Dynamic message passing and its applications for spreading processes on networks

Mateusz Wilinski
MSCA Postdoctoral Fellow

06/11/2024

Motivation

90s Argentina Cholera Epidemic

90s Argentina Cholera Epidemic

- Susceptible
- Infected

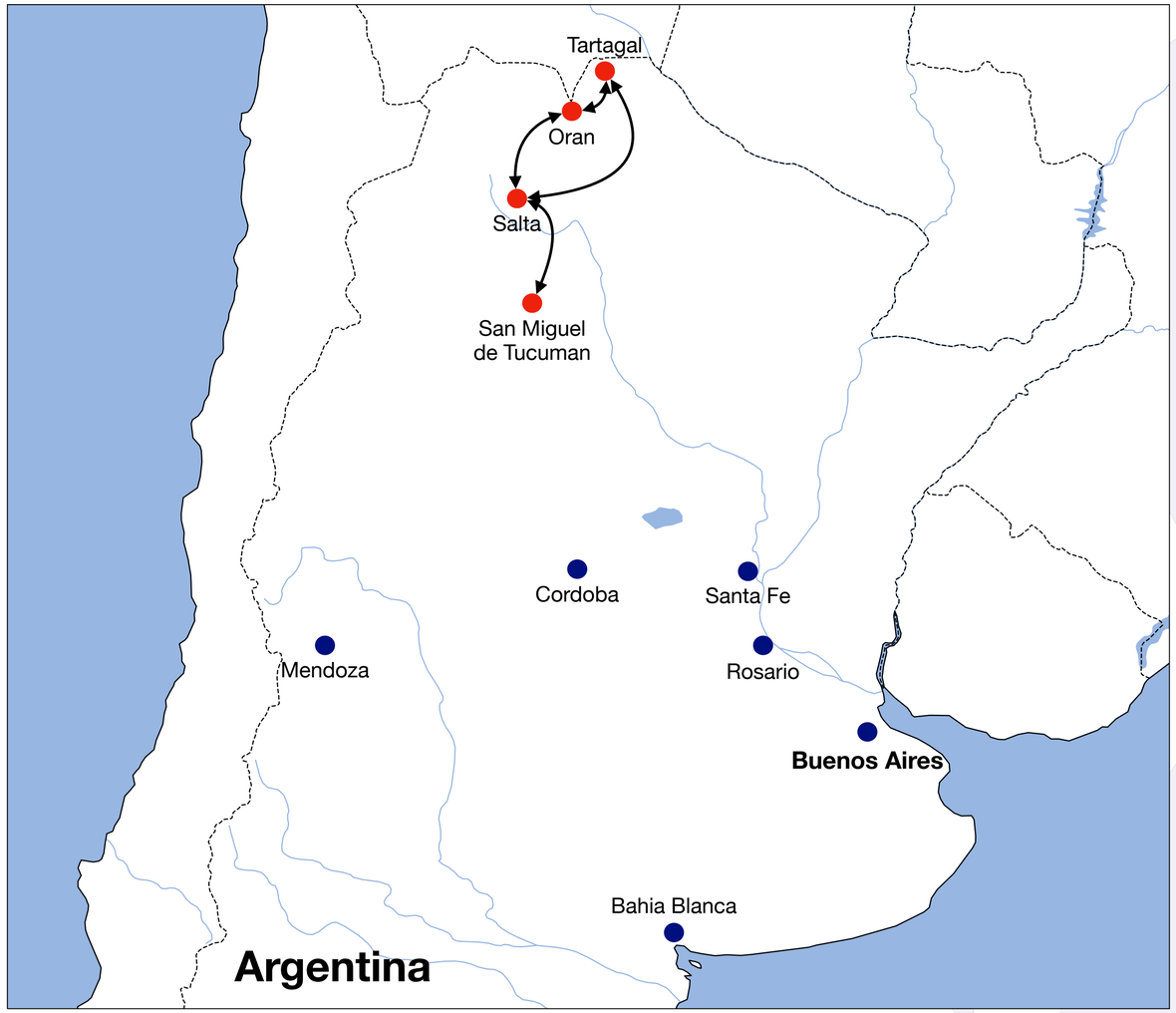
90s Argentina Cholera Epidemic

- Susceptible
- Infected

90s Argentina Cholera Epidemic

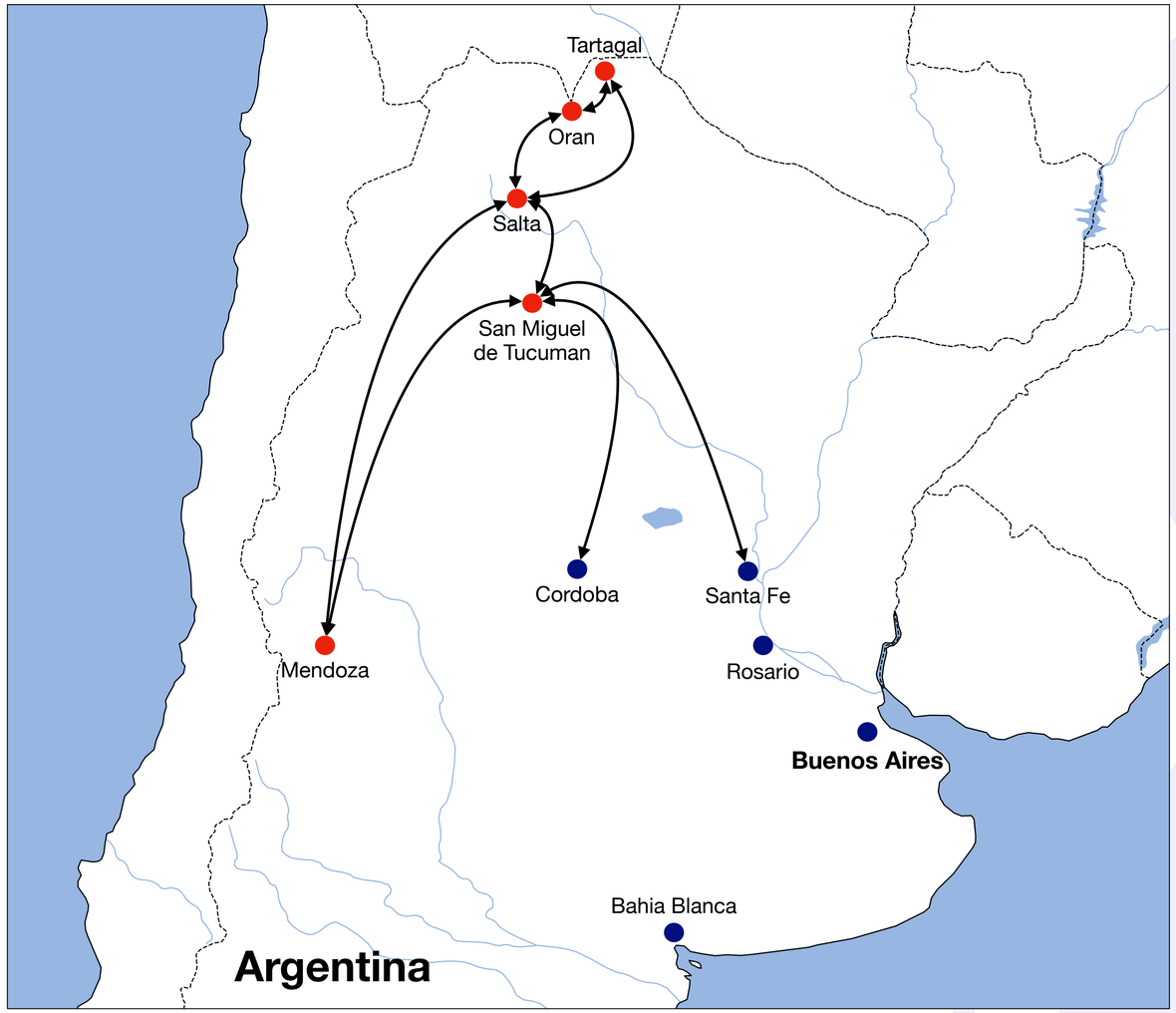
- Susceptible
- Infected

90s Argentina Cholera Epidemic



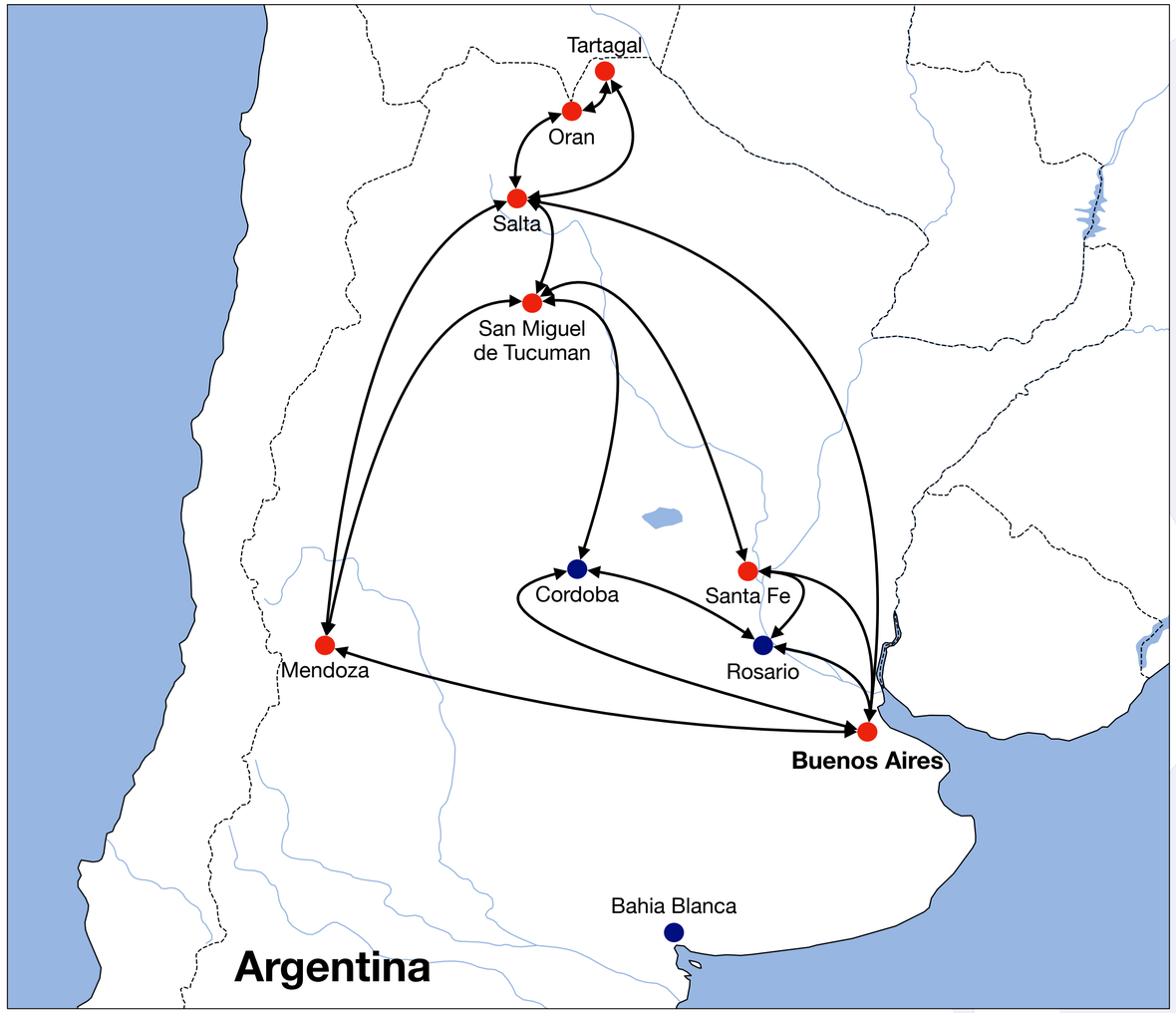
- Susceptible
- Infected

90s Argentina Cholera Epidemic



- Susceptible
- Infected

90s Argentina Cholera Epidemic

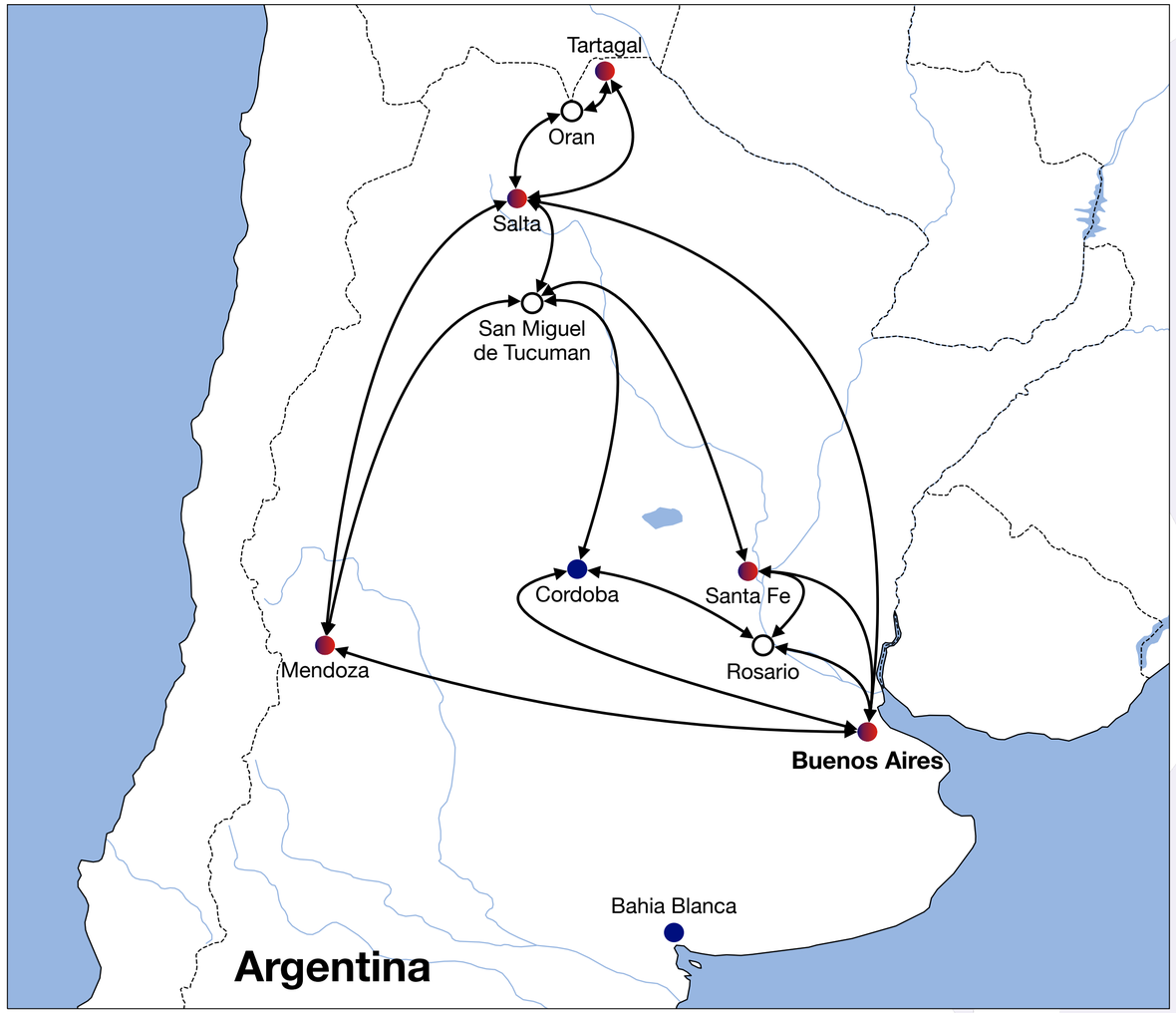


- Susceptible
- Infected

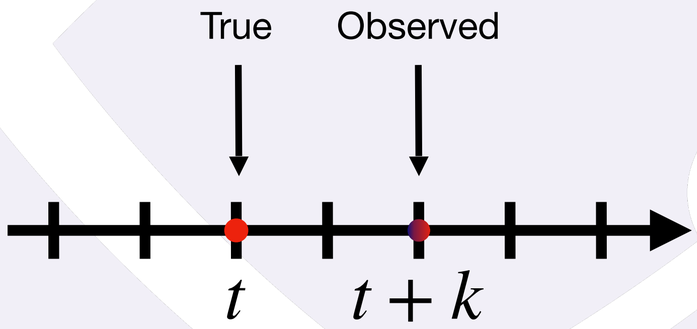
90s Argentina Cholera Epidemic

- Susceptible
- Infected
- Unobserved

90s Argentina Cholera Epidemic



- Susceptible
- Infected
- Unobserved
- Uncertain



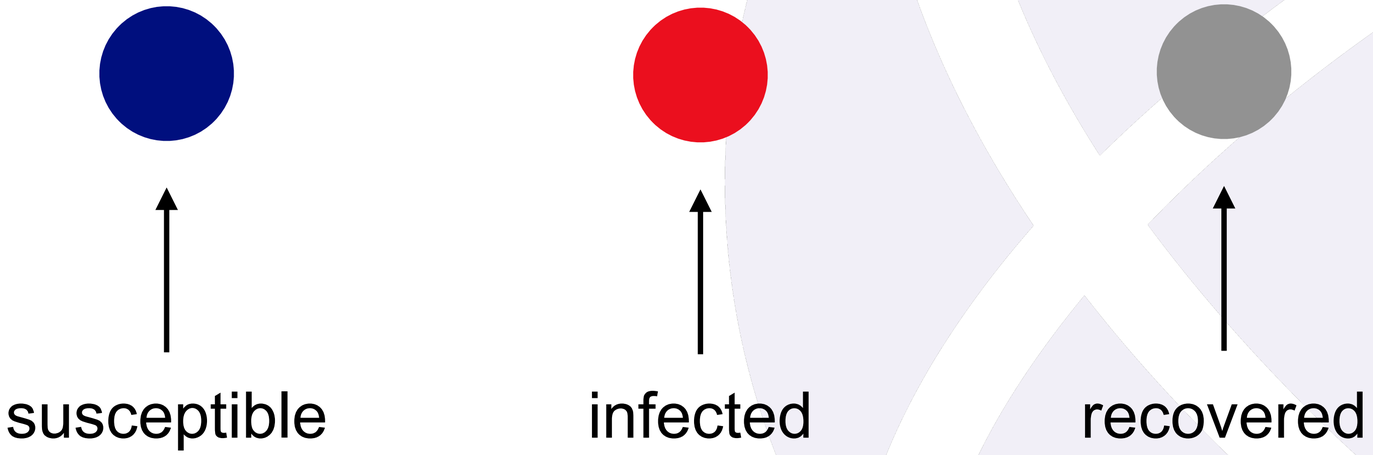
90s Argentina Cholera Epidemic

- Susceptible
- Infected
- Unobserved
- Uncertain

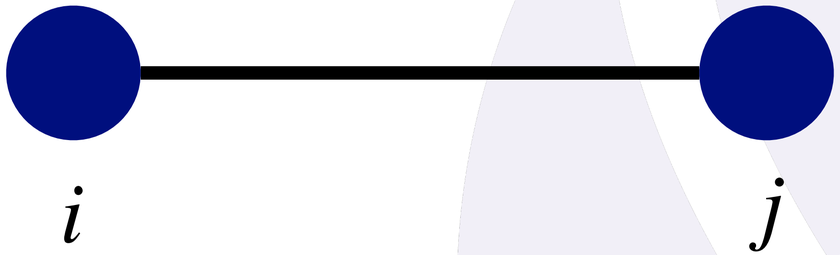
Model

Independent Cascade model

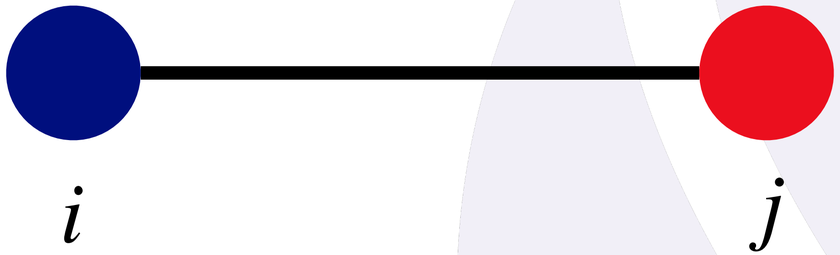
$$s_i \in \{S, I, R\}$$



Independent Cascade model

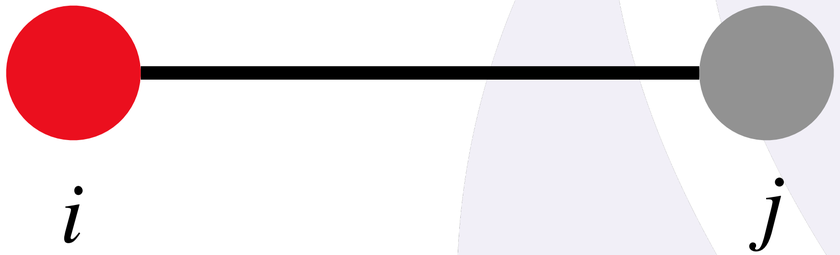


Independent Cascade model



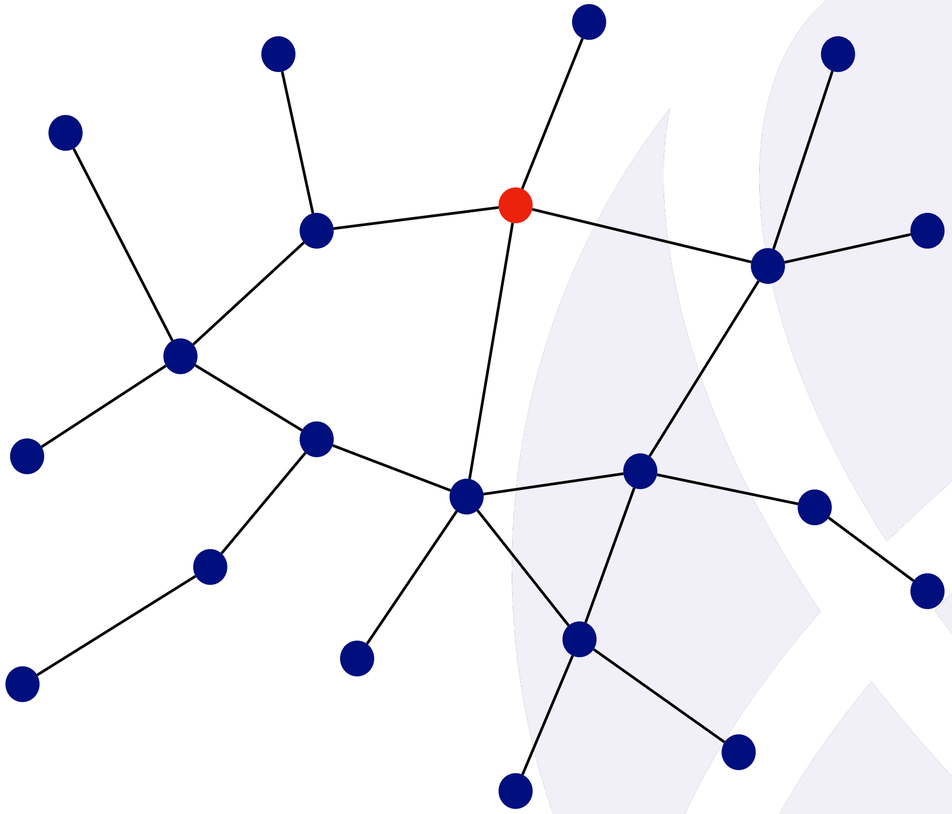
$$P(s_i(t+1) = I \,|\, s_j(t) = I) = \alpha_{ij}$$

Independent Cascade model

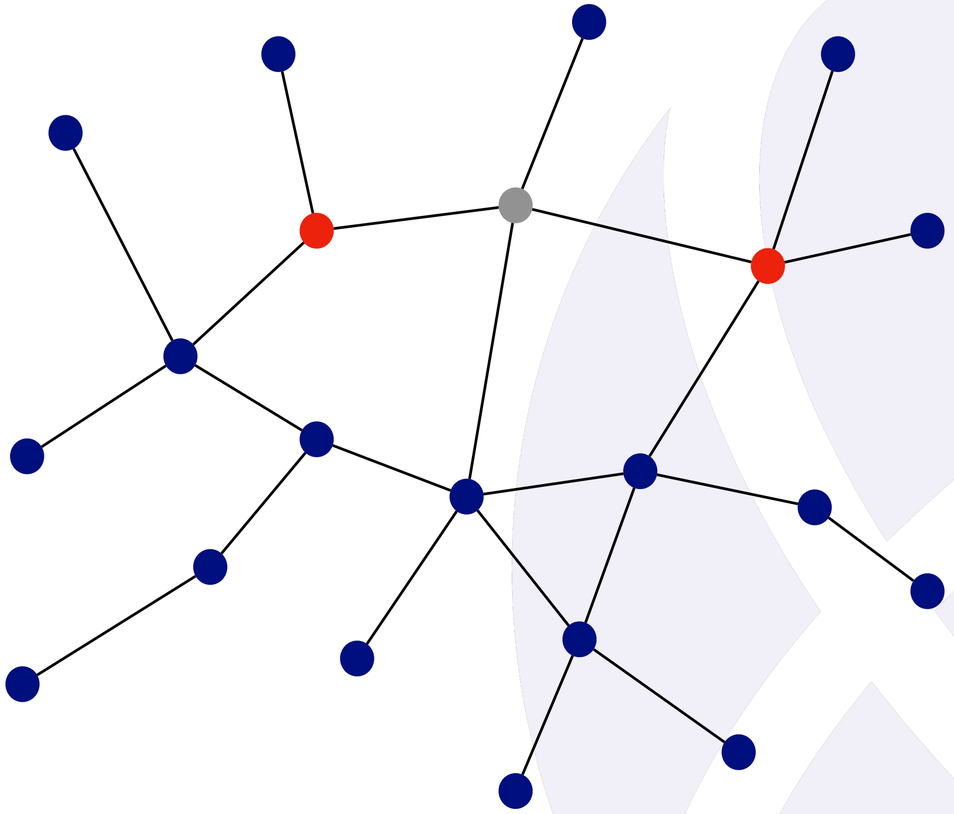


$$P(s_j(t+1) = R \mid s_j(t) = I) = 1$$

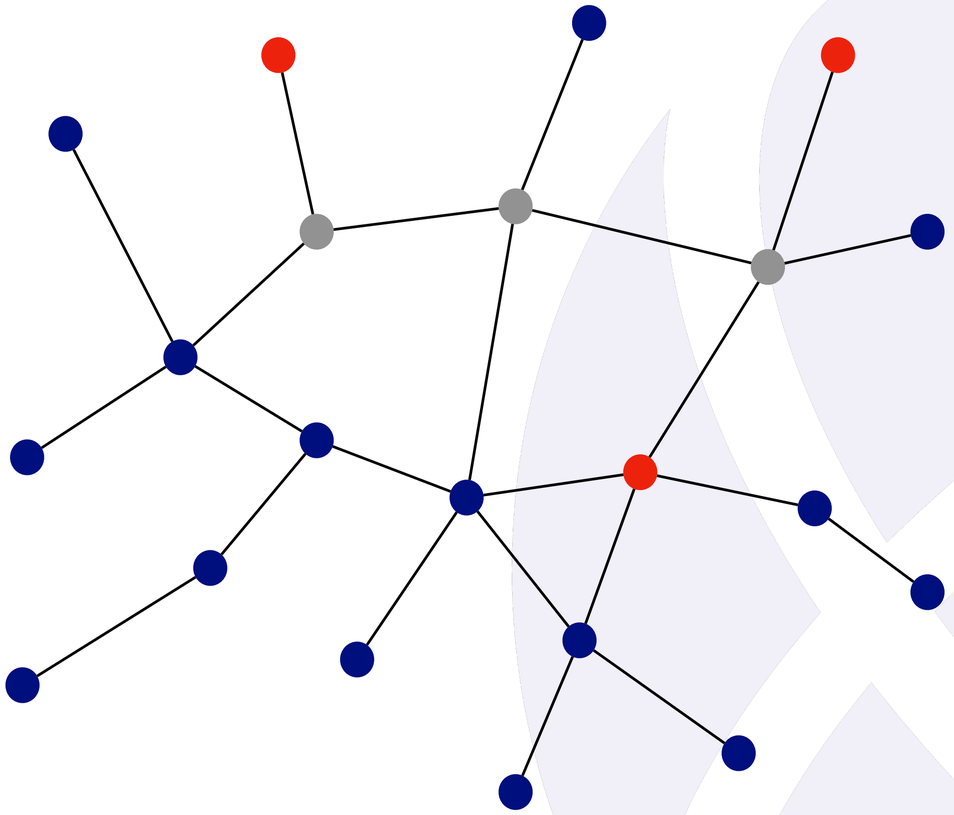
Independent Cascade model



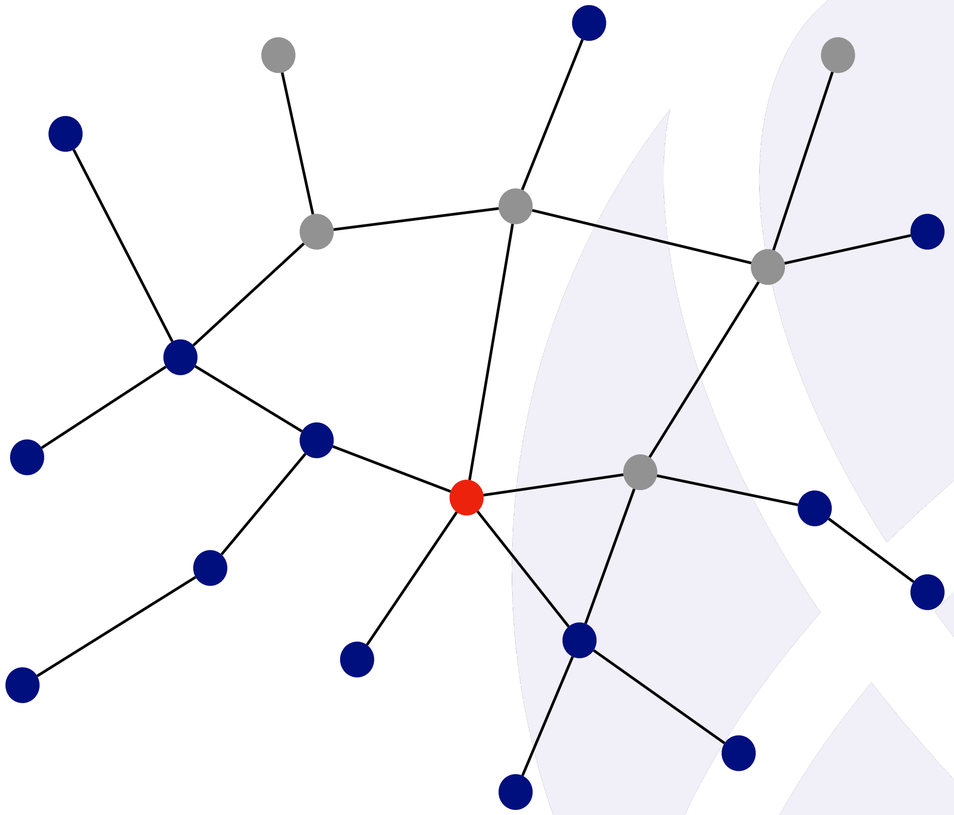
Independent Cascade model



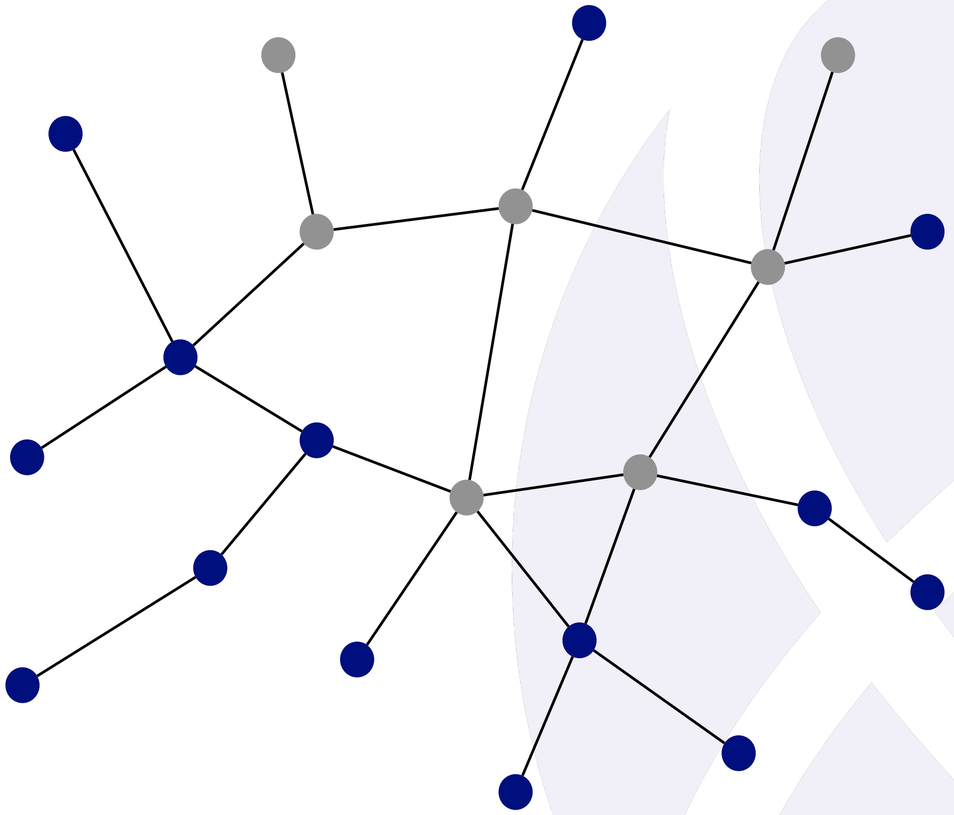
Independent Cascade model



Independent Cascade model

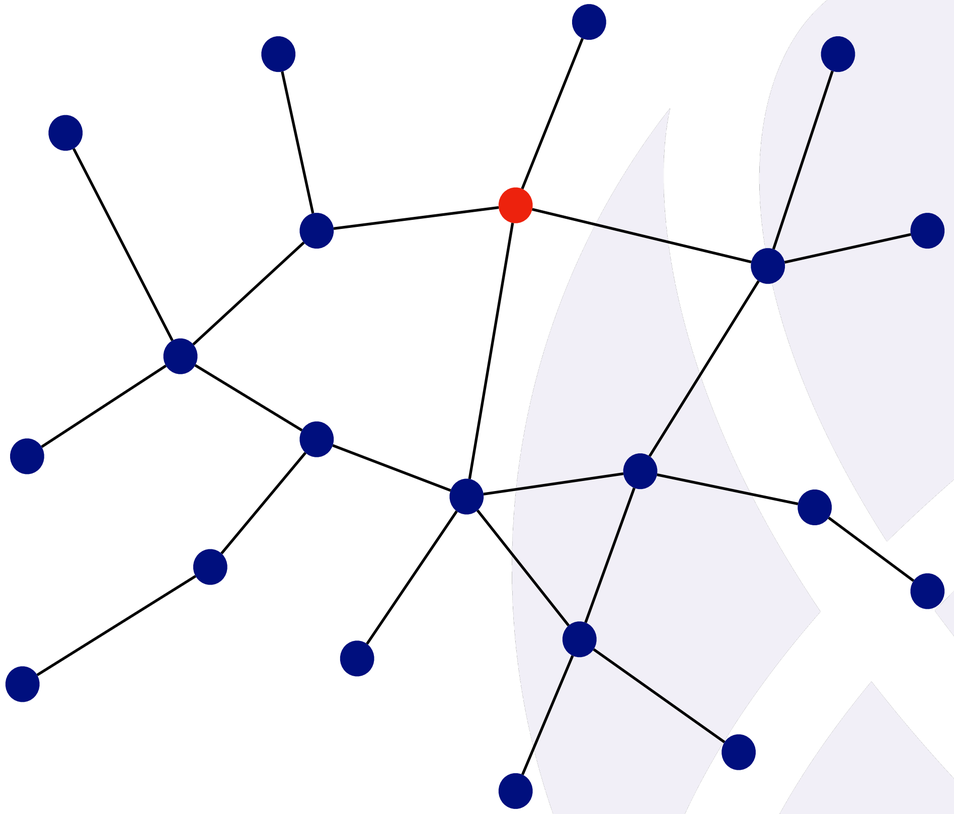


Independent Cascade model



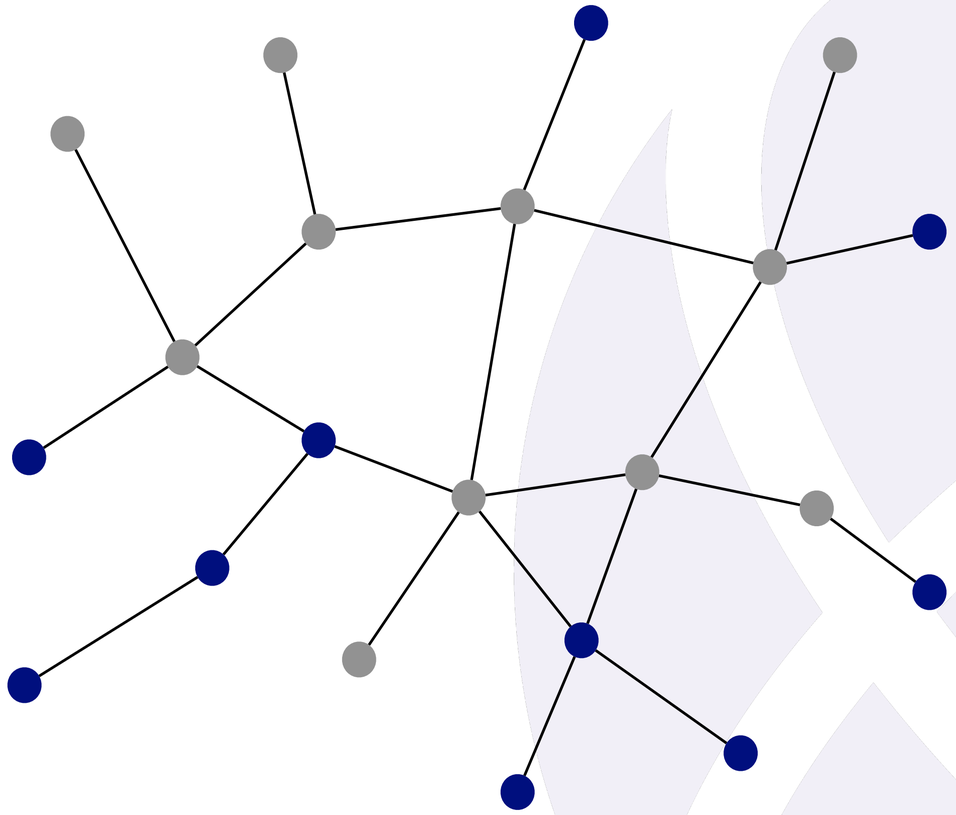
Inference

What do we want to infer?



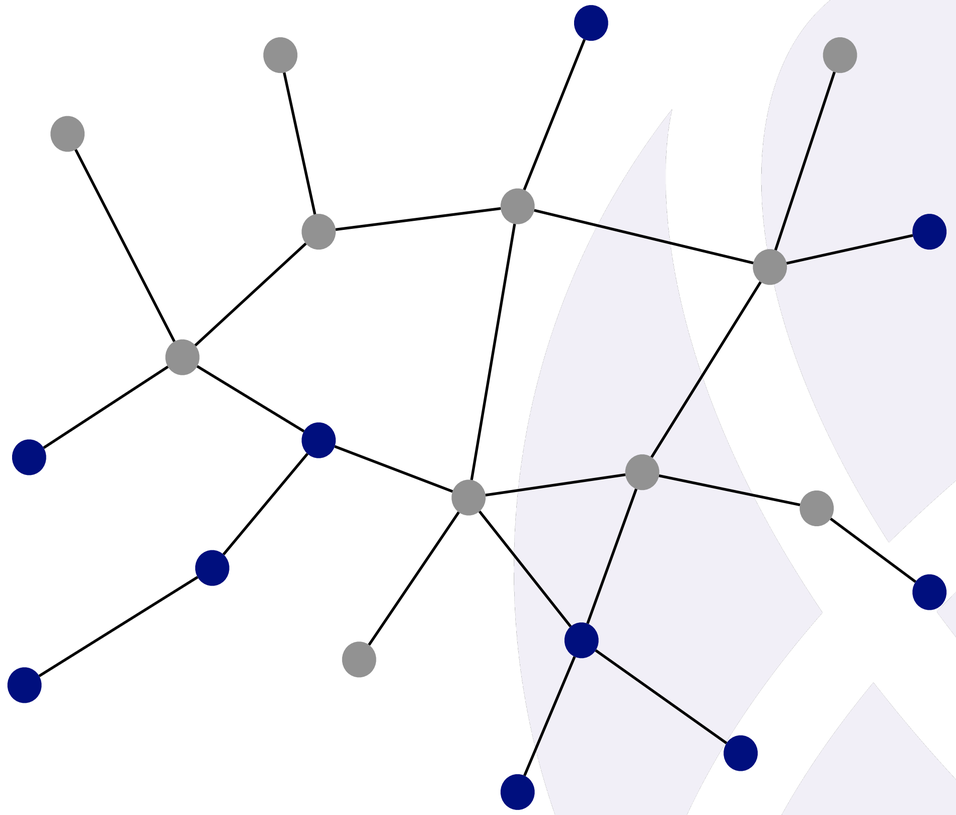
What is the expected outcome?

$$t = \infty$$



What is the expected outcome?

$$t = \tau$$



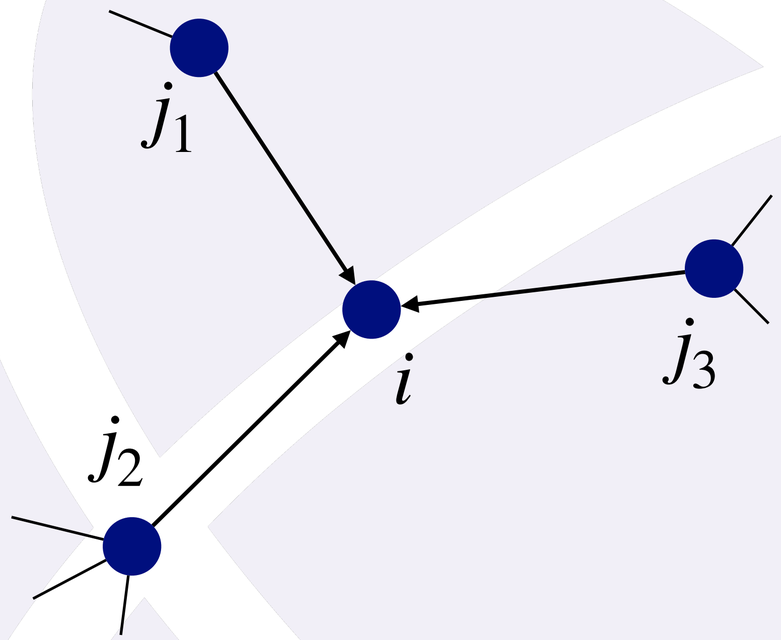
Influence function

$$\sigma_t(p_0) = \mathbb{E}\left(\sum_{i \in V} \mathbf{1}_{(t_i < t)}\right) = \sum_{i \in V} p_i(t)$$

Dynamic Message Passing

Dynamic Message Passing

$$p_i(t+1) = 1 - \left(1 - p_i(0)\right) \cdot q_i(t)$$



$p_i(t)$ ← probability of node *i* being activated at time *t* or before

$q_i(t)$ ← probability of node *i* not being activated up until time *t* by its neighbours

Dynamic Message Passing

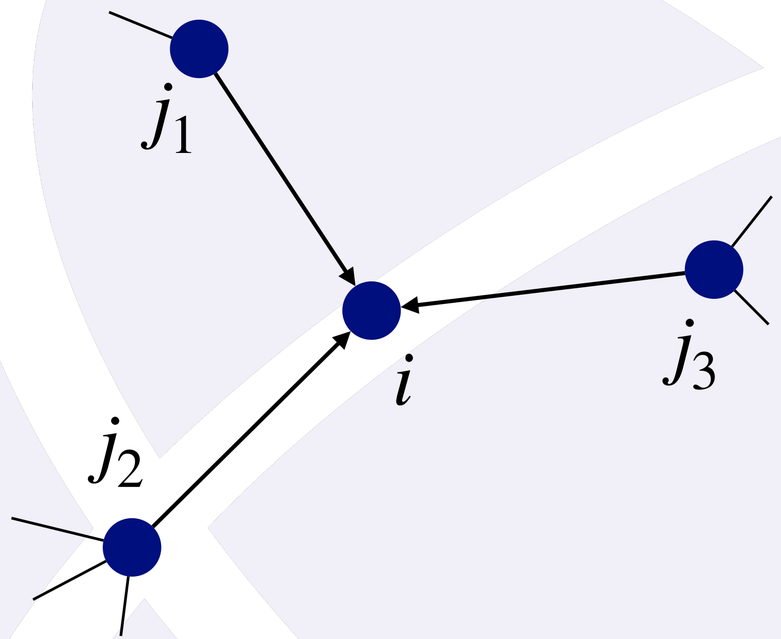
$$p_i(t+1) = 1 - \big(1 - p_i(0)\big) \cdot q_i(t)$$

$$p_i(t+1) = 1 - \big(1 - p_i(0)\big) \prod_{j \in \partial i} \big(1 - \alpha_{ji} \cdot p_{j \to i}(t)\big)$$

$p_i(t)$ ← probability of node *i* being activated at time *t* or before

$q_i(t)$ ← probability of node *i* not being activated up until time *t* by its neighbours

$p_{j \to i}(t)$ ← probability of node *j* being activated at time *t* on an auxiliary graph without *i*



Dynamic Message Passing

$$p_i(t+1) = 1 - \left(1 - p_i(0)\right) \cdot q_i(t)$$

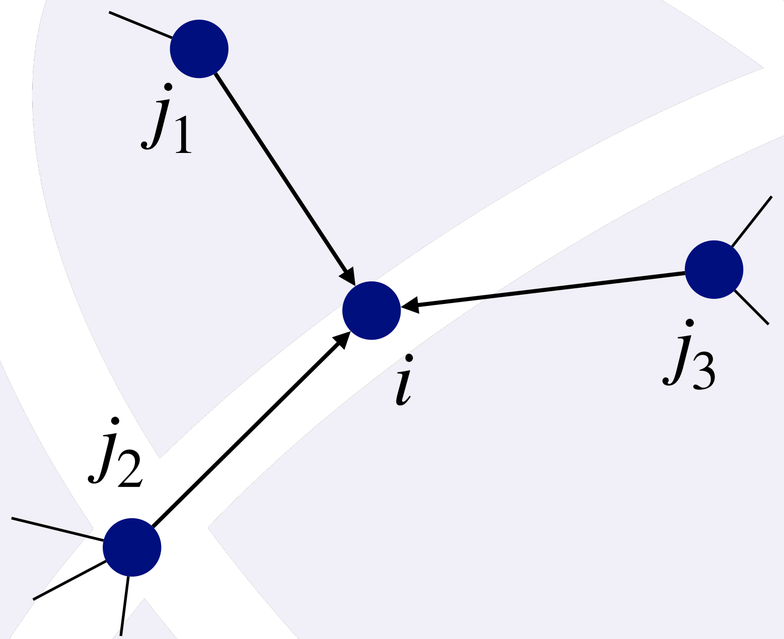
$$p_i(t+1) = 1 - \left(1 - p_i(0)\right) \prod_{j \in \partial i} \left(1 - \alpha_{ji} \cdot p_{j \to i}(t)\right)$$

$$p_{j \to i}(t+1) = 1 - \left(1 - p_i(0)\right) \prod_{k \in \partial j \setminus i} \left(1 - \alpha_{kj} \cdot p_{k \to j}(t)\right)$$

$p_i(t)$ ← probability of node *i* being activated at time *t* or before

$q_i(t)$ ← probability of node *i* not being activated up until time *t* by its neighbours

$p_{j \to i}(t)$ ← probability of node *j* being activated at time *t* on an auxiliary graph without *i*



Dynamic Message Passing

$$p_i(t+1) = 1 - \left(1 - p_i(0)\right) \cdot q_i(t)$$

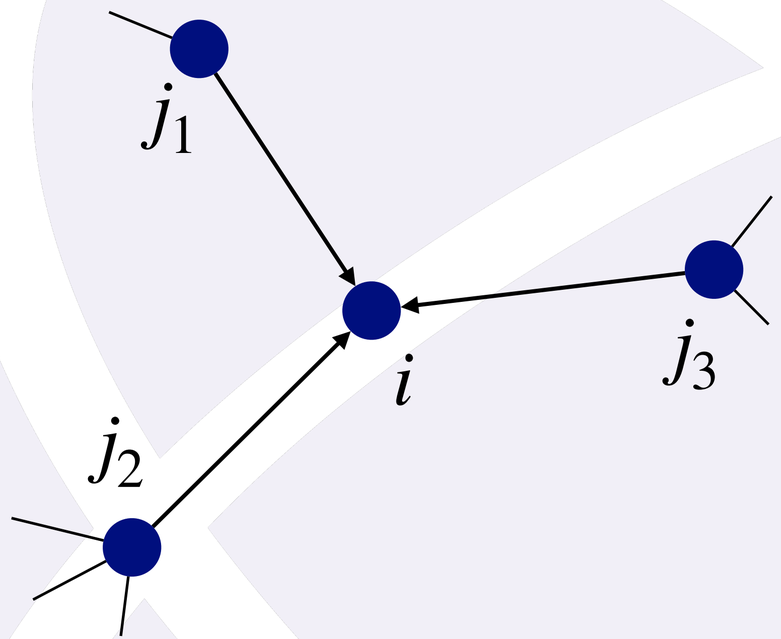
$$p_i(t+1) = 1 - \left(1 - p_i(0)\right) \prod_{j \in \partial i} \left(1 - \alpha_{ji} \cdot p_{j \to i}(t)\right)$$

$$p_{j \to i}(t+1) = 1 - \left(1 - p_i(0)\right) \prod_{k \in \partial j \setminus i} \left(1 - \alpha_{kj} \cdot p_{k \to j}(t)\right)$$

$p_i(t)$ ← probability of node *i* being activated at time *t* or before

$q_i(t)$ ← probability of node *i* not being activated up until time *t* by its neighbours

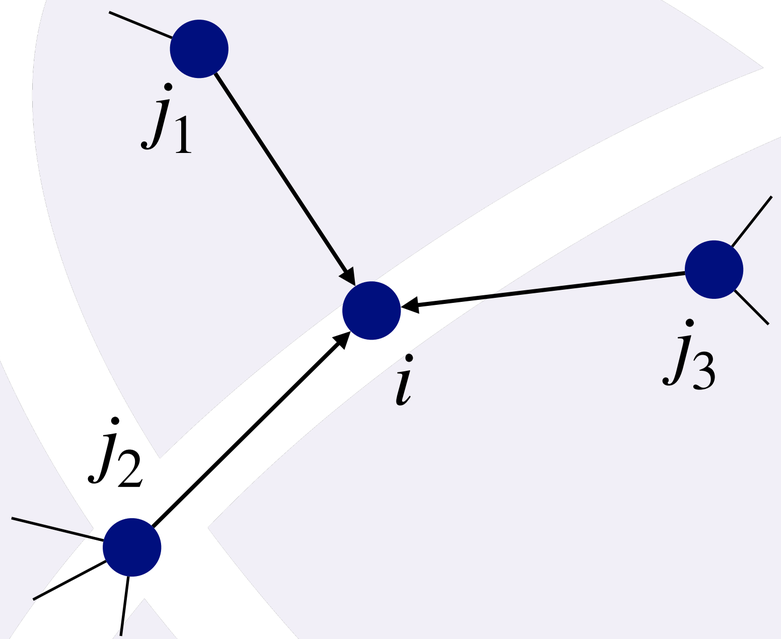
$p_{j \to i}(t)$ ← probability of node *j* being activated at time *t* on an auxiliary graph without *i*



Belief Propagation Derivation

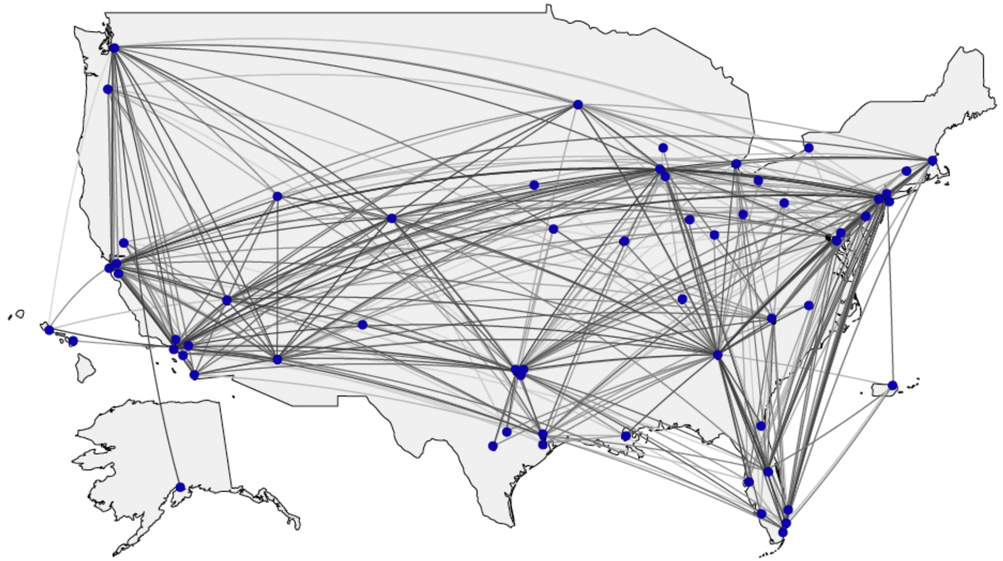
$$\mu^{i\to j}(\sigma_i) = \prod_{k\in\partial i\setminus j} \sum_{\sigma_k} \phi_{ik}(\sigma_i,\sigma_k)\cdot \mu^{k\to i}(\sigma_k)$$

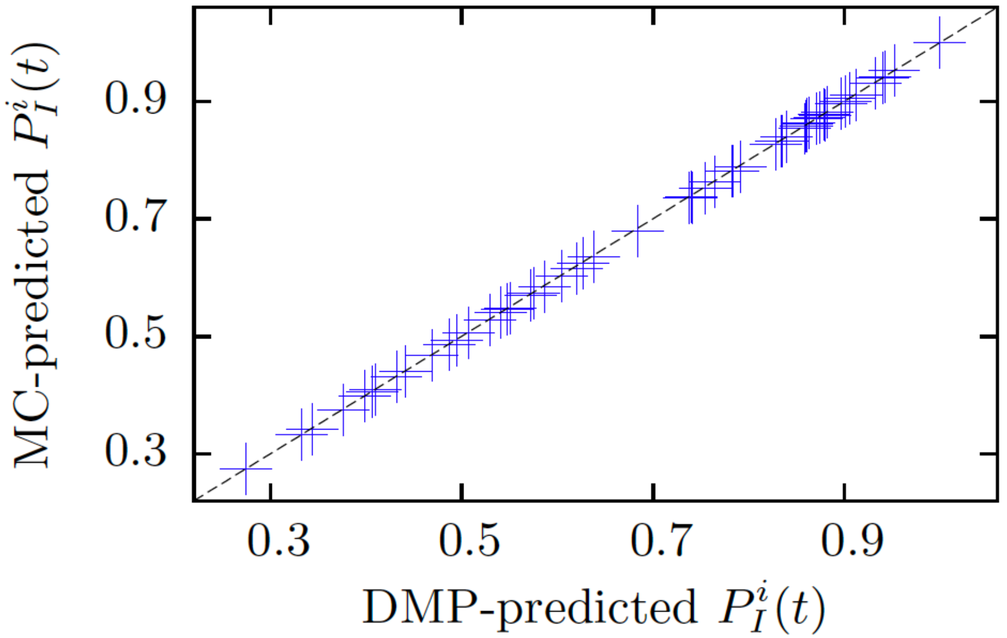
$$\mu^{i}(\sigma_i) = \prod_{k\in\partial i} \mu^{k\to i}(\sigma_k)$$



Lokhov, Mézard, Zdeborová. *PRE* 91.1 (2015): 012811.

Real Network Example



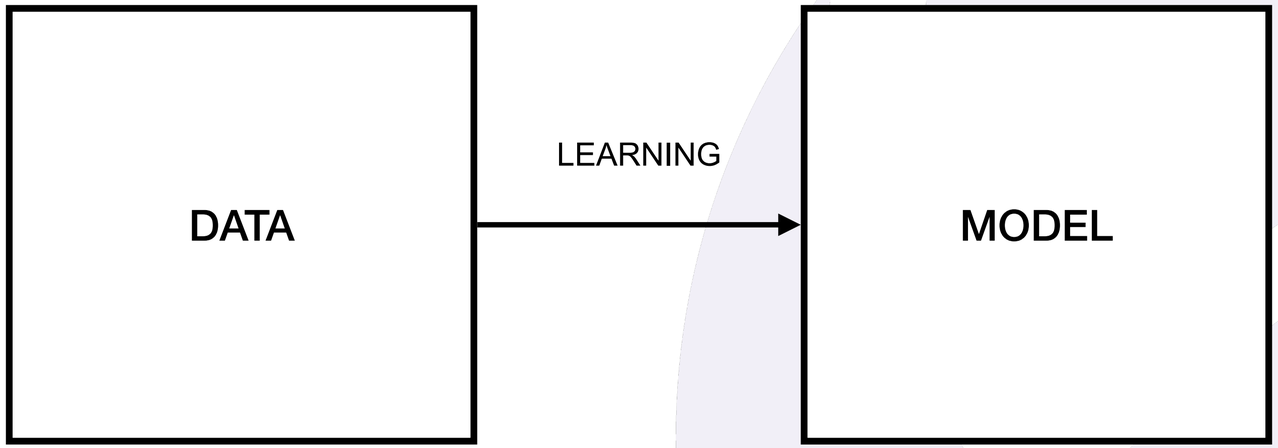


Why not just use Monte Carlo?

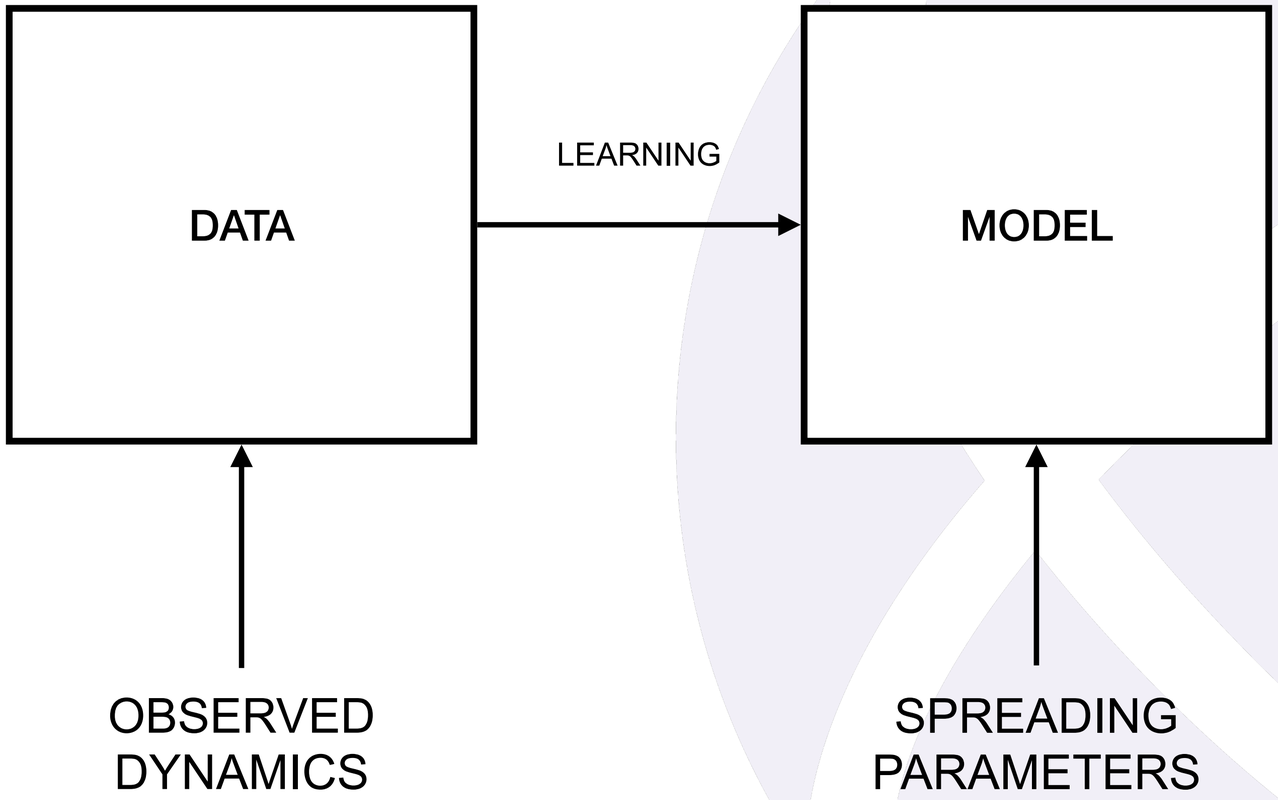
DMP has complexity of a single Monte Carlo run.

Learning

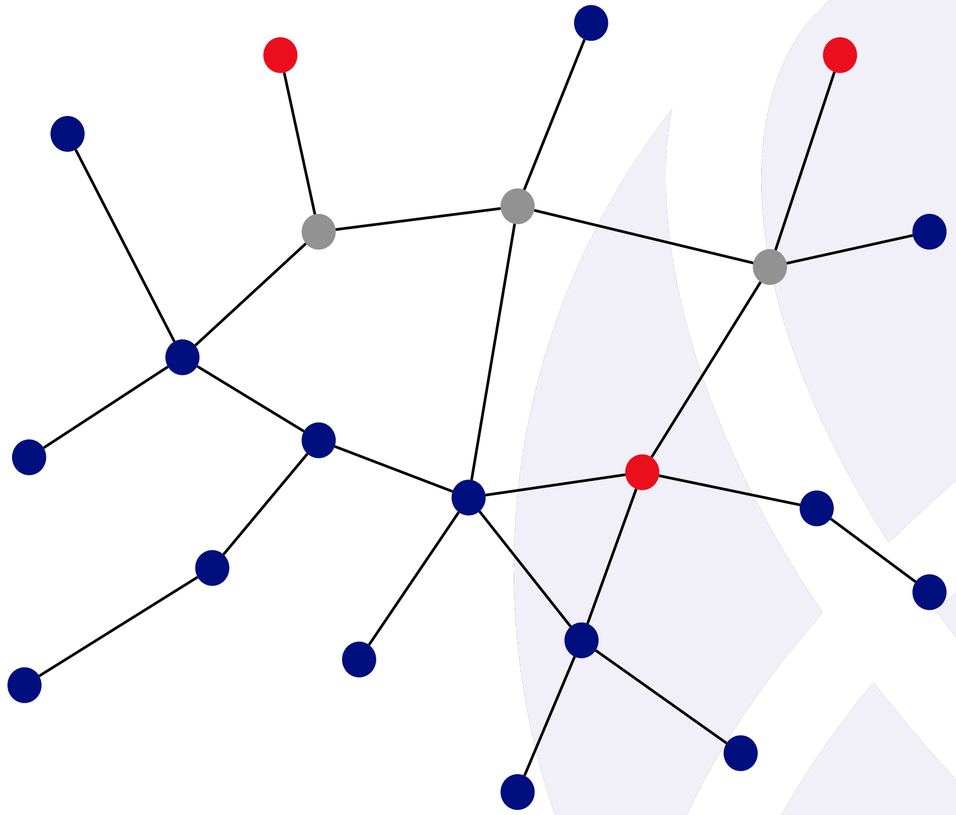
General Idea



General Idea

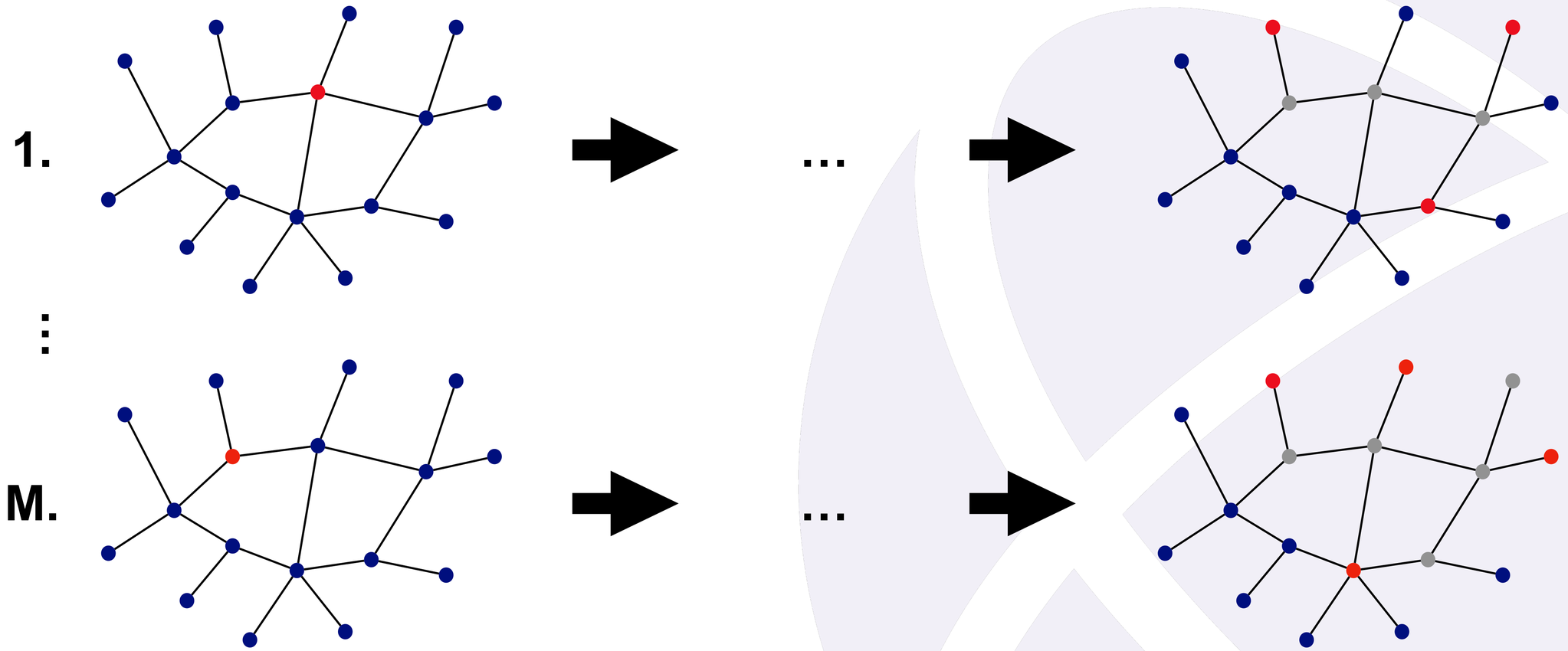


The Problem

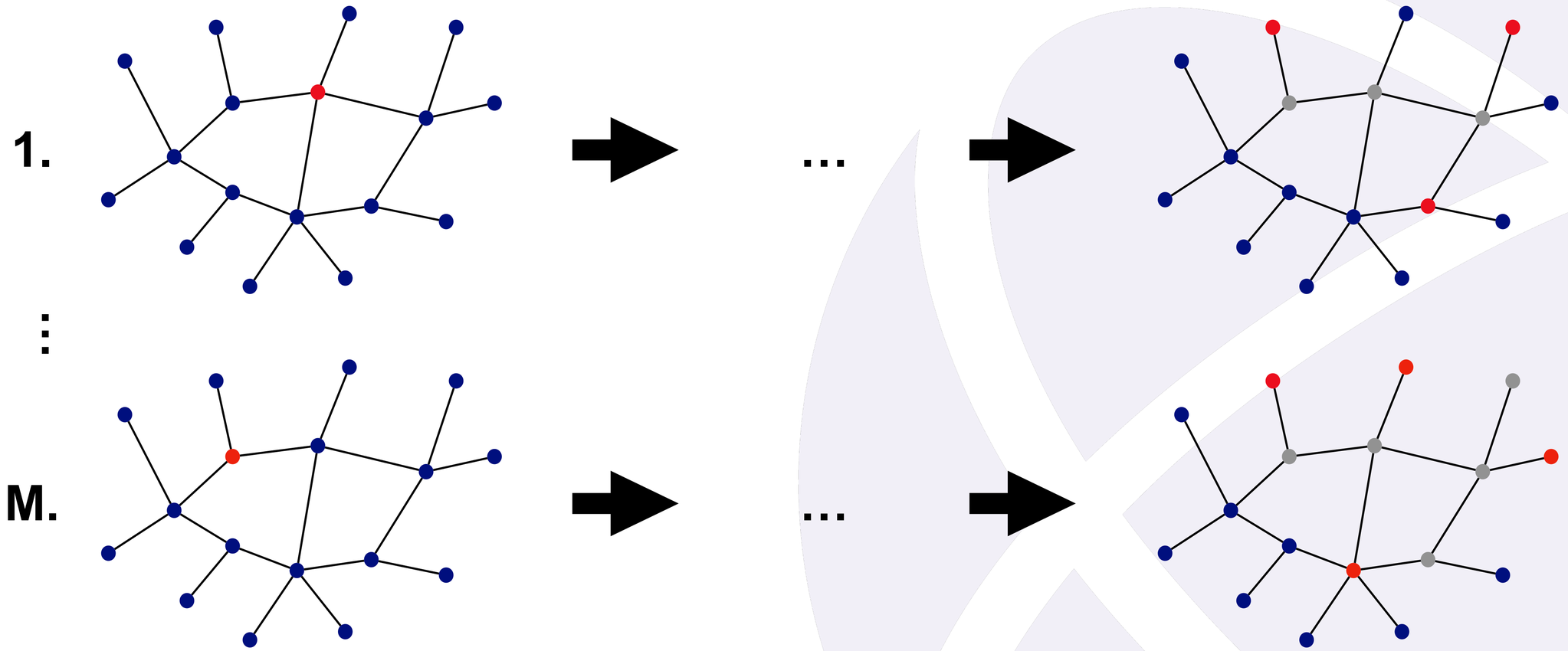


$$\forall_{(i,j)\in E}\ \alpha_{ij} = ?$$

The Data



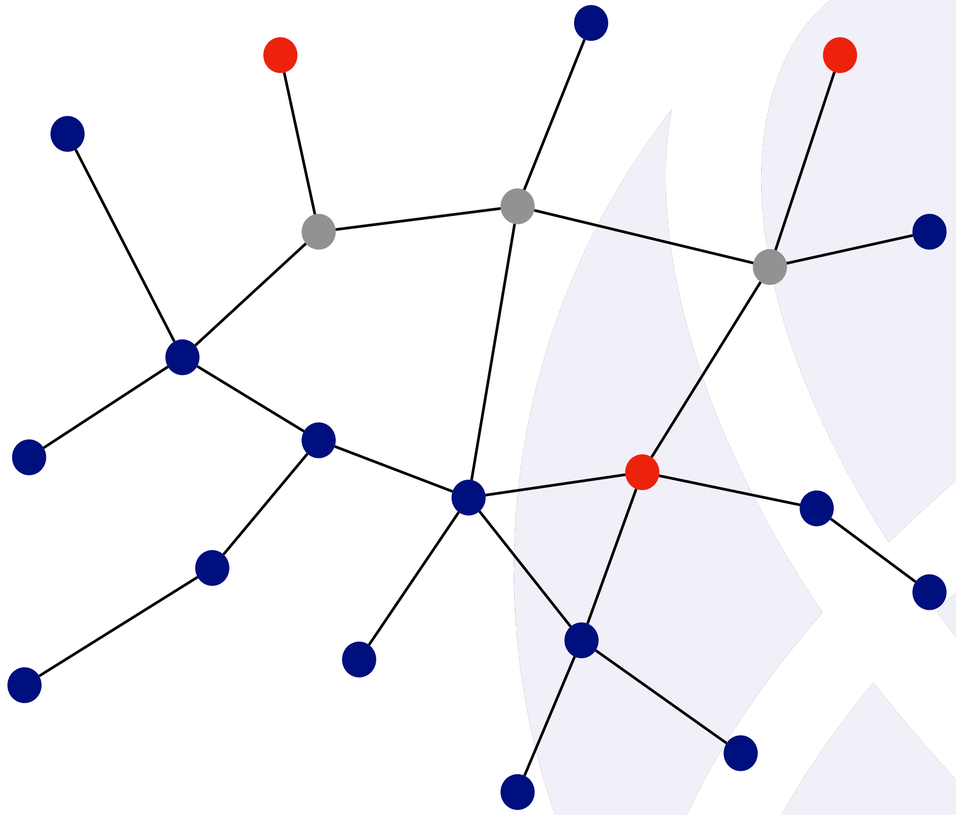
The Data



$$\Sigma = \bigcup_c \Sigma_c = \bigcup_c \{\tau_i^c\}_{i \in V}$$

Setting 1 — Full Information

- Susceptible
- Infected
- Recovered



Maximum Likelihood Approach

$$P\left(\Sigma \mid \{\alpha_{ij}\}\right) = \prod_{c=1}^{M} \prod_{i \in V} \prod_{k \in \partial i} \left(1 - \alpha_{ik} \mathbf{1}_{\tau_k^c \leq \tau_i^c - 2}\right)$$

$$\times \left(1 - \prod_{k \in \partial i} (1 - \alpha_{ik} \mathbf{1}_{\tau_k^c = \tau_i^c - 1}) \mathbf{1}_{\tau_i^c < T}\right)$$

$$\alpha_{ij}^{*} = \min_{\alpha_{ij}} \left(-\log P\left(\Sigma \mid \{\alpha_{ij}\}\right)\right)$$

Maximum Likelihood Approach

$$P\left(\Sigma \mid \{\alpha_{ij}\}\right) = \prod_{c=1}^{M} \prod_{i \in V} \prod_{k \in \partial i} \left(1 - \alpha_{ik} \mathbf{1}_{\tau_k^c \leq \tau_i^c - 2}\right)$$

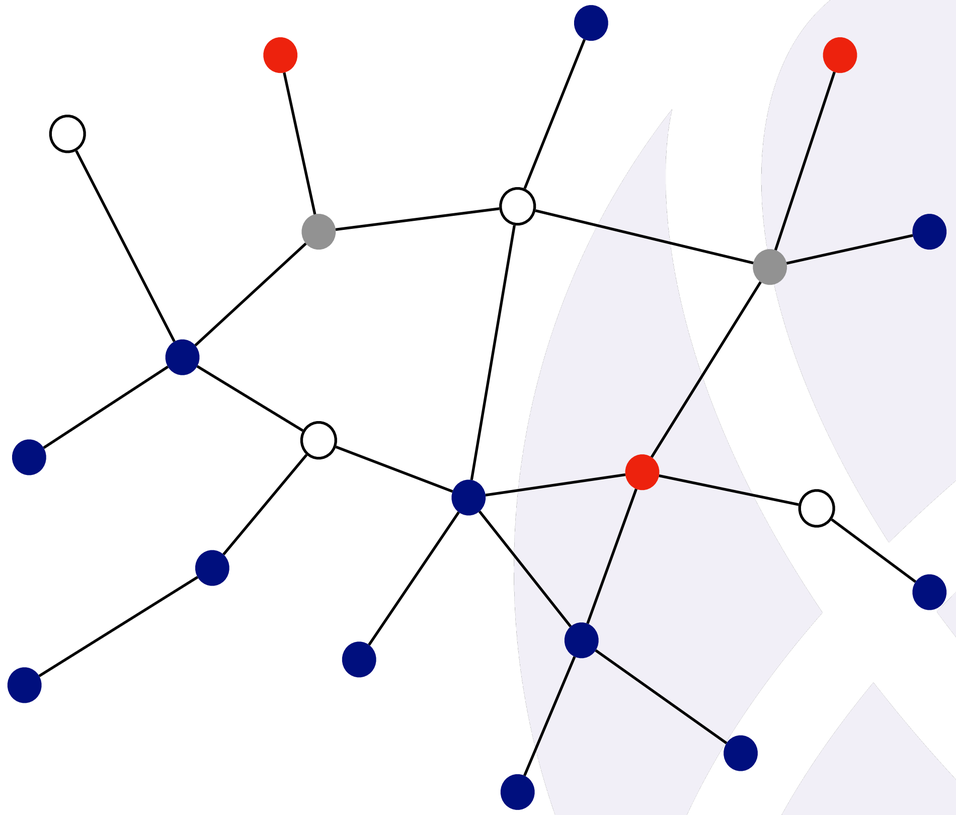
$$\times \left(1 - \prod_{k \in \partial i} (1 - \alpha_{ik} \mathbf{1}_{\tau_k^c = \tau_i^c - 1}) \mathbf{1}_{\tau_i^c < T}\right)$$

$$\alpha_{ij}^* = \min_{\alpha_{ij}} \left(-\log P\left(\Sigma \mid \{\alpha_{ij}\}\right)\right) \longleftarrow$$

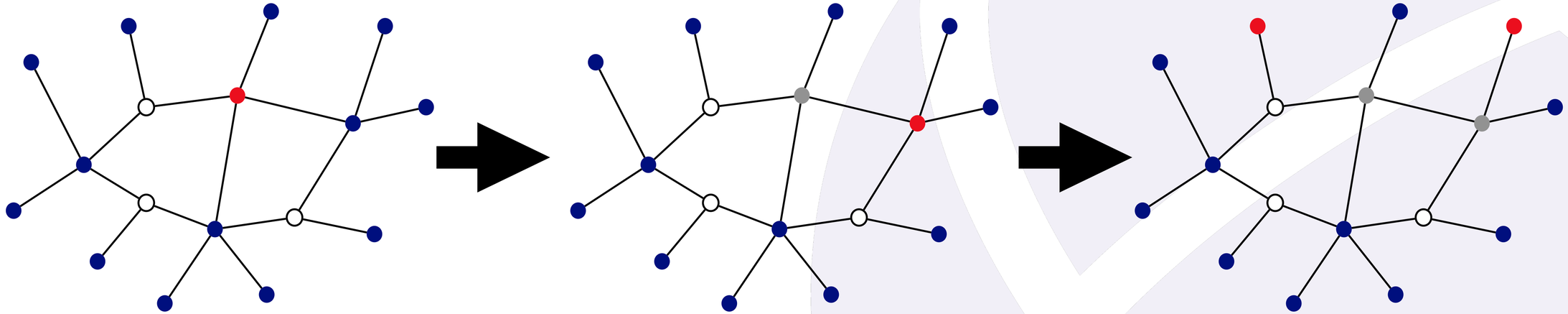
minimizing KL divergence between model and empirical probability of cascades

Setting 2 — Unobserved Nodes

- Susceptible
- Infected
- Recovered
- Unobserved



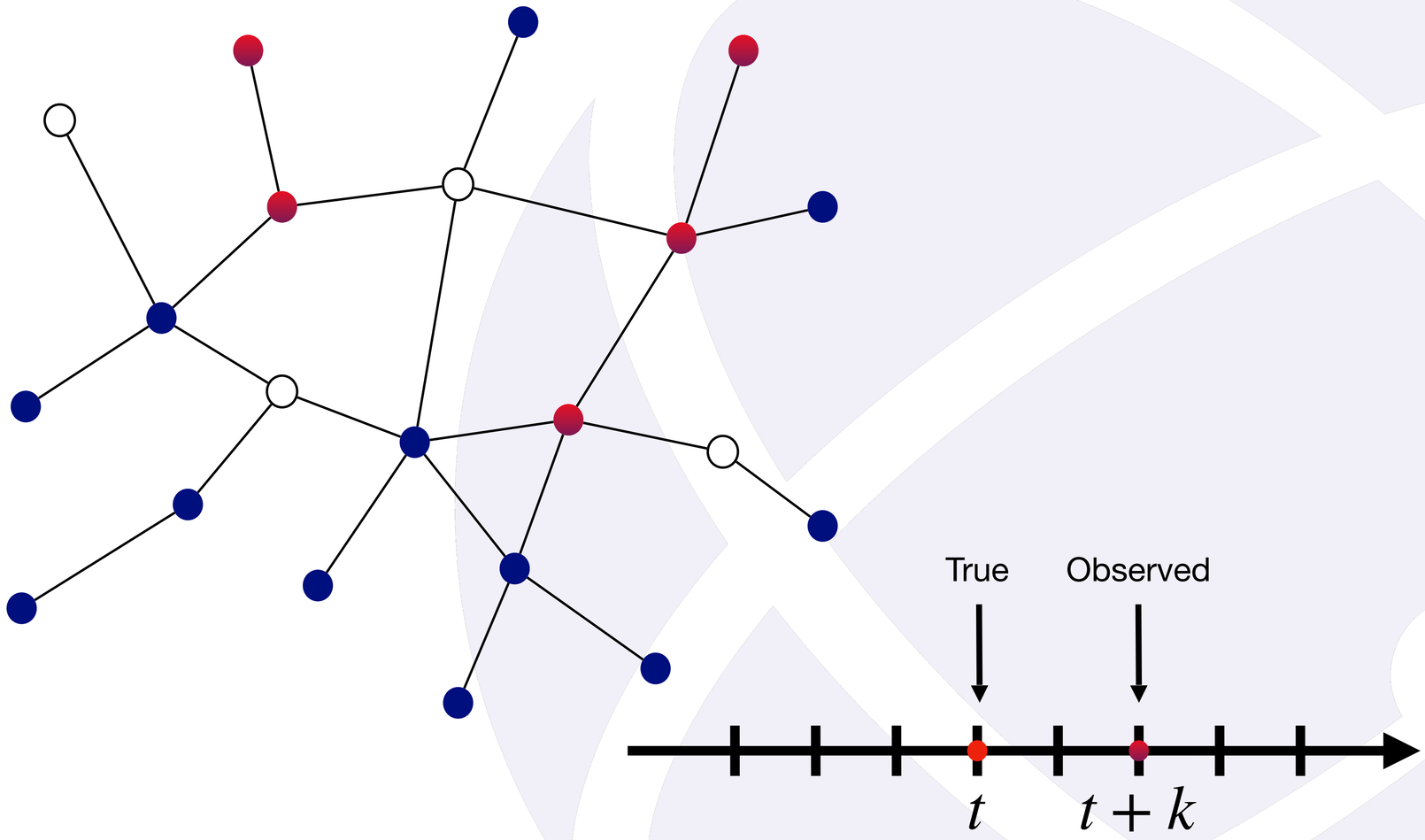
Partial observation



Maximum Likelihood Estimation
becomes numerically intractable!

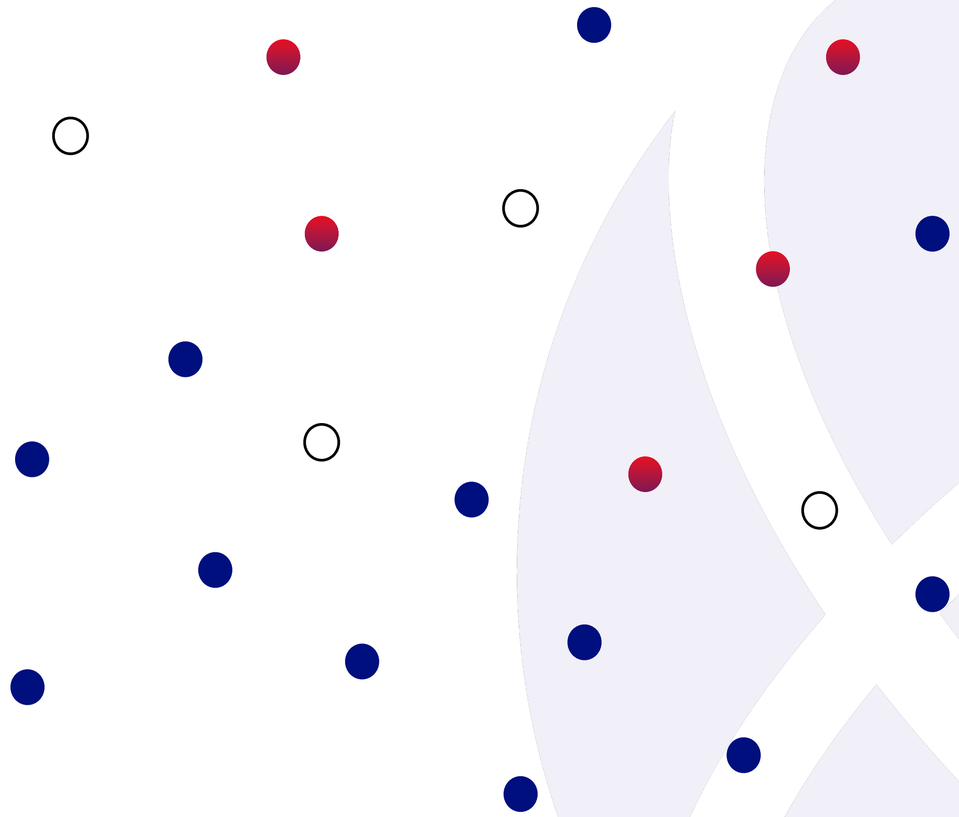
Setting 3 — Uncertainty in Activation Times

- Susceptible
- Infected
- Recovered
- Unobserved
- Uncertain

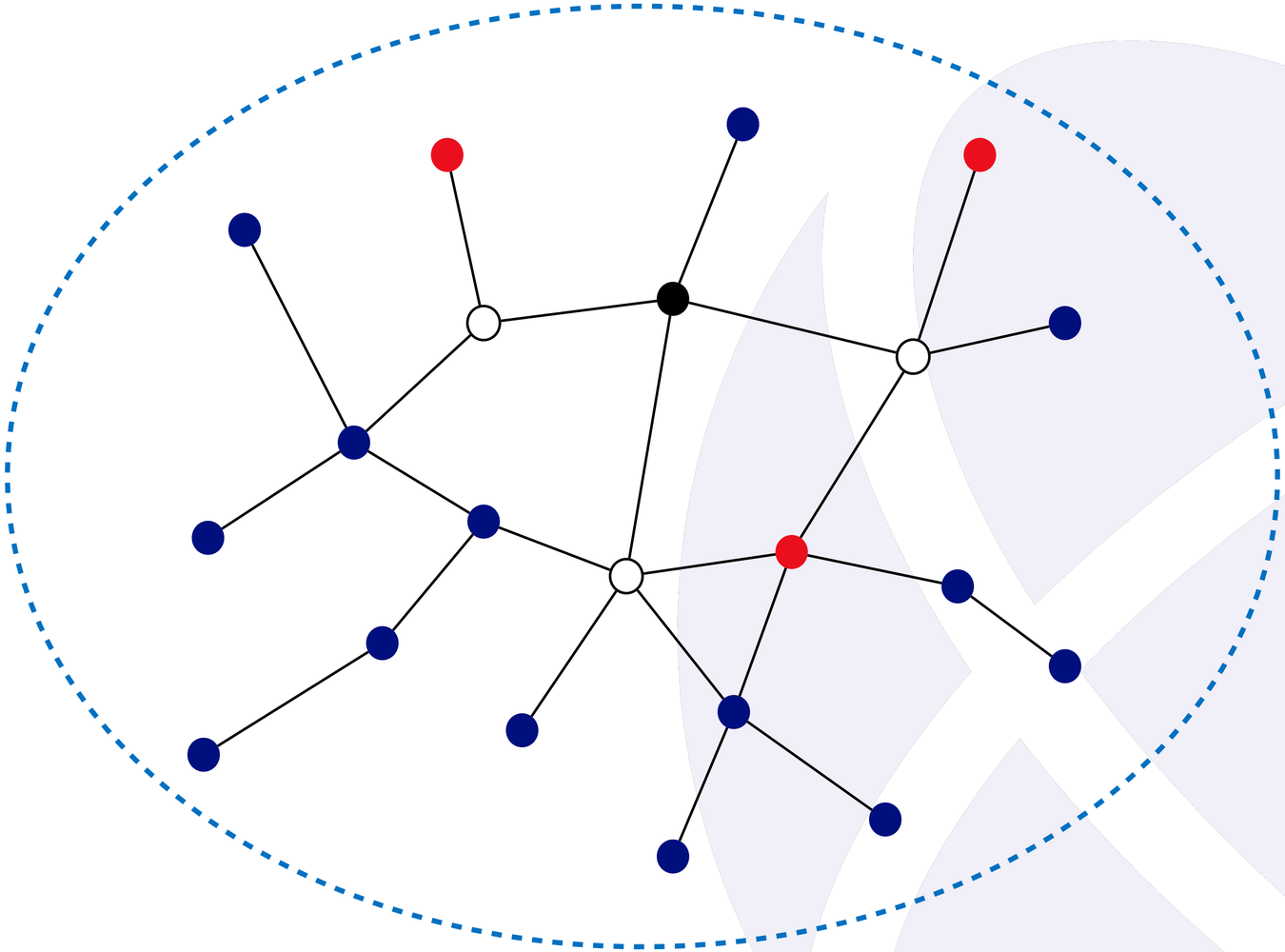


Setting 4 — Unknown Couplings and Uncertainty

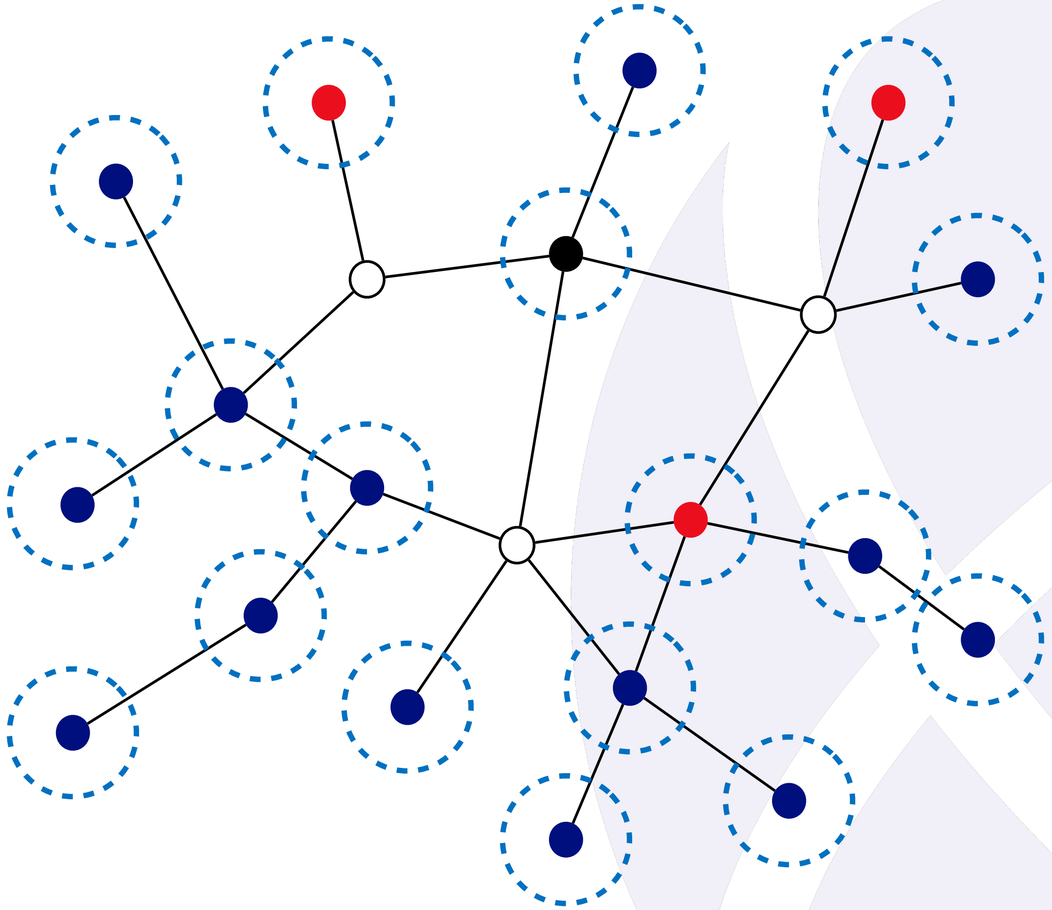
- Susceptible
- Infected
- Recovered
- Unobserved
- Uncertain



Marginals



Marginals



Objective

$$\mathcal{O} = -\sum_{c \in C} \sum_{i \in O} \log \mu_i^c(\tau_i^c)$$

$$\alpha_{ij}^* = \min_{\alpha_{ij}} \left(\mathcal{O} \right)$$

Wilinski, Lokhov. *ICML* (2021).

Objective

$$\mathcal{O} = -\sum_{c \in C} \sum_{i \in O} \log \mu_i^c(\tau_i^c)$$

marginal probability for node *i* under cascade *c*

$$\alpha_{ij}^* = \min_{\alpha_{ij}} \left(\mathcal{O} \right)$$

Wilinski, Lokhov. *ICML* (2021).

Objective

$$\mathcal{O} = -\sum_{c \in C} \sum_{i \in O} \log \mu_i^c(\tau_i^c)$$

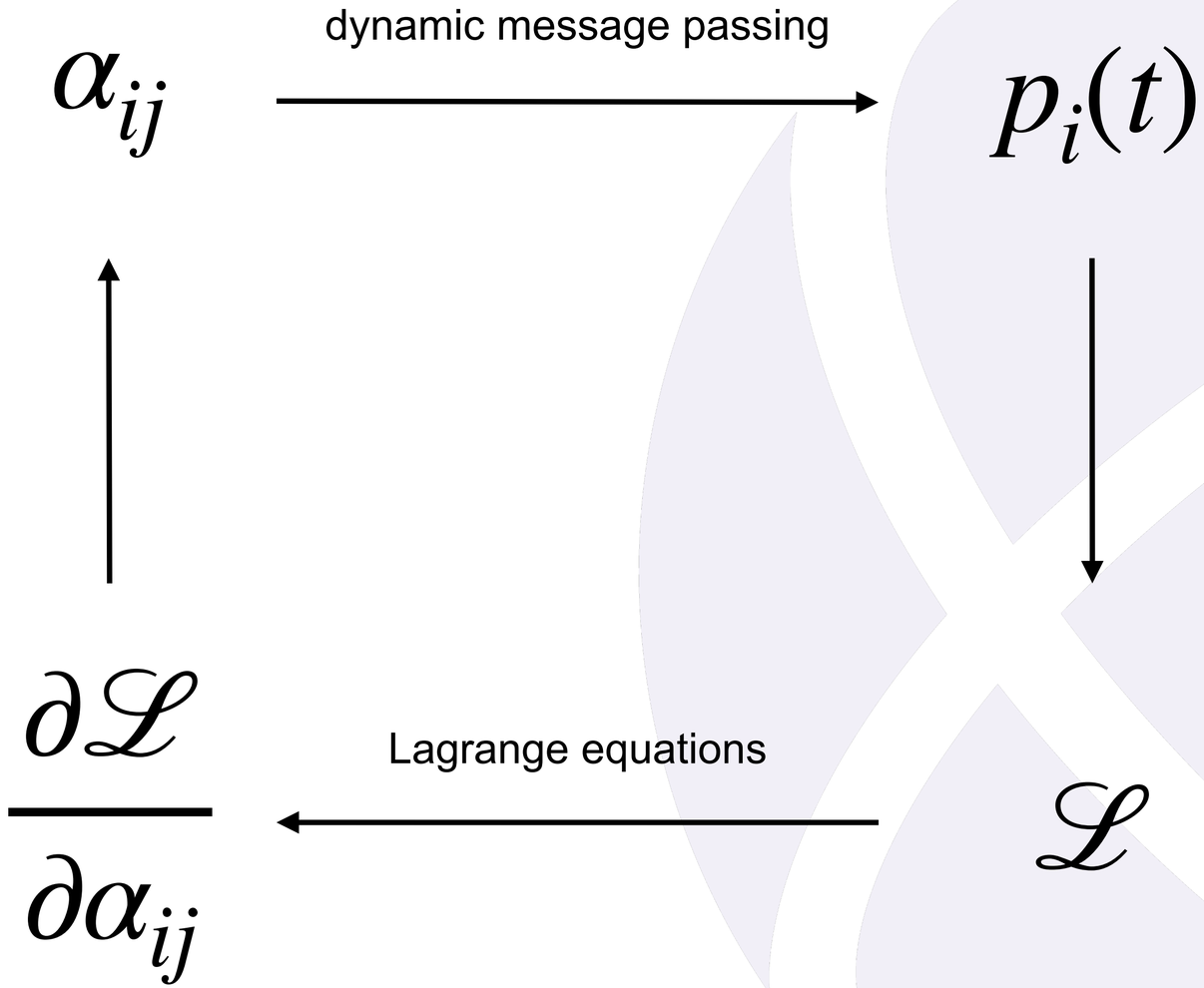
marginal probability for node *i* under cascade *c*

$$\alpha_{ij}^* = \min_{\alpha_{ij}} \left(\mathcal{O} \right)$$

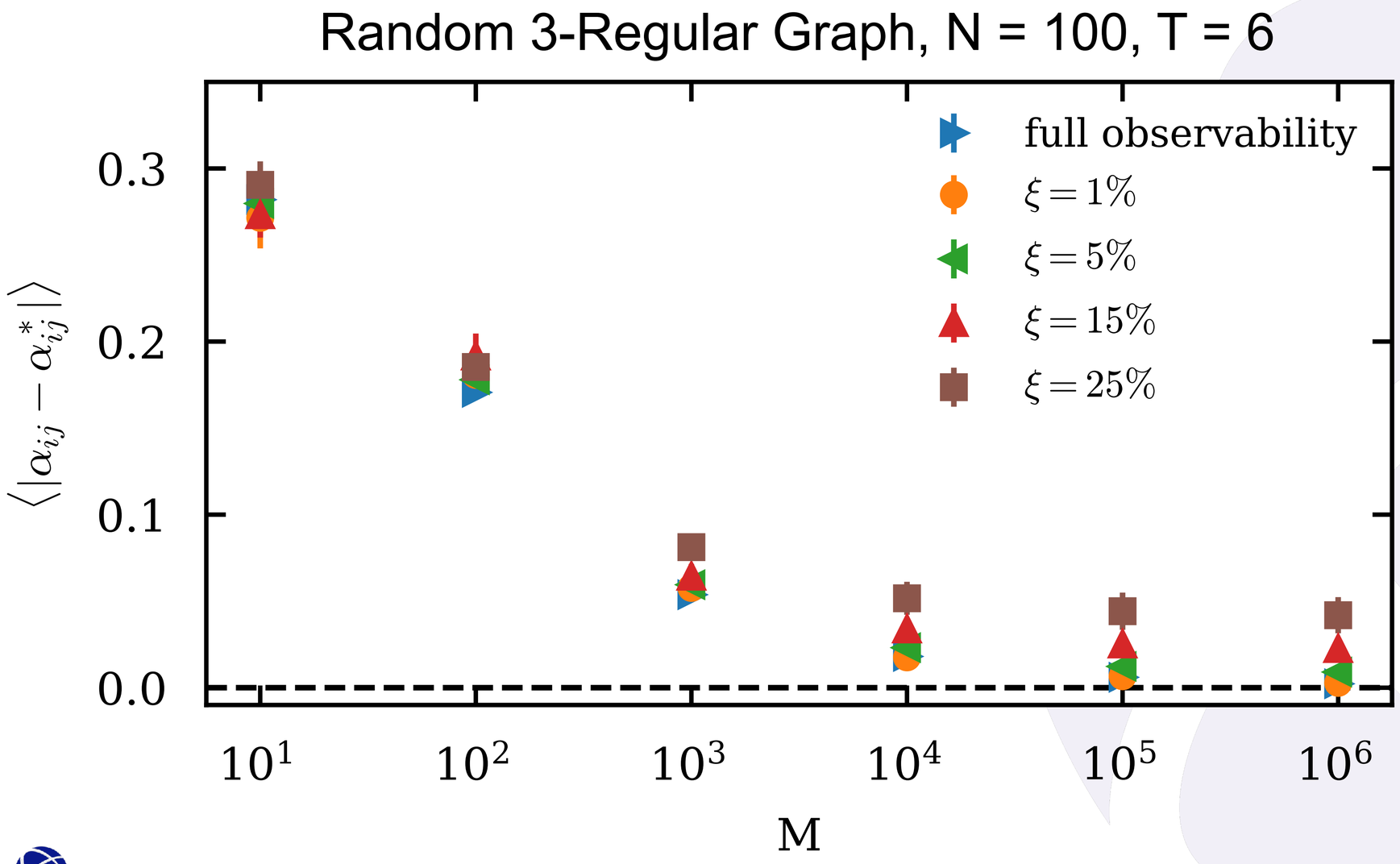
minimising KL divergence between model and empirical marginals

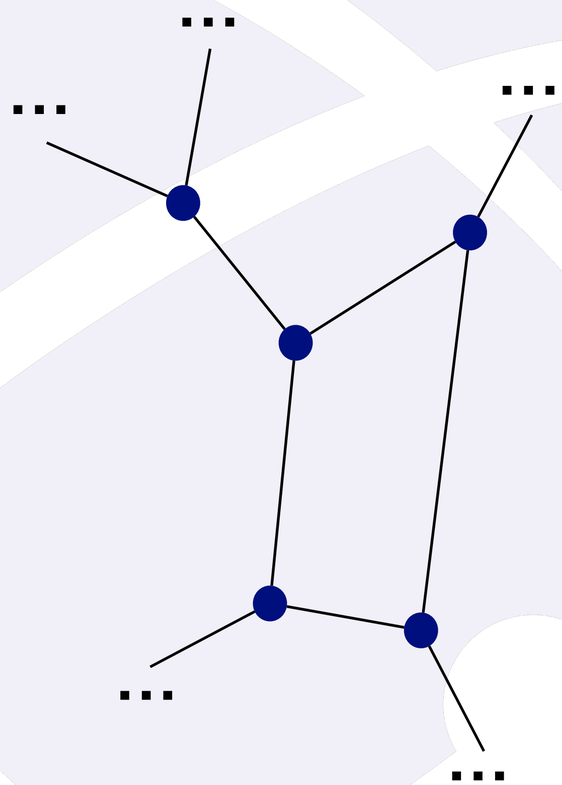
Wilinski, Lokhov. *ICML* (2021).

Forward – Backward Propagation

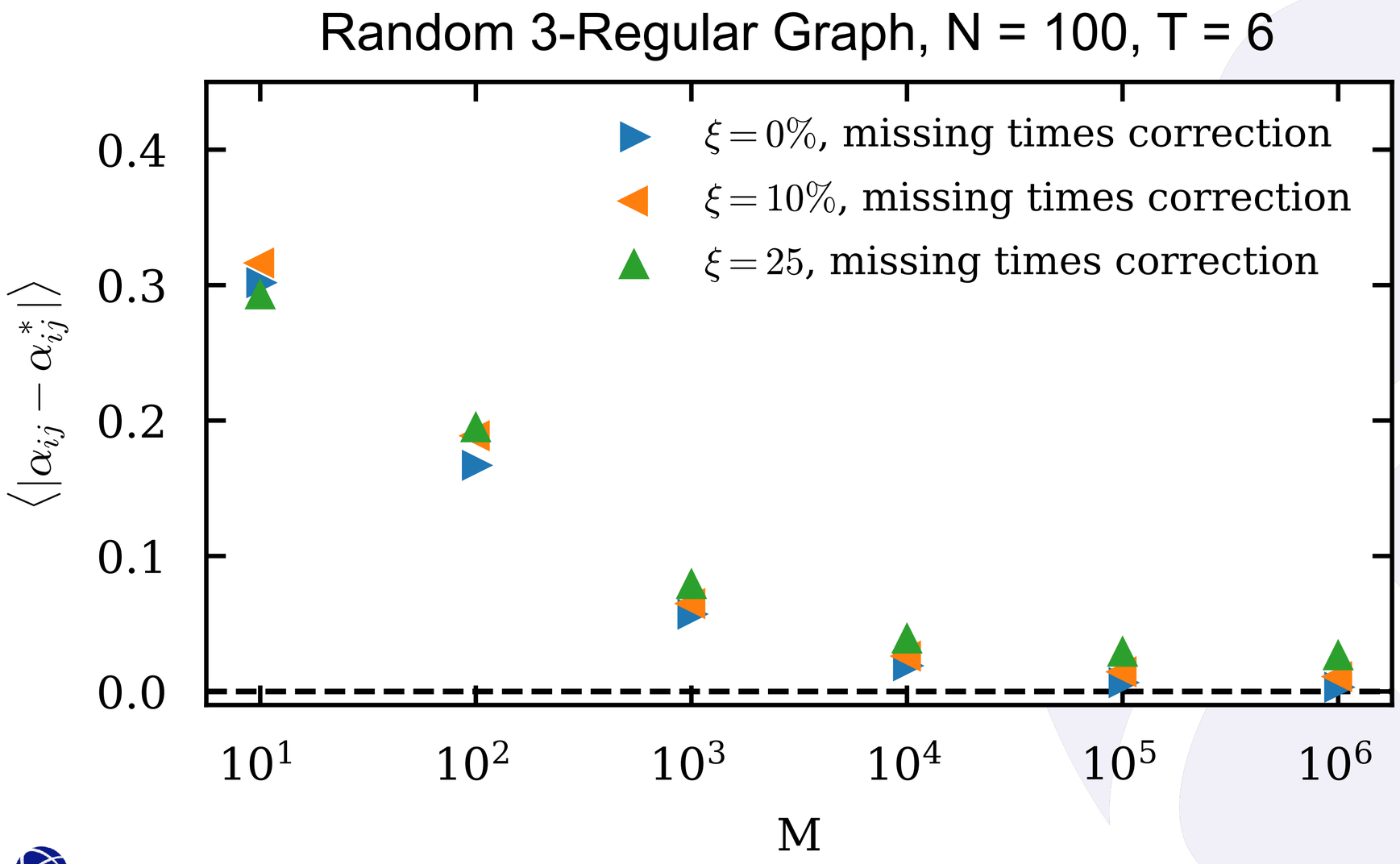


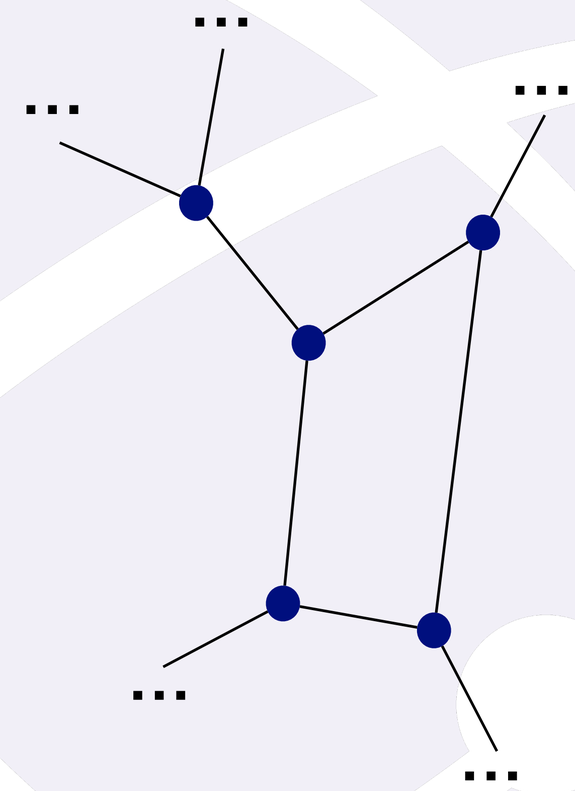
Unobserved Nodes



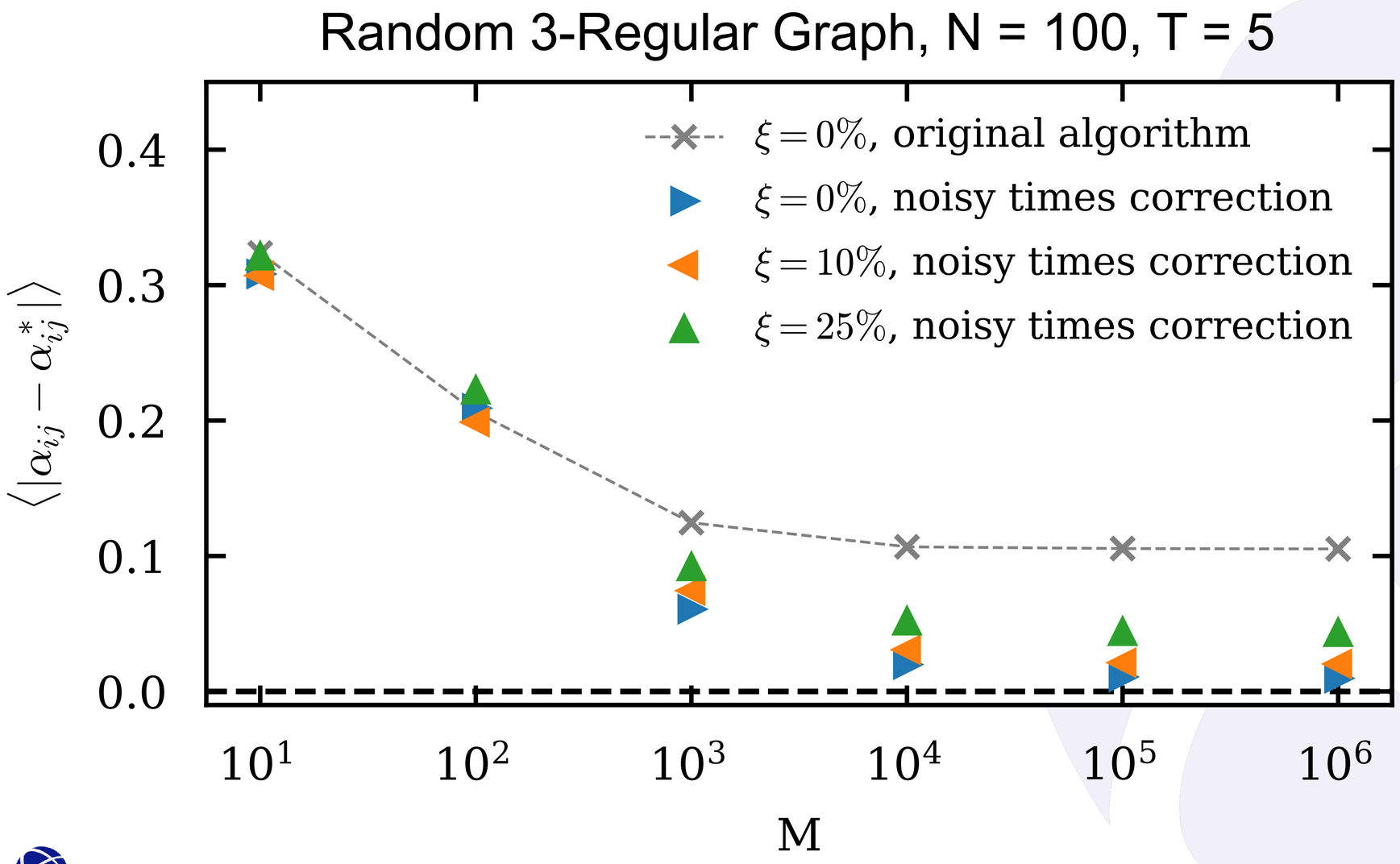


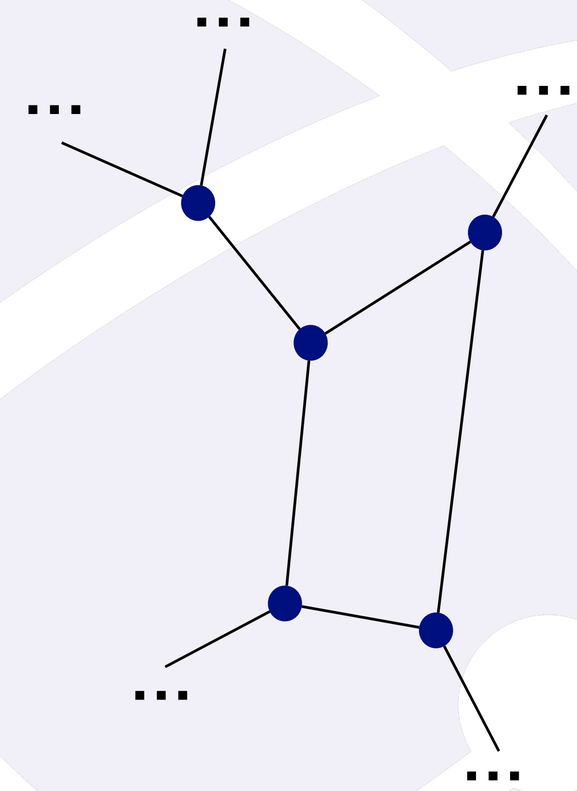
Missing Times



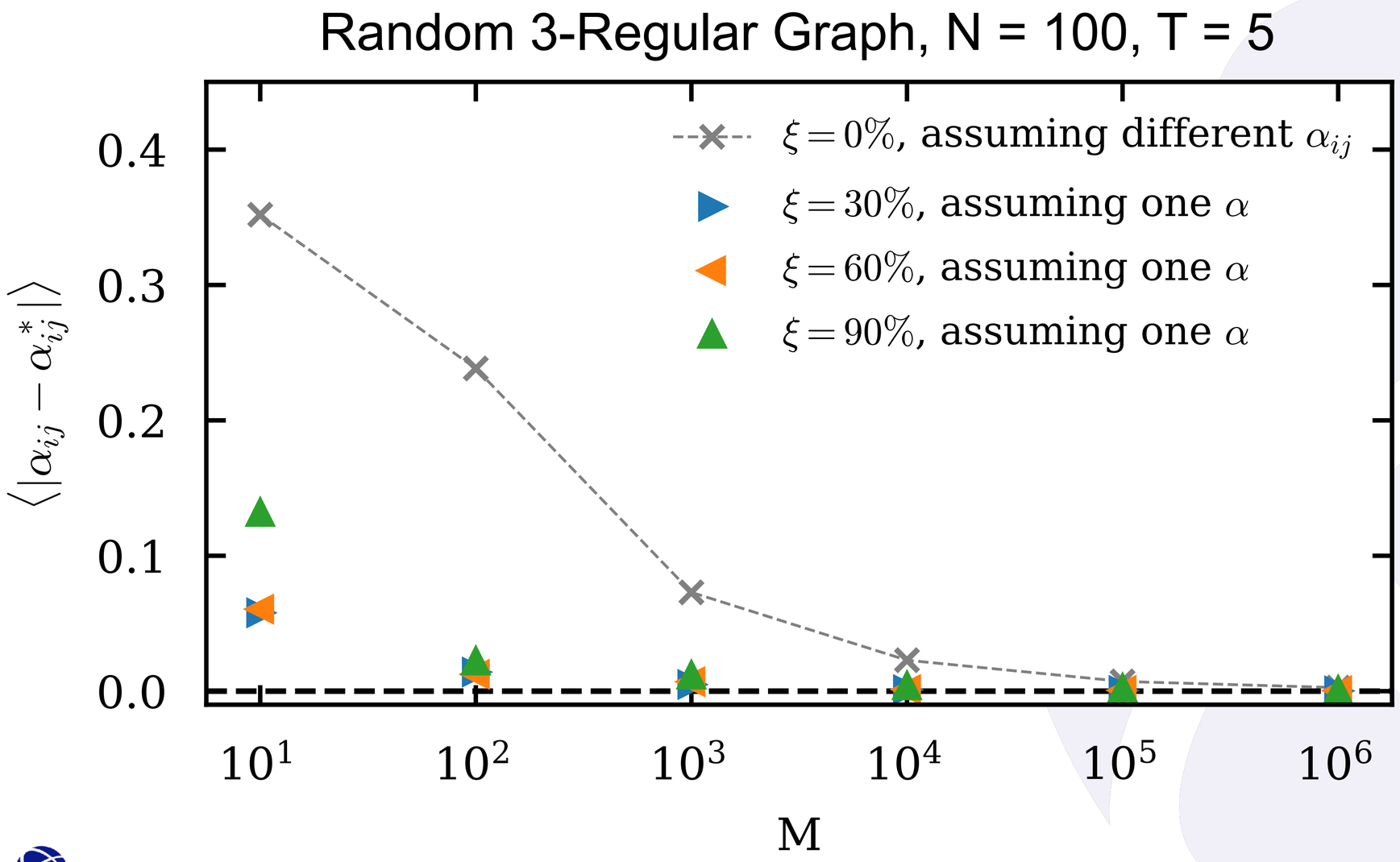


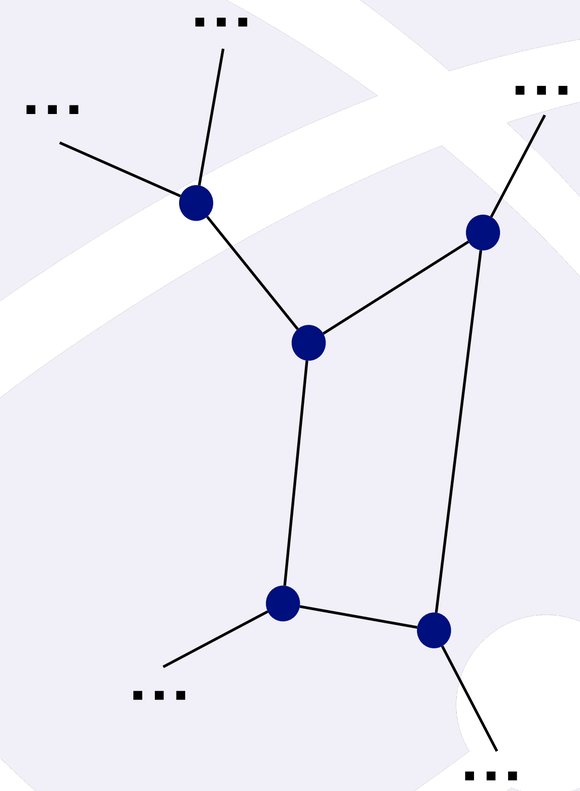
Noisy Times


Random 3-Regular Graph, N = 100, T = 5
$\xi = 0\%$, original algorithm
$\xi = 0\%$, noisy times correction
$\xi = 10\%$, noisy times correction
$\xi = 25\%$, noisy times correction
$\langle |\alpha_{ij} - \alpha_{ij}^*| \rangle$
0.0
0.1
0.2
0.3
0.4
10^1
10^2
10^3
10^4
10^5
10^6
M

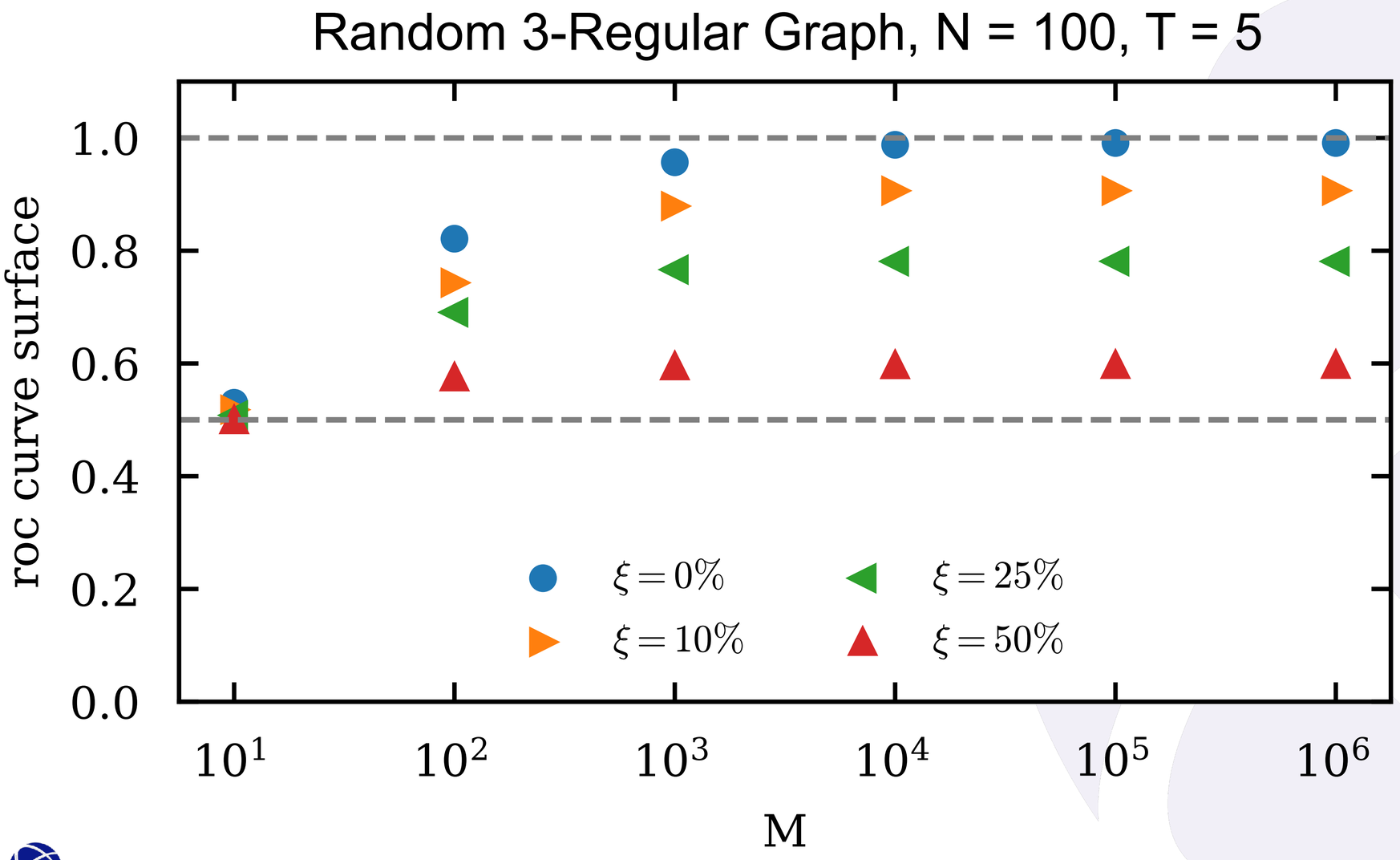


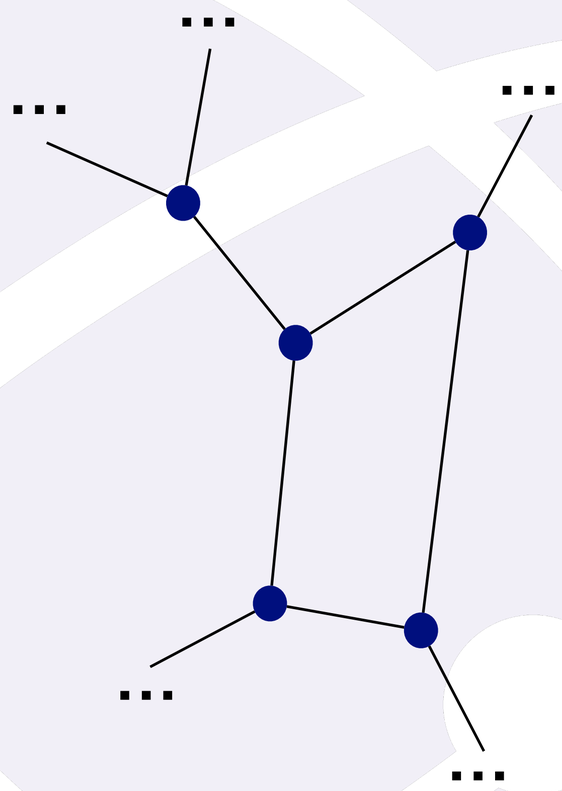
Simple Graphs



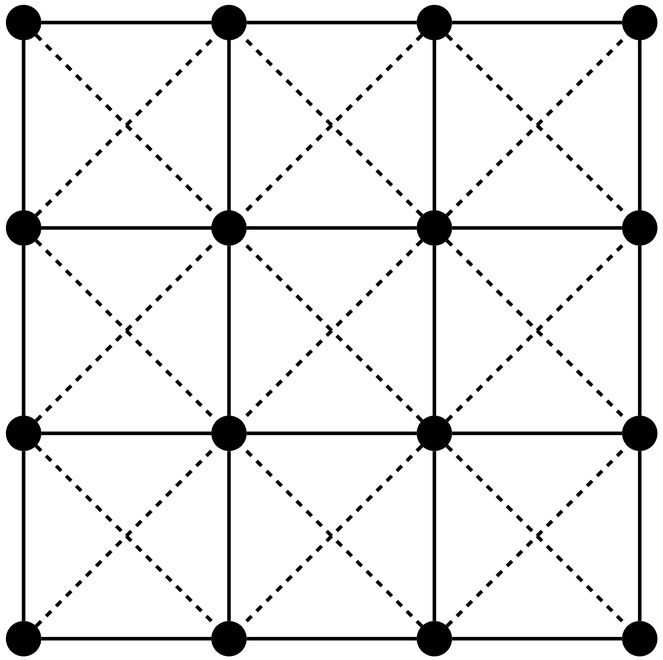


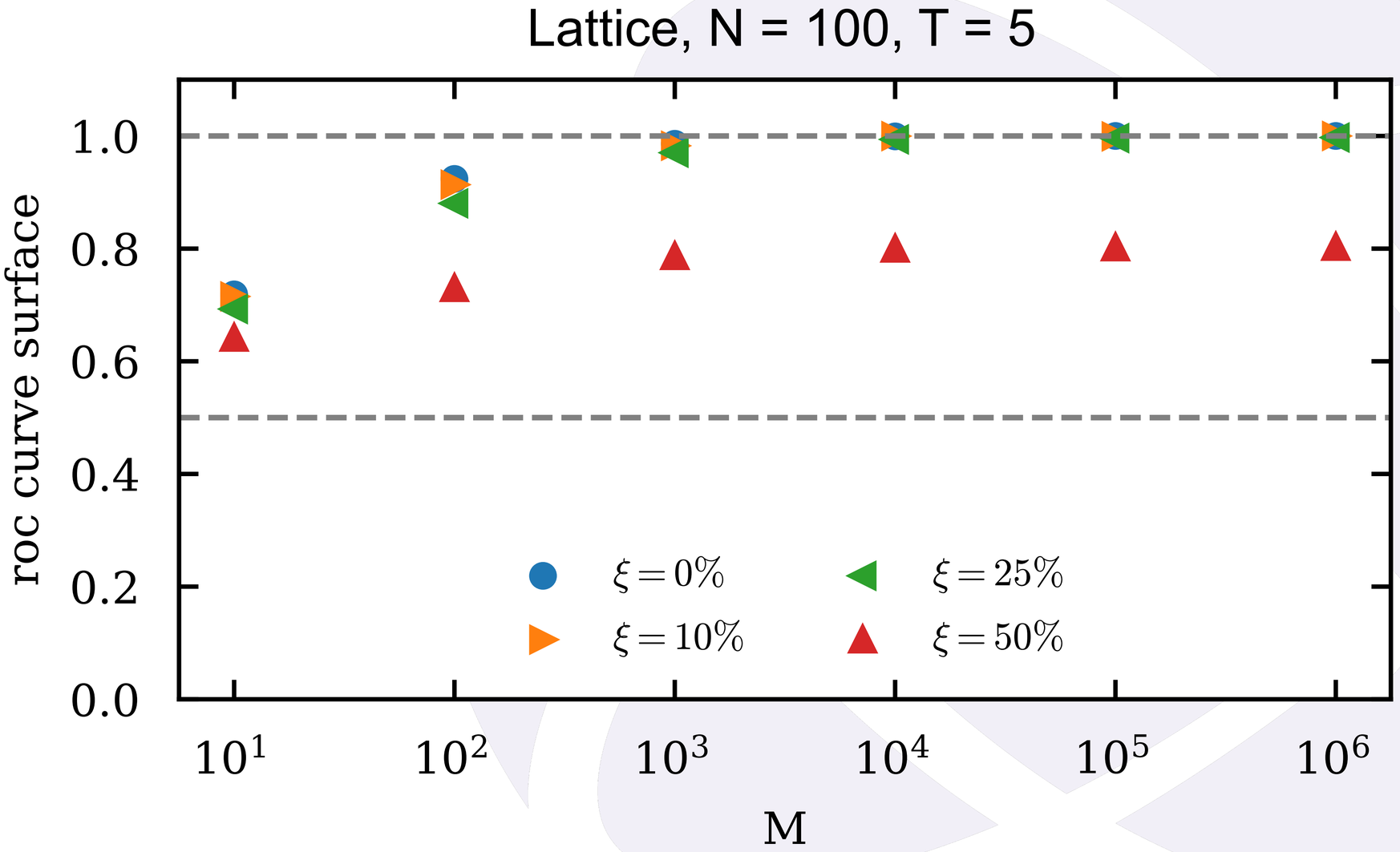
Unknown Couplings



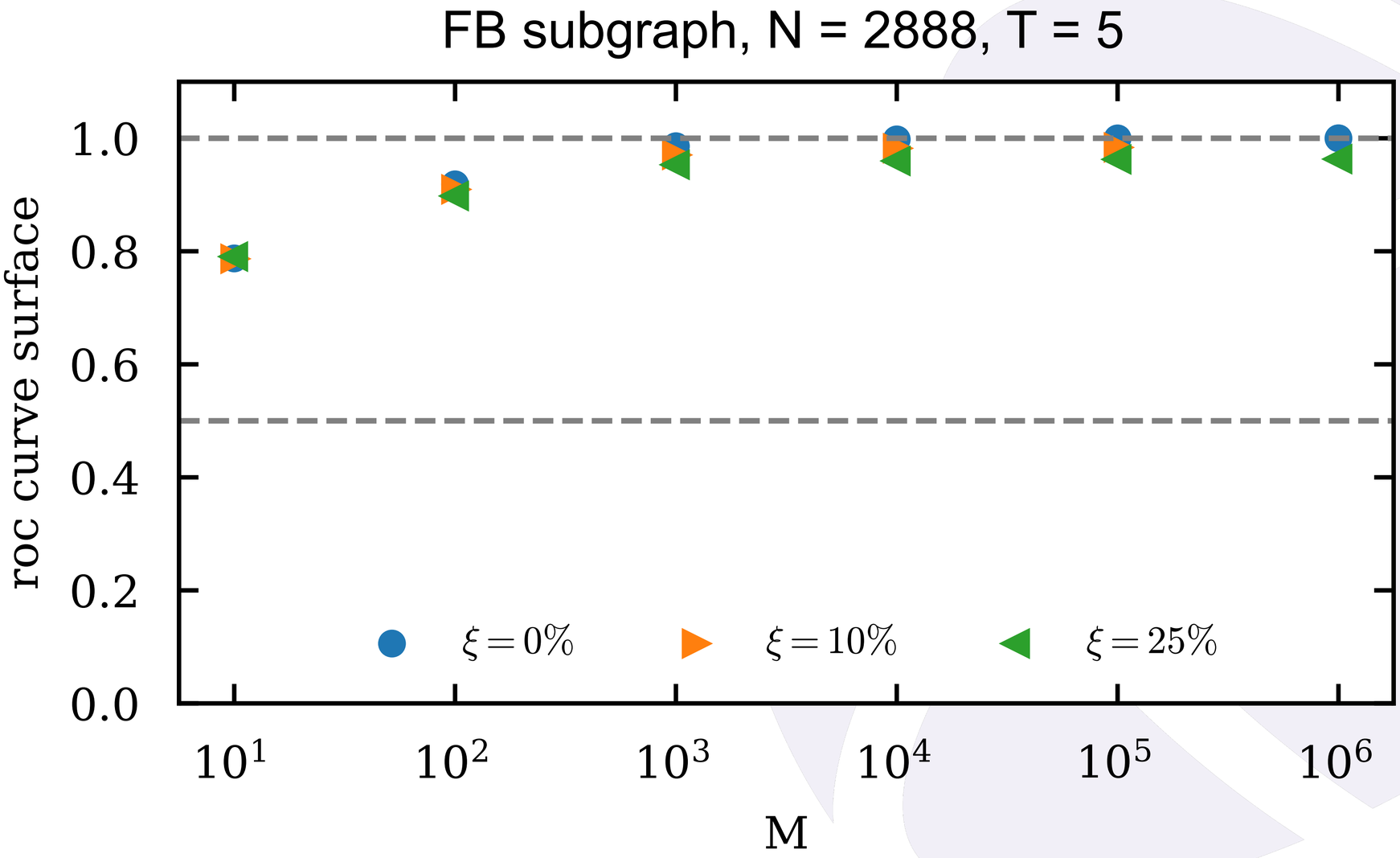


Extended Lattice



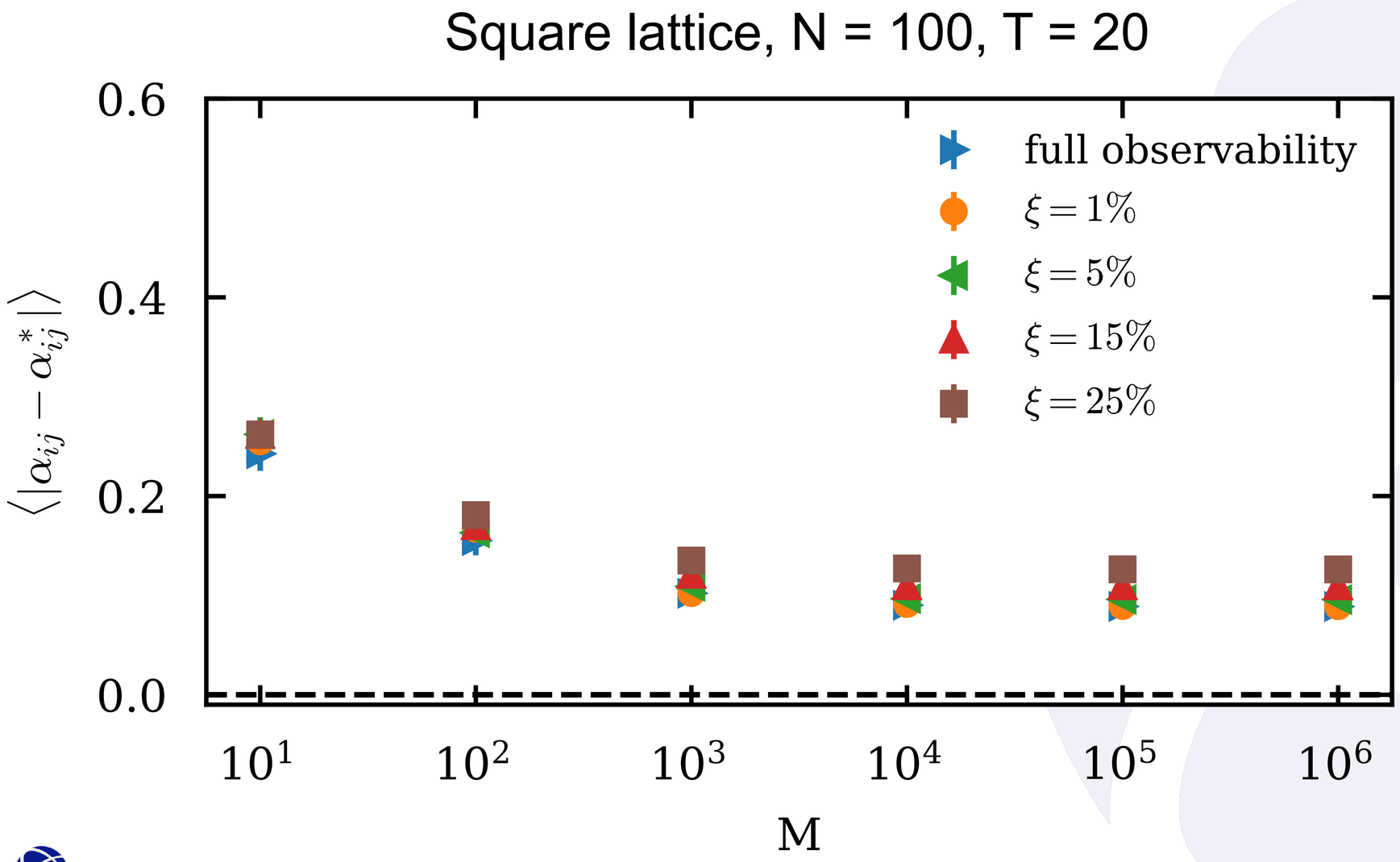


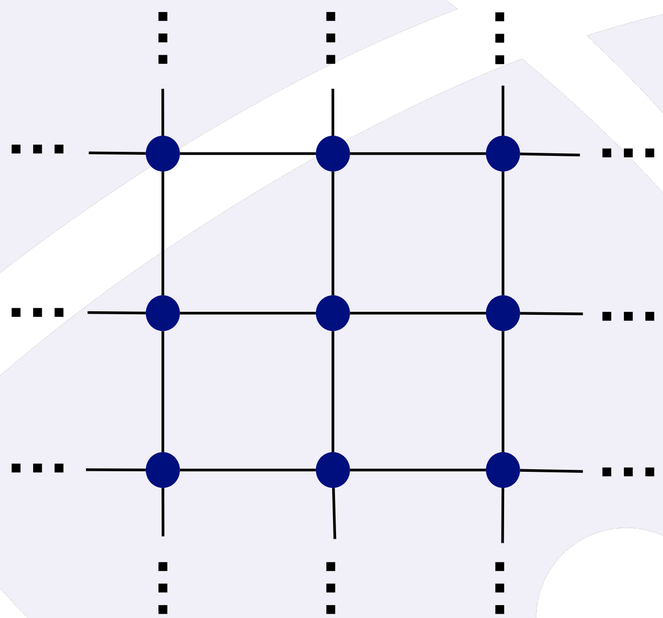
Full Setting



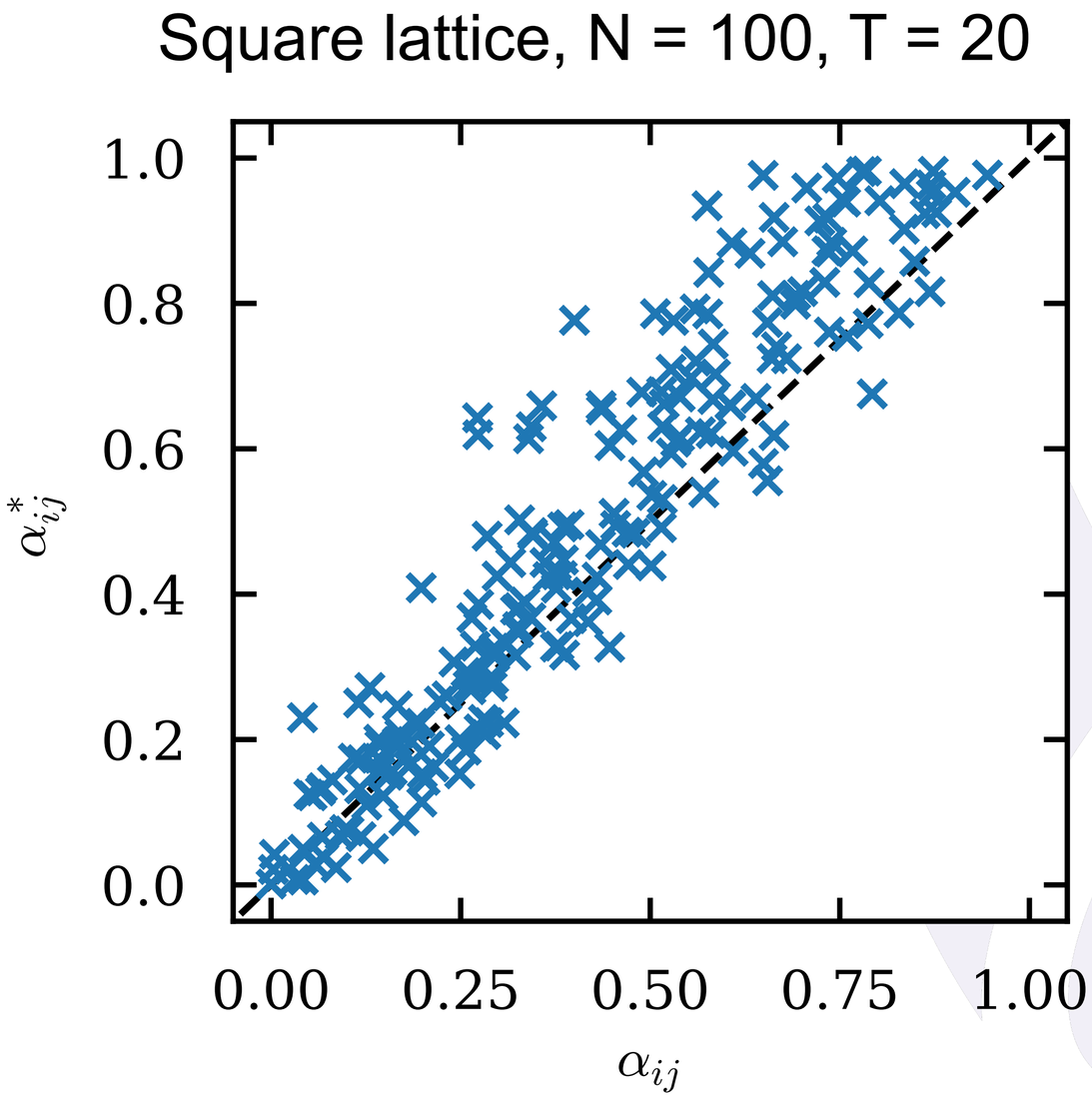
Learning for Prediction

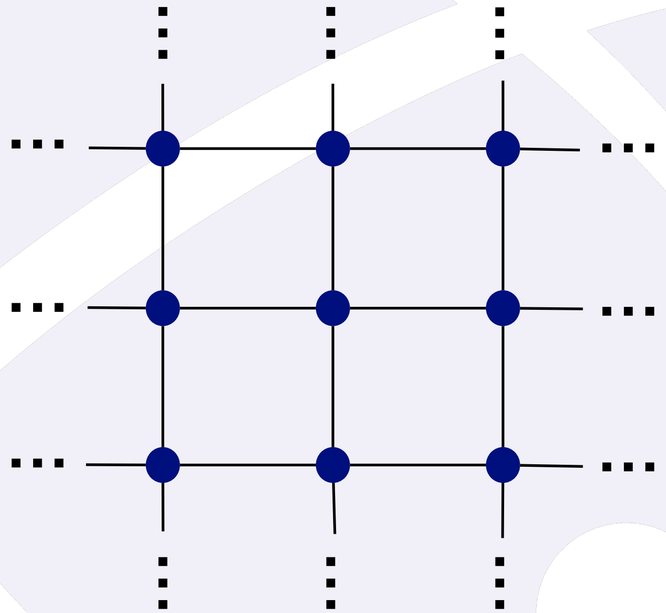
Adversarial case — square lattice





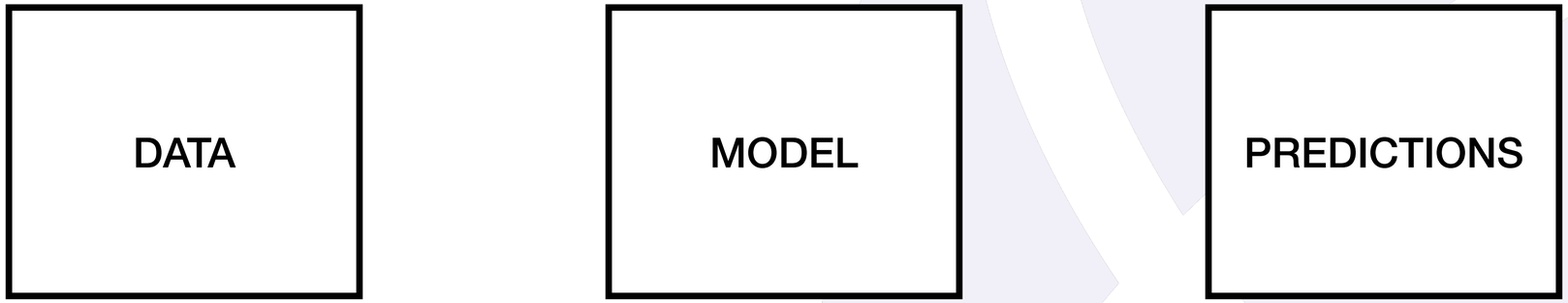
Adversarial case — square lattice





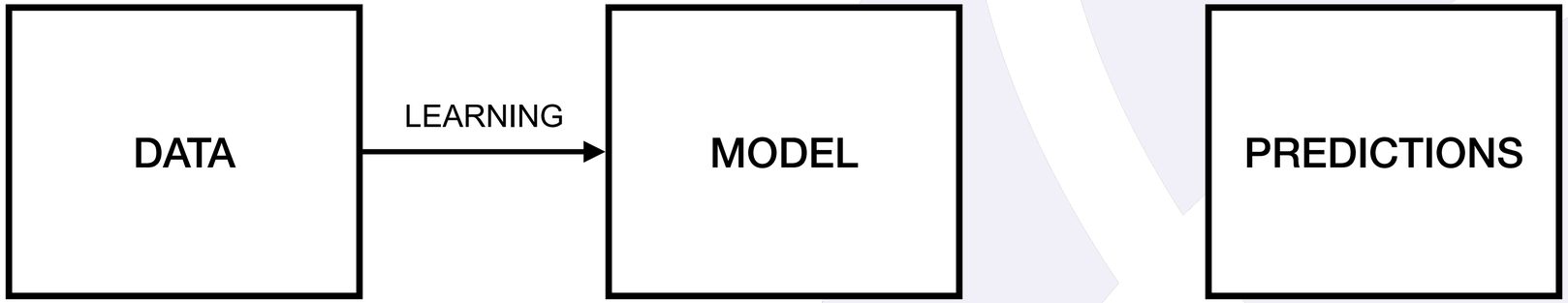
Adversarial case — square lattice

Do we really care about alphas?



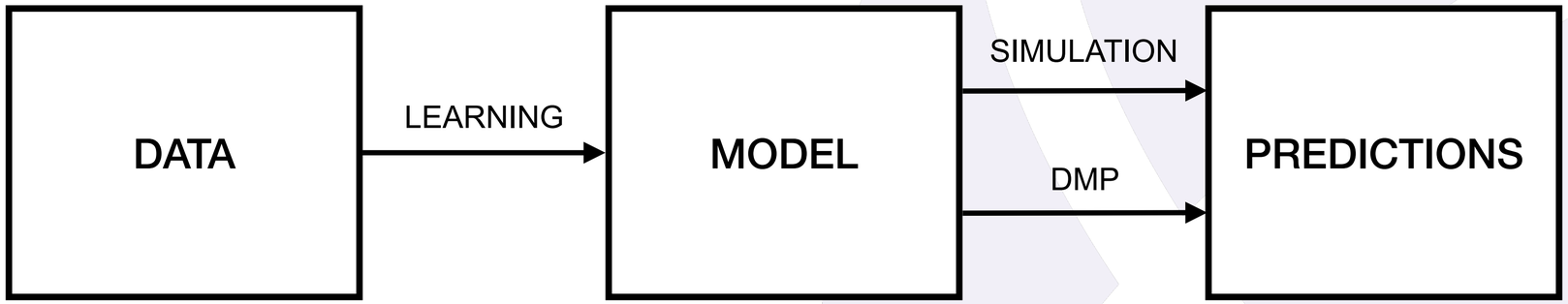
Adversarial case — square lattice

Do we really care about alphas?



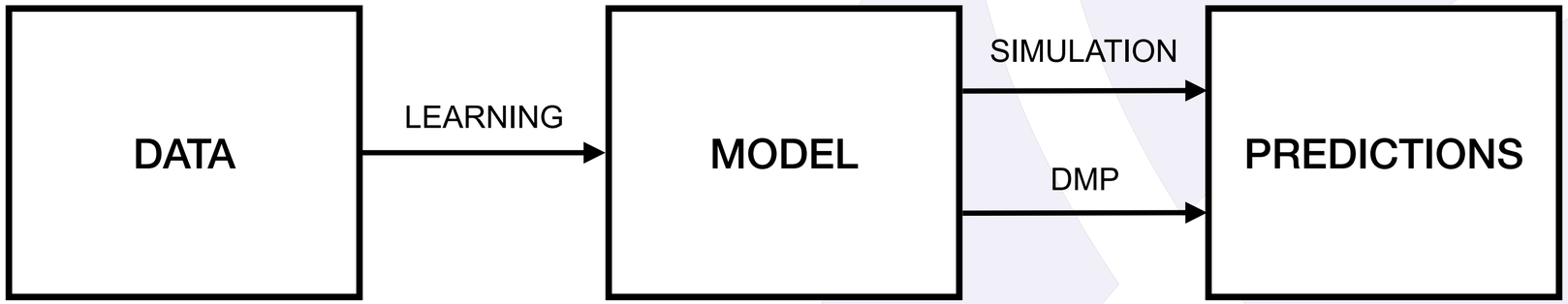
Adversarial case — square lattice

Do we really care about alphas?



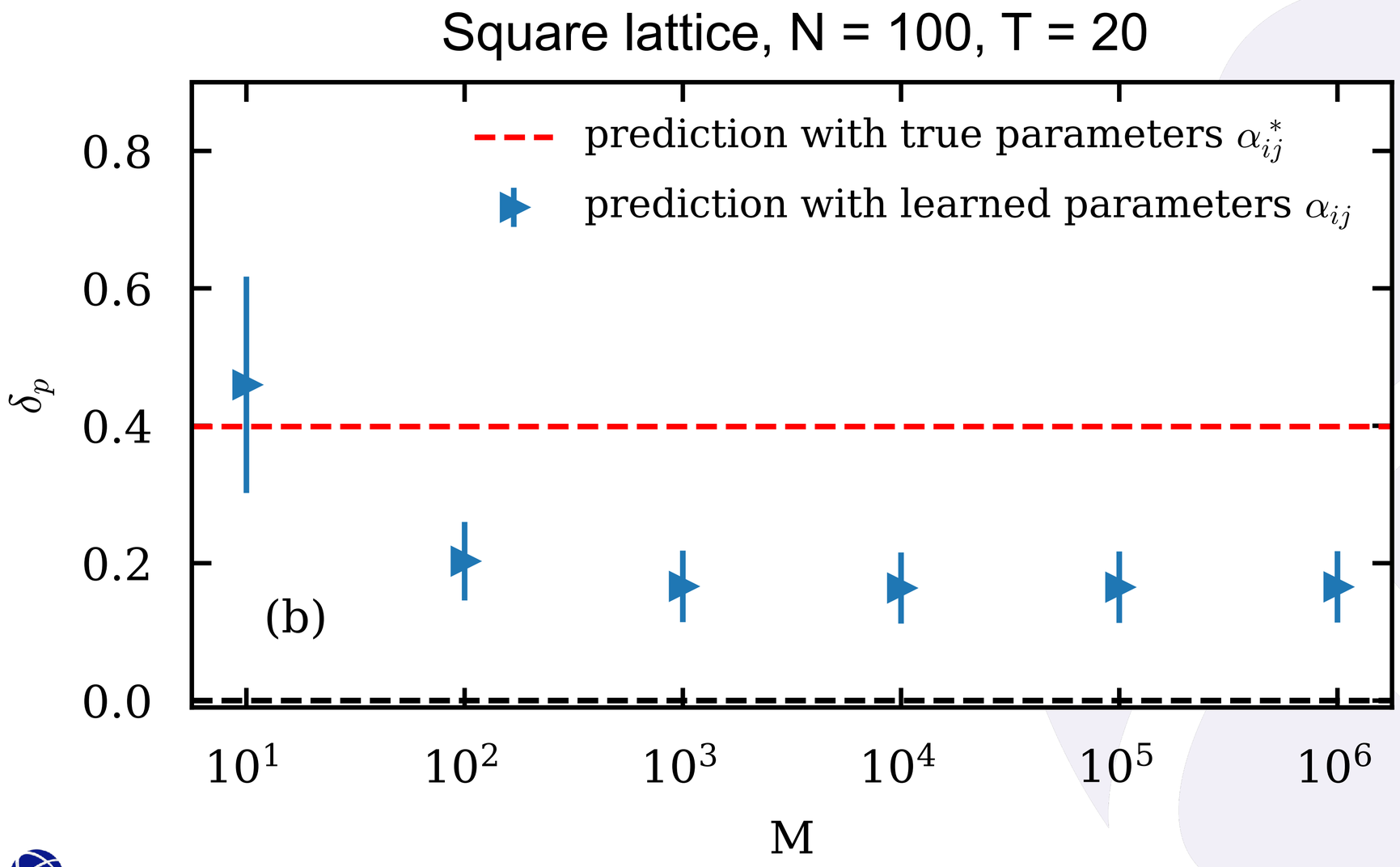
Adversarial case — square lattice

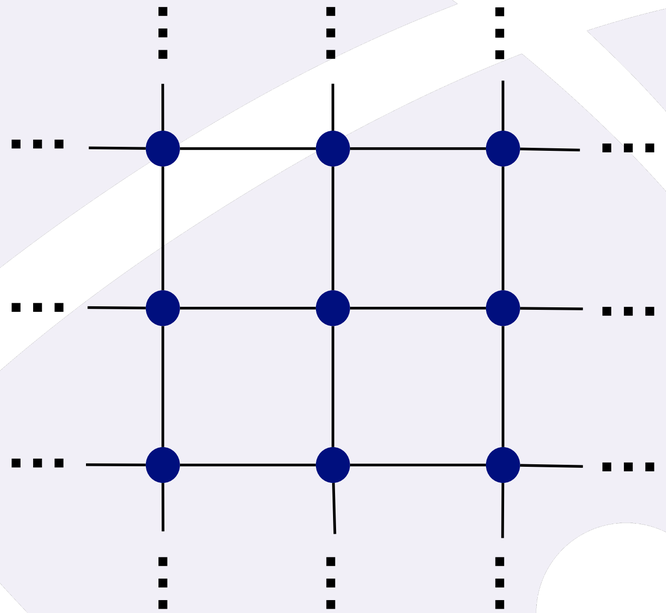
Do we really care about alphas?



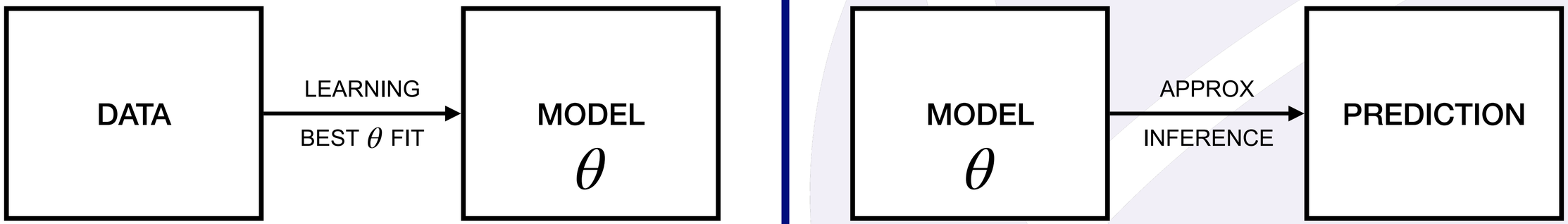
We care about prediction!

Adversarial case — square lattice

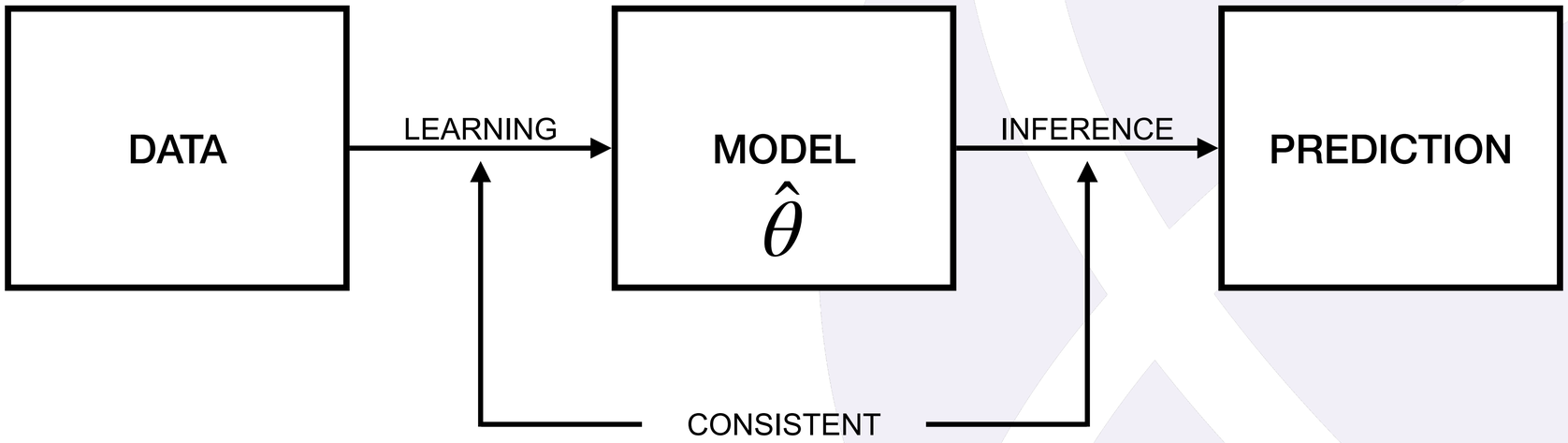




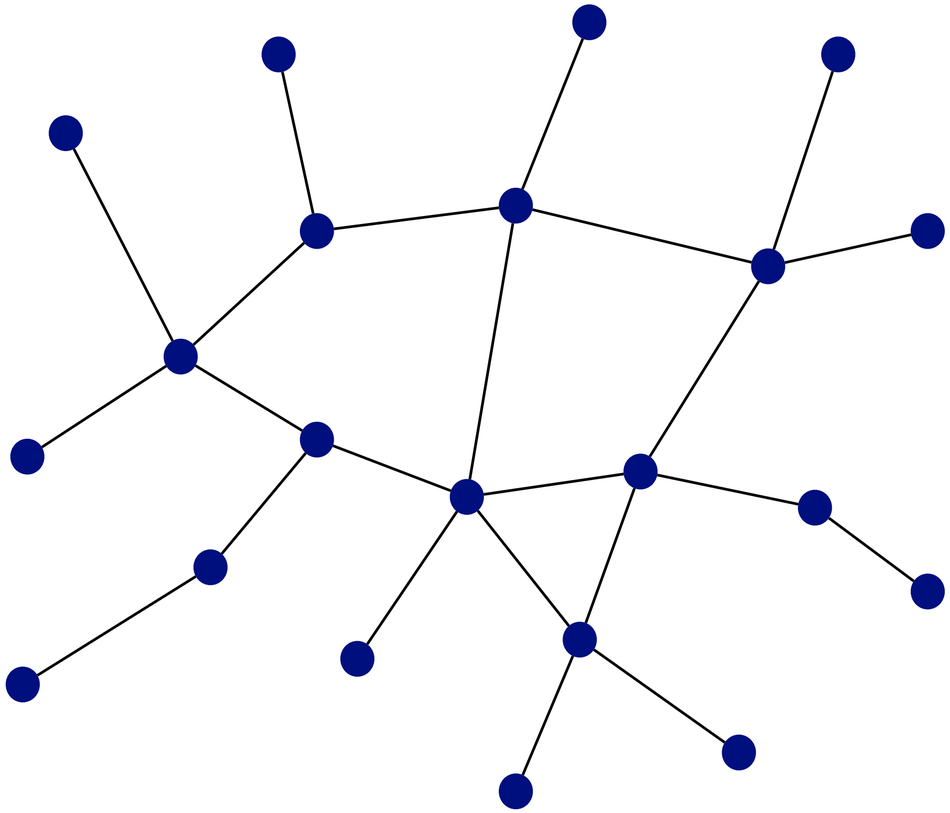
Learning *AND* Prediction



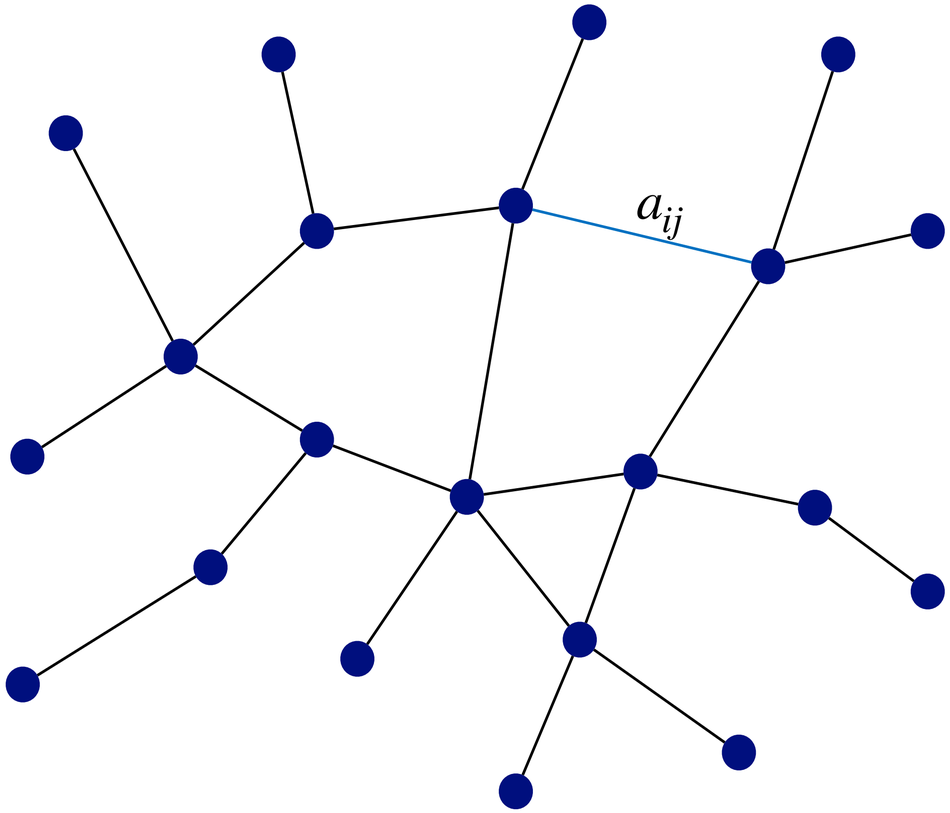
Learning *FOR* Prediction



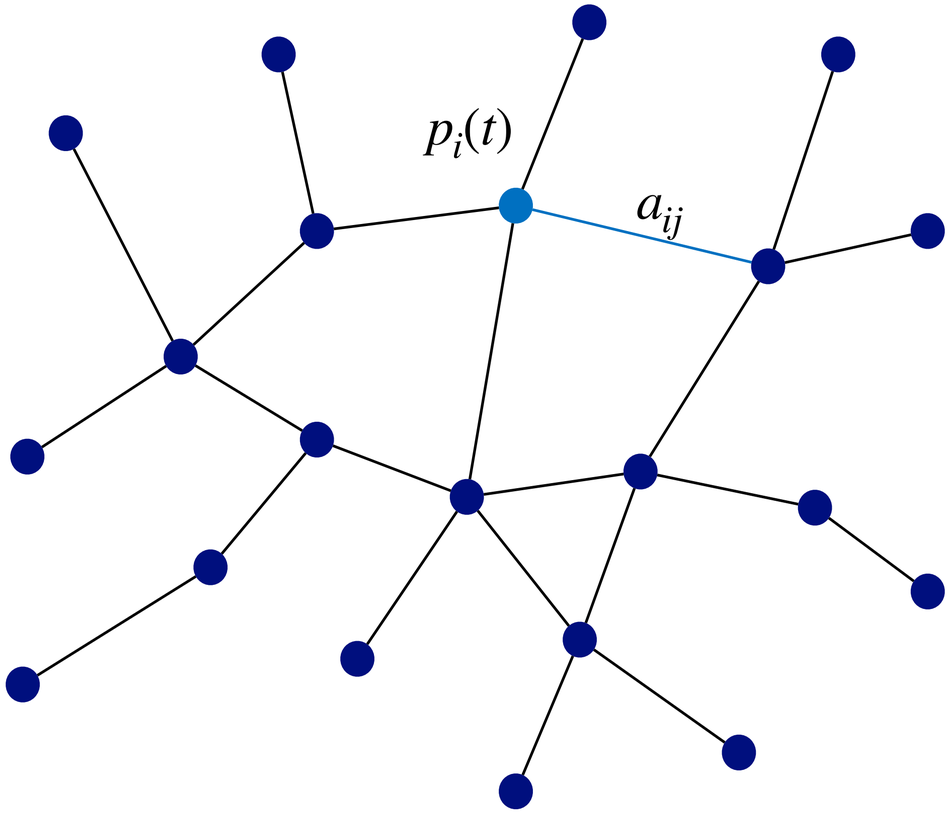
Mixture model



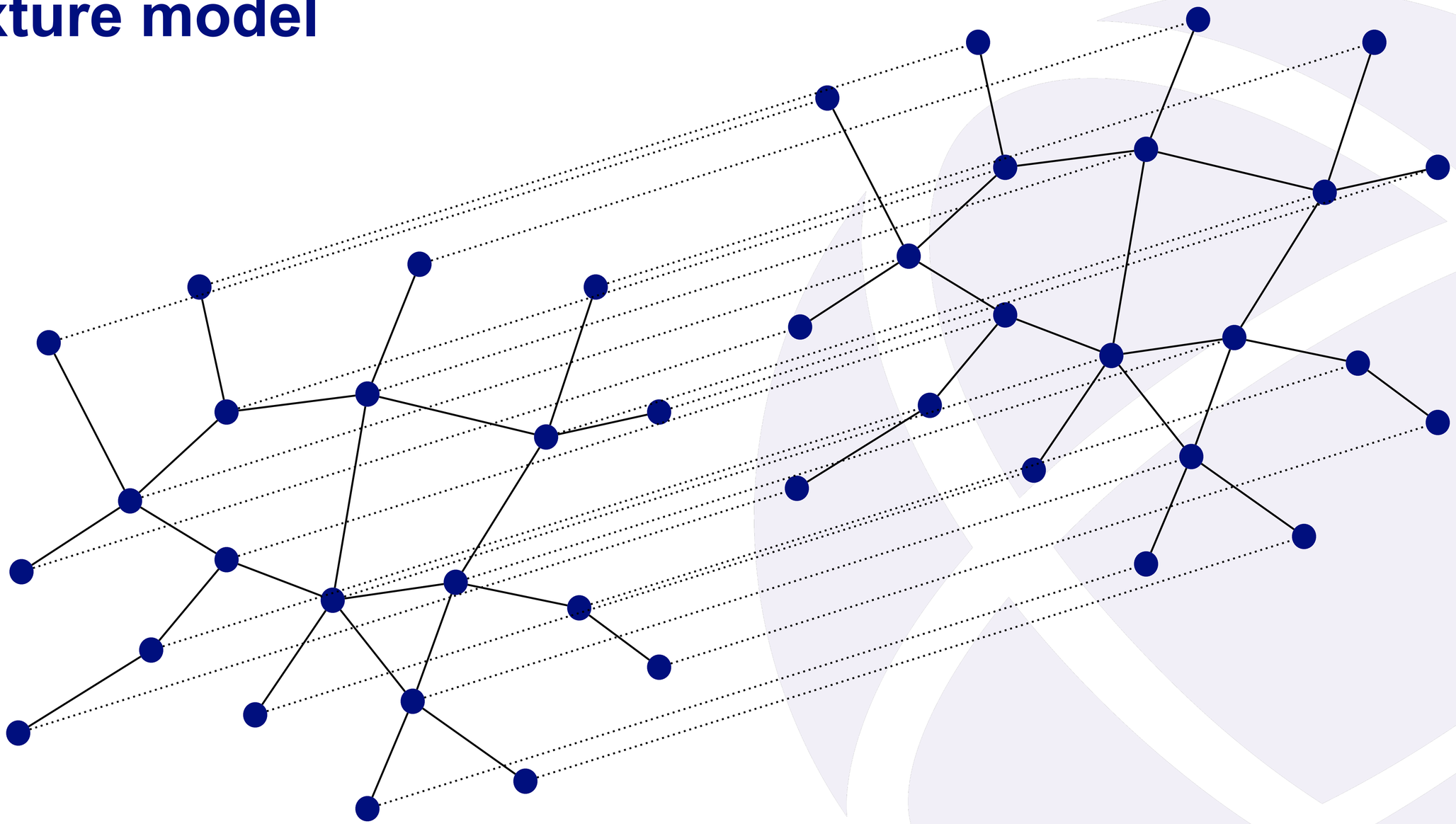
Mixture model



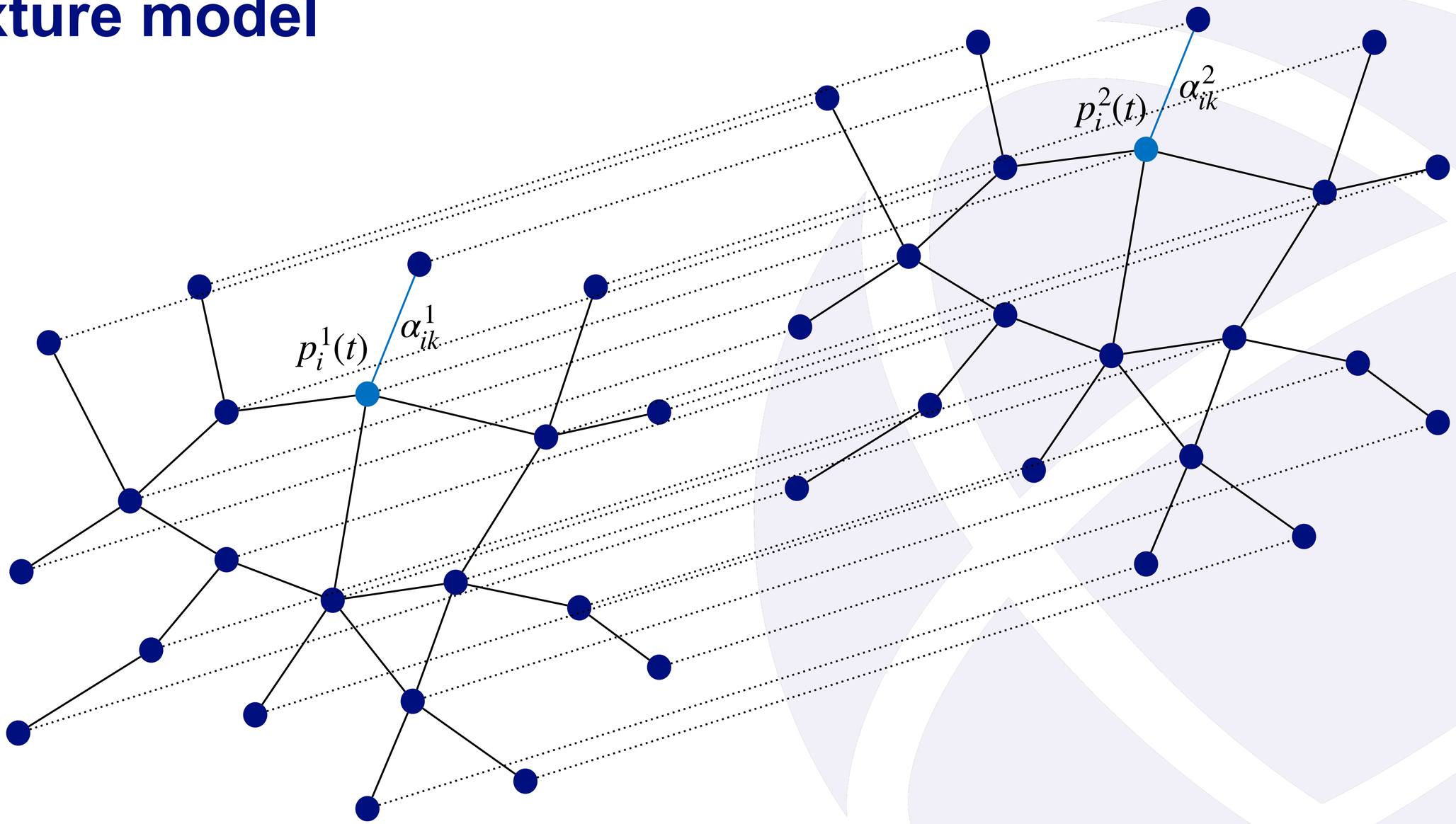
Mixture model



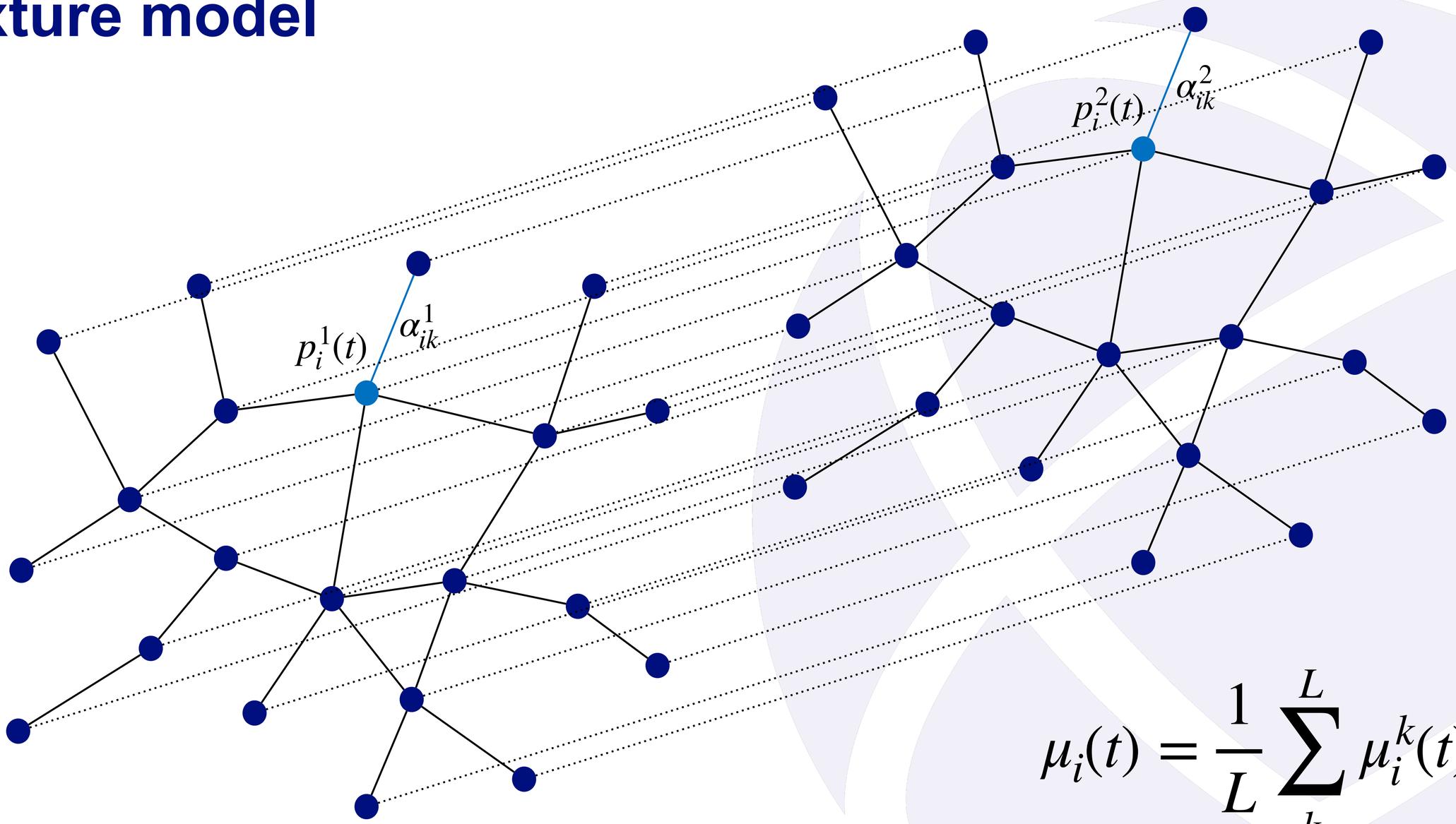
Mixture model



Mixture model

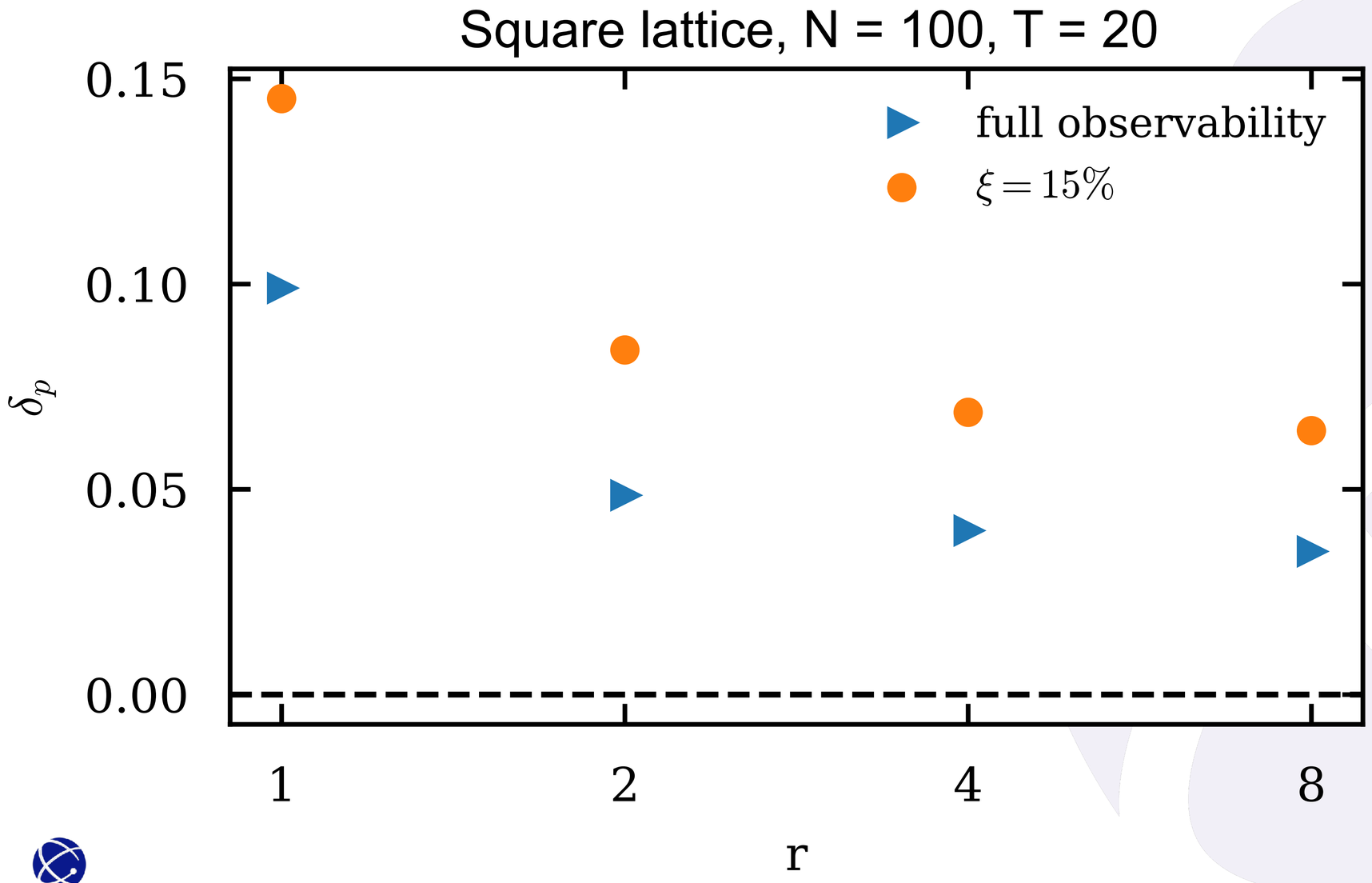


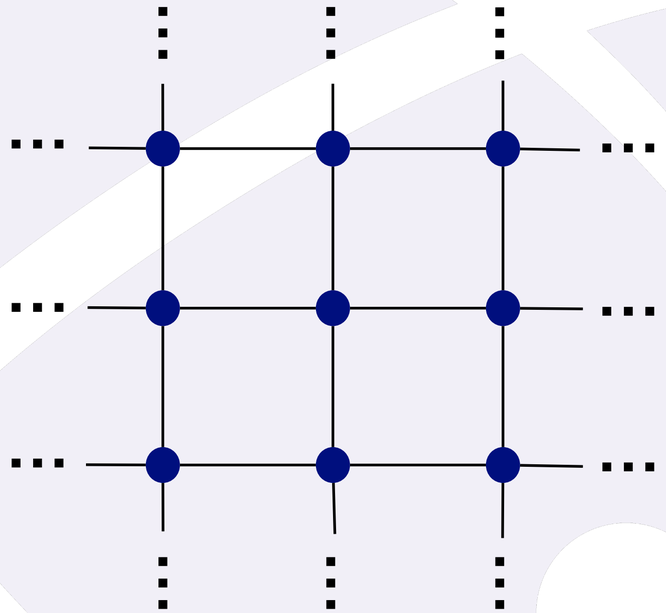
Mixture model



$$\mu_i(t) = \frac{1}{L} \sum_{k}^{L} \mu_i^k(t)$$

Mixture model

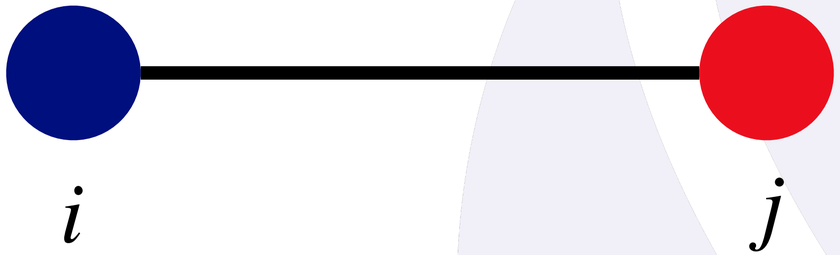




Reversible Dynamics

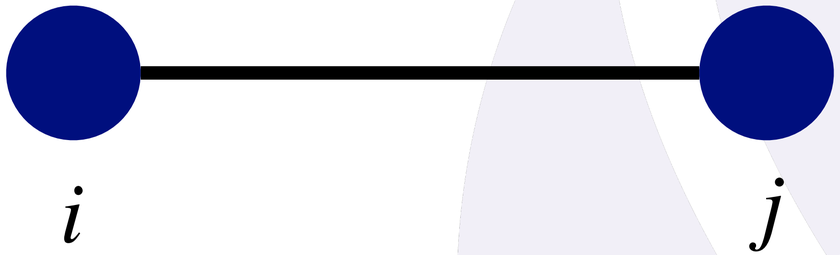
(future direction)

Susceptible-Infected-Susceptible model



$$P(s_i(t+1) = I \,|\, s_j(t) = I) = \alpha_{ij}$$

Susceptible-Infected-Susceptible model



$$P(s_j(t+1) = R \mid s_j(t) = I) = \gamma_j$$

Susceptible-Infected-Susceptible model

Ideas for Glauber dynamics:

- Del Ferraro, G., & Aurell, E. Dynamic message-passing approach for kinetic spin models with reversible dynamics. Physical Review E, 92(1), 010102, 2015.
- Barthel, T., De Bacco, C., & Franz, S. Matrix product algorithm for stochastic dynamics on networks applied to nonequilibrium Glauber dynamics. Physical Review E, 97(1), 010104, 2018.

Susceptible-Infected-Susceptible model

$$DMP(SI)$$

Susceptible-Infected-Susceptible model

$$DMP(SI) \rightarrow DMP(S_1I_1) \rightarrow DMP(S_1I_1S_2I_2)$$

Susceptible-Infected-Susceptible model

$$DMP(SI) \rightarrow DMP(S_1I_1) \rightarrow DMP(S_1I_1S_2I_2) \rightarrow DMP(S_1I_1 \ldots S_kI_k)$$

Thank you!

Questions?

Literature:

- Wilinski, M., & Lokhov, A. Prediction-centric learning of independent cascade dynamics from partial observations. ICML, 11182–11192, 2021.
- Wilinski, M., & Lokhov, A. Learning of Networked Spreading Models from Noisy and Incomplete Data. Physical Review E, 110, 054302, 2024.

My DMP github repo in Julia:
https://github.com/mateuszwilinski/dynamic-message-passing

Mixture Model Objective

$$\mathcal{O}^{mixture} = -\sum_{c \in C}\sum_{i \in O} \log\left(\frac{1}{L}\sum_{k}^{L} \mu_{i,k}^{c}(\tau_i^c)\right)$$

mixture marginal

$$\alpha_{ij}^{k*} = \min_{\alpha_{ij}^{k}} \left(\mathcal{O}^{mixture}\right)$$

we use the same Lagrange scheme as for single network

Kullback - Leibler Divergence

$$D_{KL}(P||Q) = -\sum_{x} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

Constrained Optimisation

$$\mathcal{O} = \sum_{i \in V} p_i(T)$$

$$\mathcal{J} = \lambda^{B_0} \left(B_0 - \sum_{i \in V} p_i(0) \right)$$

$$\mathcal{V} = \sum_{t=0}^{T} \lambda^{B_\mu}(t) \left(B_\mu(t) - \sum_{i \in V} \mu_i(t) \right)$$

Lokhov, Saad. *PNAS* 114.39 (2017): E8138-E8146.