes de solutions prometteuses, lesquelles pourront être développées davantage dans le futur.

Au Chapitre 5, nous avons introduit des techniques numériques pour estimer l'information mutuelle, lesquelles sont limitées à des systèmes de taille modérée (quelques milliers de nœuds). Ces méthodes Monte-Carlo sont également biaisées et coûteuses en temps de calcul. Développer de nouvelles techniques plus efficaces serait ainsi une avenue de recherche prometteuse afin de faciliter l'application du formalisme. Plusieurs pistes sont actuellement envisageables. Par exemple, les descriptions approximées (à la manière de celle proposée dans l'Annexe A) dont le calcul de l'information mutuelle pourrait être fait analytiquement (ou partiellement analytiquement) accélèreraient grandement le processus. Dans le même ordre d'idée, il serait intéressant de trouver des cas où des solutions exactes peuvent être calculées analytiquement, à l'exception des exemples simples présentés aux Chapitres 5 et 6. Des résultats préliminaires nous indiquent que les marches aléatoires sur graphes forment un bon candidat à cet égard. Le Chapitre 5 nous a également amenés à conjecturer l'existence de dualités universelles, notamment dans le cas dépendant du passé et autour de la criticalité des systèmes. Démontrer leur existence aurait certainement des implications importantes dans le cadre de notre travail.

Le succès de l'application du formalisme du Chapitre 5 au problème de reconstruction de graphes présenté au Chapitre 6 suggère que d'autres problèmes d'inférence pourraient bénéficier de notre approche. Le problème de détection de communautés [94] vient à l'esprit. Dans la Réf. [340], les auteurs proposent une méthode d'analyse de la limite de détectabilité également inspirée de la théorie de l'information, laquelle est basée sur un ratio de vraisemblances. Or, nous croyons que ce ratio est intimement lié à l'information mutuelle ; une correspondance plus formelle entre les deux mesures serait ainsi intéressante à établir. Une autre piste de recherche dans cette veine serait d'appliquer le formalisme dans le contexte de l'apprentissage profond. Une publication récente a utilisé des outils informationnels analogues aux nôtres pour quantifier la capacité de mémorisation et généralisation des grandes modèles de langage [208]. Il est possible que la présence de dualité entre prévisibilité et reconstructibilité ait un impact sur de tels modèles.

Au Chapitre 6, nous avons présenté un protocole pour estimer la reconstructibilité dans des systèmes réels, que nous avons appliqué sur des données d'activité neuronale de souris. Nous croyons que ce type d'analyses pourrait être approfondi, spécifiquement dans le contexte des neurosciences, où la reconstruction joue un rôle important. Le poisson zèbre dont le connectome est en cours de caractérisation [177] est un cas d'espèce idéal.

Reconstruction de modèles effectifs

Le succès du modèle GNN présenté au Chapitre 8 pour décrire divers processus de contagion ouvre la voie à de nombreuses applications potentielles. L'une de ces applications consiste à développer des modèles de contagion *effectifs*, c'est-à-dire des modèles décrivant partiellement la dynamique d'un système, dont certains facteurs sont inconnus. Un projet mettant en œuvre cette idée est en cours, où nous considérons des agents de contagion en interaction, dans lequel seul un agent est observé. Les résultats préliminaires indiquent que la dynamique résultante ressemble à celle d'un agent de contagion complexe, un phénomène récemment corroboré dans la Réf. [291]. Ce type d'interaction d'agents est observé empiriquement ; la pandémie de COVID-19 en est un exemple clé, dans lequel l'interaction entre la propagation du virus et celle de la désinformation était indéniable.

Une autre application potentielle implique la possibilité de reconstruire la dynamique d'un système et sa structure simultanément. Si cette tâche est réalisable, elle permettrait d'outrepasser la capacité des modèles de dynamiques actuels et d'établir une description en réseau propre aux données. À l'heure actuelle, il n'existe à notre connaissance aucun résultat théorique établissant la cohérence de cette tâche : il est possible que plusieurs couples dynamique-structure mènent à des processus résultants similaires. Qui plus est, ces modèles pourraient être profondément difficiles à interpréter, minant par le fait même leur utilité. En dépit de ces difficultés, des travaux récents basés sur les GNNs suggèrent que cette tâche est au moins réalisable, pour reconstruire complètement un réseau d'interaction [155] ou pour

en faire une reconstruction partielle [172]. Une analyse rigoureuse de leur reconstructibilité pourrait faire la lumière sur la validité de ces modèles.

Annexe A

Équations maîtresses approximées typées dans les dynamiques binaires sur réseaux corrélés

Article en préparation :

Charles Murphy, Jérémi Lesage, Antoine Allard
Département de Physique, de Génie Physique et d'Optique, Université Laval, Québec (Qc), Canada G1V 0A6

A.1 Avant-propos

Ce chapitre présente un projet de recherche en cours qui vise à généraliser le formalisme des équations maîtresses approximées (AME) de la Réf. [114] pour les réseaux corrélés. L'objectif initial de ce projet était d'obtenir une description plus granulaire des dynamiques binaires, dans le but de pouvoir simplifier les calculs numériques de quantités informationnelles, telles que l'information mutuelle $I(X;G)$, introduit au Chapitre 5. J. Lesage a intégré le projet durant l'été 2022 lors d'un stage qui consistait à solutionner les équations numériquement. Depuis, il a contribué de manière significative au développement du projet au-delà de la réalisation de l'algorithme numérique.

Ce projet a débuté en 2021 et nous n'avons pu l'achever à temps pour la rédaction de cette thèse, c'est pourquoi nous ne l'incluons pas dans la partie principale de la thèse. Néanmoins, nous avons déjà obtenu des résultats préliminaires prometteurs, lesquels sont présentés ici. Nous utiliserons donc cette annexe comme une archive de ces résultats, qui permettront de poursuivre le projet dans un avenir proche.

Symbole	Description
k	Degré, nombre de voisins
m	Degré actif, nombre de voisins actifs
$\alpha(k,m)$, $\beta(k,m)$	Fonction des taux d'activation et de désactivation, respectivement
$\boldsymbol{\pi}$	Distribution stationnaire de la description exacte
$\mathbf{\Gamma}$	Matrice de taux de transition
$s_{k,m}$, $i_{k,m}$	Fraction des noeuds de degré k, inactifs et actifs ayant m voisins actifs
i_0	Probabilité qu'un noeud soit initialement actif
θ_s, ϕ_s, θ_i, ϕ_i	Taux de transition moyen des voisins inactifs et actifs
μ, ν	Étiquettes des groupes
$\mathcal{Q}$	Ensemble des groupes
$\boldsymbol{k}$, $\boldsymbol{m}$	Degrés généralisés
$k^\nu \equiv [\boldsymbol{k}]_\nu$, $m^\nu \equiv [\boldsymbol{m}]_\nu$	Nombre de voisins dans le groupe ν
$s^\mu_{\boldsymbol{k},\boldsymbol{m}}$, $i^\mu_{\boldsymbol{k},\boldsymbol{m}}$	Fraction des noeuds inactifs et actifs de degré généralisée $\boldsymbol{k}$ dans le groupe μ ayant un degré actif généralisé $\boldsymbol{m}$

$\theta_s^{\mu,\nu}$, $\phi_s^{\mu,\nu}$, $\theta_i^{\mu,\nu}$, $\phi_i^{\mu,\nu}$	Taux de transition moyen des voisins inactifs et actifs du groupe ν connectés à un noeud de groupe μ
$\boldsymbol{w}$	Matrice de connectivité entre les groupes
$w^{\mu,\nu} \equiv [\boldsymbol{w}]_{\mu,\nu}$	Probabilité qu'un voisin d'un noeud dans le groupe μ soit dans le groupe ν
$\mathbb{M}(\boldsymbol{k}\|k, \boldsymbol{p})$	Distribution multinomiale de paramètre $\boldsymbol{p} = (p_\nu)_{\nu\in\mathcal{Q}}$
$\mathbb{B}(m\|k, p)$	Distribution binomiale de paramètre p
$s_{k,m}^{\mu}$, $i_{k,m}^{\mu}$	Fraction des noeuds inactifs et actifs de degré total k dans le groupe μ ayant un degré total actif m
$i_{k;0}^{\mu}$	Probabilité qu'un noeud dans le groupe μ de degré k soit initialement actif
$\boldsymbol{\chi}_{k,m}^{\mu}$, $\boldsymbol{\psi}_{k,m}^{\mu}$	Distribution moyenne par groupe des voisins actifs d'un noeud dans le groupe μ ayant un degré total k et m voisins actifs
q	Paramètre de modularité
$\boldsymbol{e}$	Matrice binaire de connectivité des groupes dans le graphe aléatoire modulaire

TABLEAU A.1 – Glossaire des symboles utilisés à l'Annexe A.

A.2 Introduction

The mesoscopic structure, which encapsulates connectivity patterns emerging at the scale of groups, plays a fundamental role in information diffusion on complex networks. For instance, it has been shown to partly govern the localization phenomenon in spreading processes, a realization behind the design of better confinement measures [292]. Memory capacity of the brain has also been linked to the optimal modularity of the mesoscale organization of connectomes [216, 256]. However, the mathematical models behind these results are usually tailored to the problem at hand, which limits their scope and calls for more generalized frameworks.

We propose a general framework for describing arbitrary binary-state dynamics on mesoscopic structures. Our work is a direct generalization of the approximate master equation (AME) framework from Refs. [113, 114], but where the underlying network structure is specified by a configuration model with node types [5]. In our framework, the nodes are compartmentalized by their state, their type, their generalized degree $\boldsymbol{k}$ and their generalized active degree $\boldsymbol{m}$—both of which encode the number of neighbors of each type. In turn, this compartmentalization enables us to write multi-Type Approximate Master Equations (TAME) that accurately approximate the time evolution of the dynamics for each type of node.

We use our framework to investigate the optimality of information diffusion for the threshold model on modular graphs with a particular mesoscopic structure. We show that TAME captures accurately the optimal modularity region of Ref. [216], and the time evolution for the fraction of active nodes in each community. The generality of our framework may naturally extend the concept of optimal modularity to richer mesoscopic structures and to other binary-state dynamics, including complex contagions and consensus models.

A.3 Binary-state dynamics

The type of systems we are interested in are binary-state dynamics evolving on graphs. In these processes, nodes can either be active (or in state 1) or inactive (or in state 0). Binary-state dynamics are time-homogeneous, locally homogeneous and Markovian. As a result, they are completely determined by two functions, namely the activation rate function $\alpha(k, m)$ and the inactivation rate function $\beta(k, m)$. There functions define all transition rates in the dynamics and depend on the nodes degree k and their number of active neighbors m—i.e., their active degree. Naturally, $0 \leq m \leq k$.

These systems are exactly described by the following master equation, assuming the graph g is composed of N nodes :

$$\frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}t} = \boldsymbol{\pi}\mathbf{\Gamma} \tag{A.1}$$

where $\boldsymbol{\pi} \in [0,1]^{2^N}$ is the probability vector of the complete state of the N nodes, and $\mathbf{\Gamma} \in \mathbb{R}^{2^N \times 2^N}$ is the transition rate matrix, whose entries are combinations of the activation and inactivation rate functions. This description is exact, but also computationally intractable for moderately large systems. To circumvent this issue, mean-field approximations are typically used to reduce the dimensionality of the system. Approximate master equations are a class of mean-field approximations that have been shown to be accurate for a wide range of systems [113, 114]. We describe this framework in the next section.

A.4 Approximate master equations

The AME idea is to compartmentalize the node states to reduce the dimensionality of the system, where each compartment defines a state in which a node can be. The evolution of the AME system is then described by a set of rate equations analogous to Eq. (A.1), where additional mean-fields are introduced for closure. The way in which the AME are laid down should give us some intuition to generalize them later on.

These equations describe the temporal evolution of the probabilities $s_{k,m}(t)$ and $i_{k,m}(t)$ corresponding to the probability that (or, equivalently, fraction of) nodes of degree k with active degree m are inactive and active at time t, respectively. In this framework, nodes with identical degrees and active degrees are considered indistinguishable. Also, note that the

probabilities $s_{k,m}$ and $i_{k,m}$ are related by

$$\sum_{m=0}^{k} s_{k,m}(t) + i_{k,m}(t) = 1\,, \tag{A.2}$$

for all t. The rate equation that describes the evolution of these probabilities is

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}s_{k,m} =& -\alpha(k,m)s_{k,m} + \beta(k,m)i_{k,m} \\ &- \theta_s(k-m)s_{k,m} + \theta_s(k-m+1)s_{k,m-1} \\ &- \phi_s m s_{k,m} + \phi_s(m+1)s_{k,m+1}\,,\end{aligned} \tag{A.3}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}i_{k,m} =& -\beta(k,m)i_{k,m} + \alpha(k,m)s_{k,m} \\ &- (k-m)\theta_i i_{k,m} + (k-m+1)\theta_i i_{k,m-1} \\ &- m\phi_i i_{k,m} + (m+1)\phi_i i_{k,m+1}\,,\end{aligned} \tag{A.4}$$

where the mean-fields θ_s, ϕ_s, θ_i and ϕ_i are defined as follows :

$$\theta_s = \frac{\mathbb{E}_k\left[\sum_{m=0}^{k}(k-m)\alpha(k,m)s_{k,m}\right]}{\mathbb{E}_k\left[\sum_{m=0}^{k}(k-m)s_{k,m}\right]}\,, \tag{A.5a}$$

$$\phi_s = \frac{\mathbb{E}_k\left[\sum_{m=0}^{k}(k-m)\beta(k,m)i_{k,m}\right]}{\mathbb{E}_k\left[\sum_{m=0}^{k}(k-m)i_{k,m}\right]}\,, \tag{A.5b}$$

$$\theta_i = \frac{\mathbb{E}_k\left[\sum_{m=0}^{k} m\alpha(k,m)s_{k,m}\right]}{\mathbb{E}_k\left[\sum_{m=0}^{k} m s_{k,m}\right]}\,, \tag{A.5c}$$

$$\phi_i = \frac{\mathbb{E}_k\left[\sum_{m=0}^{k} m\beta(k,m)i_{k,m}\right]}{\mathbb{E}_k\left[\sum_{m=0}^{k} m i_{k,m}\right]}\,, \tag{A.5d}$$

such that $\mathbb{E}_k[\cdot]$ denotes the expectation over the degree distribution $p(k)$. In both cases, each term can be interpreted as different events changing the compartments' proportion of nodes. The first term corresponds to a node inactive of degree k with m infected neighbors activating—the second term is the same event but for an active node that deactivates. The four remaining terms correspond to events where the neighborhood of a node changes state. For instance, the third and fourth terms in Eq. (A.3) correspond to the event where a susceptible neighbor of a node of degree k with m infected neighbors activates, resulting in the loss of an inactive neighbor. Two of these terms appear because these events can happen in the compartment (k,m) or in the compartment $(k,m-1)$. The remaining terms are completely analogous, but instead involve active neighbors that deactivate. The mean-fields θ_s, ϕ_s, θ_i and ϕ_i are therefore approximate rates at which the neighborhood events occur.

The number of equations in the AME system is $(k_{\max}+1)^2$, where $k_{\max}$ is the maximum degree in the network. This is because there is one equation for each compartment (k,m)

and, for each degree k, there are $2(k+1)-1$ possible values of m—the -1 accounts for the normalization constraint.

The initial conditions of the AME system, i.e. $s_{k,m}(0)$ and $i_{k,m}(0)$, are seeded with an initial fraction of active nodes, denoted i_0, from which we can deduce the other variables of the system. For instance, to evaluate $i_{k,m}(0)$, we factor this probability as a product $p(i|k)p(m|i,k)$, where $p(i|k) = i_0$ is the probability that a node of degree k is selected to be active, and $p(m|i,k) = \mathbb{B}(m|k,i_0) \equiv \binom{k}{m} i_0^m (1-i_0)^{k-m}$, the probability that this node has m active neighbors, is a binomial distribution of parameter i_0. In fixing $i_{k,m}$ using a binomial distribution, we assume that active nodes are truly selected at random. Similarly, we fix $s_{k,m} = (1-i_0)\mathbb{B}(m|k,i_0)$.

The resolution of AME regarding the types of graph structure it can model is limited to the degree distribution $p(k)$ of the graphs. In fact, the implicit random graph model used in the AME framework is the configuration model, which is a graph ensemble where the degree distribution is fixed. For graphs with more intricate structural properties, such as correlations or community structure, the AME framework is therefore not appropriate. In the next section, we generalize the AME framework to such graphs.

A.5 Multi-typed approximate master equations

Suppose the nodes of a graph g are partitioned, such that each node belongs to a group $\mu \in \mathcal{Q}$, where $\mathcal{Q}$ is the set of all node groups. The fraction of nodes in group μ is denoted by $p(\mu)$—hereafter, we use Greek letters for group labels. The configuration model can be extended with node types assuming that we also partition the degrees according to the group membership [5]. For instance, assume a graph with two groups, one of its nodes of degree k may have k_1 neighbors in the first group and k_2 neighbors in the second group. Hence, this typed configuration model is constrained by the generalized membership-degree distribution, denoted $p(\mu, \boldsymbol{k})$, such that $\boldsymbol{k}$ defines the generalized degree of a node in group μ, i.e., a vector of degrees where $k^\nu \equiv [\boldsymbol{k}]_\nu$ is the number of neighbors of a node in group ν. Naturally, we have $k = \sum_\nu k^\nu$, which defines the total degree in this context. The membership-degree distribution can be factored, without loss of generality, as $p(\mu, \boldsymbol{k}) = p(\mu)p(k|\mu)p(\boldsymbol{k}|\mu,k)$, where $p(k|\mu)$ is the group-dependent degree distribution, and $p(\boldsymbol{k}|\mu,k)$ captures the correlations between groups.

To generalize the AME for the typed configuration model, we let $s^\mu_{\boldsymbol{k},\boldsymbol{m}}$ ($i^\mu_{\boldsymbol{k},\boldsymbol{m}}$) be the probability that a node of group μ with generalized degree $\boldsymbol{k}$ and generalized active degree $\boldsymbol{m}$ is inactive (active). Here, $\boldsymbol{m}$ received the same treatment as $\boldsymbol{k}$, where $m^\nu \equiv [\boldsymbol{m}]_\nu$ is the number of active neighbors of a node in group ν such that $m = \sum_\nu m^\nu$ defines the total active degree. The compartment probabilities are also normalized such that $\sum_{\boldsymbol{m}} s^\mu_{\boldsymbol{k},\boldsymbol{m}} + i^\mu_{\boldsymbol{k},\boldsymbol{m}} = 1$, where the sum is taken over all possible values of $\boldsymbol{m}$ such that $0 \leq m^\nu \leq k^\nu$ for all $\nu \in \mathcal{Q}$. The evolution of

these probabilities is governed by the following set of rate equations :

$$\begin{aligned}\frac{d}{dt}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} =& -\alpha(k,m)s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + \beta(k,m)i^{\mu}_{\boldsymbol{k},\boldsymbol{m}} \\ &+ \sum_{\nu}\left[-(k^{\nu}-m^{\nu})\theta^{\mu,\nu}_{s}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + (k^{\nu}-m^{\nu}+1)\theta^{\mu,\nu}_{s}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}-\boldsymbol{e}_{\nu}}\right] \\ &+ \sum_{\nu}\left[-m^{\nu}\phi^{\mu,\nu}_{s}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + (m^{\nu}+1)\phi^{\mu,\nu}_{s}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}+\boldsymbol{e}_{\nu}}\right]\end{aligned} \tag{A.6}$$

$$\begin{aligned}\frac{d}{dt}i^{\mu}_{\boldsymbol{k},\boldsymbol{m}} =& \alpha(k,m)s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} - \beta(k,m)i^{\mu}_{\boldsymbol{k},\boldsymbol{m}} \\ &+ \sum_{\nu}\left[-(k^{\nu}-m^{\nu})\theta^{\mu,\nu}_{i}i^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + (k^{\nu}-m^{\nu}+1)\theta^{\mu,\nu}_{i}i^{\mu}_{\boldsymbol{k},\boldsymbol{m}-\boldsymbol{e}_{\nu}}\right] \\ &+ \sum_{\nu}\left[-m^{\nu}\phi^{\mu,\nu}_{i}i^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + (m^{\nu}+1)\phi^{\mu,\nu}_{i}i^{\mu}_{\boldsymbol{k},\boldsymbol{m}+\boldsymbol{e}_{\nu}}\right]\end{aligned} \tag{A.7}$$

where $\boldsymbol{e}_{\nu}$ is a unit vector whose only non-zero component is the one with index ν. Furthermore, the mean-fields are defined as follows :

$$\theta^{\mu,\nu}_{s} = \frac{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}(k^{\mu}-m^{\mu})\alpha(k,m)s^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}(k^{\mu}-m^{\mu})s^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}, \tag{A.8a}$$

$$\phi^{\mu,\nu}_{s} = \frac{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}(k^{\mu}-m^{\mu})\beta(k,m)i^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}(k^{\mu}-m^{\mu})i^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}, \tag{A.8b}$$

$$\theta^{\mu,\nu}_{i} = \frac{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}m^{\mu}\alpha(k,m)s^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}m^{\mu}s^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}, \tag{A.8c}$$

$$\phi^{\mu,\nu}_{i} = \frac{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}m^{\mu}\beta(k,m)i^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}{\mathbb{E}_{\boldsymbol{k}|\nu}\left[\sum_{\boldsymbol{m}}m^{\mu}i^{\nu}_{\boldsymbol{k},\boldsymbol{m}}\right]}, \tag{A.8d}$$

such that the expectation $\mathbb{E}_{\boldsymbol{k}|\nu}[\cdot]$ is taken over $p(\boldsymbol{k}|\nu) = p(k|\nu)p(\boldsymbol{k}|\nu,k)$. The Eqs. (A.6) and (A.7) are the direct generalization of the AME to the typed configuration model, which are obtained, similarly as before, by accounting for each possible event that can change the state of a node. In fact, the same exact terms can be found in both sets of equations. The only difference is the dependency on the membership and the summation over the groups in the neighborhood event terms. Indeed, the neighbor groups ν must be considered separately, as a different event (i.e., neighbor activation or deactivation) can happen in each group. Also, note that the mean-fields depend on both groups μ and ν, which indicates that the TAME actually take into account the connectivity between groups.

The number of equations in the TAME system can be significantly larger than in the AME system, depending on the number of groups $|\mathcal{Q}|$. Like before, we have one equation for each compartment $(\boldsymbol{k},\boldsymbol{m})$, such that $\boldsymbol{m}$ can take any value that satisfies $0 \leq m^{\nu} \leq k^{\nu}$ for all $\nu \in \mathcal{Q}$. Consequently, we have $2\prod_{\nu}(k^{\nu}+1)-1$ possible values of $\boldsymbol{m}$ for each $\boldsymbol{k}$. Assuming that

$k^\nu \leq k_{\max}$ for all $\nu \in \mathcal{Q}$, we may have as many as $\mathcal{O}\left(k_{\max}^{2|\mathcal{Q}|}\right)$ equations if all values of $\boldsymbol{k}$ are considered. This exponential scaling in the number of groups may be intractable, which calls for further approximations. In the next section, we show how the cardinality of the TAME system can be reduced with additional assumptions.

A.6 Conditionally independent generalized degrees

Consider a special case of the typed configuration model where the generalized degrees are conditionally independent. In this case, a node of group μ and total degree k may be connected to another node in group ν with a fixed probability $w^{\mu,\nu}$. The matrix $\boldsymbol{w}$ whose elements are these connexion probabilities is a stochastic matrix, such that $\sum_\nu w^{\mu,\nu} = 1$ for all $\mu \in \mathcal{Q}$. This assumption reduces the conditional generalized degree distribution to a multinomial distribution :

$$p(\boldsymbol{k}|\mu, k) = \mathbb{M}(\boldsymbol{k}|k, \boldsymbol{w}^\mu) \equiv \binom{k}{\boldsymbol{k}} \prod_\nu (w^{\mu,\nu})^{k^\nu} \tag{A.9}$$

where $\mathbb{M}(\boldsymbol{k}|k, \boldsymbol{p})$ is a multinomial distribution of parameter $\boldsymbol{p}$, $\binom{k}{\boldsymbol{k}} = \frac{k!}{\prod_\nu k^\nu!}$ is the multinomial coefficient and $\boldsymbol{w}^\mu = (w^{\mu,\nu})_{\nu\in\mathcal{Q}}$ is the μ-th row of $\boldsymbol{w}$. The conditional independence of the generalized degrees also suggest the following approximation of the probabilities $s^\mu_{\boldsymbol{k},\boldsymbol{m}}$ and $i^\mu_{\boldsymbol{k},\boldsymbol{m}}$:

$$s^\mu_{\boldsymbol{k},\boldsymbol{m}} = s^\mu_{k,m} \mathbb{M}\left(\boldsymbol{m}\middle|m, \boldsymbol{\chi}^\mu_{k,m}\right), \tag{A.10a}$$

$$i^\mu_{\boldsymbol{k},\boldsymbol{m}} = i^\mu_{k,m} \mathbb{M}\left(\boldsymbol{m}\middle|m, \boldsymbol{\psi}^\mu_{k,m}\right), \tag{A.10b}$$

where $s^\mu_{k,m}$ and $i^\mu_{k,m}$ are marginal compartment probabilities; and $\boldsymbol{\chi}^\mu_{k,m}$ and $\boldsymbol{\psi}^\mu_{k,m}$ are vectors containing the probabilities that one of the active neighbors of a node (active or inactive, respectively) of group μ with degree k and active degree m is each groups $\nu \in \mathcal{Q}$. We denote $\chi^{\mu,\nu}_{k,m} \equiv [\boldsymbol{\chi}^\mu_{k,m}]_\nu$ and $\psi^{\mu,\nu}_{k,m} \equiv [\boldsymbol{\psi}^\mu_{k,m}]_\nu$, which we note are normalized according to $\sum_\nu \chi^{\mu,\nu}_{k,m} = \sum_\nu \psi^{\mu,\nu}_{k,m} = 1$. The marginal probabilities $s^\mu_{k,m}$ and $i^\mu_{k,m}$ are completely analogous to the ones in the AME system, except they retain the group dependency. Hence, we call this system the *conditionally independent typed approximate master equations* (CTAME) system.

Before going further, we must examine the CTAME more closely. Indeed, eqs. (A.10) result from assuming that the state of a node and its active generalized degree $\boldsymbol{m}$ is independent from its generalized degree $\boldsymbol{k}$. Hence, some values of $s^\mu_{\boldsymbol{k},\boldsymbol{m}}$ or $i^\mu_{\boldsymbol{k},\boldsymbol{m}}$ such that, for some ν, $m^\nu > k^\nu$ may be non-zero, which should not be physically possible. This might not be a problem in practice, since the variables of interest are $s^\mu_{k,m}$ and $i^\mu_{k,m}$ in the CTAME system and those constraints do not apply.

The evolution equations for the CTAME system can be obtained by substituting Eqs. (A.10) into Eqs. (A.6) and (A.7) and taking the expectation over the generalized degrees :

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}s^{\mu}_{k,m} = & -\alpha(k,m)s^{\mu}_{k,m} + \beta(k,m)i^{\mu}_{k,m} \\ & -(k-m)\theta^{\mu}_{s}s^{\mu}_{k,m} + (k-m+1)\theta^{\mu}_{s}s^{\mu}_{k,m-1} \\ & -m\phi^{\mu}_{s}s^{\mu}_{k,m} + (m+1)\phi^{\mu}_{s}s^{\mu}_{k,m+1}\,,\end{aligned} \tag{A.11}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}i^{\mu}_{k,m} = & \alpha(k,m)s^{\mu}_{k,m} - \beta(k,m)i^{\mu}_{k,m} \\ & -(k-m)\theta^{\mu}_{i}i^{\mu}_{k,m} + (k-m+1)\theta^{\mu}_{i}i^{\mu}_{k,m-1} \\ & -m\phi^{\mu}_{i}i^{\mu}_{k,m} + (m+1)\phi^{\mu}_{i}i^{\mu}_{k,m+1}\end{aligned} \tag{A.12}$$

where

$$\theta^{\mu}_{s} = \frac{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k}(kw^{\nu,\mu} - m\chi^{\nu,\mu}_{k,m})\alpha(k,m)s^{\nu}_{k,m}\right]}{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k}(kw^{\nu,\mu} - m\chi^{\nu,\mu}_{k,m})s^{\nu}_{k,m}\right]}\,, \tag{A.13a}$$

$$\phi^{\mu}_{s} = \frac{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k}(kw^{\nu,\mu} - m\psi^{\nu,\mu}_{k,m})\beta(k,m)i^{\nu}_{k,m}\right]}{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k}(kw^{\nu,\mu} - m\psi^{\nu,\mu}_{k,m})i^{\nu}_{k,m}\right]}\,, \tag{A.13b}$$

$$\theta^{\mu}_{i} = \frac{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k} m\chi^{\nu,\mu}_{k,m}\alpha(k,m)s^{\nu}_{k,m}\right]}{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k} m\chi^{\nu,\mu}_{k,m}s^{\nu}_{k,m}\right]}\,, \tag{A.13c}$$

$$\phi^{\mu}_{i} = \frac{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k} m\psi^{\nu,\mu}_{k,m}\beta(k,m)i^{\nu}_{k,m}\right]}{\mathbb{E}_{k,\nu}\left[\sum_{m=0}^{k} m\psi^{\nu,\mu}_{k,m}i^{\nu}_{k,m}\right]}\,. \tag{A.13d}$$

Some evolution equations for $\chi^{\nu,\mu}_{k,m}$ and $\psi^{\nu,\mu}_{k,m}$ may also be obtained by using the equation

$$\mathbb{E}_{\boldsymbol{k}|\mu,k}\left[\sum_{|\boldsymbol{m}|=m} m^{\nu}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}}\right] = m\chi^{\mu,\nu}_{k,m}s^{\mu}_{k,m}\,, \tag{A.14a}$$

$$\mathbb{E}_{\boldsymbol{k}|\mu,k}\left[\sum_{|\boldsymbol{m}|=m} m^{\nu}i^{\mu}_{\boldsymbol{k},\boldsymbol{m}}\right] = m\psi^{\mu,\nu}_{k,m}i^{\mu}_{k,m} \tag{A.14b}$$

such that $\mathbb{E}_{\boldsymbol{k}|\mu,k}[\cdot]$ is the expectation over the multinomial distribution $p(\boldsymbol{k}|\mu,k) = \mathbb{M}(\boldsymbol{k}|k,\boldsymbol{w}^{\mu})$, and by taking the time derivative and the expectation over the generalized degrees. Note that the summation $\sum_{|\boldsymbol{m}|=m}$ is taken over all possible values of $\boldsymbol{m}$ such that $\sum_{\nu} m^{\nu} = m$. As

an example, consider the case of $\chi^{\mu,\nu}_{k,m}$:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}[m\chi^{\mu,\nu}_{k,m}s^{\mu}_{k,m}] = \mathbb{E}_{\boldsymbol{k}|\mu,k}\Big[\sum_{\boldsymbol{m}=m}\Big(&- m^{\nu}\alpha(k,m)s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + m^{\nu}\beta(k,m)i^{\mu}_{\boldsymbol{k},\boldsymbol{m}} \\
&+ m^{\nu}\sum_{\nu'} -(k^{\nu'}-m^{\nu'})\theta^{\mu}_{s}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + (k^{\nu'}-m^{\nu'}+1)\theta^{\mu}_{s}s^{\mu}_{\boldsymbol{m}-\boldsymbol{e}_{\nu'}|\boldsymbol{k}} \\
&+ m^{\nu}\sum_{\nu'} -m^{\nu'}\phi^{\mu}_{s}s^{\mu}_{\boldsymbol{k},\boldsymbol{m}} + (m^{\nu'}+1)\phi^{\mu}_{s}s^{\mu}_{\boldsymbol{m}+\boldsymbol{e}_{\nu'}|\boldsymbol{k}}\Big]\,, \\
= ms^{\mu}_{k,m}\chi^{\mu,\nu}_{k,m}&\left[-\alpha(k,m) + \frac{i^{\mu}_{k,m}\psi^{\mu,\nu}_{k,m}}{s^{\mu}_{k,m}\chi^{\mu,\nu}_{k,m}}\beta(k,m) - (k-m)\theta^{\mu}_{s} - m\phi^{\mu}_{s}\right] \\
&+ \theta^{\mu}_{s}s^{\mu}_{k,m-1}\sum_{\nu'}\sum_{|\boldsymbol{m}|=m} m^{\nu}(kw^{\mu,\nu'} - m^{\nu'}+1)\mathbb{M}\Big(\boldsymbol{m}-\boldsymbol{e}_{\nu'}|m-1,\chi^{\mu}_{k,m-1}\Big) \\
&+ \theta^{\mu}_{s}s^{\mu}_{k,m+1}\sum_{\nu'}\sum_{|\boldsymbol{m}|=m} m^{\nu}(m^{\nu'}+1)\mathbb{M}\Big(\boldsymbol{m}+\boldsymbol{e}_{\nu'}|m+1,\chi^{\mu}_{k,m+1}\Big) \\
= ms^{\mu}_{k,m}\chi^{\mu,\nu}_{k,m}&\left[-\alpha(k,m) + \frac{i^{\mu}_{k,m}\psi^{\mu,\nu}_{k,m}}{s^{\mu}_{k,m}\chi^{\mu,\nu}_{k,m}}\beta(k,m) - (k-m)\theta^{\mu}_{s} - m\phi^{\mu}_{s}\right] \\
&+ \theta^{\mu}_{s}s^{\mu}_{k,m-1}\left[m(k-m+1)\chi^{\mu,\nu}_{k,m-1} + k\left(w^{\mu,\nu}-\chi^{\mu,\nu}_{k,m-1}\right)\right] \\
&+ m(m+1)\theta^{\mu}_{s}s^{\mu}_{k,m+1}\chi^{\mu,\nu}_{k,m+1}\,.
\end{aligned}$$

After some simplifications and arranging the terms conveniently, we obtain

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\chi^{\mu,\nu}_{k,m} =&\beta(k,m)\frac{i^{\mu}_{k,m}}{s^{\mu}_{k,m}}\left(\psi^{\mu,\nu}_{k,m}-\chi^{\mu,\nu}_{k,m}\right) \\
&+ \theta^{\mu}_{s}\frac{s^{\mu}_{k,m-1}}{s^{\mu}_{k,m}}\left[(k-m+1)\left(\chi^{\mu,\nu}_{k,m-1}-\chi^{\mu,\nu}_{k,m}\right) + \frac{k}{m}\left(w^{\mu,\nu}-\chi^{\mu,\nu}_{k,m-1}\right)\right] \\
&+ \phi^{\mu}_{s}\frac{s^{\mu}_{k,m+1}}{s^{\mu}_{k,m}}(m+1)\left(\chi^{\mu,\nu}_{k,m+1}-\chi^{\mu,\nu}_{k,m}\right)\,.
\end{aligned} \tag{A.15}$$

For $\psi^{\mu,\nu}_{k,m}$, we can proceed in an identical (and tedious) fashion,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\psi^{\mu,\nu}_{k,m} =&\alpha(k,m)\frac{s^{\mu}_{k,m}}{i^{\mu}_{k,m}}\left(\chi^{\mu,\nu}_{k,m}-\psi^{\mu,\nu}_{k,m}\right) \\
&+ \theta^{\mu}_{i}\frac{i^{\mu}_{k,m-1}}{i^{\mu}_{k,m}}\left[(k-m+1)\left(\psi^{\mu,\nu}_{k,m-1}-\psi^{\mu,\nu}_{k,m}\right) + \frac{k}{m}\left(w^{\mu,\nu}-\psi^{\mu,\nu}_{k,m-1}\right)\right] \\
&+ \phi^{\mu}_{i}\frac{i^{\mu}_{k,m+1}}{i^{\mu}_{k,m}}(m+1)\left(\psi^{\mu,\nu}_{k,m+1}-\psi^{\mu,\nu}_{k,m}\right)\,.
\end{aligned} \tag{A.16}$$

In comparison with the previous analysis where the number of differential equations was exponential with the number of groups, we now have a quadratic number of equations.

The CTAME systems of equation can be further simplified. First, note that Eqs. (A.15) and (A.16) both are exclusively composed of terms expressing a difference between some combinations of $\chi^{\mu,\nu}_{k,m}$, $\psi^{\mu,\nu}_{k,m}$ and $w^{\mu,\nu}$. As a result, the point where $\chi^{\mu,\nu}_{k,m} = \psi^{\mu,\nu}_{k,m} = w^{\mu,\nu}$ for all m and

k is an equilibrium point of the CTAME system. Whether it is stable or not is not clear at the moment, but numerical experiments suggest that it is. We leave a more thorough analysis of the stability of this equilibrium for future work. For now, we assume that this equilibrium is always stable, which lets us fix the values of $\chi_{k,m}^{\mu,\nu}$ and $\psi_{k,m}^{\mu,\nu}$ to $w^{\mu,\nu}$. At this point, we have a system of equations linear in the number of groups, which is a significant improvement over the exponential scaling of the TAME system.

Like the AME system, the CTAME system can also be initialized with a fraction of active nodes, but the mathematical treatment to get the initial conditions is slightly more complicated. It can also be generalized to depend also on the group, where i_0^μ is the initial fraction of active nodes in group μ. We start from the same procedure as before, where for instance $i_{k,m}^\mu(0) = i_0^\mu \mathbb{B}\big(m|k, \lambda_k^\mu\big)$, such that λ_k^μ is the parameter to determine, i.e., the probability that a randomly selected neighbor of node in group μ with degree k is active. We can also write this probability as the following marginal probability

$$\lambda_k^\mu = \sum_{\nu,k'} p(i,\nu,k'|\mu,k) \tag{A.17}$$

where $p(i,\nu,k'|\mu,k)$ is the probability that a node in group μ with degree k is connected to a node that is in group ν, has degree k' and is active. This probability can be factored without loss of generality into $p(i,\nu,k'|\mu,k) = p(i|\mu,k,\nu,k')p(\nu,k'|\mu,k)$, which, in turn, can be further simplified. First, since the active nodes are randomly selected, then $p(i|\mu,k,\nu,k') = i_0^\nu$. Also, for the typed configuration model with conditionally independent generalized degrees, we have

$$p(\nu,k'|\mu,k) = p(\nu|\mu,k)p(k'|\nu,\mu,k) = w^{\mu,\nu}\frac{k'p(k'|\nu)}{\mathbb{E}_{k'|\nu}[k']}\,. \tag{A.18}$$

Hence, we are left with

$$\lambda_k^\mu = \sum_\nu w^{\mu,\nu} i_0^\nu\,. \tag{A.19}$$

If we let $i_{k;0}^\mu$ have a dependency to the degree, we can also use

$$\lambda_k^\mu = \sum_{\nu,k'} \frac{w^{\mu,\nu}k'p(k'|\nu)}{\mathbb{E}_{k'|\nu}[k']} i_{k';0}^\nu\,. \tag{A.20}$$

As before, we also have $s_{k,m}^\mu = (1-i_0^\mu)\mathbb{B}\big(m|k,\lambda_k^\mu\big)$, where λ_k^μ is replaced with Eq. (A.19) or Eq. (A.20).

A.7 Optimal mesoscopic structure for information diffusion

We now apply the CTAME framework to study the optimal mesoscopic structure for information diffusion. We consider a threshold model such that the activation and deactivation rate functions are $\alpha(k,m) = H(m-\theta k)$ and $\beta = 0$, where $H(x)$ is the Heaviside step function. Here, θ is the threshold parameter, which controls the propensity of a node to activate.

This model has been shown to exhibit a region that depends on the graph structure where information flows optimally between two connected communities [216], where a signal emitted from one community propagates to the other one. This region is characterized in terms of the modularity of the graph, i.e. the fraction of edges connecting nodes of different groups, as opposed to nodes of the same group.

We consider a random graph with Q communities such that the probability that two nodes in groups μ and ν are connected is proportional to q if $\mu \neq \nu$; otherwise it is proportional to $1 - q$. Additionally, we overlay the connectivity matrix with a binary matrix $\boldsymbol{e}$, that determines which groups are connected together, where $e^{\mu,\nu} = 1$ if μ and ν can be connected and 0 otherwise. This random graph is constrained on its degrees, such that the group-dependent degree distribution $p(k|\mu)$ is fixed. In this set up, the connectivity matrix is given by

$$w^{\mu,\nu} \propto e^{\mu,\nu}[q + (1 - 2q)\delta(\mu, \nu)] \tag{A.21}$$

where $\delta(x, y)$ is the Kronecker delta. When $q = 0$, the graph is divided into Q disconnected components, and when $q = 1$ (assuming $Q = 2$), the graph is perfectly bipartite. The regime when $q = 0.5$ is equivalent to the case of a typical random graph without community structure. For the sake of illustration, we fix $p(k|\mu) = \delta(\kappa, k)$, where κ is the degree of all nodes in the graph, and choose $Q = 3$ such that $p(\mu) = \frac{1}{3}$ and

$$\boldsymbol{e} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \tag{A.22}$$

i.e., three groups with a central group connected to two other, disconnected, groups [see the diagram in Fig. A.1(b)]. These specifications correspond to the uniform (with $p(\mu) = \frac{1}{3}$), regular (with $p(k|\mu) = \delta(\kappa, k)$) and modular (with Eq. (A.21)) random graph model. The groups with labels 1 and 3 are illustrated in red, while the group with label 2 is in blue. To simulate information diffusion, we randomly select a fraction of nodes in group 2 that will be active, while the other two groups are completely inactive :

$$i_0^\mu = \begin{cases} i_0 & \mu = 2 \\ 0 & \text{otherwise}\,. \end{cases} \tag{A.23}$$

Hence, information diffusion will occur if any nodes in groups 1 or 3 activates as a result of interacting with group 2.

The results of this experiment are summarized in Fig. A.1. We show that the CTAME framework can accurately predict the time evolution of the threshold model on this type of

1. This figure has been presented during NetSci 2022 and was reused in this appendix, without modification. Back then, the notation was different which explains a typo in the legends, and should read CTAME instead of TAME.

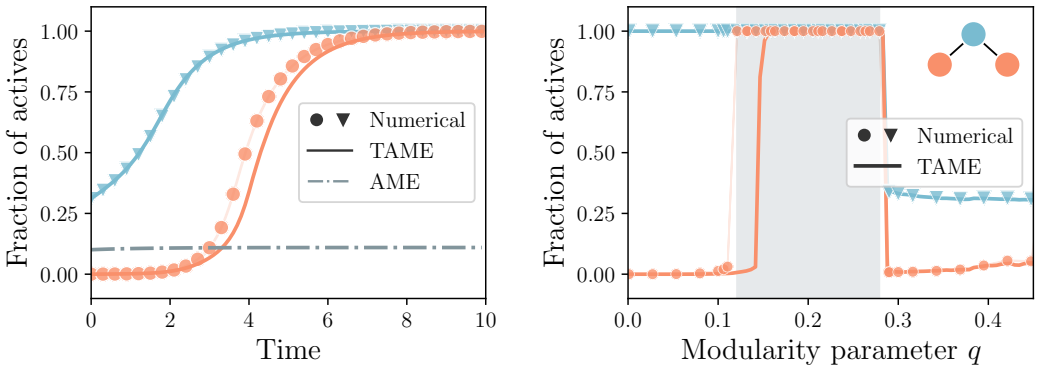


FIGURE A.1 – Information diffusion on uniform, regular and modular graphs of size $N = 10^5$ with $M = 10^6$ edges and three groups ($\kappa = 20$). [1] The numerical simulations are averaged over 50 realisations in both panels.
Left panel : Time evolution of the threshold dynamics predicted by TAME where the threshold is $\theta = 0.3$ and the modularity parameter is $q = 0.2$. We show numerical simulations (symbols), the numerical solution of Eqs. A.11 and A.12 (solid lines) and the prediction of the AME from Refs. [113, 114] as a baseline (dotted line). The symbols and lines are color-coded according to the diagram in the left panel, specifying the group to which it corresponds.
Right panel : Phase diagram of the threshold model with threshold $\theta = 0.3$. The shaded area denotes an optimal modularity region akin to that of Ref. [216].

random graph, a task that is typically difficult for mean-field approaches [114]. Additionally, the CTAME system is also shown to predict the optimal modularity region, which is in agreement with the results of Ref. [216]. However, note that the community structure considered here is marginally different than the one in Ref. [216]—now involving three groups instead of two with a non-trivial community structure—which illustrates the flexibility of the CTAME framework.

A.8 Conclusion

In this chapter, we introduced a new approximate master equation framework for studying binary-state dynamics evolving on a general class of graphs. Even though our framework is general, it is also particularly difficult to solve in practice, due to the size of the system of equations. For this reason, we introduced a conditionally independent approximation that reduces the number of equations to a linear scaling with the number of groups. We showed that this approximation is capable of predicting the optimality region in the modularity space for information diffusion.

This work is far from complete, and there are many directions to explore before it can be considered for publication. First, we need to investigate the stability of the equilibrium point allowing to simplify the CTAME system. Second, we need to further explore the capacities of TAME and CTAME frameworks, considering other case studies and comparing them

to simulations. These case studies may include different binary-dynamics, but also other graph structures, such as graphs with degree-degree correlations or more complex community structures. It is also tempting to use real networks in these experiments, which our framework is, in principle, able to represent.

Bibliographie

[1] ABBEEL P., KOLLER D. ET NG A. Y., *Learning factor graphs in polynomial time and sample complexity*, J. Mach. Learn. Res., 7 (2006), p. 1743–1788.

[2] AJELLI M., ZHANG Q., SUN K., MERLER S., FUMANELLI L., CHOWELL G., SIMONSEN L., VIBOUD C. ET VESPIGNANI A., *The RAPIDD Ebola forecasting challenge : Model description and synthetic data generation*, Epidemics, 22 (2018), p. 3–12.

[3] ALBERT R. ET BARABÁSI A.-L., *Statistical mechanics of complex networks*, Rev. Mod. Phys., 74 (2002), p. 47.

[4] ALETA A. ET MORENO Y., *Evaluation of the potential incidence of COVID-19 and effectiveness of containment measures in Spain : A data-driven approach*, BMC Med., 18 (2020), p. 157.

[5] ALLARD A., HÉBERT-DUFRESNE L., YOUNG J.-G. ET DUBÉ L. J., *General and exact approach to percolation on random graphs*, Phys. Rev. E, 92 (2015), p. 062807.

[6] AMIN M. H., ANDRIYASH E., ROLFE J., KULCHYTSKYY B. ET MELKO R., *Quantum Boltzmann Machine*, Phys. Rev. X, 8 (2018), p. 021050.

[7] ANAND K. ET BIANCONI G., *Entropy measures for networks : Toward an information theory of complex topologies*, Phys. Rev. E, 80 (2009), p. 045102(R).

[8] ——, *Gibbs entropy of network ensembles by cavity methods*, Phys. Rev. E, 82 (2010), p. 011116.

[9] ANAND K., BIANCONI G. ET SEVERINI S., *Shannon and von Neumann entropy of random networks with heterogeneous expected degree*, Phys. Rev. E, 83 (2011), p. 036109.

[10] ANDERSON P. W., *More is different*, Science, 177 (1972), p. 393–396.

[11] ANDERSON R. M. ET MAY R. M., *Infectious Diseases of Humans : Dynamics and control*, Oxford University press, 1992.

[12] ANDREWS G. E., ASKEY R., ROY R. ET ASKEY R., *Special functions*, vol. 71, Cambridge University Press, 1999.

[13] ANGULO M. T., MORENO J. A., LIPPNER G., BARABÁSI A.-L. ET LIU Y.-Y., *Fundamental limitations of network reconstruction from temporal data*, J. R. Soc. Interface, 14 (2017), p. 20160966.

[14] ARENAS A., DÍAZ-GUILERA A., KURTHS J., MORENO Y. ET ZHOU C., *Synchronization in complex networks*, Phys. Rep., 469 (2008), p. 93–153.

[15] ATHREYA K. B. ET LAHIR S. N., *Measure Theory and Probability Theory*, Springer, 2006.

[16] BALCAN D., GONÇALVES B., HU H., RAMASCO J. J., COLIZZA V. ET VESPIGNANI A., *Modeling the spatial spread of infectious diseases : The global epidemic and mobility computational model*, J. Comput. Sci., 1 (2010), p. 132–145.

[17] BARABÁSI A.-L., *The network takeover*, Nat. Phys., 8 (2012), p. 14–16.

[18] ——, *Network science*, Phil. Trans. R. Soc. A, 371 (2013), p. 20120375.

[19] ——, *Network Science*, Cambridge University Press, 2016.

[20] BARABÁSI A.-L. ET ALBERT R., *Emergence of scaling in random networks*, Science, 286 (1999).

[21] BARABÁSI D. L., BIANCONI G., BULLMORE E., BURGESS M., CHUNG S., ELIASSI-RAD T., GEORGE D., KOVÁCS I. A., MAKSE H., NICHOLS T. E., PAPADIMITRIOU C., SPORNS O., STACHENFELD K., TOROCZKAI Z., TOWLSON E. K., ZADOR A. M., ZENG H., BARABÁSI A.-L., BERNARD A. ET BUZSÁKI G., *Neuroscience needs network science*, J. Neurosci., 43 (2023), p. 5989–5995.

[22] BARNETT L., LIZIER J. T., HARRÉ M., SETH A. K. ET BOSSOMAIER T., *Information flow in a kinetic Ising model peaks in the disordered phase*, Phys. Rev. Lett., 111 (2013), p. 177203.

[23] BARZEL B. ET BARABÁSI A.-L., *Universality in network dynamics*, Nat. Phys., 9 (2013), p. 673–681.

[24] BASS F. M., *A New Product Growth for Model Consumer Durables*, Manage. Sci., 15 (1969), p. 215–227.

[25] BASSETT D. S. ET SPORNS O., *Network neuroscience*, Nat. Neurosci, 20 (2017), p. 353.

[26] BASSETT D. S., ZURN P. ET GOLD J. I., *On the nature and use of models in network neuroscience*, Nat. Rev. Neurosci., 19 (2018), p. 566.

[27] BAUM G. L., CUI D. R., Z.AND ROALF, CIRIC R., BETZEL R. ., LARSEN B., CIESLAK M., COOK P. A., XIA C. H., MOORE T. M. *et al.*, *Development of structure–function coupling in human brain networks during youth*, Proc. Natl. Acad. Sci. U. S. A., 117 (2020), p. 771–778.

[28] BENSON A. R., ABEBE R., SCHAUB M. T., JADBABAIE A. ET KLEINBERG J., *Simplicial closure and higher-order link prediction*, Proc. Nat. Acad. Sci. USA, 115 (2018), p. E11221–E11230.

[29] BENTO J. ET MONTANARI A., *Which graphical models are difficult to learn ?*, dans Advances in neural information processing systems, 2009, p. 1303–1311.

[30] BERGER J. O., BERNARDO J. M. ET SUN D., *Objective priors for discrete parameter spaces*, J. Am. Stat. Assoc., 107 (2012), p. 636–648.

[31] BIANCONI G., *Entropy of network ensembles*, Phys. Rev. E, 79 (2009), p. 036114.

[32] BIGGERSTAFF M., JOHANSSON M., ALPER D., BROOKS L. C., CHAKRABORTY P., FARROW D. C., HYUN S., KANDULA S., MCGOWAN C., RAMAKRISHNAN N., ROSENFELD R., SHAMAN J., TIBSHIRANI R., TIBSHIRANI R. J., VESPIGNANI A., YANG W., ZHANG Q. ET REED C., *Results from the second year of a collaborative effort to forecast influenza seasons in the united states*, Epidemics, 24 (2018), p. 26–33.

[33] BINDER K. ET HEERMANN D., *Monte Carlo Simulation in Statistical Physics*, Springer, 2010.

[34] BLUM A. ET RIVEST R., *Training a 3-node neural network is np-complete*, Adv. Neural Inf. Process. Syst., 1 (1988).

[35] BOCCALETTI S., ALMENDRAL J., GUAN S., LEYVA I., LIU Z., SENDIÑA-NADAL I., WANG Z. ET ZOU Y., *Explosive transitions in complex networks' structure and dynamics : Percolation and synchronization*, Phys. Rep., 660 (2016), p. 1–94.

[36] BOCCALETTI S., ALMENDRAL J. A., GUAN S., LEYVA I., LIU Z., SENDIÑA-NADAL I., WANG Z. ET ZOU Y., *Explosive transitions in complex networks' structure and dynamics : Percolation and synchronization*, Phys. Rep., 660 (2016), p. 1–94.

[37] BOGUÑÁ M., BONAMASSA I., DE DOMENICO M., HAVLIN S., KRIOUKOV D. ET SERRANO M. A., *Network geometry*, Nat. Rev. Phys., 3 (2021), p. 114–135.

[38] BOGUÑÁ M., PAPADOPOULOS F. ET KRIOUKOV D., *Sustaining the internet with hyperbolic mapping*, Nat. Commun., 1 (2010), p. 1–8.

[39] BOGUÑÁ M., PASTOR-SATORRAS R. ET VESPIGNANI A., *Cut-offs and finite size effects in scale-free networks*, Eur. Phys. J. B, 38 (2004), p. 205–209.

[40] BOLLOBÁS B., *Modern Graph Theory*, Springer, 1998.

[41] BORCHERING R. K., VIBOUD C., HOWERTON E. *et al.*, *Modeling of future covid-19 cases, hospitalizations, and deaths, by vaccination rates and nonpharmaceutical intervention scenarios — united states, april–september 2021*, MMWR. Morbidity and Mortality Weekly Report, 70 (2021), p. 719–724.

[42] BRAUER F., CASTILLO-CHAVEZ C. ET FENG Z., *Mathematical Models in Epidemiology*, Springer, 2019.

[43] BREAKSPEAR M., *Dynamic models of large-scale brain activity*, Nat. Neurosci., 20 (2017), p. 340–352.

[44] BRESLER G., MOSSEL E. ET SLY A., *Reconstruction of markov random fields from samples : some observations and algorithms*, SIAM J. Comput., 42 (2013), p. 563–578.

[45] BRONSTEIN M. M., BRUNA J., LECUN Y., SZLAM A. ET VANDERGHEYNST P., *Geometric deep learning : Going beyond euclidean data*, IEEE Signal Process. Mag., 34 (2017), p. 18–42.

[46] BRÜEL GABRIELSSON R., *Universal function approximation on graphs*, dans Advances in Neural Information Processing Systems, Larochelle H., Ranzato M., Hadsell R., Balcan M. et Lin H., éds., vol. 33, Curran Associates, Inc., 2020, p. 19762–19772.

[47] BRUGERE I., GALLAGHER B. ET BERGER-WOLF T. Y., *Network structure inference, a survey : Motivations, methods, and applications*, ACM Comput. Surv., 51 (2018), p. 1–39.

[48] BRUNA J., ZAREMBA W., SZLAM A. ET LECUN Y., *Spectral networks and locally connected networks on graphs*, arXiv preprint, arXiv :1312.6203 (2013).

[49] BRUNTON S. L., PROCTOR J. L. ET KUTZ J. N., *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proc. Natl. Acad. Sci. USA, 113 (2016), p. 3932–3937.

[50] CAO H., TAN C., GAO Z., XU Y., CHEN G., HENG P.-A. ET LI S. Z., *A survey on generative diffusion models*, IEEE Trans. Knowl. Data Eng., 36 (2024), p. 2814–2830.

[51] CARDILLO A., G'OMEZ-GARDENES J., ZANIN M., ROMANCE M., PAPO D., POZO F.D. ET BOCCALETTI S., *Emergence of network features from multiplexity*, Sci. Rep., 3 (2013), p. 1344.

[52] CARROLL S. M. ET PAROLA A., *What emergence can possibly mean*, arXiv preprint, arXiv :2410.15468 (2024).

[53] CASTELLANO C. ET PASTOR-SATORRAS R., *Relating topological determinants of complex networks to their spectral properties : Structural and dynamical effects*, Phys. Rev. X, 7 (2017), p. 041024.

[54] CENTOLA D., *The spread of behavior in an online social network experiment*, Sci. 80-, 329 (2010), p. 1194–1197.

[55] CHEN X., WENG T., YANG H., GU C., ZHANG J. ET SMALL M., *Mapping topological characteristics of dynamical systems into neural networks : A reservoir computing approach*, Phys. Rev. E, 102 (2020), p. 33314.

[56] CHIB S. ET KUFFNER T. A., *Bayes factor consistency*, arXiv preprint, arXiv :1607.00292 (2016).

[57] CIMINI G., SQUARTINI T., SARACCO F., GARLASCHELLI D., GABRIELLI A. ET CALDARELLI G., *The statistical physics of real-world networks*, Nat. Rev. Phys., 1 (2019), p. 58.

[58] CLAUSET A., MOORE C. ET NEWMAN M. E. J., *Hierarchical structure and the prediction of missing links in networks*, Nature, 453 (2008), p. 98–101.

[59] CLIFFORD P. ET SUDBURY A., *A model for spatial conflict*, Biometrika, 60 (1973), p. 581–588.

[60] COLIZZA V., PASTOR-SATORRAS R. ET VESPIGNANI A., *Reaction–diffusion processes and metapopulation models in heterogeneous networks*, Nat. Phys., 3 (2007), p. 276–282.

[61] CONOVER W. J., *Practical Nonparametric Statistics*, John Wiley & Sons, 1998.

[62] COOK S. J., JARRELL T. A., BRITTIN C. A., WANG Y., BLONIARZ A. E., YAKOVLEV M. A., NGUYEN K., TANG L., BAYER E. A., DUERR J. S., BÜLOW H., HOBERT O., HALL D. H. ET EMMONS S. W., *Whole-animal connectomes of both caenorhabditis elegans sexes*, Nature, 571 (2019), p. 63–71.

[63] COOLEN A., ANNIBALE A. ET ROBERTS E., *Markov Chain Monte Carlo sampling of graphs*, dans Generating Random Networks and Graphs, Oxford University Press, 03 2017.

[64] CORSO G., CAVALLERI L., BEAINI D., LIÒ P. ET VELIČKOVIĆ P., *Principal neighbourhood aggregation for graph nets*, arXiv preprint, arXiv :2004.05718 (2020).

[65] COVER T. M. ET THOMAS J. A., *Elements of Information Theory*, Wiley-Interscience, 2nd éd., 2006.

[66] COWAN J. D., *Stochastic neurodynamics*, dans Advances in Neural Information Processing Systems, vol. 3, 1990, p. 62.

[67] CROPPER E. C., DACKS A. M. ET WEISS K. R., *Consequences of degeneracy in network function*, Curr. Opin. Neurobiol., 41 (2016), p. 62–67.

[68] CRUTCHFIELD J. ET WIESNER K., *Simplicity and complexity*, Phys. World, 23 (2010), p. 36.

[69] CRUTCHFIELD J. P. ET YOUNG K., *Inferring statistical complexity*, Phys. Rev. Lett., 63 (1989), p. 105.

[70] CYBENKO G., *Approximation by superpositions of a sigmoidal function*, Math. Control. Signals, Syst., 2 (1989), p. 303–314.

[71] DALEY D. J. ET KENDALL D. G., *Epidemics and Rumours*, Nature, 204 (1964), p. 1118–1118.

[72] ——, *Stochastic Rumours*, IMA J. Appl. Math., 1 (1965), p. 42–55.

[73] DAVIS P. J., *Interpolation and Approximation*, Dover, 1975.

[74] DE SILVA B. M., HIGDON D. M., BRUNTON S. L. ET KUTZ J. N., *Discovery of physics from data : Universal laws and discrepancies*, Front. Artif. Intell., 3 (2020), p. 25.

[75] DECELLE A., KRZAKALA F., MOORE C. ET ZDEBOROV\'A L., *Inference and phase transitions in the detection of modules in sparse networks*, Phys. Rev. Lett., 107 (2011), p. 065701.

[76] DELSOLE T. ET TIPPETT M. K., *Predictability : Recent insights from information theory*, Rev. Geophys., 45 (2007), p. RG4002.

[77] DESROSIERS P., LABRECQUE S., TREMBLAY M., BÉLANGER M., DE DORLODOT B. ET CÔTÉ D. C., *Network inference from functional experimental data (Conference Presentation)*, dans Clinical and Translational Neurophotonics ; Neural Imaging and Sensing ; and Optogenetics and Optical Manipulation, vol. 9690, International Society for Optics and Photonics, SPIE, 2016, p. 969019.

[78] DESTEXHE A. ET SEJNOWSKI T. J., *The wilson–cowan model, 36 years later*, Biol. Cybern., 101 (2009), p. 1.

[79] DODDS P. S. ET WATTS D. J., *Universal Behavior in a Generalized Model of Contagion*, Phys. Rev. Lett., 92 (2004), p. 218701.

[80] DOMB C., *Phase Transitions and Critical Phenomena*, Elsevier, 2000.

[81] DOSOVITSKIY A., BEYER L., KOLESNIKOV A., WEISSENBORN D., ZHAI X., UNTERTHINER T., DEHGHANI A., MINDERER M., HEIGOLD G., GELLY S., USZKOREIT J. ET HOULSBY N., *An image is worth 16x16 words : Transformers for image recognition at scale*, arXiv preprint, arXiv :2010.11929 (2021).

[82] DUTTA R., MIRA A. ET ONNELA J.-P., *Bayesian inference of spreading processes on networks*, Proc. R. Soc. A, 474 (2018), p. 20180129.

[83] EDWARDS D., *Introduction to Graphical Modelling*, Springer Science & Business Media, 2012.

[84] EICHELSBACHER P. ET GANESH A., *Bayesian inference for markov chains*, J. Appl. Probab., 39 (2002), p. 91–99.

[85] EICHLER M., *Causal Inference in Time Series Analysis*, Wiley Online Library, 2012.

[86] ERDŐS P. ET RÉNYI A., *On the evolution of random graphs*, Publ. Math. Inst. Hung. Acad. Sci, 5 (1960), p. 17.

[87] FEDER M. ET MERHAV N., *Relations between entropy and error probability*, IEEE Trans. Inf. Theory, 40 (1994), p. 259.

[88] FELDMAN D. P. ET CRUTCHFIELD J. P., *Measures of statistical complexity : Why ?*, Phys. Lett. A, 238 (1998), p. 244–252.

[89] FERREIRA S. C., CASTELLANO C. ET PASTOR-SATORRAS R., *Epidemic thresholds of the susceptible-infected-susceptible model on networks : A comparison of numerical and theoretical results*, Phys. Rev. E, 86 (2012), p. 041125.

[90] FERREIRA CACERES M. M., SOSA J. P., LAWRENCE J. A., SESTACOVSCHI C., TIDD-JOHNSON A., RASOOL M. H. U., GADAMIDI V. K., OZAIR S., PANDAV K., CUEVAS-LOU C., PARRISH M., RODRIGUEZ I. ET FERNANDEZ J. P., *The impact of misinformation on the covid-19 pandemic*, AIMS Public Health, 9 (2022), p. 262–277.

[91] FEY M. ET LENSSEN J. E., *Fast graph representation learning with pytorch geometric*, arXiv preprint, arXiv :1903.02428 (2019).

[92] FORNITO A., ZALESKY A. ET BREAKSPEAR M., *The connectomics of brain disorders*, Nat. Rev. Neurosci., 16 (2015), p. 159–172.

[93] FORRESTER M., CROFTS J. J., SOTIROPOULOS S. N., COOMBES S. ET O'DEA R. D., *The role of node dynamics in shaping emergent functional connectivity patterns in the brain*, Network Neuroscience, 4 (2020), p. 467.

[94] FORTUNATO S., *Community detection in graphs*, Phys. Rep., 486 (2010), p. 75–174.

[95] FOSDICK B. K., LARREMORE D. B., NISHIMURA J. ET UGANDER J., *Configuring random graph models with fixed degree sequences*, SIAM Rev., 60 (2018), p. 315–355.

[96] FOUT A., BYRD J., SHARIAT B. ET BEN-HUR A., *Protein interface prediction using graph convolutional networks*, dans Adv. Neural Inf. Process. Syst. 30, 2017, p. 6530–6539.

[97] FRITSCH F. N. ET BUTLAND J., *A method for constructing local monotone piecewise cubic interpolants*, SIAM J. Sci. Stat. Comput., 5 (1984), p. 300.

[98] FRITZ C., DORIGATTI E. ET RÜGAMER D., *Combining graph neural networks and spatio-temporal disease models to predict COVID-19 cases in germany*, arXiv preprint, arXiv :2101.00661 (2021).

[99] GAO J., BARZEL B. ET BARABÁSI A.-L., *Universal resilience patterns in complex networks*, Nature, 530 (2016), p. 307.

[100] GAO J., SHARMA R., QIAN C., GLASS L. M., SPAEDER J., ROMBERG J., SUN J. ET XIAO C., *STAN : Spatio-temporal attention network for pandemic prediction using real-world evidence*, J. Am. Med. Inform. Assoc, 28 (2021), p. 733–743.

[101] GARCÍA-PÉREZ G., ALLARD A., SERRANO M. A. ET BOGUÑÁ M., *Mercator : uncovering faithful hyperbolic embeddings of complex networks*, New J. Phys., 21 (2019), p. 123033.

[102] GARLAND J., JAMES R. ET BRADLEY E., *Model-free quantification of time-series predictability*, Phys. Rev. E, 90 (2014), p. 052910.

[103] GELMAN A., CARLIN J. B., STERN H. S., DUNSON D. B., VEHTARI A. ET RUBIN D. B., *Bayesian Data Analysis*, CRC Press, 1995.

[104] GÉNOIS M. ET BARRAT A., *Can co-location be used as a proxy for face-to-face contacts ?*, EPJ Data Sci., 7 (2018), p. 11.

[105] GENTLE J. E., *Random Number Generation and Monte Carlo Methods*, Springer New York, NY, 2003.

[106] GERSTNER W., KISTLER W. M., NAUD R. ET PANINSKI L., *Neuronal Dynamics : From Single Neurons to Networks and Models of Cognition*, Cambridge University Press, 2014.

[107] GHASEMIAN A., ZHANG P., CLAUSET A., MOORE C. ET PEEL L., *Detectability thresholds and optimal algorithms for community structure in dynamic networks*, Phys. Rev. X, 6 (2016), p. 031005.

[108] GIANNAKIS D., MAJDA A. J. ET HORENKO I., *Information theory, model error, and predictive skill of stochastic models for complex nonlinear systems*, Physica D, 241 (2012), p. 1735–1752.

[109] GILMER J., SCHOENHOLZ S. S., RILEY P. F., VINYALS O. ET DAHL G. E., *Neural message passing for quantum chemistry*, dans Proceedings of the 34th International Conference on Machine Learning, Precup D. et Teh Y. W., éds., vol. 70 de Proceedings of Machine Learning Research, PMLR, 06–11 Aug 2017, p. 1263–1272.

[110] GIRVAN M. ET NEWMAN M. E. J., *Community structure in social and biological networks*, Proc. Natl. Acad. Sci., 99 (2002), p. 7821–7826.

[111] GIVEON A., PORRATI M. ET RABINOVICI E., *Target space duality in string theory*, Phys. Rep., 244 (1994), p. 77–202.

[112] GLAUBER R. J., *Time-Dependent Statistics of the Ising Model*, J. Math. Phys., 4 (1963), p. 294–307.

[113] GLEESON J. P., *High-Accuracy Approximation of Binary-State Dynamics on Networks*, Phys. Rev. Lett., 107 (2011), p. 068701.

[114] ——, *Binary-state dynamics on complex networks : Pair approximation and beyond*, Phys. Rev. X, 3 (2013), p. 021004.

[115] GLOBAL HEALTH , *Global.health : A data science initiative*, 2020.

[116] GOBIERNO DE ESPAÑA , *Observatorio del transporte y la logística en españa*, 2018.

[117] ——, *COVID-19 en españa*, 2020.

[118] GOLDBERG D. S. ET ROTH F. P., *Assessing experimentally derived interactions in a small world*, Proc. Natl. Acad. Sci. U.S.A., 100 (2003), p. 4372–4376.

[119] GOODFELLOW I., BENGIO Y. ET COURTVILLE A., *Deep Learning*, MIT Press, 2016.

[120] GRASSLY N. C. ET FRASER C., *Mathematical models of infectious disease transmission*, Nat. Rev. Microbiol., 6 (2008), p. 477–487.

[121] GROSS T. ET BLASIUS B., *Adaptive coevolutionary networks : A review*, J. R. Soc. Interface, 5 (2008), p. 259–271.

[122] GU S.-J., SUN C.-P. ET LIN H.-Q., *Universal role of correlation entropy in critical phenomena*, J. Phys. A, 41 (2007), p. 025002.

[123] GUI J., CHEN T., ZHANG J., CAO Q., SUN Z., LUO H. ET TAO D., *A survey on self-supervised learning : Algorithms, applications, and future trends*, arXiv preprint, arXiv :2301.05712 (2024).

[124] GUIMERÀ R. ET SALES-PARDO M., *Missing and spurious interactions and the reconstruction of complex networks*, Proc. Natl. Acad. Sci. U.S.A., 106 (2009), p. 22073–22078.

[125] HACKING I., *An Introduction to Probability and Inductive Logic*, Cambridge University Press, 2001.

[126] HAMILTON W. L., YING R. ET LESKOVEC J., *Inductive representation learning on large graphs*, dans Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, p. 1025–1035.

[127] ——, *Representation learning on graphs : Methods and applications*, arXiv preprint, arXiv :1709.05584 (2017).

[128] HAMMOND D. K., VANDERGHEYNST P. ET GRIBONVAL R., *Wavelets on graphs via spectral graph theory*, Appl. Comput. Harmon. Anal., 30 (2011), p. 129–150.

[129] HASTINGS W. K., *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), p. 97.

[130] HÉBERT-DUFRESNE L. ET ALTHOUSE B. M., *Complex dynamics of synergistic coinfections on realistically clustered networks*, Proc. Natl. Acad. Sci. USA, 112 (2015), p. 10551–10556.

[131] HÉBERT-DUFRESNE L., NOËL P.-A., MARCEAU V., ALLARD A. ET DUBÉ L. J., *Propagation dynamics on networks featuring complex topologies*, Phys. Rev. E, 82 (2010), p. 036115.

[132] HÉBERT-DUFRESNE L., SCARPINO S. V. ET YOUNG J.-G., *Macroscopic patterns of interacting contagions are indistinguishable from social reinforcement*, Nat. Phys., 16 (2020), p. 426–431.

[133] HÉBERT-DUFRESNE L., SCARPINO S. V. ET YOUNG J.-G., *Macroscopic patterns of interacting contagions are indistinguishable from social reinforcement*, Nat. Phys., 16 (2020), p. 426–431.

[134] HETHCOTE H. W., *The mathematics of infectious diseases*, SIAM Rev., 42 (2000), p. 599–653.

[135] HINNE M., HESKES T., BECKMANN C. F. ET VAN GERVEN M. A. J., *Bayesian inference of structural brain networks*, NeuroImage, 66 (2013), p. 543–552.

[136] HLINKA J. ET COOMBES S., *Using computational models to relate structural and functional brain connectivity*, European Journal of Neuroscience, 36 (2012), p. 2137.

[137] HOLLEY R. A. ET LIGGETT T. M., *Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model*, Ann. Probab., 3 (1975), p. 643–663.

[138] HU C. K., *Percolation, clusters, and phase transitions in spin models*, Phys. Rev. B, 29 (1984), p. 5103.

[139] IACOPINI I., PETRI G., BARRAT A. ET LATORA V., *Simplicial models of social contagion*, Nat. Commun., 10 (2019), p. 2485.

[140] INSTITUTO NACIONAL DE ESTADÍSTICA , *Instituto nacional de estadística*, 2020.

[141] JAYNES E. T., *Probability Theory : The Logic of Science*, Cambridge University Press, Cambridge, 2003.

[142] JENSEN F. V., *An Introduction to Bayesian Networks*, UCL press, 1996.

[143] JERDEE M., KIRKLEY A. ET NEWMAN M. E. J., *Mutual information and the encoding of contingency tables*, Phys. Rev. E, 110 (2024), p. 064306.

[144] JIN W., BARZILAY R. ET JAAKKOLA T., *Junction tree variational autoencoder for molecular graph generation*, arXiv preprint, arXiv :1802.04364 (2019).

[145] JOHNSON S., TORRES J. J., MARRO J. ET MUNOZ M. A., *Entropic origin of disassortativity in complex networks*, Phys. Rev. Lett., 104 (2010), p. 108702.

[146] JONES P. W. ET SMITH P., *Stochastic Processes : An Introduction*, Chapman and Hall/CRC, 2017.

[147] KAPOOR A., BEN X., LIU L., PEROZZI B., BARNES M., BLAIS M. ET O'BANION S., *Examining COVID-19 forecasting using spatio-temporal graph neural networks*, arXiv preprint, arXiv :2007.03113 (2020).

[148] KARRER B. ET NEWMAN M. E. J., *Stochastic blockmodels and community structure in networks*, Phys. Rev. E, 83 (2011), p. 016107.

[149] KARRER B., NEWMAN M. E. J. ET ZDEBOROVÁ L., *Percolation on sparse networks*, Phys. Rev. Lett., 113 (2014), p. 208702.

[150] KASS R. E. ET RAFTERY A. E., *Bayes factors*, J. Am. Stat. Assoc., 90 (1995), p. 773–795.

[151] KELLY F. P., *Reversibility and Stochastic Networks*, Cambridge University Press, 2011.

[152] KERMACK W. O. ET MCKENDRICK A. G., *A contribution to the mathematical theory of epidemics*, Proc. R. Soc. A, 115 (1927), p. 700–721.

[153] KHALEDI-NASAB A., KROMER J. A. ET TASS P. A., *Long-Lasting Desynchronization of Plastic Neural Networks by Random Reset Stimulation*, Front. Physiol., 11 (2021), p. 622620.

[154] KINGMA D. P. ET BA J., *Adam : A method for stochastic optimization*, arXiv preprint, arXiv :1412.6980 (2014).

[155] KIPF T., FETAYA E., WANG K.-C., WELLING M. ET ZEMEL R., *Neural relational inference for interacting systems*, dans Proceedings of the 35th International Conference on Machine Learning, vol. 80 de Proceedings of Machine Learning Research, PMLR, 2018, p. 2688–2697.

[156] KIPF T. N. ET WELLING M., *Semi-supervised classification with graph convolutional networks*, arXiv preprint, arXiv :1609.02907 (2016).

[157] KISS I. Z., MILLER J. C. ET SIMON P. L., *Mathematics of Epidemics on Networks*, Springer, 2017.

[158] KLEEMAN R., *Information theory and dynamical system predictability*, Entropy, 13 (2011), p. 612.

[159] KNUTH D. E., *Big Omicron and Big Omega and Big Theta*, SIGACT News, (1976), p. 18.

[160] KRAMER M. A., EDEN U. T., CASH S. S. ET KOLACZYK E. D., *Network inference with confidence from multivariate time series*, Phys. Rev. E, 79 (2009), p. 061916.

[161] KRAUSE A., SINGH A. ET GUESTRIN C., *Near-Optimal Sensor Placements in Gaussian Processes : Theory, Efficient Algorithms and Empirical Studies*, J. Mach. Learn. Res., 9 (2008), p. 235–284.

[162] KREBS V., *The political books network*, 2004. En ligne ; visité le 28 octobre 2024.

[163] KRIOUKOV D., PAPADOPOULOS F., KITSAK M., VAHDAT A. ET BOGUÑÁ M., *Hyperbolic geometry of complex networks*, Phys. Rev. E, 82 (2010), p. 036106.

[164] KRIZHEVSKY A., SUTSKEVER I. ET HINTON G. E., *Imagenet classification with deep convolutional neural networks*, dans Advances in Neural Information Processing Systems, Pereira F., Burges C., Bottou L. et Weinberger K., éds., vol. 25, Curran Associates, Inc., 2012.

[165] KUHN T. S., *The Structure of Scientific Revolutions*, University of Chicago Press, Chicago, 1962.

[166] KUTZ J. N., *Deep learning in fluid dynamics*, J. Fluid Mech., 814 (2017), p. 1–4.

[167] LAKSHMIKANTHAM V. ET LEELA S., *Differential and Integral Inequalities-Ordinary Differential Equations, vol. I*, Academic Press, 1969.

[168] LANDAU L., *The Theory of Phase Transitions*, Nature, 138 (1936), p. 840–841.

[169] ——, *On the theory of phase transitions*, Zh. Eksp. Teor. Fiz., 7 (1937), p. 19–32.

[170] LATORA V., NICOSIA V. ET RUSSO G., *Complex Networks : Principles, Methods and Applications*, Cambridge University Press, 2017.

[171] LAURENCE E., DOYON N., DUBÉ L. J. ET DESROSIERS P., *Spectral dimension reduction of complex dynamical networks*, Phys. Rev. X, 9 (2019), p. 011042.

[172] LAURENCE E., MURPHY C., ST-ONGE G., ROY-POMERLEAU X. ET THIBEAULT V., *Detecting structural perturbations from time series using deep learning*, arXiv preprint, arXiv :2006.05232 (2020).

[173] LAZER D., PENTLAND A., ADAMIC L., ARAL S., BARABASI A., BREWER D., CHRISTAKIS N., CONTRACTOR N., FOWLER J., GUTMANN M., JEBARA T., KING G., MACY M., ROY D. ET VAN ALSTYNE M., *Computational social science*, Science, 323 (2009), p. 721–3.

[174] LE ROUX N. ET BENGIO Y., *Representational power of restricted boltzmann machines and deep belief networks*, Neural Comput., 20 (2008), p. 1631–1649.

[175] LECUN Y., BOTTOU L., BENGIO Y. ET HAFFNER P., *Gradient-based learning applied to document recognition*, Proc. IEEE, 86 (1998), p. 2278–2324.

[176] LEGARE A., LEMIEUX M., BOILY V., POULIN S., LEGARE A., DESROSIERS P. ET DE KONINCK P., *Structural and genetic determinants of zebrafish functional brain networks*, bioRxiv, (2024), p. 2024–12.

[177] LÉGARÉ A., LEMIEUX M., DESROSIERS P. ET DE KONINCK P., *Zebrafish brain atlases : a collective effort for a tiny vertebrate brain*, Neurophotonics, 10 (2023), p. 044409.

[178] LEHMANN E. L., *Elements of Large-Sample Theory*, Springer-Verlag, 1 éd., 1999.

[179] LEHMANN S. ET AHN Y.-Y., éds., *Complex Spreading Phenomena in Social Systems*, Springer, 2018.

[180] LEVIN D. A., PERES Y. ET WILMER E. L., *Markov Chains and Mixing Times*, American Mathematical Society, 2006.

[181] LIN A., YANG R., DORKENWALD S., MATSLIAH A., STERLING A. R., SCHLEGEL P., YU S., MCKELLAR C. E., COSTA M., EICHLER K. *et al.*, *Network statistics of the whole-brain connectome of drosophila*, Nature, 634 (2024), p. 153–165.

[182] LIN T.-Y., GOYAL P., GIRSHICK R., HE K. ET DOLLÁR P., *Focal loss for dense object detection*, arXiv preprint, arXiv :1708.02002 (2018).

[183] LIU L., JIANG H., HE P., CHEN W., LIU X., GAO J. ET HAN J., *On the variance of the adaptive learning rate and beyond*, arXiv preprint, arXiv :1908.03265 (2019).

[184] LIZOTTE S., YOUNG J.-G. ET ALLARD A., *Hypergraph reconstruction from uncertain pairwise observations*, Sci. Rep., 13 (2023), p. 21364.

[185] ——, *Hypergraph reconstruction from uncertain pairwise observations*, Sci. Rep., 13 (2023), p. 21364.

[186] LORENZ E. N., *Deterministic nonperiodic flow*, J. Atmos. Sci., 20 (1963), p. 130–141.

[187] LÜ L. ET ZHOU T., *Link prediction in complex networks : A survey*, Physica A Stat., 390 (2011), p. 1150–1170.

[188] LU Z., PU H., WANG F., HU Z. ET WANG L., *The expressive power of neural networks : A view from the width*, dans Advances in Neural Information Processing Systems, Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S. et Garnett R., éds., vol. 30, Curran Associates, Inc., 2017.

[189] LUSCH B., KUTZ J. N. ET BRUNTON S. L., *Deep learning for universal linear embeddings of nonlinear dynamics*, Nat. Commun., 9 (2018), p. 1–10.

[190] LYNN C. W. ET BASSETT D. S., *The physics of brain network structure, function and control*, Nat. Rev. Phys., 1 (2019), p. 318.

[191] MARCEAU V., NOËL P.-A., HÉBERT-DUFRESNE L., ALLARD A. ET DUBÉ L. J., *Adaptive networks : Coevolution of disease and topology*, Phys. Rev. E, 82 (2010), p. 036116.

[192] MARTINEZ N., *Artifacts or attributes ? effects of resolution on the little rock lake food web.*, Ecol. Monogr., 61 (1991), p. 367–392.

[193] MASTRANDREA R., FOURNET J. ET BARRAT A., *Contact patterns in a high school : A comparison between data collected using wearable sensors, contact diaries and friendship surveys*, PLoS One, 10 (2015), p. e0136497.

[194] MATSUDA H., KUDO K., NAKAMURA R., YAMAKAWA O. ET MURATA T., *Mutual information of ising systems*, Int. J. Theor. Phys., 35 (1996), p. 839–845.

[195] MAY R. M., *Simple mathematical models with very complicated dynamics*, Nature, 261 (1976), p. 459–467.

[196] MCCABE S., TORRES L., LAROCK T., HAQUE S., YANG C.-H., HARTLE H. ET KLEIN B., *netrd : A library for network reconstruction and graph distances*, J. Open Source Softw., 6 (2021), p. 2990.

[197] MCCULLOCH W. S. ET PITTS W., *A logical calculus of the ideas immanent in nervous activity*, Bull. Math. Biophys., 5 (1943), p. 115–133.

[198] MEIJERS M., ITO S. ET WOLDE P. R.T. , *Behavior of information flow near criticality*, Phys. Rev. E, 103 (2021), p. L010102.

[199] METROPOLIS N., ROSENBLUTH A. W., ROSENBLUTH M. N., TELLER A. H. ET TELLER E., *Equation of state calciations by fast computing machines*, J. Chem. Phys., 21 (1953), p. 2384.

[200] MÉZARD M. ET MONTANARI A., *Information, Physics, and Computation*, Oxford University Press, jan. 2009.

[201] MEZARD M., PARISI G. ET VIRASORO M., *Spin Glass Theory and Beyond*, vol. 9, WORLD SCIENTIFIC, nov. 1986.

[202] MINSKY M. ET SEYMOUR P., *Perceptrons : An Introduction to Computational Geometry*, MIT press, 1969.

[203] MISTRY D., LITVINOVA M., PIONTTI A.P. Y. , CHINAZZI M., FUMANELLI L., GOMES M. F. C., HAQUE S. A., LIU Q., MU K., XIONG X., HALLORAN M. E., LONGINI I. M., MERLER S., AJELLI M. ET VESPIGNANI A., *Inferring high-resolution human mixing patterns for disease modeling*, Nat. Commun., 12 (2021), p. 323.

[204] MITCHELL, T. M. , *Machine Learning*, McGraw-Hill, 1997.

[205] MONTI F., BOSCAINI D., MASCI J., RODOLÀ E., SVOBODA J. ET BRONSTEIN M. M., *Geometric deep learning on graphs and manifolds using mixture model cnns*, arXiv preprint, arXiv :1611.08402 (2016).

[206] MORENS D. M., TAUBENBERGER J. K. ET FAUCI A. S., *Predominant role of bacterial pneumonia as a cause of death in pandemic influenza : Implications for pandemic influenza preparedness*, J. Infect. Dis., 198 (2008), p. 962–970.

[207] MORRIS C., RITZERT M., FEY M., HAMILTON W. L., LENSSEN J. E., RATTAN G. ET GROHE M., *Weisfeiler and leman go neural : Higher-order graph neural networks*, arXiv preprint, arXiv :1810.02244 (2018).

[208] MORRIS J. X., SITAWARIN C., GUO C., KOKHLIKYAN N., SUH G. E., RUSH A. M., CHAUDHURI K. ET MAHLOUJIFAR S., *How much do language models memorize ?*, 2025.

[209] MURPHY C., *Mcmc sampling for the coin toss*. https://gist.github.com/charlesmurphy1/f1063fc4509214f3e67adf0ff961726e, 2025.

[210] MURPHY C., LAURENCE E. ET ALLARD A., *Deep learning of contagion dynamics on complex networks*, Nat Commun., 12 (2021), p. 4720.

[211] MURPHY C., LIZOTTE S., THIBAULT F., THIBEAULT V., DESROSIERS P. ET ALLARD A., *On the reconstruction limits of complex networks*, arXiv preprint, arXiv :2501.01437 (2024).

[212] MURPHY C., THIBEAULT V., ALLARD A. ET DESROSIERS P., *Duality between predictability and reconstructability in complex systems*, Nat. Commun., 15 (2024), p. 4478.

[213] MURPHY K. P., *Probabilistic Machine Learning : Advanced Topics*, MIT Press, 2023.

[214] MUSMECI N., BATTISTON S., CALDARELLI G., PULIGA M. ET GABRIELLI A., *Bootstrapping topological properties and systemic risk of complex networks using the fitness model*, J. Stat. Phys., 151 (2013), p. 720–734.

[215] NEAL R. M., *Annealed importance sampling*, Stat. Comput., 11 (2001), p. 125.

[216] NEMATZADEH A., FERRARA E., FLAMMINI A. ET AHN Y.-Y., *Optimal network modularity for information diffusion*, Phys. Rev. Lett., 113 (2014), p. 088701.

[217] NEWMAN M. E. J., *The structure of scientific collaboration networks*, Proc. Natl. Acad. Sci., 98 (2001), p. 404–409.

[218] ——, *Assortative mixing in networks*, Phys. Rev. Lett., 89 (2002), p. 208701.

[219] ——, *Mixing patterns in networks*, Phys. Rev. E, 67 (2003), p. 026126.

[220] ——, *Network structure from rich but noisy data*, Nat. Phys., 14 (2018), p. 542–545.

[221] ——, *Networks*, Oxford University Press, second éd., 2018.

[222] NEWMAN M. E. J., CANTWELL G. T. ET YOUNG J.-G., *Improved mutual information measure for clustering, classification, and community detection*, Phys. Rev. E, 101 (2020), p. 042304.

[223] NEWMAN M. E. J., STROGATZ S. H. ET WATTS D. J., *Random graphs with arbitrary degree distributions and their applications*, Phys. Rev. E, 64 (2001), p. 026118.

[224] NEWMAN M. E. J., WATTS D. J. ET STROGATZ S. H., *Random graph models of social networks*, Proc. Natl. Acad. Sci., 99 (2002), p. 2566–2572.

[225] NEWTON M. A. ET RAFTERY A. E., *Approximate bayesian inference with the weighted likelihood bootstrap*, J. Roy. Stat. Soc. B, 56 (1994), p. 3.

[226] NICKBAKHSH S., MAIR C., MATTHEWS L., REEVE R., JOHNSON P. C. D., THORBURN F., VON WISSMANN B., REYNOLDS A., MCMENAMIN J., GUNSON R. N. ET MURCIA P. R., *Virus-virus interactions impact the population dynamics of influenza and the common cold*, Proc. Natl. Acad. Sci. U.S.A., 116 (2019), p. 27142–27150.

[227] ORTEGA A., FROSSARD P., KOVAČEVIĆ J., MOURA J. M. F. ET VANDERGHEYNST P., *Graph signal processing : Overview, challenges, and applications*, Proc. IEEE, 106 (2018), p. 808–828.

[228] PAINCHAUD V., DOYON N. ET DESROSIERS P., *Beyond Wilson-Cowan dynamics : oscillations and chaos without inhibition*, Biol. Cybern., 116 (2022), p. in press.

[229] PAPADOPOULOS F., ALDECOA R. ET KRIOUKOV D., *Network geometry inference using common neighbors*, Phys. Rev. E, 92 (2015), p. 022807.

[230] PASTOR-SATORRAS R. ET CASTELLANO C., *Eigenvector localization in real networks and its implications for epidemic spreading*, J. Stat. Phys., 173 (2018), p. 1110–1123.

[231] PASTOR-SATORRAS R., CASTELLANO C., VAN MIEGHEM P. ET VESPIGNANI A., *Epidemic processes in complex networks*, Rev. Mod. Phys., 87 (2015), p. 925–979.

[232] PASTOR-SATORRAS R. ET VESPIGNANI A., *Epidemic spreading in scale-free networks*, Phys. Rev. Lett., 86 (2001), p. 3200.

[233] PASTORE Y PIONTTI A., PERRA N., ROSSI L., SAMAY N. ET VESPIGNANI A., *Charting the next Pandemic : Modeling Infectious Disease Spreading in the Data Science Age*, Springer, 2019.

[234] PATHAK J., HUNT B., GIRVAN M., LU Z. ET OTT E., *Model-free prediction of large spatiotemporally chaotic systems from data : A reservoir computing approach*, Phys. Rev. Lett., 120 (2018), p. 024102.

[235] PATHAK J., LU Z., HUNT B. R., GIRVAN M. ET OTT E., *Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data*, Chaos, 27 (2017), p. 121102.

[236] PEEL L., PEIXOTO T. P. ET DE DOMENICO M., *Statistical inference links data and theory in network science*, Nat. Commun., 13 (2022), p. 6794.

[237] PEIXOTO T. P., *Entropy of stochastic blockmodel ensembles*, Phys. Rev. E, 85 (2012), p. 056122.

[238] ——, *Parsimonious module inference in large networks*, Phys. Rev. Lett., 110 (2013), p. 148701.

[239] ——, *Hierarchical block structures and high-resolution model selection in large networks*, Phys. Rev. X, 4 (2014), p. 011047.

[240] ——, *Nonparametric bayesian inference of the microcanonical stochastic block model*, Phys. Rev. E, 95 (2017), p. 012317.

[241] ——, *Reconstructing networks with unknown and heterogeneous errors*, Phys. Rev. X, 8 (2018), p. 041011.

[242] ——, *Network reconstruction and community detection from dynamics*, Phys. Rev. Lett., 123 (2019), p. 128301.

[243] ——, *Revealing Consensus and Dissensus between Network Partitions*, Phys. Rev. X, 11 (2021), p. 021003.

[244] ——, *The netzschleuder network catalogue and repository*, 2023.

[245] ——, *Network reconstruction via the minimum description length principle*, arXiv preprint, arXiv :2405.01015 (2024).

[246] ——, *Scalable network reconstruction in subquadratic time*, arXiv preprint, arXiv :2401.01404 (2024).

[247] PENNEKAMP F., ILES A. C., GARLAND J., BRENNAN G., BROSE U., GAEDKE U., JACOB U., KRATINA P., MATTHEWS B., MUNCH S., NOVAK M., PALAMARA G. M., RALL B. C., ROSENBAUM B., TABI A., WARD C., WILLIAMS R., YE H. ET PETCHEY O. L., *The intrinsic predictability of ecological time series and its potential to guide forecasting*, Ecol. Monogr., 89 (2019), p. e01359.

[248] PEROZZI B., AL-RFOU R. ET SKIENA S., *DeepWalk : Online learning of social representations*, dans Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., New York, NY, USA, 2014, p. 701–710.

[249] PIETRAS B. ET DAFFERTSHOFER A., *Network dynamics of coupled oscillators and phase reduction techniques*, Phys. Rep., 819 (2019), p. 1–109.

[250] PRASSE B., ACHTERBERG M. A., MA L. ET VAN MIEGHEM P., *Network-inference-based prediction of the COVID-19 epidemic outbreak in the Chinese province Hubei*, Appl. Netw. Sci., 5 (2020), p. 35.

[251] PRASSE B. ET VAN MIEGHEM P., *Predicting network dynamics without requiring the knowledge of the interaction graph*, Proc. Natl. Acad. Sci. U.S.A., 119 (2022), p. e2205517119.

[252] PRINZ A. A., BUCHER D. ET MARDER E., *Similar network activity from disparate circuit parameters*, Nat. Neurosci., 7 (2004), p. 1345–1352.

[253] RADICCHI F. ET CASTELLANO C., *Uncertainty Reduction for Stochastic Processes on Complex Networks*, Phys. Rev. Lett., 120 (2018), p. 198301.

[254] RAFTERY A. E., *Bayesian model selection in social research*, Soc. Methodol., (1995), p. 111–163.

[255] RODRIGUES F. A., PERON T., CONNAUGHTON C., KURTHS J. ET MORENO Y., *A machine learning approach to predicting dynamical observables from network structure*, arXiv preprint, arXiv :1910.00544 (2019).

[256] RODRIGUEZ N., IZQUIERDO E. ET AHN Y.-Y., *Optimal modularity and memory capacity of neural reservoirs*, Netw. Neurosc., 3 (2019), p. 551–566.

[257] ROGERS E. M., *Diffusion of Innovations*, Free Press, 2003.

[258] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P. ET OMMER B., *High-resolution image synthesis with latent diffusion models*, arXiv preprint, arXiv :2112.10752 (2022).

[259] ROSAS F. E., MEDIANO P. A. M., JENSEN H. J., SETH A. K., BARRETT A. B., CARHART-HARRIS R. L. ET BOR D., *Reconciling emergences : An information-theoretic approach to identify causal emergence in multivariate data*, PLOS Comput. Biol., 16 (2020), p. 1–22.

[260] ROSENBLATT F., *The perceptron : a probabilistic model for information storage and organization in the brain.*, Psychol. Rev., 65 (1958), p. 386.

[261] ——, *Principles of neurodynamics : Perceptrons and the theory of brain mechanisms*, rap. tech., Cornell Aeronautical Laboratory, mars 1961. The work reported in this volume has been carried out under Contract Nonr-2381 (00) (Project PARA) at C.A.L. and Contract Nonr-401(40), at Cornell University.

[262] RUBINSTEIN R. Y. ET KROESE D. P., *Simulation and the Monte Carlo Method*, Wiley, third éd., 2016.

[263] ——, *Simulation and the monte carlo method : Third edition*, dans Simul. Monte Carlo Method, Wiley, third éd., 2016, p. 414.

[264] RUDER S., *An overview of gradient descent optimization algorithms*, arXiv preprint, arXiv :1609.04747 (2017).

[265] RUDIN W., *Real and Complex Analysis*, McGraw-Hill, 3 éd., 1986.

[266] RUSTAM F., RESHI A. A., MEHMOOD A., ULLAH S., ON B., ASLAM W. ET CHOI G. S., *COVID-19 future forecasting using supervised machine learning models*, IEEE Access, 8 (2020), p. 101489–101499.

[267] SALAKHUTDINOV R. ET LAROCHELLE H., *Efficient learning of deep boltzmann machines*, dans Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, p. 693–700.

[268] SALAKHUTDINOV R. ET MURRAY I., *On the quantitative analysis of deep belief networks*, dans Proceedings of the 25th international conference on Machine learning, 2008, p. 872–879.

[269] SALOVA A., EMENHEISER J., RUPE A., CRUTCHFIELD J. P. ET D'SOUZA R. M., *Koopman operator and its approximations for systems with symmetries*, Chaos, 29 (2019), p. 93128.

[270] SANHEDRAI H., GAO J., BASHAN A., SCHWARTZ M., HAVLIN S. ET BARZEL B., *Reviving a failed network through microscopic interventions*, Nat. Phys., 18 (2022), p. 338–349.

[271] SANZ J., XIA C.-Y., MELONI S. ET MORENO Y., *Dynamics of interacting diseases*, Phys. Rev. X, 4 (2014), p. 41005.

[272] SCARPINO S. V., ALLARD A. ET HÉBERT-DUFRESNE L., *The effect of a prudent adaptive behaviour on disease transmission*, Nat. Phys., 12 (2016), p. 1042–1046.

[273] SCARPINO S. V. ET PETRI G., *On the predictability of infectious disease outbreaks*, Nat. Commun., 10 (2019), p. 1.

[274] SCHREIBER T., *Measuring Information Transfer*, Phys. Rev. Lett., 85 (2000), p. 461–464.

[275] SERRANO M. A., KRIOUKOV D. ET BOGUÑÁ M., *Self-similarity of complex networks and hidden metric spaces*, Phys. Rev. Lett., 100 (2008), p. 078701.

[276] SERRANO S. ET SMITH N. A., *Is attention interpretable ?*, dans Proc. 57th Annu. Meet. Assoc. Comput. Linguist., Florence, Italy, juil. 2019, Association for Computational Linguistics, p. 2931–2951.

[277] SETH A. K., *Causal connectivity of evolved neural networks during behavior*, Netw. Comput. Neural Syst., 16 (2005), p. 35–54.

[278] SHAH C., DEHMAMY N., PERRA N., CHINAZZI M., BARABÁSI A.-L., VESPIGNANI A. ET YU R., *Finding patient zero : Learning contagion source with graph neural networks*, arXiv preprint, arXiv :2006.11913 (2020).

[279] SHANNON C. E., *A mathematical theory of communication*, Bell Syst. Tech. J. BELL, 27 (1948), p. 379–423.

[280] SHERRINGTON D. ET KIRKPATRICK S., *Solvable Model of a Spin-Glass*, Phys. Rev. Lett., 35 (1975), p. 1792–1796.

[281] SHUMAN D. I., NARANG S. K., FROSSARD P., ORTEGA A. ET VANDERGHEYNST P., *The emerging field of signal processing on graphs : Extending high-dimensional data analysis to networks and other irregular domains*, IEEE Signal Process. Mag., 30 (2013), p. 83–98.

[282] SIETTOS C. I. ET RUSSO L., *Mathematical modeling of infectious disease dynamics*, Virulence, 4 (2013), p. 295–306.

[283] SIMON H. A., *The architecture of complexity*, Proc. Am. Philos. Soc., 106 (1962), p. 467–482.

[284] SIMS C. A., *Macroeconomics and reality*, Econometrica, (1980), p. 1–48.

[285] SKARDING J., GABRYS B. ET K. M., *Foundations and modelling of dynamic networks using Dynamic Graph Neural Networks : A survey*, arXiv preprint, arXiv :2005.07496 (2020).

[286] SONG C., QU Z., BLUMM N. ET BARABÁSI A.-L., *Limits of predictability in human mobility*, Science, 327 (2010), p. 1018.

[287] SOOD V. ET REDNER S., *Voter model on heterogeneous graphs*, Phys. Rev. Lett., 94 (2005), p. 178701.

[288] SORIANO-PAÑOS D., LOTERO L., ARENAS A. ET GÓMEZ-GARDEÑES J., *Spreading processes in multiplex metapopulations containing different mobility networks*, Phys. Rev. X, 8 (2018), p. 031039.

[289] SPORNS O., *Structure and function of complex brain networks*, Dialogues Clin. Neurosci., 15 (2013), p. 247–262.

[290] ——, *Structure and function of complex brain networks*, Dialogues Clin. Neurosci., 15 (2013), p. 247–262.

[291] ST-ONGE G., HÉBERT-DUFRESNE L. ET ALLARD A., *Nonlinear bias toward complex contagion in uncertain transmission settings*, Proc. Nat. Acad. Sci., 121 (2024), p. e2312202121.

[292] ST-ONGE G., SUN H., ALLARD A., HÉBERT-DUFRESNE L. ET BIANCONI G., *Universal nonlinear infection kernel from heterogeneous exposure on higher-order networks*, Phys. Rev. Lett., 127 (2021), p. 158301.

[293] ST-ONGE G., SUN H., ALLARD A., HÉBERT-DUFRESNE L. ET BIANCONI G., *Universal nonlinear infection kernel from heterogeneous exposure on higher-order networks*, Phys. Rev. Lett., 127 (2021), p. 158301.

[294] ST-ONGE G., THIBEAULT V., ALLARD A., DUBÉ L. J. ET HÉBERT-DUFRESNE L., *Master equation analysis of mesoscopic localization in contagion dynamics on higher-order networks*, Phys. Rev. E, 103 (2021), p. 032301.

[295] ST-ONGE G., THIBEAULT V., ALLARD A., DUBÉ L. J. ET HÉBERT-DUFRESNE L., *Master equation analysis of mesoscopic localization in contagion dynamics on higher-order networks*, Phys. Rev. E, 103 (2021), p. 032301.

[296] ST-ONGE G., THIBEAULT V., ALLARD A., DUBÉ L. J. ET HÉBERT-DUFRESNE L., *Social confinement and mesoscopic localization of epidemics on networks*, Phys. Rev. Lett., 126 (2021), p. 098301.

[297] ST-ONGE G., YOUNG J.-G., LAURENCE E., MURPHY C. ET DUBÉ L. J., *Phase transition of the susceptible-infected-susceptible dynamics on time-varying configuration model networks*, Phys. Rev. E, 97 (2018), p. 022305.

[298] STANLEY H. E., *Introduction to Phase Transitions and Critical Phenomena*, Oxford University Press, 1987.

[299] STEINMETZ N., PACHITARIU M., STRINGER C., CARANDINI M. ET HARRIS K., *Eight-probe Neuropixels recordings during spontaneous behaviors*, 3 2019.

[300] STRINGER C., PACHITARIU M., STEINMETZ N., BAI REDDY C., CARANDINI M. ET HARRIS K. D., *Spontaneous behaviors drive multidimensional, brainwide activity*, Science, 364 (2019), p. eaav7893.

[301] STROGATZ S. H., *Exploring complex networks*, Nature, 410 (2001), p. 268–276.

[302] STROGATZ S. H., *Nonlinear Dynamics and Chaos : With Applications to Physics, Biology, Chemistry, and Engineering*, CRC Press, 2018.

[303] SUPEKAR K., MENON V., RUBIN D., MUSEN M. ET GREICIUS M. D., *Network analysis of intrinsic functional brain connectivity in alzheimer's disease*, PLOS Comput. Biol., 4 (2008), p. e1000100.

[304] THIBEAULT V., ALLARD A. ET DESROSIERS P., *The low-rank hypothesis of complex systems*, Nat. Phys., (2024).

[305] THIBEAULT V., ST-ONGE G., DUBÉ L. J. ET DESROSIERS P., *Threefold way to the dimension reduction of dynamics on networks : An application to synchronization*, Phys. Rev. Research, 2 (2020), p. 043215.

[306] THIERRIN F. C., ALAJAJI F. ET LINDER T., *On the rényi cross-entropy*, arXiv preprint, arXiv :2206.14329 (2022).

[307] TIAN J., LIU Y.-C., GLASER N., HSU Y.-C. ET KIRA Z., *Posterior re-calibration for imbalanced datasets*, dans Advances in Neural Information Processing Systems, Larochelle H., Ranzato M., Hadsell R., Balcan M. et Lin H., éds., vol. 33, Curran Associates, Inc., 2020, p. 8101–8113.

[308] TIAN Y., LUTHRA I. ET ZHANG X., *Forecasting COVID-19 cases using machine learning models*, medRxiv, (2020).

[309] TIBSHIRANI R., *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. B, 58 (1996), p. 267–288.

[310] HEUVEL M. P.V. D. ET SPORNS O., *A cross-disorder connectome landscape of brain dysconnectivity*, Nat. Rev. Neurosci., 20 (2019), p. 435–446.

[311] VAN MIEGHEM P. ET CATOR E., *Epidemics in networks with nodal self-infection and the epidemic threshold*, Phys. Rev. E, 86 (2012), p. 016116.

[312] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. ET POLOSUKHIN I., *Attention is all you need*, dans Adv. Neural Inf. Process. Syst. 30 (NIPS 2017), 2017, p. 5998–6008.

[313] VEGUÉ M., THIBEAULT V., DESROSIERS P. ET ALLARD A., *Dimension reduction of dynamics on modular and heterogeneous directed networks*, PNAS nexus, 2 (2023), p. pgad150.

[314] VELIČKOVIĆ P., CUCURULL G., CASANOVA A., ROMERO A., LIÒ P. ET BENGIO Y., *Graph attention networks*, arXiv preprint, arXiv :1710.10903 (2018).

[315] VESPIGNANI A., *Twenty years of network science*, Nature, 558 (2018).

[316] VESPIGNANI A., TIAN H., DYE C., LLOYD-SMITH J. O., EGGO R. M., SHRESTHA M., SCARPINO S. V., GUTIERREZ B., KRAEMER M. U. G., WU J., LEUNG K. ET LEUNG G. M., *Modelling covid-19*, Nat. Rev. Phys., 2 (2020), p. 279–281.

[317] VIBOUD C. ET VESPIGNANI A., *The future of influenza forecasts*, Proc. Natl. Acad. Sci. U.S.A., 116 (2019), p. 2802–2804.

[318] VISWANATH B., MISLOVE A., CHA M. ET GUMMADI K. P., *On the evolution of user interaction in facebook*, dans Proceedings of the 2nd ACM Workshop on Online Social Networks, New York, NY, USA, 2009, Association for Computing Machinery, p. 37–42.

[319] VOITALOV I., VAN DER HOORN P., VAN DER HOFSTAD R. ET KRIOUKOV D., *Scale-free networks well done*, Phys. Rev. Res., 1 (2019), p. 33034.

[320] WANG Q., XIE S., WANG Y. ET ZENG D., *Survival-convolution models for predicting COVID-19 cases and assessing effects of mitigation strategies*, Front. Public Heal., 8 (2020), p. 325.

[321] WANG Y., JOSHI T., ZHANG X.-S., XU D. ET CHEN L., *Inferring gene regulatory networks from multiple microarray datasets*, Bioinformatics, 22 (2006), p. 2413–2420.

[322] WATTS D. J., *A simple model of global cascades on random networks*, Proc. Natl. Acad. Sci. U.S.A., 99 (2002), p. 5766.

[323] WATTS D. J. ET STROGATZ S. H., *Collective dynamics of 'small-world' networks*, Nature, 393 (1998).

[324] WEAVER W., *Science and complexity*, Am. Sci., 36 (1948), p. 536–544.

[325] WEI W. W. S., *Multivariate Time Series Analysis and Applications*, John Wiley & Sons, 2018.

[326] WEISFEILER B. ET LEHMAN A. A., *The reduction of a graph to canonical form and the algebra which appears therein*, Nauchno-Technicheskaya Informatsia, 2 (1968), p. 12–16.

[327] WILSON H. R. ET COWAN J. D., *Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons*, Biophys. J., 12 (1972), p. 1–24.

[328] WILSON H. R. ET COWAN J. D., *Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons*, Biophys. J., 12 (1972), p. 1.

[329] WOLBERG G. ET ALFY I., *An energy-minimization framework for monotonic cubic spline interpolation*, J. Comput. Appl. Math., 143 (2002), p. 145.

[330] WORLDOMETER , *COVID-19 Coronavirus Pandemic*, 2022. Dernière mise à jour : 13 avril 2024.

[331] WU F. Y., *The potts model*, Rev. Mod. Phys., 54 (1982), p. 235–268.

[332] XIE W., LEWIS P. O., FAN Y., KUO L. ET CHEN M.-H., *Improving marginal likelihood estimation for bayesian phylogenetic model selection*, Syst. Biol., 60 (2011), p. 150.

[333] XU K., HU W., LESKOVEC J. ET JEGELKA S., *How powerful are graph neural networks ?*, arXiv preprint, arXiv :1810.00826 (2018).

[334] YANG L., ZHANG Z., SONG Y., HONG S., XU R., ZHAO Y., ZHANG W., CUI B. ET YANG M.-H., *Diffusion models : A comprehensive survey of methods and applications*, ACM Comput. Surv., 56 (2023).

[335] YAO J. ET NELSON K. E., *An unconditionally monotone C^2 quartic spline method with nonoscillation derivatives*, Advances in Pure Mathematics, 8 (2018), p. 25.

[336] YEN T. C. ET LARREMORE D. B., *Community detection in bipartite networks with stochastic block models*, Phys. Rev. E, 102 (2020), p. 032309.

[337] YING R., HE R., CHEN K., EKSOMBATCHAI P., HAMILTON W. L. ET LESKOVEC J., *Graph convolutional neural networks for web-scale recommender systems*, dans Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 2018, Association for Computing Machinery, p. 974–983.

[338] YOUNG J.-G., *Inférence et réseaux complexes*, thèse de doctorat, Université Laval, 2018.

[339] YOUNG J.-G., CANTWELL G. T. ET NEWMAN M. E. J., *Bayesian inference of network structure from unreliable data*, J. Complex Netw., 8 (2020), p. cnaa046.

[340] YOUNG J.-G., DESROSIERS P., L. HÉBERT-DUFRESNE , LAURENCE E. ET DUB\'E L. J., *Finite-size analysis of the detectability limit of the stochastic block model*, Phys. Rev. E, 95 (2017), p. 062304.

[341] YOUNG J.-G., PETRI G. ET PEIXOTO T. P., *Hypergraph reconstruction from network data*, Commun. Phys., 4 (2021), p. 135.

[342] YOUNG J.-G., VALDOVINOS F. S. ET NEWMAN M. E. J., *Reconstruction of plant–pollinator networks from observational data*, Nat. Commun., 12 (2021), p. 3911.

[343] ZACHARY W. W., *An information flow model for conflict and fission in small groups*, J. Anthropol. Res., 33 (1977), p. 452–473.

[344] ZHANG Z., CUI P. ET ZHU W., *Deep learning on graphs : A survey*, arXiv preprint, arXiv :1812.04202 (2018).

[345] ZHANG Z., ZHAO Y., LIU J., WANG S., TAO R., XIN R. ET ZHANG J., *A general deep learning framework for network reconstruction and dynamics learning*, Appl. Netw. Sci., 4 (2019), p. 110.

[346] ZHOU G., *Mixed hamiltonian monte carlo for mixed discrete and continuous variables*, dans Advances in Neural Information Processing Systems, Larochelle H., Ranzato M., Hadsell R., Balcan M. et Lin H., éds., vol. 33, Curran Associates, Inc., 2020, p. 17094–17104.

[347] ZHOU J., CUI G., HU S., ZHANG Z., YANG C., LIU Z., WANG L., LI C. ET SUN M., *Graph neural networks : A review of methods and applications*, AI Open, 1 (2020), p. 57–81.

[348] ZHOU J., CUI G., ZHANG Z., YANG C., LIU Z., WANG L., LI C. ET SUN M., *Graph neural networks : A review of methods and applications*, arXiv preprint, arXiv :1812.08434 (2018).

[349] ZITNIK M., AGRAWAL M. ET LESKOVEC J., *Modeling polypharmacy side effects with graph convolutional networks*, Bioinformatics, 34 (2018), p. i457–i466.