Probabilistic hyperbolic embedding of networks

Combining network geometry with Bayesian inference

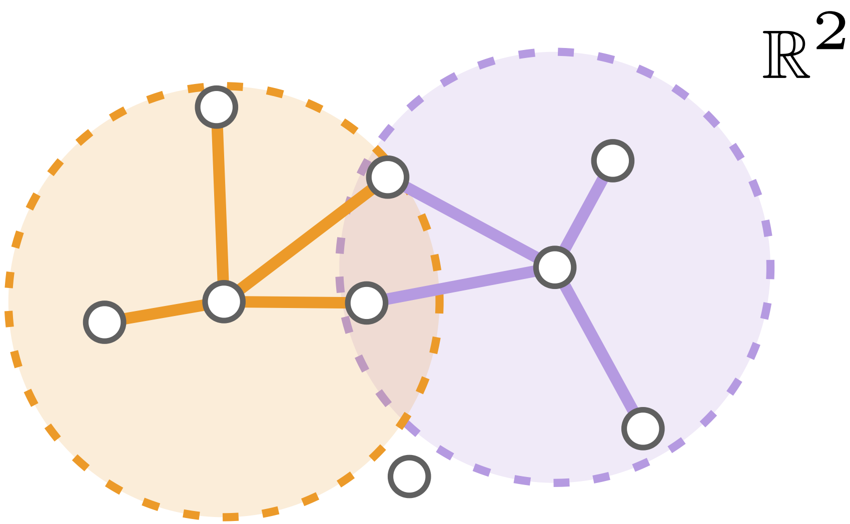
Simon Lizotte

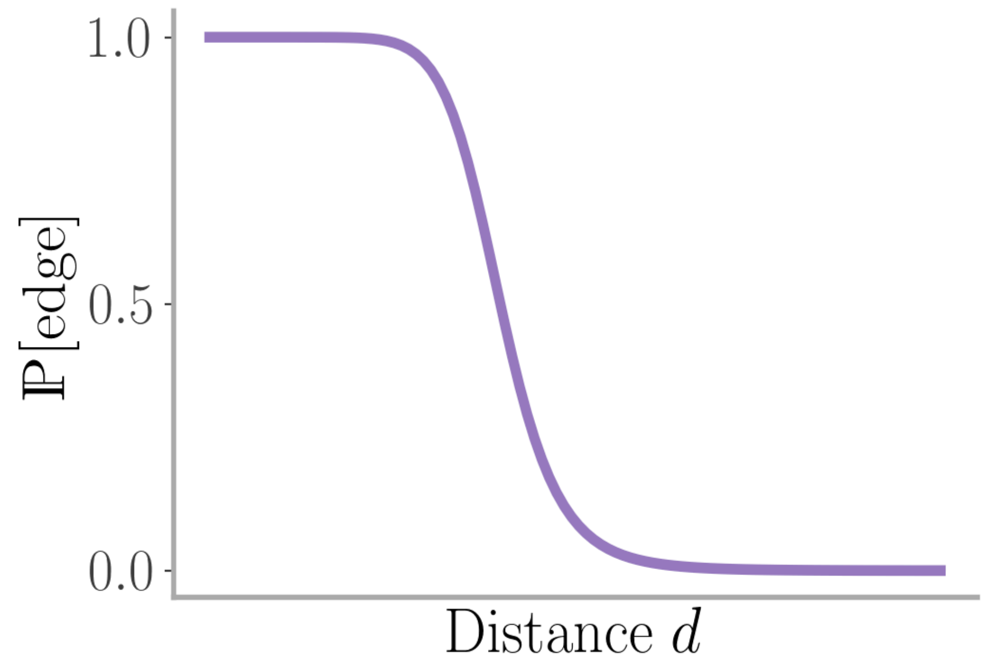
2025-09-01

Creating edges requires a cost

Creating edges requires a cost

Vertices are placed in metric space. The edge cost **increases with its length**.

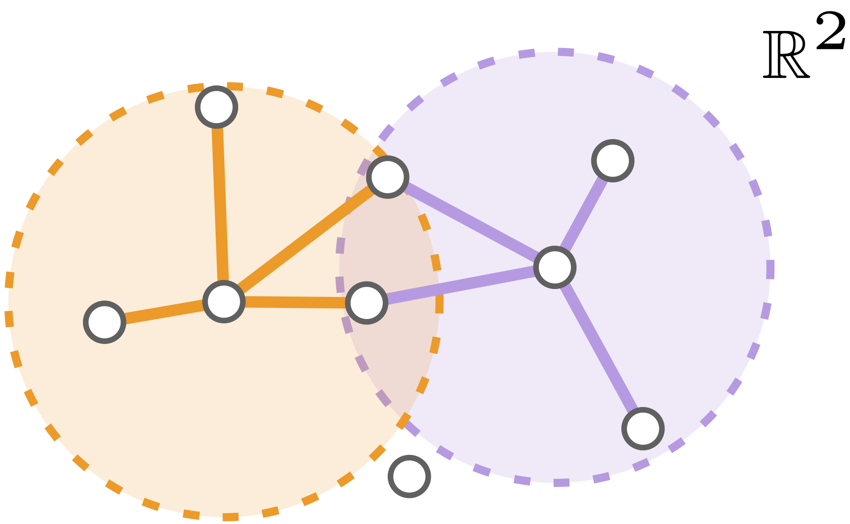


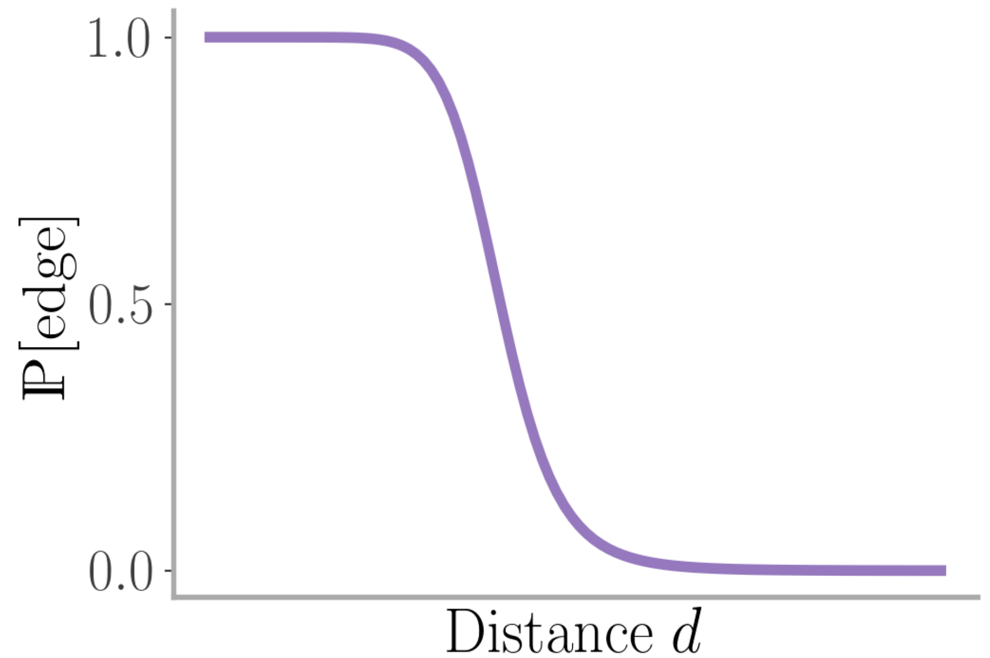


Creating edges requires a cost

Vertices are placed in metric space. The edge cost **increases with its length**.

The metric space can be physical (e.g. transportation network, brain network) or not.

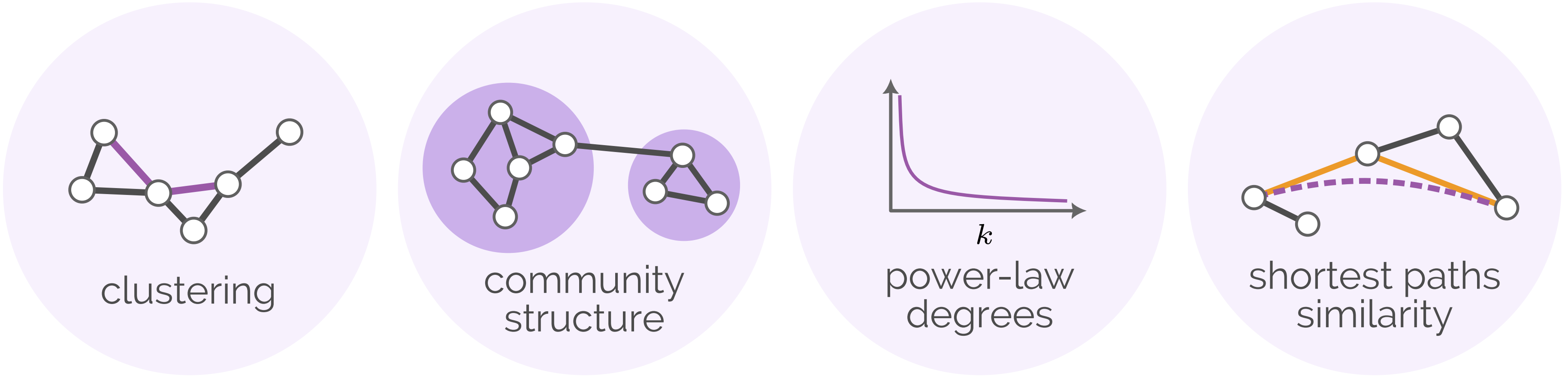




Latent hyperbolic space

Graphs obtained from hyperbolic space reproduce many empirically observed properties.

- Krioukov, D., et al. *Hyperbolic geometry of complex networks*. Phys. Rev. E **82**, 036106 (2010).
- Zuev, K., et al. *Emergence of Soft Communities from Geometric Preferential Attachment*. Sci. Rep. **5**, 9421 (2015).
- Krioukov, D., et al. *Clustering Implies Geometry in Networks*. Phys. Rev. Lett. **116**, 208302 (2016).
- Faqeeh, A., et al. *Characterizing the Analogy Between Hyperbolic Embedding and Community Structure of Complex Networks*. Phys. Rev. Lett. **121**, 098301 (2018).



... and others!

$\mathbb{S}^1$ random graph model

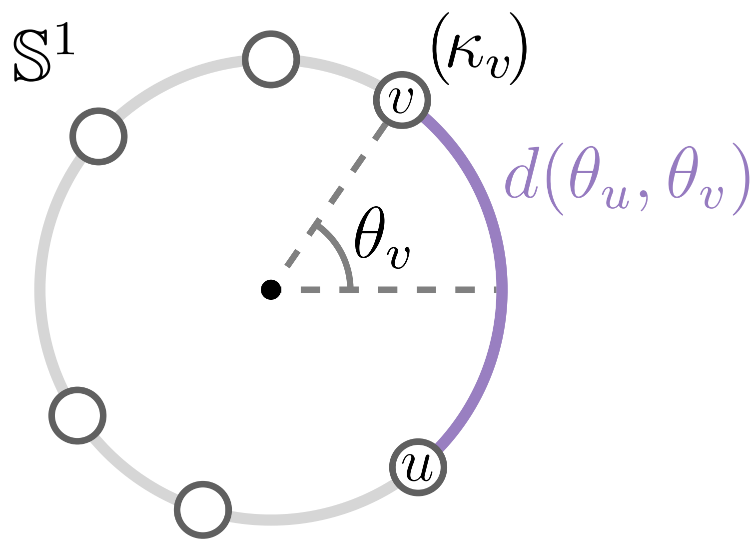
The connection probability is

$\mathbb{S}^1$ model

$$p_{uv} = \frac{1}{1 + \left(\frac{d(\theta_u, \theta_v)}{\mu \kappa_u, \kappa_v}\right)^{\beta}}$$

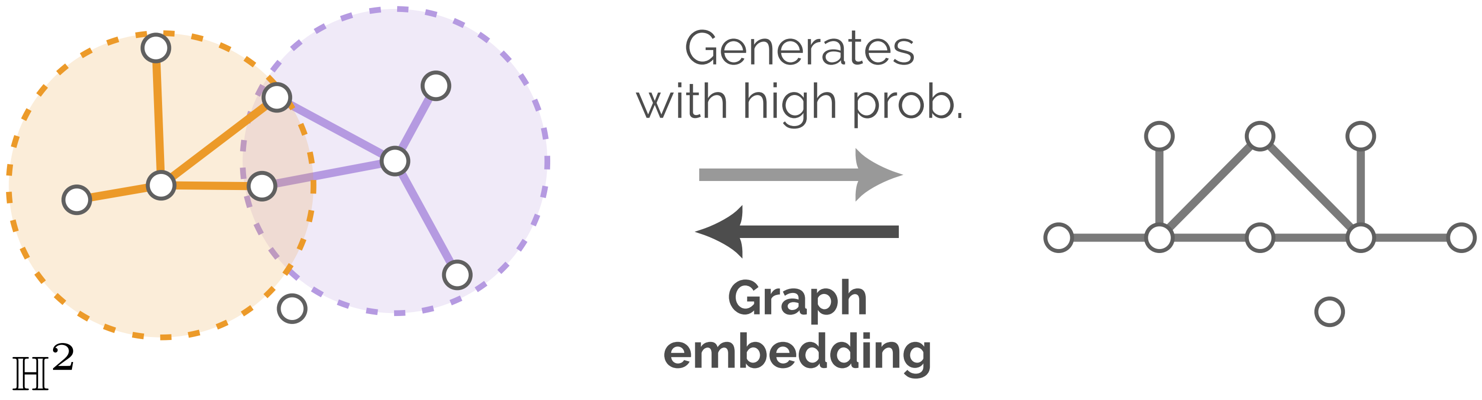
where

- d is the arc length,
- β controls the steepness of the decline,
- θ_v is the position of vertex v,
- κ_v is the expected degree of vertex v,
- μ is a scaling factor.



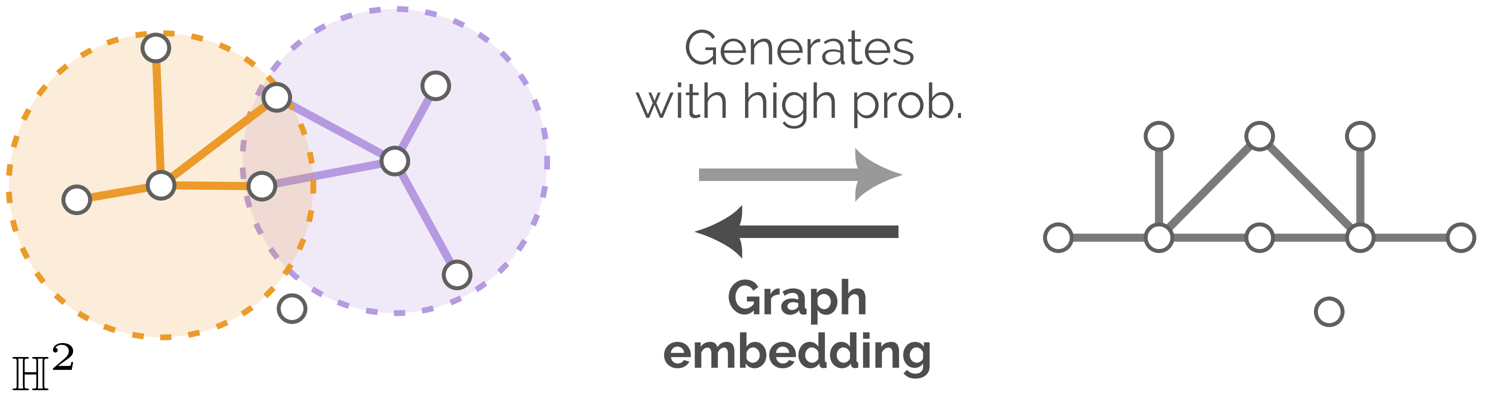
Graph vertex embedding in a nutshell

We want to **represent a given graph** using a hyperbolic embedding.



Graph vertex embedding in a nutshell

We want to **represent a given graph** using a hyperbolic embedding.



This amounts to:

$$\text{Pairs of vertices are} \begin{cases} \text{close} & \text{if connected in graph,} \\ \text{far} & \text{if not connected in graph.} \end{cases}$$

Current algorithms miss some crucial information

Algorithms

- Papadopoulos, F. et al. Phys. Rev. E **92**, 022807 (2015).
- Alanis-Lobato, G. et al. Appl. Netw. Sci. **1**, 1–14 (2016).
- Muscoloni, A. et al. Nat. Commun. **8**, 1615 (2017).
- García-Pérez, G. et al. New J. Phys. **21**, 123033 (2019).
- Wang, Z. et al. J. Stat. Mech.: Theory Exp. 123401 (2019).
- . . .

yield a **single embedding**.

Current algorithms miss some crucial information

Algorithms

- Papadopoulos, F. et al. Phys. Rev. E **92**, 022807 (2015).
- Alanis-Lobato, G. et al. Appl. Netw. Sci. **1**, 1–14 (2016).
- Muscoloni, A. et al. Nat. Commun. **8**, 1615 (2017).
- García-Pérez, G. et al. New J. Phys. **21**, 123033 (2019).
- Wang, Z. et al. J. Stat. Mech.: Theory Exp. 123401 (2019).
- . . .

yield a **single embedding**.

We currently ignore

- if there exists many plausible embeddings;
- how precise the vertex coordinates are.

We address both issues using a Bayesian approach.

$\mathbb{S}^1$ Bayesian model

The likelihood of the $\mathbb{S}^1$ model is:

$$\mathbb{P}[G|\theta, \kappa, \beta] = \prod_{(u,v)\in E} p_{uv} \prod_{(u,v)\in E^c} (1 - p_{uv}),$$

where $p_{uv} = \left[1 + \left(\frac{d_{\mathbb{S}}(\theta_u,\theta_v)}{\mu\kappa_u,\kappa_v}\right)^{\beta}\right]^{-1}$. The posterior of the Bayesian $\mathbb{S}^1$ model is

$\mathbb{S}^1$ Bayesian model

The likelihood of the $\mathbb{S}^1$ model is:

$$\mathbb{P}[G|\theta, \kappa, \beta] = \prod_{(u,v)\in E} p_{uv} \prod_{(u,v)\in E^c} (1 - p_{uv}),$$

where $p_{uv} = \left[1 + \left(\frac{d_{\mathbb{S}}(\theta_u, \theta_v)}{\mu\kappa_u, \kappa_v}\right)^{\beta}\right]^{-1}$. The posterior of the Bayesian $\mathbb{S}^1$ model is

$$p(\theta, \kappa, \beta|G) \propto \mathbb{P}[G|\theta, \kappa, \beta]\; p(\beta) \prod_{v\in V} p(\theta_v) p(\kappa_v),$$

$\mathbb{S}^1$ Bayesian model

The likelihood of the $\mathbb{S}^1$ model is:

$$\mathbb{P}[G|\theta, \kappa, \beta] = \prod_{(u,v)\in E} p_{uv} \prod_{(u,v)\in E^c} (1 - p_{uv}),$$

where $p_{uv} = \left[1 + \left(\frac{d_{\mathbb{S}}(\theta_u,\theta_v)}{\mu\kappa_u,\kappa_v}\right)^{\beta}\right]^{-1}$. The posterior of the Bayesian $\mathbb{S}^1$ model is

$$p(\theta, \kappa, \beta|G) \propto \mathbb{P}[G|\theta, \kappa, \beta]\; p(\beta) \prod_{v\in V} p(\theta_v)p(\kappa_v),$$

where the priors are

$$\begin{aligned} \theta_v &\overset{iid}{\sim} \text{Uniform}[-\pi, \pi), \\ \kappa_v &\overset{iid}{\sim} \text{Cauchy}, \quad \kappa_v > \epsilon, \\ \beta &\sim \text{Normal}, \quad \beta > 1. \end{aligned}$$

$\mathbb{S}^1$ Bayesian model

The likelihood of the $\mathbb{S}^1$ model is:

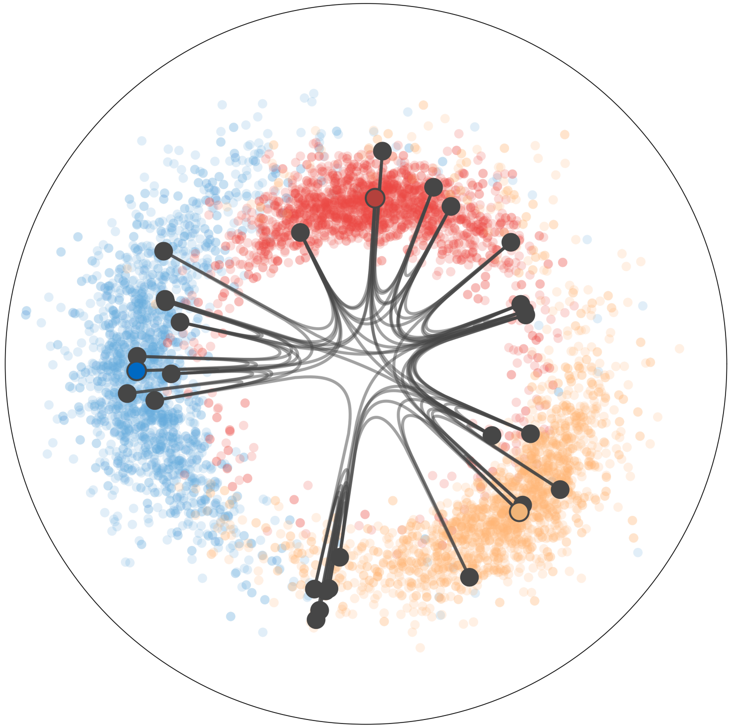
$$\mathbb{P}[G|\theta, \kappa, \beta] = \prod_{(u,v)\in E} p_{uv} \prod_{(u,v)\in E^c} (1 - p_{uv}),$$

where $p_{uv} = \left[1 + \left(\frac{d_{\mathbb{S}}(\theta_u, \theta_v)}{\mu \kappa_u, \kappa_v}\right)^{\beta}\right]^{-1}$. The posterior of the Bayesian $\mathbb{S}^1$ model is

$$p(\theta, \kappa, \beta|G) \propto \mathbb{P}[G|\theta, \kappa, \beta]\; p(\beta) \prod_{v\in V} p(\theta_v) p(\kappa_v),$$

where the priors are

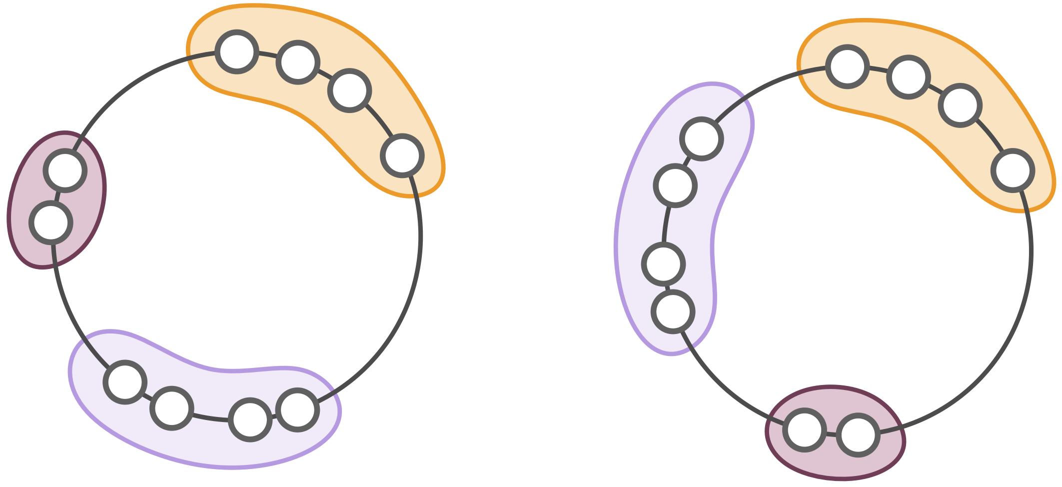
$$\begin{aligned} \theta_v &\overset{iid}{\sim} \text{Uniform}[-\pi, \pi), \\ \kappa_v &\overset{iid}{\sim} \text{Cauchy}, \quad \kappa_v > \epsilon, \\ \beta &\sim \text{Normal}, \quad \beta > 1. \end{aligned}$$



We **sample embeddings** from the posterior using Markov chain Monte Carlo (MCMC).

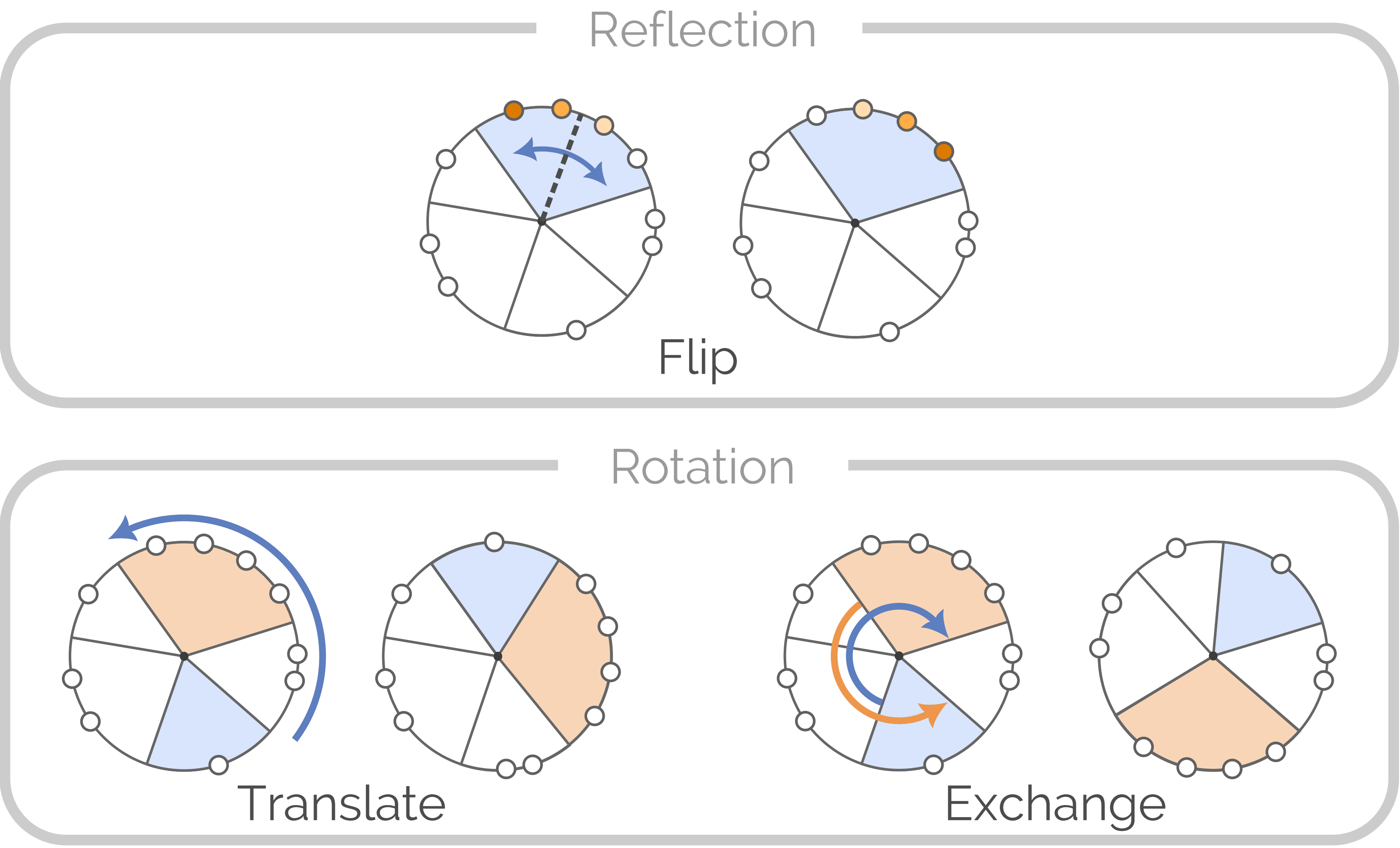
Locality implies clusters

Since edges are local, groups of nearby vertices should be moved together.

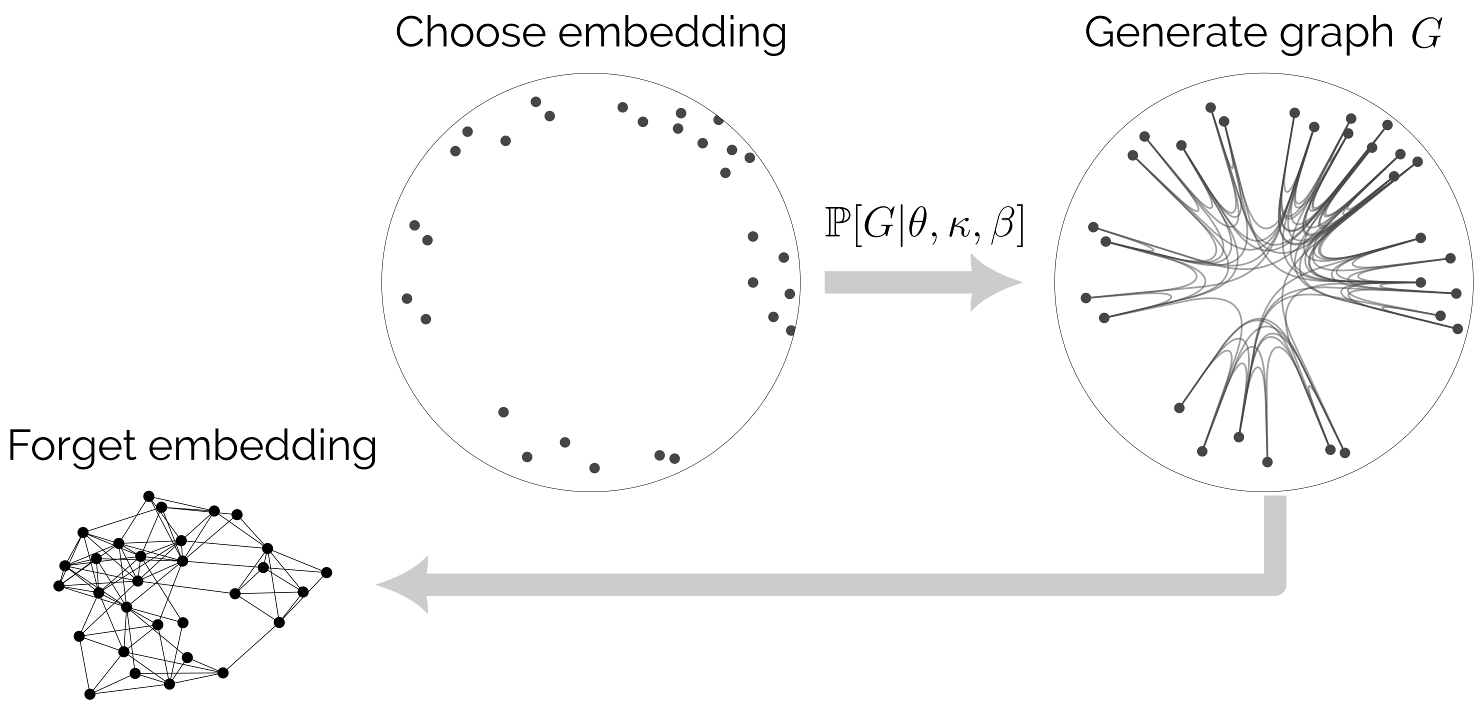


Locality implies clusters

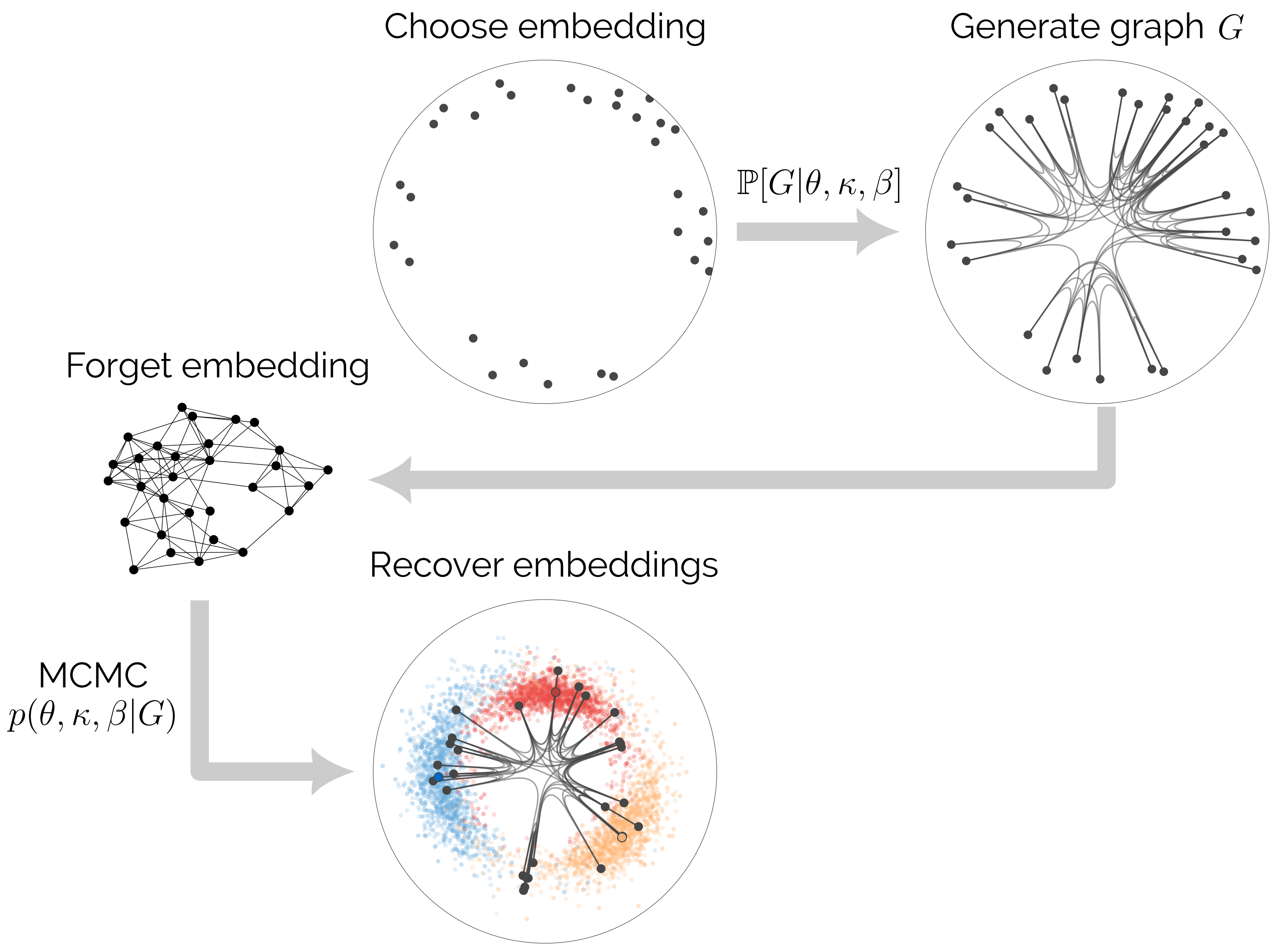
Since edges are local, groups of nearby vertices should be moved together.



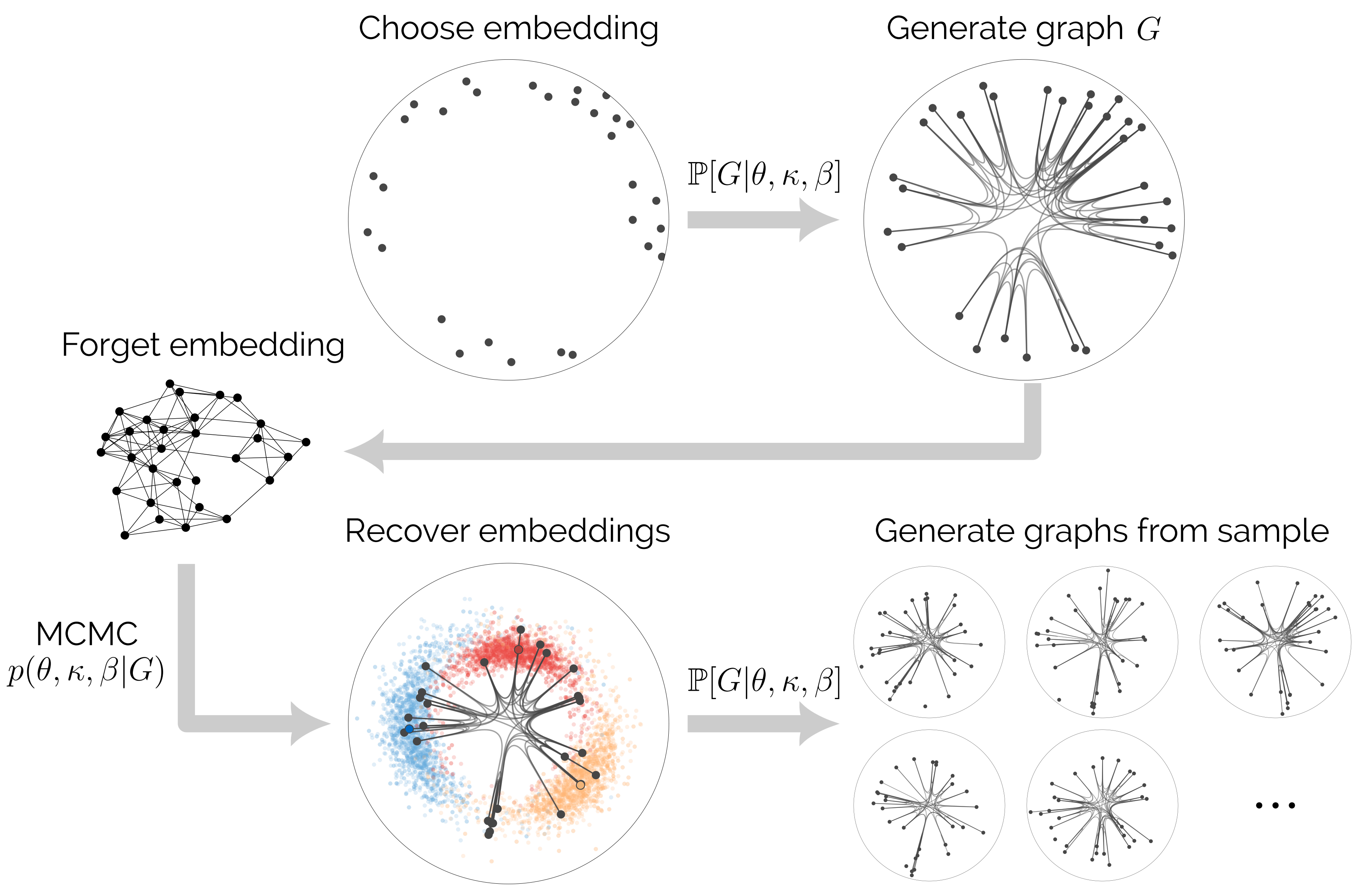
Sanity check with synthetic data



Sanity check with synthetic data

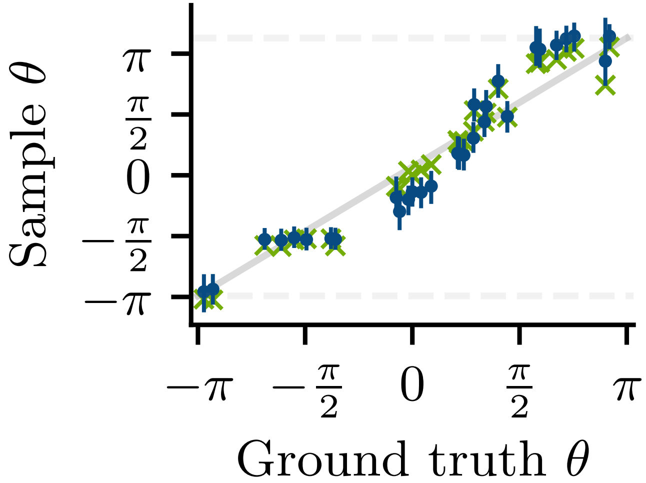


Sanity check with synthetic data

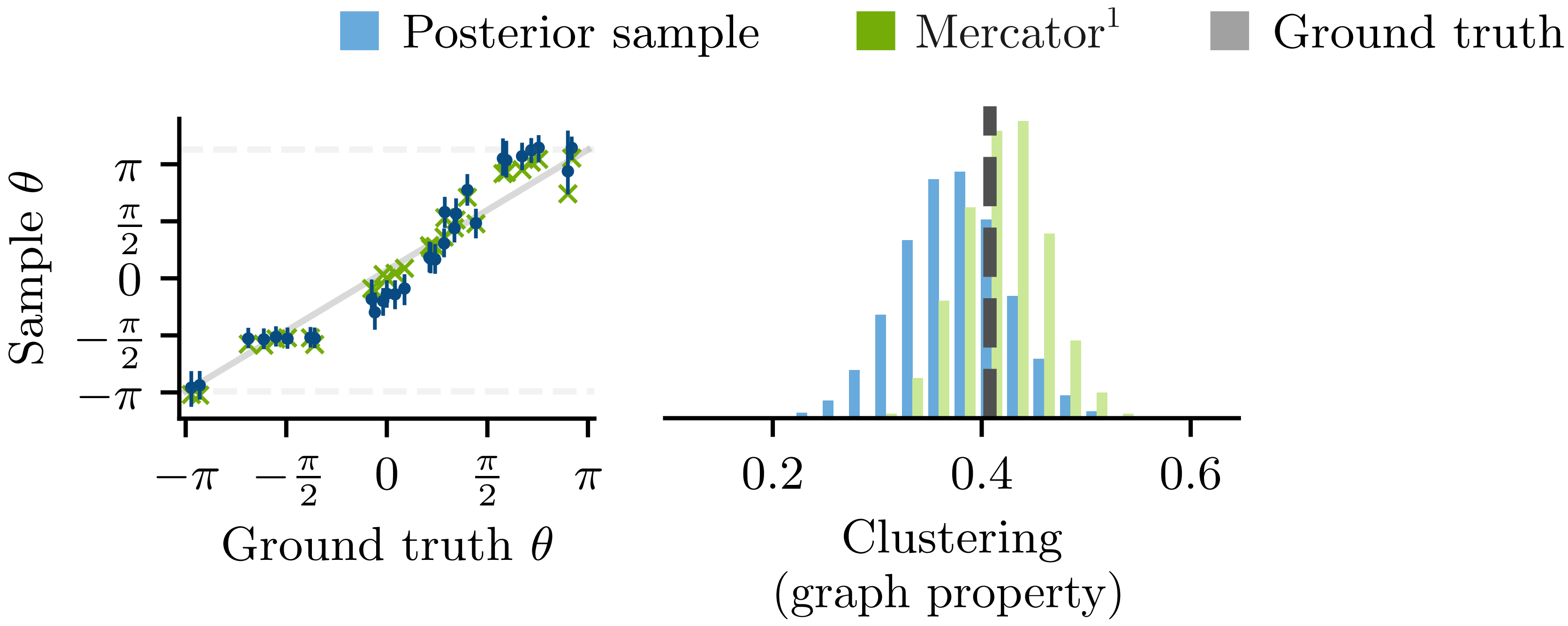


Embedding error bars

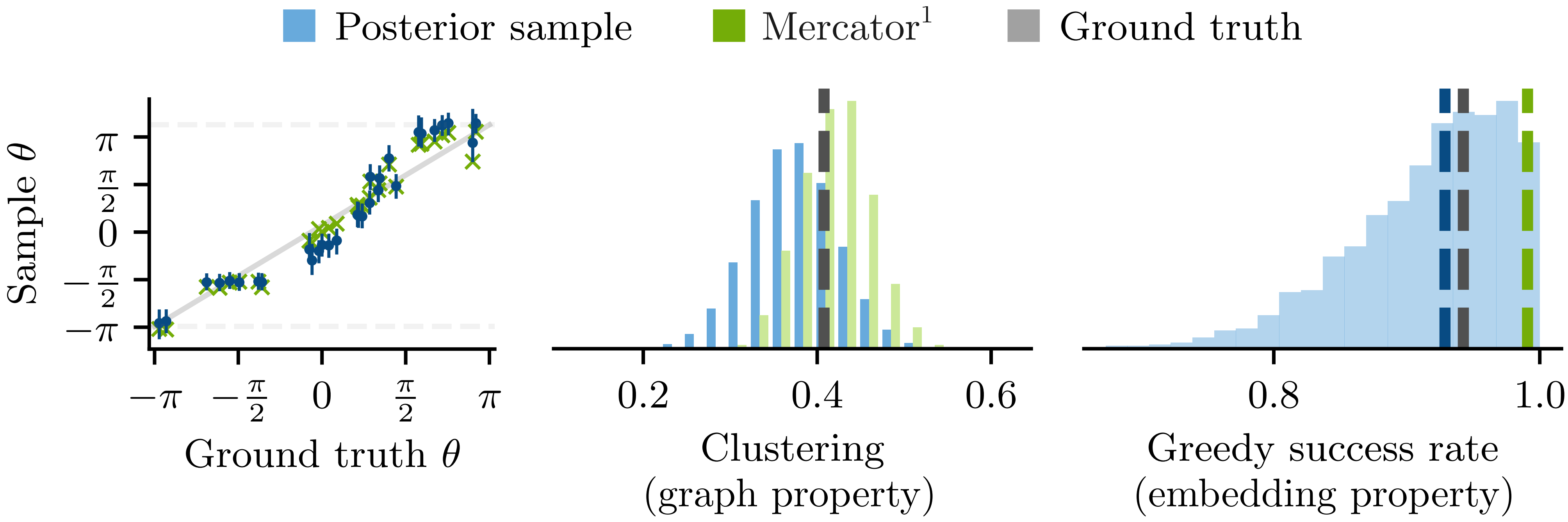
Posterior sample　Mercator[1]　Ground truth



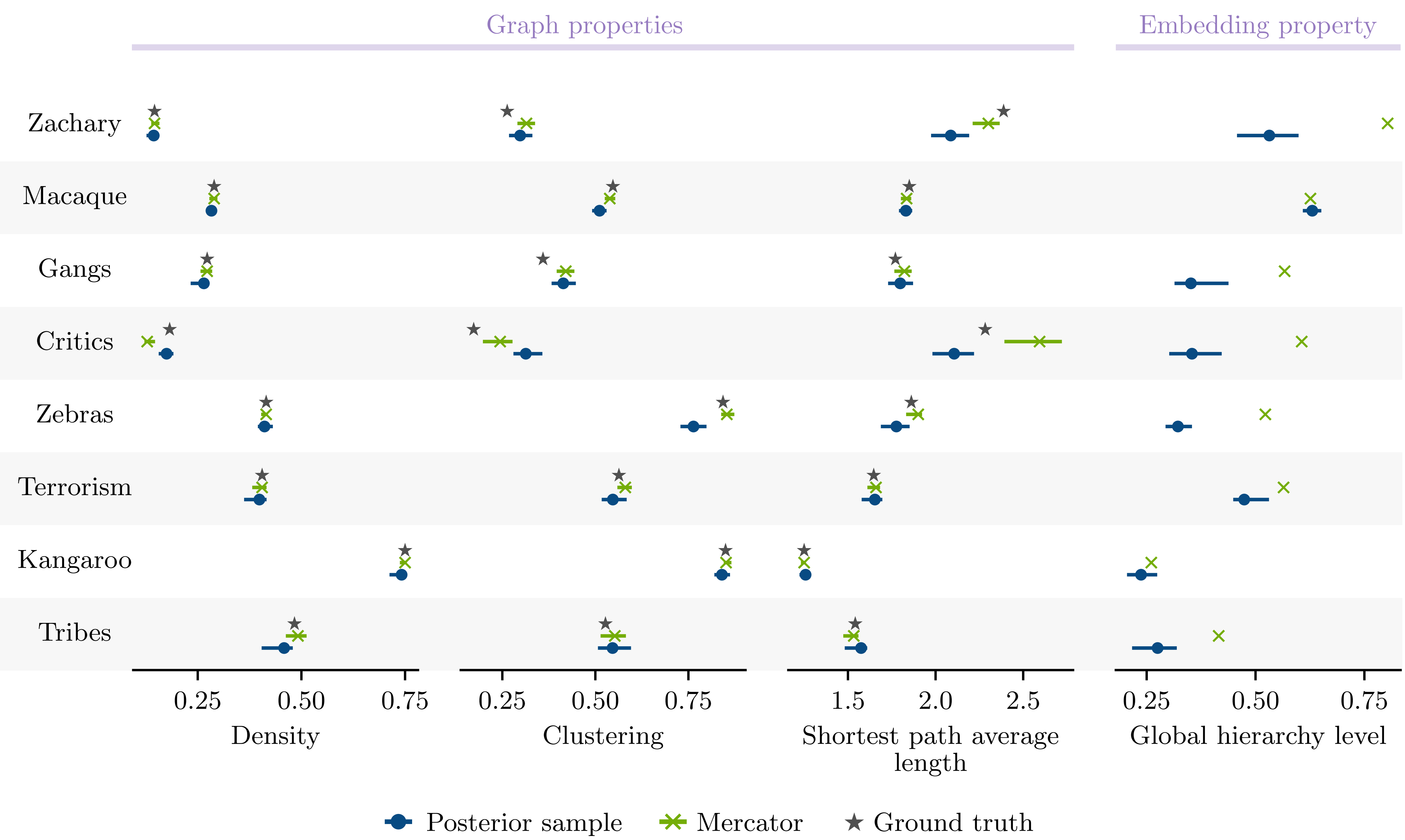
Embedding error bars



Embedding error bars



Empirical networks properties



Induced multimodal distribution

Conflicting ground truth model:

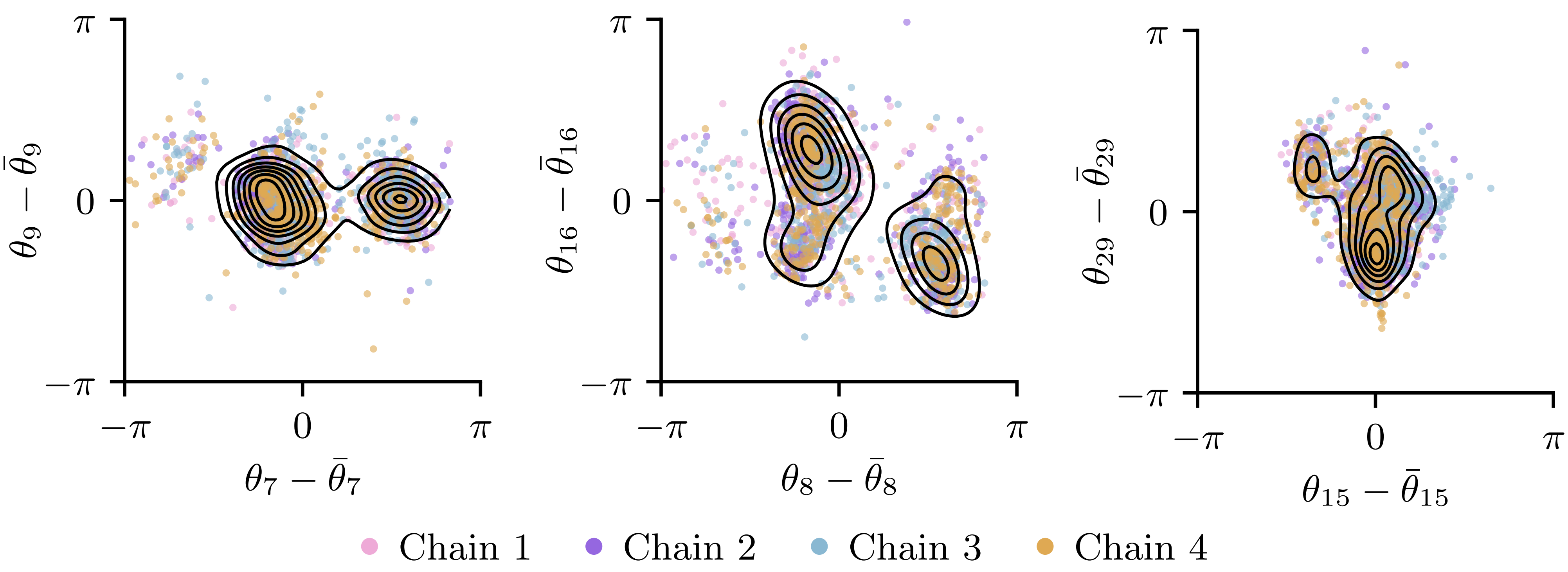
- A vertex v is given two positions $\theta_v^{(1)}$ and $\theta_v^{(2)}$.
- When generating G with the $\mathbb{S}^1$ model, each edge probability including v uses randomly $\theta_v^{(1)}$ or $\theta_v^{(2)}$.

Induced multimodal distribution

Conflicting ground truth model:

- A vertex v is given two positions $\theta_v^{(1)}$ and $\theta_v^{(2)}$.
- When generating G with the $\mathbb{S}^1$ model, each edge probability including v uses randomly $\theta_v^{(1)}$ or $\theta_v^{(2)}$.

Marginal posterior distributions



Takeaways

- Hyperbolic random graphs reproduce many empirically observed properties.
- Current embedding algorithms give a single embedding $\Longrightarrow$ no error bars;
- Our Bayesian approach can identify multiple good embeddings;
- Locality of edges $\Longrightarrow$ clusters as coarse-graining of the embedding.

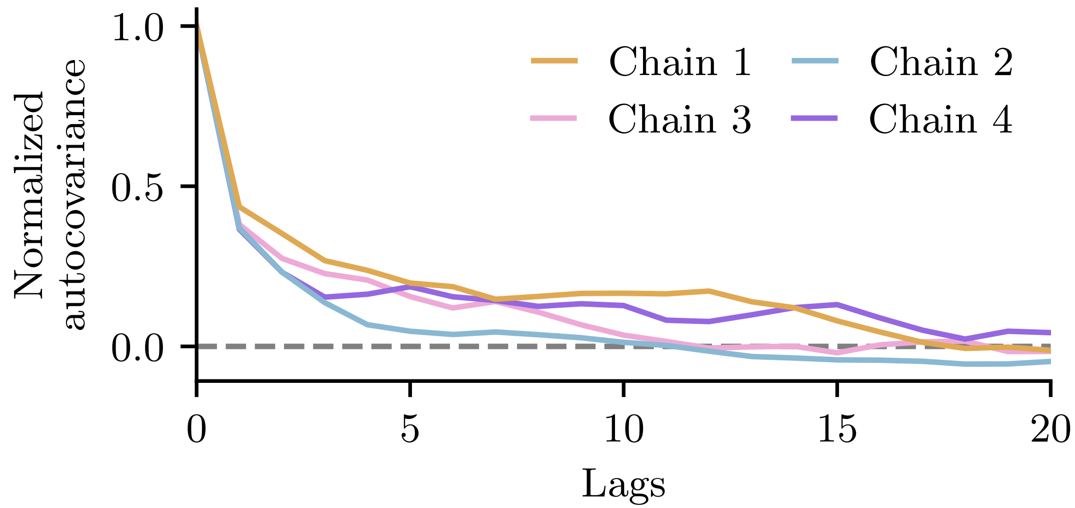
Paper:

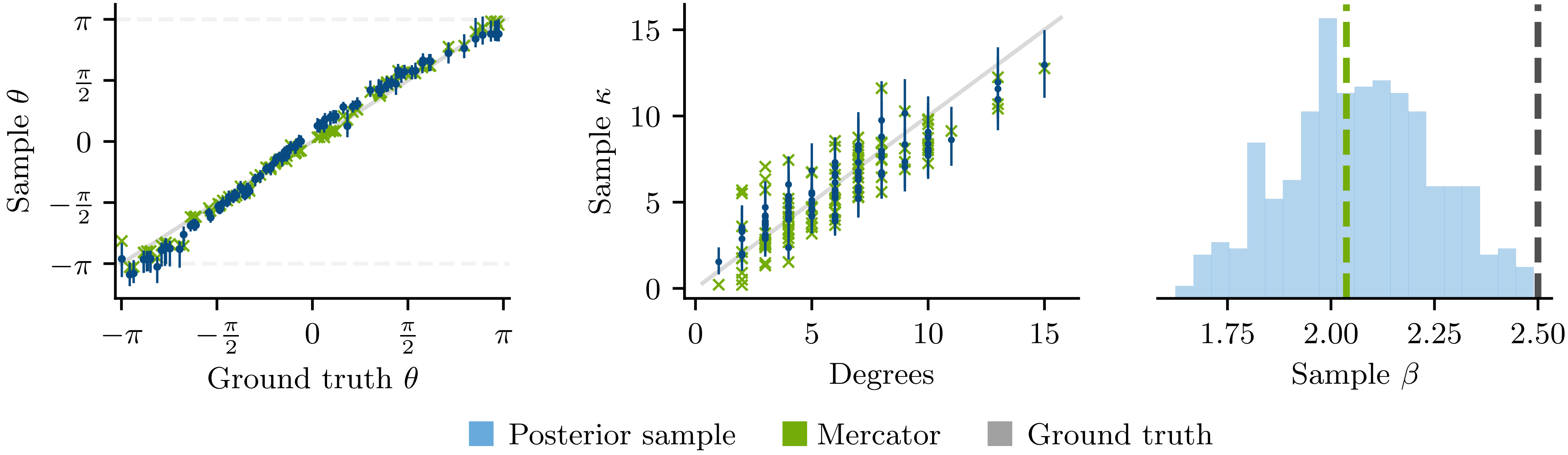
Lizotte, S., Young, J.-G. and Allard A. *Symmetry-driven embedding of networks in hyperbolic space*. Commun. Phys. **8**, 199 (2025).

Antoine Allard

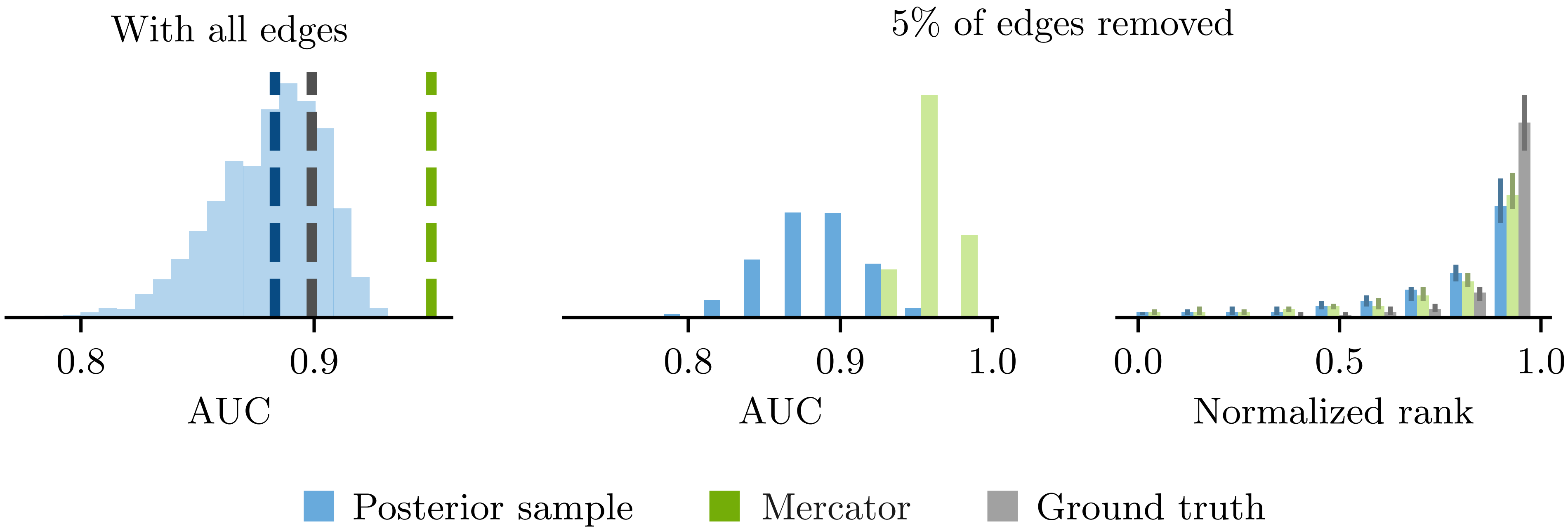
Jean-Gabriel Young

Synthetic graph of 100 vertices

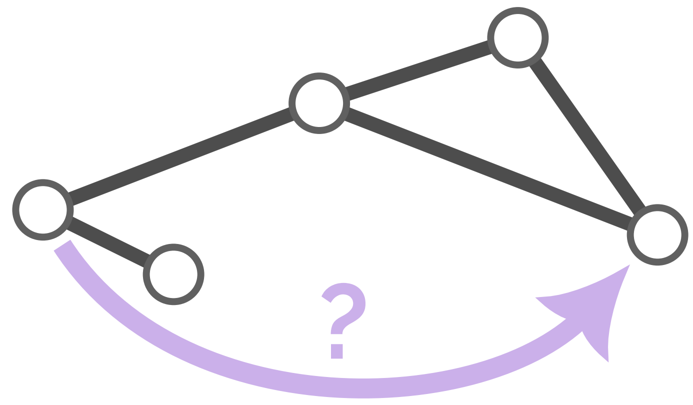




Link prediction is equivalent when sampling



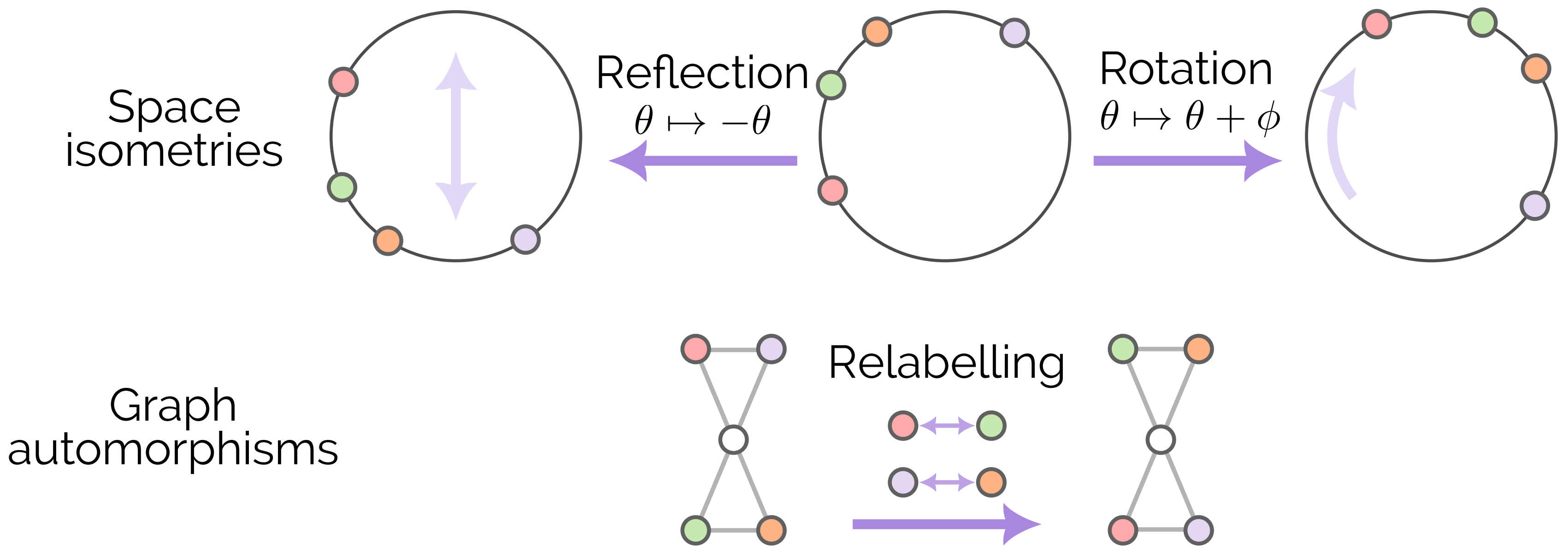
Greedy routing



Go to the neighbour closest to the destination.

Model symmetries

The $\mathbb{S}^1$ model is not identifiable because of graph and space symmetries.



Comparing embeddings requires alignment.