Learning Latent Graph Structures and their Uncertainty

Alessandro Manenti [1] **Daniele Zambon** [1] **Cesare Alippi** [1 2]

Abstract

Within a prediction task, Graph Neural Networks (GNNs) use relational information as an inductive bias to enhance the model's accuracy. As task-relevant relations might be unknown, graph structure learning approaches have been proposed to learn them while solving the downstream prediction task. In this paper, we demonstrate that minimization of a point-prediction loss function, e.g., the mean absolute error, does not guarantee proper learning of the latent relational information and its associated uncertainty. Conversely, we prove that a suitable loss function on the stochastic model outputs simultaneously grants (i) the unknown adjacency matrix latent distribution and (ii) optimal performance on the prediction task. Finally, we propose a sampling-based method that solves this joint learning task. Empirical results validate our theoretical claims and demonstrate the effectiveness of the proposed approach.

1. Introduction

Relational information processing has provided breakthroughs in the analysis of rich and complex data coming from, e.g., social networks, natural language, and biology. This side information takes various forms, from structuring the data into clusters, to defining causal relations and hierarchies, and enables machine learning models to condition their predictions on dependency-related observations. In this context, predictive models $y = f_\psi(x, A)$ condition the input-output relation $x \mapsto y$ – modeled by f_ψ and its parameters in ψ – on the relational information encoded in variable A. This paper focuses on Graph Neural Networks (GNNs) [Scarselli et al., 2008], a successful example of models that rely on a graph structure represented as an adjacency matrix A, e.g., see [Fout et al., 2017; Shlomi et al., 2020].

Not rarely, such topological information is inadequate to address the problem at hand or even completely unavailable. Therefore, Graph Structure Learning (GSL) emerges as an approach for learning the graph topology alongside the predictive model f_ψ. This entails formulating a joint learning process that learns a parametrization of A altogether with the predictor's parameters ψ. Examples of GSL appear within graph deep learning methods for both static [Jiang et al., 2019; Yu et al., 2021; Kazi et al., 2022] and temporal data [Wu et al., 2019; 2020; Cini et al., 2023; De Felice et al., 2024]; a recent review is provided by Zhu et al. [2021].

Different sources of uncertainty affect the GSL process, including epistemic uncertainty in the data and variability inherent in the data-generating process. To capture it, probabilistic approaches have been devised and model $A \sim P^\theta$ as a random variable with parametric distribution P^θ [Kipf et al., 2018; Franceschi et al., 2019; Elinas et al., 2020; Shang et al., 2021; Niepert et al., 2021; Cini et al., 2023].

In this paper, we address the joint problem of learning a predictive model yielding optimal point predictions of the outputs y and, contextually, a calibrated distribution for the latent adjacency matrix A; to the best of our knowledge, no prior work on GSL has studied such joint problem before. The novel contributions can be summarized as:

1. We demonstrate that models trained to achieve optimal point predictions do *not* guarantee calibration of the adjacency matrix distribution [Section 3].
2. We provide theoretical conditions on the predictive model and loss function that guarantee both distribution calibration and optimal point-predictions [Section 4].
3. We propose a theoretically-grounded sampling-based learning method to address the joint learning problem [Section 4].
4. We empirically validate major paper's theoretical developments and claims and show that the proposed method is indeed able to solve the joint learning task [Section 5].

[1]The Swiss AI Lab IDSIA USI-SUPSI and Università della Svizzera italiana, Switzerland. [2]Politecnico di Milano, Italy.. Correspondence to: Alessandro Manenti <alessandro.manenti@usi.ch>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria.

2. Problem formulation

Consider a set of N interacting entities and the data-generating process

$$\begin{cases} y = f^*(x, A) \\ A \sim P_A^* \end{cases} \tag{1}$$

where $y \in \mathcal{Y} \subseteq \mathbb{R}^{N \times d_{out}}$ is the system output obtained from input $x \in \mathcal{X} \subseteq \mathbb{R}^{N \times d_{in}}$ through function f^* and conditioned on a realization of the latent adjacency matrix $A \in \mathcal{A} \subseteq \{0,1\}^{N \times N}$ drawn from P_A^*; superscript $*$ refers to unknown entities. Each entry of the adjacency matrix A is a binary value encoding the existence of a pairwise relation between two nodes.

Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ of n input-output observations from (1), we aim at learning a probabilistic predictive model

$$\begin{cases} \hat{y} = f_\psi(x, A) \\ A \sim P_A^\theta \end{cases} \tag{2}$$

from $\mathcal{D}$, while learning at the same time distribution P_A^θ approximating P_A^*. The two parameter vectors θ and ψ approximate distribution P_A^* and function f^*, respectively. We assume

Assumption 2.1. The family $\{P_A^\theta\}$ of probability distributions P_A^θ parametrized by θ and the family of predictive functions $\{f_\psi\}$ are expressive enough to contain the true latent distribution P_A^* and function f^*, respectively.

Assumption 2.1 implies that $f^* \in \{f_\psi\}$ and $P_A^* \in \{P_A^\theta\}$. Under such assumption the minimum function approximation error is null and we can focus on the theoretical conditions requested to guarantee a successful learning, i.e., achieving both optimal point predictions and latent distribution calibration.

Optimal point predictions Outputs y and $\hat{y}$ of probabilistic model (1) and (2) are random variables following push-forward distributions[1] $P_{y|x}^*$ and $P_{y|x}^{\theta,\psi}$, respectively. A single point prediction $y_{PP} \in \mathcal{Y}$ can be obtained through an appropriate functional $T[\cdot]$ as

$$y_{PP} = y_{PP}(x, \theta, \psi) \equiv T\left[P_{y|x}^{\theta,\psi}\right]. \tag{3}$$

For example, T can be the expected value or the value at a specific quantile. We then define an *optimal predictor* as one whose parameters θ and ψ minimize the expected *point-prediction loss*

$$\mathcal{L}^{point}(\theta, \psi) = \mathbb{E}_{x \sim P_x^*}\left[\mathbb{E}_{y \sim P_{y|x}^*}\left[\ell\big(y, y_{PP}(x, \theta, \psi)\big)\right]\right] \tag{4}$$

between the system output y and the point-prediction y_{PP}, as measured by of a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$.

Statistical functional T is coupled with the loss ℓ as the optimal functional T to employ given a specific loss ℓ is often known [Berger, 1990; Gneiting, 2011], when $P_{y|x}^{\theta,\psi}$ approximates well $P_{y|x}^*$. For instance, if ℓ is the Mean Absolute Error (MAE) the associated functional T is the median, if ℓ is the Mean Squared Error (MSE) the associated functional is the expected value.

Latent distribution calibration Calibration of a parametrized distribution requires learning parameters θ, so that distribution P_A^θ aligns with P_A^*. Quantitatively, a dissimilarity measure $\Delta^{cal} : \mathcal{P}_A \times \mathcal{P}_A \to \mathbb{R}_+$, defined over a set $\mathcal{P}_A$ of distributions on $\mathcal{A}$, assesses how close two distributions are. The family of f-divergences [Rényi, 1961] and the integral probability metrics [Müller, 1997] are examples of such dissimilarity measures. In this paper, we are interested in those discrepancies for which $\Delta^{cal}(P_1, P_2) = 0 \iff P_1 = P_2$ holds. It follows that the latent distribution P_A^θ is *calibrated* on P_A^* if it minimizes the *latent distribution loss*

$$\mathcal{L}^{cal} = \mathbb{E}_{x \sim P_x^*}\left[\Delta^{cal}\left(P_A^*, P_A^\theta\right)\right], \tag{5}$$

The problem of designing a predictive model (2) that both yields optimal point predictions and calibrates the latent distribution is non-trivial

3. Limitations of point-prediction optimization

In this section, we demonstrate that the optimization of a point prediction loss Equation (4) does not generally grant calibration of the latent random variable.

Proposition 3.1. *Consider Assumption 2.1. Loss function $\mathcal{L}^{point}(\theta, \psi)$ in (4) is minimized by all θ and ψ s.t. $T\left[P_{y|x}^{\theta,\psi}\right] = T\left[P_{y|x}^*\right]$ almost surely on x and, in particular,*

$$\mathcal{L}^{point}(\theta, \psi) \text{ is minimal} \quad \begin{matrix} \Longleftarrow \\ \not\Longrightarrow \end{matrix} \quad P_{y|x}^{\theta,\psi} = P_{y|x}^*.$$

The proof of the proposition is given in Appendix A.1; we provide a counterexample for which calibration is not granted even when the processing function f_ψ is equal to f^* in Appendix A.2. Figure 1 supports the proposition's claim by showing a view of the optimization landscape of $\mathcal{L}^{point}$ with MAE as ℓ that is flatter than the proposed $\mathcal{L}^{dist}$ discussed in Section 4.

Given the provided negative result and the impossibility of assessing loss $\mathcal{L}^{cal}$ in (5), in the next section, we propose another optimization objective that, as we will prove, allows us to both calibrate the latent random variable and to have optimal point predictions.

[1] The distribution of $y = f^*(x, A)$ originated from P_A^* and of $\hat{y} = f_\psi(x, A)$ originated from P_A^θ.

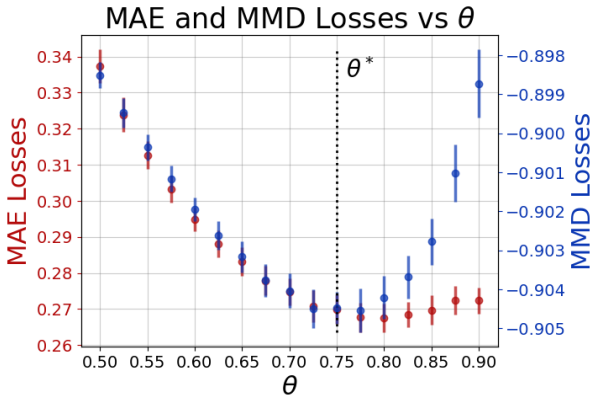

Figure 1: Comparison of the landscape of loss $\mathcal{L}^{point}$ in (4) with MAE as ℓ (red) and loss $\mathcal{L}^{dist}$ in (6) with MMD as Δ (blue). The losses are evaluated for different values of θ on the data generated from parameter $\theta^* = 3/4$. Further details are in Appendix C.2.

4. Predictive distribution optimization: two birds with one stone

Optimal point predictions (2) and latent distribution calibration can be achieved by comparing push-forward distributions $P^*_{y|x}$ and $P^{\theta,\psi}_{y|x}$. In particular, Theorem 4.2 below proves that, under appropriate conditions, minimization of the *output distribution loss*

$$\mathcal{L}^{dist}(\theta,\psi) = \mathbb{E}_{x\sim P^*_x}\Big[\Delta(P^*_{y|x}, P^{\theta,\psi}_{y|x})\Big] \tag{6}$$

provides calibrated P^θ_A, even when P^*_A is not available; $\Delta : \mathcal{P}_y \times \mathcal{P}_y \to \mathbb{R}_+$ is a dissimilarity measure between distributions over space $\mathcal{Y}$. We assume the following on Δ.

Assumption 4.1. $\Delta(P_1, P_2) \geq 0$ for all distributions P_1 and P_2 in $\mathcal{P}_y$ and $\Delta(P_1, P_2) = 0$ if and only if $P_1 = P_2$.

Several choices of Δ meet Assumption 4.1. As detailed below, we propose considering the Maximum Mean Discrepancy (MMD) [Gretton et al., 2012].

Theorem 4.2. *Let $I = \{x : A \mapsto f^*(x, A)$ is injective$\} \subseteq \mathcal{X}$ be the set of points $x \in \mathcal{X}$ such that map $A \mapsto f^*(x, A)$ is injective. Under Assumptions 2.1 and 4.1, if $\mathbb{P}_{x\sim P^*_x}(I) > 0$, then*

$$\mathcal{L}^{dist}(\theta,\psi^*) = 0 \implies \begin{cases} \mathcal{L}^{point}(\theta,\psi^*) \text{ is minimal} \\ \mathcal{L}^{cal}(\theta) = 0, \end{cases}$$

where ψ^* is such that $f_{\psi^*} = f^*$.

Theorem 4.2 is proven in Appendix A.3. Under the theorem's hypotheses, a predictor that minimizes $\mathcal{L}^{dist}$ is both *calibrated* on the latent random distribution and provides *optimal point predictions*. This overcomes limits of Proposition 3.1 where optimization of $\mathcal{L}^{point}(\theta,\psi^*)$ does not grant $\mathcal{L}^{cal}(\theta) = 0$.

The hypotheses under which Theorem 4.2 holds are rather mild. In fact, condition $\mathbb{P}_{x\sim P^*_x}(I) > 0$ pertains to the data-generating process. Instead, condition $f_\psi = f^*$ is set to avoid scenarios of different, yet equivalent,[2] representations of the latent distribution. Assumptions 2.1 and 4.1 can be met with an appropriate choice of model (2) and measure Δ; as such they are controllable by the designer.

Maximum mean discrepancy Given two distributions $P_1, P_2 \in \mathcal{P}_y$, the MMD can be defined from a kernel function $\kappa(\cdot,\cdot) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ as

$$\begin{aligned}\mathrm{MMD}^2_\kappa[P_1, P_2] = &\mathop{\mathbb{E}}_{y_1,y_1'\sim P_1}\Big[\kappa(y_1, y_1')\Big] + \\ &- 2\mathop{\mathbb{E}}_{\substack{y_1\sim P_1\\ y_2\sim P_2}}\Big[\kappa(y_1, y_2)\Big] + \mathop{\mathbb{E}}_{y_2,y_2'\sim P_2}\Big[\kappa(y_2, y_2')\Big]\end{aligned} \tag{7}$$

and provides a versatile choice of Δ that allows Monte Carlo (MC) computation without requiring evaluations of the likelihood w.r.t. the output distributions $P^*_{y|x}$ and $P^{\theta,\psi}_{y|x}$. In particular, when universal kernels are considered (e.g., the Gaussian one), then Assumption 4.1 is fulfilled [Gretton et al., 2012].

Finite-sample loss optimization To compute the gradients of $\mathcal{L}^{dist}(\theta,\psi)$ w.r.t. parameter vectors ψ and θ, we rely on MC sampling, obtaining

$$\sum_{(x,y)\in\mathcal{B}}\left(\frac{2\sum_{i=1}^{N_{adj}}\sum_{j=1}^{i-1}\kappa(\hat{y}_i,\hat{y}_j)}{N_{adj}(N_{adj}-1)} - 2\frac{\sum_{i=1}^{N_{adj}}\kappa(y,\hat{y}_i)}{N_{adj}}\right) \tag{8}$$

as an approximation of $\mathcal{L}^{dist}$, where $N_{adj} > 1$ is the number of adjacency matrices sampled from P^θ_A to obtain output samples $\hat{y}_i = f_\psi(x, A_i) \sim P^{\theta,\psi}_{y|x}$, whereas the pairs (x, y) are from a mini-batch $\mathcal{B}$ of the training set $\mathcal{D}$. Note that the third term of (7) is dropped as it is independent of ψ and θ.

While gradient $\nabla_\psi\mathcal{L}^{dist}(\theta,\psi)$ is computed directly via automatic differentiation, $\nabla_\theta\mathcal{L}^{dist}(\theta,\psi)$ special care, the gradient is computed with respect to the same parameter vector θ that defines the integrated distribution. Here, we rely on a score-function gradient estimator (SFE) [Williams, 1992; Mohamed et al., 2020] as in [Cini et al., 2023]. A version of the algorithm with reduced variance is detailed in Appendix B.

5. Experiments

Experiments employ a synthetic dataset to evaluate the discrepancy between the true latent distribution and the learned one. We remark that the latent distribution P^*_A is used *only*

[2] E.g., $f_\psi(A, x) = f_*(\mathbf{1} - A, x)$ and P^θ_A encoding the absence of edges instead of their presence as in P^*_A.

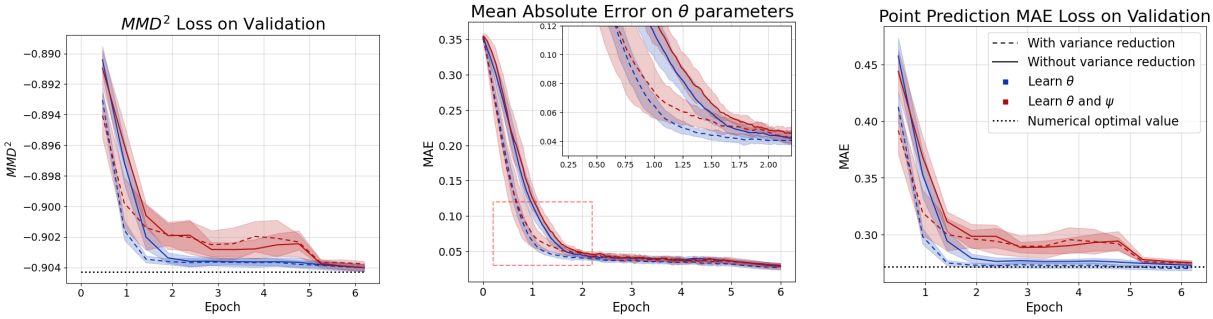

Figure 2: Validation losses $\mathcal{L}^{dist}$, $\mathcal{L}^{cal}$ and $\mathcal{L}^{point}$ during training. At epoch 5, the learning rate is decreased to ensure convergence. Results are averaged over 20 model initializations and shaded areas indicate ± 1 standard deviation from the mean. Results are reported with and without applying the variance reduction technique (see Appendix B), by training only parameters θ while freezing ψ to ψ^* (same setting of Theorem 4.2), and by joint training of both ψ and θ.

to assess performance and does not drive the model training in any way.

Dataset and models Consider data-generating process (1) with latent distribution $P_A^* = P_A^{\theta^*}$ producing N-node adjacency matrices. P_A^* is defined by a set of $N \times N$ independent Bernoulli distributions, each of which corresponds to the sampling probability of an edge. Function $f_* = f_{\psi^*}$ is a generic GNN with node-level readout, i.e., $f_{\psi^*}(\cdot, A) : \mathbb{R}^{N \times d_{in}} \to \mathbb{R}^{N \times d_{out}}$. As predictive model family (2), we follow the same architecture of f_{ψ^*} and $P_A^{\theta^*}$ ensuring that during all the experiments Assumption 2.1 is fulfilled. Additional specifics are detailed in Appendix C.

Solving the joint learning problem The left panel of Figure 2 shows that the training succeeded and the MMD loss $\mathcal{L}^{dist}$ approached its minimum (dotted line). Once $\mathcal{L}^{dist}$ reaches its minimum, also the calibration of latent distribution P_A^θ is successful. Specifically, the central panel shows that the validation MAE ($N^{-2}\|\theta^* - \theta\|_1$) approaches zero as training proceeds (MAE < 0.04). Regarding the point predictions, the right-hand side of Figure 2 confirms that $\mathcal{L}^{point}$ reached its minimum value; recall that optimal prediction MAE is not 0, as the target variable y is random, and note that a learning rate reduction is applied at epoch number 5. Moreover, we observe that calibration is achieved regardless of the variance reduction (see Appendix B), although variance reduction increases convergence speed.

Optimization landscape of $\mathcal{L}^{point}$ and $\mathcal{L}^{dist}$ In this experiment, we analyze the values of $\mathcal{L}^{point}(\theta, \psi^*)$ and $\mathcal{L}^{dist}(\theta, \psi^*)$ for different values of θ. $\mathcal{L}^{point}$ is computed employing MAE as loss function ℓ. Specifically, we let a scalar p vary from $1/2$ to 1 and set $\theta_{ij} = p$ for all i, j where $\theta^*_{ij} = 3/4$. Figure 1 reports the obtained results, highlighting an almost flat $\mathcal{L}^{point}$ for values $p \geq 0.725$. In contrast, $\mathcal{L}^{dist}$ displays a pronounced concave shape with a clear minimum around θ^* which suggests that calibration is easier when we minimize $\mathcal{L}^{dist}$ instead of $\mathcal{L}^{point}$.

Overall, we conclude that our approach is effective in solving the joint learning problem of calibrating the latent variable while producing optimal point predictions.

6. Conclusions

Graph structure learning has emerged as a research field focused on learning graph topologies in support of solving downstream predictive tasks. Assuming stochastic latent graph structures, we are led to a joint optimization objective: (i) learning the correct distribution of the latent topology while (ii) achieving optimal point predictions on the downstream task. In this paper, at first, we prove both positive and negative theoretical results to demonstrate that appropriate loss functions must be chosen to solve this joint learning problem. Second, we propose a sampling-based learning method that does not require the computation of the predictive likelihood. Our empirical results demonstrate that this approach achieves optimal point predictions on the considered downstream task while also yielding calibrated latent graph distributions.

Finally, we acknowledge that the proposed method requires sampling and processing multiple adjacency matrices for each input and, although the model and prediction accuracy is enhanced, a computation overhead is requested. We plan future research to explore the applicability of this method to real-world datasets and to other classes of neural networks beyond GNNs; the current study, in fact, focuses on a set of controlled experiments on synthetic data to validate all the theoretical claims.

Acknowledgments

This research was funded by the Swiss National Science Foundation under grant 204061: *High-Order Relations and Dynamics in Graph Neural Networks*.

References

Berger, J. O. Statistical decision theory. In *Time Series and Statistics*, pp. 277–284. Springer, 1990.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.

Cini, A., Zambon, D., and Alippi, C. Sparse graph learning from spatiotemporal time series. *Journal of Machine Learning Research*, 24:1–36, 2023.

De Felice, G., Cini, A., Zambon, D., Gusev, V., and Alippi, C. Graph-based Virtual Sensing from Sparse and Partial Multivariate Observations. In *The Twelfth International Conference on Learning Representations*, 2024.

Elinas, P., Bonilla, E. V., and Tiao, L. Variational inference for graph convolutional networks in the absence of graph data and adversarial settings. *Advances in Neural Information Processing Systems*, 33:18648–18660, 2020.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Fout, A., Byrd, J., Shariat, B., and Ben-Hur, A. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.

Franceschi, L., Niepert, M., Pontil, M., and He, X. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pp. 1972–1982. PMLR, 2019.

Gneiting, T. Making and Evaluating Point Forecasts. *Journal of the American Statistical Association*, 106 (494):746–762, June 2011. ISSN 0162-1459. doi: 10.1198/jasa.2011.r10138.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95, 2007.

Jiang, B., Zhang, Z., Lin, D., Tang, J., and Luo, B. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11313–11320, 2019.

Kazi, A., Cosmo, L., Ahmadi, S.-A., Navab, N., and Bronstein, M. M. Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1606–1617, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. In *International conference on machine learning*, pp. 2688–2697. PMLR, 2018.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte carlo gradient estimation in machine learning. *The Journal of Machine Learning Research*, 21(1):5183–5244, 2020.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29 (2):429–443, 1997.

Niepert, M., Minervini, P., and Franceschi, L. Implicit MLE: Backpropagating Through Discrete Exponential Family Distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14567–14579. Curran Associates, Inc., 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Rényi, A. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, volume 4, pp. 547–562. University of California Press, 1961.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Shang, C., Chen, J., and Bi, J. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021.

Shlomi, J., Battaglia, P., and Vlimant, J.-R. Graph neural networks in particle physics. *Machine Learning: Science and Technology*, 2(2):021001, 2020.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Yu, D., Zhang, R., Jiang, Z., Wu, Y., and Yang, Y. Graph-revised convolutional network. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*, pp. 378–393. Springer, 2021.

Zhu, Y., Xu, W., Zhang, J., Liu, Q., Wu, S., and Wang, L. Deep graph structure learning for robust representations: A survey. *arXiv preprint arXiv:2103.03036*, 14: 1–1, 2021.

A. Proofs of the theoretical results

A.1. Minimizing $\mathcal{L}^{point}$ does not guarantee calibration

Proof of Proposition 3.1.

Proof. Recall the definition of $\mathcal{L}^{point}$ in (4) using (3)

$$\mathcal{L}^{point}(\psi,\theta) = \mathbb{E}_x\Big[\mathbb{E}_{y^*\sim P^*_{y|x}}\Big[\ell\big(y^*, T\big[P^{\theta,\psi}_{y|x}\big]\big)\Big]\Big]$$

Given loss function ℓ, T is, by definition [Berger, 1990; Gneiting, 2011], the functional that minimizes

$$\mathbb{E}_{y^*\sim P^*_{y|x}}\Big[\ell\big(y^*, T\big[P^*_{y|x}\big]\big)\Big]$$

Therefore, if $P^{\theta,\psi}_{y|x} = P^*_{y|x} \implies \mathcal{L}^{point}$ is minimal. If another distribution over y, namely, $P^{\psi',\theta'}_{y|x}$ parametrized by θ' and ψ' satisfies:

$$T\Big[P^{\psi',\theta'}_{y|x}\Big] = T\Big[P^*_{y|x}\Big]$$

almost surely on x, then,

$$\begin{aligned}\mathcal{L}^{point}(\theta',\psi') &= \mathbb{E}_x\Big[\mathbb{E}_{y^*\sim P^*_{y|x}}\Big[\ell\big(y^*, T\big[P^{\psi',\theta'}_{y|x}\big]\big)\Big]\Big]\\ &= \mathbb{E}_x\Big[\mathbb{E}_{y^*\sim P^*_{y|x}}\Big[\ell\big(y^*, T\big[P^*_{y|x}\big]\big)\Big]\Big]\end{aligned}$$

Thus, $P^{\psi',\theta'}_{y|x}$ minimizes $\mathcal{L}^{point}$.

Appendix A.2 discusses graph distributions where $T\big[P^{\psi',\theta'}_{y|x}\big] = T\big[P^*_{y|x}\big]$ but $P^{\psi',\theta'}_{y|x} \neq P^*_{y|x}$. We conclude that reaching the minimum of $\mathcal{L}^{point}(\psi,\theta)$ does not imply $P^{\psi,\theta}_{y|x} = P^*_{y|x}$. □

A.2. Minimizing $\mathcal{L}^{point}$ does not guarantee calibration: an example with MAE

In this section, we show that $\mathcal{L}^{point}$ equipped with MAE as ℓ admits multiple global minima for different parameters θ, even for simple models and $f_\psi = f^*$.

Consider a single Bernoulli of parameter $\theta^* > 1/2$ as latent variable A and a scalar function $f^*(x,A)$ such that $f^*(x,1) > f^*(x,0)$ for all x. Given input x the value of functional $T(P^*_{y|x})$ that minimizes

$$\mathbb{E}_{y\sim P^*_{y|x}}\Big[\Big|y - T\big[P^*_{y|x}\big]\Big|\Big] = \theta^*\Big|f^*(x,1) - T\big[P^*_{y|x}\big]\Big| + (1-\theta^*)\Big|f^*(x,0) - T\big[P^*_{y|x}\big]\Big|$$

is $T(P^*_{y|x}) = f^*(x,1)$; this derives from the fact that range of f^* is $\{f^*(x,0), f^*(x,1)\}$ and the likelihood of $f^*(x,1)$ is larger than that of $f^*(x,0)$.

Note that $T\big[P^*_{y|x}\big] = f^*(x,1)$ for all x, therefore also $\mathcal{L}^{point}$ is minimized by such T. Moreover, $T\big[P^*_{y|x}\big]$ is function of θ^* and equal to $f^*(x,1)$ for all $\theta > 1/2$. We conclude that for any $\theta \neq \theta^*$ distributions $P^{\theta,\psi}_{y|x}$ and $P^*_{y|x}$ are different, yet both of them minimize $\mathcal{L}^{point}$ if $\theta > 1/2$.

A similar reasoning applies for $\theta^* < 1/2$.

A.3. Minimizing $\mathcal{L}^{dist}$ guarantees calibration and optimal point predictions.

Proof of Theorem 4.2

Proof. Recall from Equation (6) that

$$\mathcal{L}^{dist}(\theta) = \mathbb{E}_x\Big[\Delta(P^*_{y|x}, P^\theta_{y|x})\Big]$$

We start by proving that if $\mathcal{L}^{dist}(\theta,\psi) = 0 \implies \mathcal{L}^{point}(\theta,\psi)$ is minimal.

Note that $\mathcal{L}^{dist}(\theta,\psi)=0$ implies that $\Delta(P^*_{y|x},P^\theta_{y|x})=0$ almost surely in x. Then, by Assumption 4.1, $P^*_{y|x}=P^{\psi,\theta}_{y|x}$ almost surely on x and, in particular, $T[P^*_{y|x}]=T[P^{\psi,\theta}_{y|x}]$, which leads to $\mathcal{L}^{point}(\psi,\theta)$ being minimal (Proposition 3.1).

We now prove that if $\mathcal{L}^{dist}(\theta,\psi^*)=0 \implies \mathcal{L}^{cal}(\theta)=0$.

From the previous step, we have that $\mathcal{L}^{dist}(\theta,\psi)=0$ implies $P^*_{y|x}=P^{\psi,\theta}_{y|x}$ almost surely for $x\in I$. Under the assumption that $f_\psi=f_*$ and the injectivity of f_* in such $x\in I$, for any output y a single A exists such that $f_*(x,A)=y$. Therefore, the probability mass function of y equals that of A. Accordingly, $P^*_{y|x}=P^{\psi,\theta}_{y|x}$ implies $P^*_A=P^\theta_A$.

□

Here, we also prove a corollary of Theorem 4.2.

Corollary A.1. *Under Assumptions 2.1 and 4.1, if*

1. $\exists \bar{x}\in Supp(P^*_x)\subseteq\mathcal{X}$ *such that* $f^*(\bar{x};\cdot)$ *is injective,*
2. $f^*(x,A)$ *is continuous in* $\bar{x}$ $\forall A\in\mathcal{A}$*,*

then

$$\mathcal{L}^{dist}(\theta,\psi^*)=0 \implies \begin{cases}\mathcal{L}^{point}(\theta,\psi^*) \textit{ is minimal}\\ \mathcal{L}^{cal}(\theta)=0,\end{cases}$$

The corollary shows that it is sufficient that f^* is continuous in x and there exists one point $\bar{x}$ where $f^*(\bar{x},\,\cdot\,)$ is injective to meet theorem's hypothesis $\mathbb{P}_{x\sim P^*_x}(I)>0$; we observe that, as $\mathcal{A}$ is discrete, the injectivity assumption is not as restrictive as if the domain were continuous.

Proof. As $\mathcal{A}$ is a finite set, the minimum $\bar{\epsilon}=\min_{A,A'\in\mathcal{A}}\|f^*(\bar{x},A)-f^*(\bar{x},A')\|>0$ exists and, by the injectivity assumption, is strictly positive.

By continuity of $f^*(\,\cdot\,,A)$, for every $\epsilon<\frac{1}{2}\bar{\epsilon}$ there exists δ, such that for all $x\in B(\bar{x},\delta)$ we have $\|f^*(\bar{x},A)-f^*(x,A)\|<\epsilon$. It follows that, $\forall x\in B$,

$$\begin{aligned}&\|f^*(x,A)-f^*(x,A')\|\\ &\qquad\geq\|f^*(\bar{x},A)-f^*(\bar{x},A')\|-\|f^*(\bar{x},A)-f^*(x,A)\|-\|f^*(\bar{x},A')-f^*(x,A')\|\\ &\qquad\geq\|f^*(\bar{x},A)-f^*(\bar{x},A')\|-2\epsilon\\ &\qquad\geq\|f^*(\bar{x},A)-f^*(\bar{x},A')\|-\bar{\epsilon}>0\end{aligned}$$

Finally, as $\bar{x}\in\mathrm{Supp}(P^*_x)$ and $B(\bar{x},\delta)\subseteq I$, we conclude that

$$\mathbb{P}_x(I)\geq\mathbb{P}_x(B(\bar{x},\delta))>0,$$

therefore, we are in the hypothesis of Theorem 4.2 and can conclude that

$$\mathcal{L}^{dist}(\theta,\psi^*)=0 \implies \begin{cases}\mathcal{L}^{point}(\theta,\psi^*) \text{ is minimal}\\ \mathcal{L}^{cal}(\theta)=0,\end{cases}$$

□

B. Reducing the variance of the gradient estimator

B.1. Score-function gradient estimator

For computing $\nabla_\theta\mathcal{L}^{dist}(\theta,\psi)$, we rely on a score-function gradient estimator (SFE) [Williams, 1992; Mohamed et al., 2020] which uses the log derivative trick to rewrite the gradient of an expected loss L as $\nabla_\theta\mathbb{E}_{A\sim P^\theta}[L(A)]=$

$\mathbb{E}_{A\sim P^\theta}[L(A)\nabla_\theta \log P^\theta(A)]$, with $P^\theta(A)$ denoting the likelihood of $A \sim P^\theta$. Applying the SFE to our problem the gradient of the loss function w.r.t. θ reads:

$$\nabla_\theta \mathcal{L}^{dist}(\psi,\theta) = \mathop{\mathbb{E}}_{(x,y^*)\sim P^*_{x,y}}\Bigg[\mathop{\mathbb{E}}_{\hat{y}_1,\hat{y}_2\sim P^{\theta,\psi}_{y|x}}\Big[\kappa(\hat{y}_1,\hat{y}_2)\nabla_\theta \log\Big(P^{\theta,\psi}_{y|x}(\hat{y}_1)P^{\theta,\psi}_{y|x}(\hat{y}_2)\Big)\Big] - 2\mathop{\mathbb{E}}_{\hat{y}\sim P^{\theta,\psi}_{y|x}}\Big[\kappa(y^*,\hat{y})\nabla_\theta \log P^{\theta,\psi}_{y|x}(\hat{y})\Big]\Bigg] \quad (9)$$

SFEs are known to suffer of high variance [Mohamed et al., 2020]. Following Section B.2 derives a variance-reduction technique based on control variates that requires negligible computational overhead.

B.2. Variance-reduced loss for SFE

Two natural approaches to reduce the variance of MC estimates of (9) involve (i) increasing the number B of training data points in the mini-batch used for each gradient estimate and (ii) increasing the number N_{adj} of adjacency matrices sampled for each data point in (7). These techniques act on two different sources of noise. Increasing B decreases the variance coming from the data-generating process, whereas increasing N_{adj} improves the approximation of the predictive distribution $P^{\theta,\psi}_{y|x}$. Nonetheless, by fixing B and N_{adj}, it is possible to further reduce the latter source of variance by employing the *control variates* method [Mohamed et al., 2020] that, in our case, requires only a negligible computational overhead but sensibly improves the training speed (see Section 5).

Consider the expectation $\mathbb{E}_{A\sim P^\theta}[L(A)\nabla_\theta \log P^\theta(A)]$ of the SFE – both terms in (9) can be cast into that form. With the control variates method, $L(A)$ is replaced by a surrogate function

$$\tilde{L}(A) = L(A) - \beta\Big(h(A) - \mathbb{E}_{A\sim P^\theta}[h(A)]\Big) \quad (10)$$

that leads to a reduced variance in MC estimator while maintaining it unbiased. In this paper, we set function $h(A)$ to $\nabla_\theta \log P^\theta(A)$ and show how to compute a near-optimal choice for scalar value β, often called *baseline* in the literature. As the expected value of $\nabla_\theta \log P^\theta(A)$ is zero, gradient (9) rewrites as

$$\nabla_\theta \mathcal{L}^{dist} = \mathop{\mathbb{E}}_{(x,y^*)\sim P^*_{x,y}}\Bigg[\mathop{\mathbb{E}}_{A_1,A_2\sim P^\theta_A}\Big[\big(\kappa(f_\psi(x,A_1),f_\psi(x,A_2)) - \beta_1\big)\ \nabla_\theta \log\big(P^\theta_A(A_1)P^\theta_A(A_2)\big)\Big] - 2\mathop{\mathbb{E}}_{A\sim P^\theta_A}\Big[\big(\kappa(y^*,f_\psi(x,A)) - \beta_2\big)\ \nabla_\theta \log P^\theta_A(A)\Big]\Bigg]. \quad (11)$$

In Appendix B.3, we show that in our setup the best values of β_1 and β_2 are approximated by

$$\tilde{\beta}_1 = \mathop{\mathbb{E}}_{\substack{x\sim P^*_x\\ A_1,A_2\sim P^\theta_A}}\Big[\kappa\big(f_\psi(x,A_1),f_\psi(x,A_2)\big)\Big], \qquad \tilde{\beta}_2 = \mathop{\mathbb{E}}_{\substack{(x,y^*)\sim P^*_{x,y}\\ A\sim P^\theta_A}}\Big[\kappa\big(y^*,f_\psi(x,A)\big)\Big], \quad (12)$$

which can be efficiently computed via MC, as kernel values in (12) are already computed to estimate (11).

B.3. Estimation of optimal β_1 and β_2

Here we show that, when reducing the variance of the SFE via control variates in (11), the best β_1 and β_2 can be approximated by

$$\tilde{\beta}_1 = \mathop{\mathbb{E}}_{\substack{x\sim P^*_x\\ A_1,A_2\sim P^\theta_A}}\Big[\kappa\left(f_\psi(x,A_1),f_\psi(x,A_2)\right)\Big], \qquad \tilde{\beta}_2 = \mathop{\mathbb{E}}_{\substack{(x,y^*)\sim P^*_{x,y}\\ A\sim P^\theta_A}}\Big[\kappa\left(y^*,f_\psi(x,A)\right)\Big], \quad (13)$$

Consider generic function $L(A)$ depending on a sample A of a parametric distribution $P^\theta_A(A)$ and the surrogate loss $\tilde{L}(A)$ in (10), i.e.,

$$\tilde{L}(A) = L(A) - \beta\Big(h(A) - \mathbb{E}_{A\sim P^\theta}[h(A)]\Big); \quad (14)$$

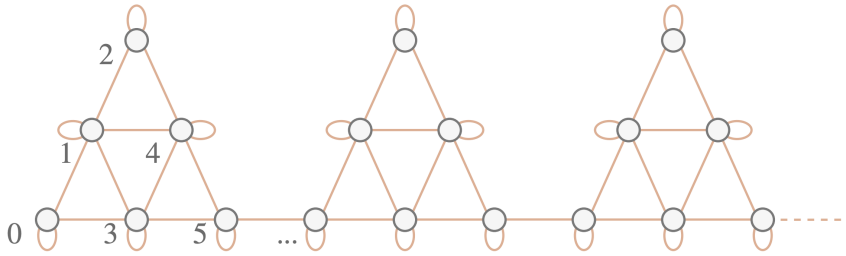

Figure 3: The adjacency matrices used in this paper are sampled from this graph. Each edge in orange is independently sampled with probability θ^*. In the picture 3 communities of an arbitrarly large graph are shown.

This choice is not new in the literature [Sutton et al., 1999; Mnih et al., 2016] where β is often referred to as *baseline*. The 1-sample MC approximation of the loss becomes

$$\nabla_\theta \mathbb{E}_{A\sim P^\theta}[L(A)] \approx \tilde{L}(A')\nabla_\theta \log P^\theta(A') = (L(A') - \beta)\nabla_\theta \log P^\theta(A'), \tag{15}$$

with A' sampled from P_A^θ. The variance of the estimator is

$$\begin{aligned}\mathbb{V}_{A\sim P^\theta}\left[(L(A) - \beta)\nabla_\theta \log P^\theta(A)\right] = \mathbb{V}_{A\sim P^\theta}\left[L(A)\nabla_\theta \log P^\theta(A)\right] + \\ + \beta^2\, \mathbb{E}_{A\sim P^\theta}\left[\left(\nabla_\theta \log P^\theta(A)\right)^2\right] - 2\beta\, \mathbb{E}_{A\sim P^\theta}\left[L(A)\left(\nabla_\theta \log P^\theta(A)\right)^2\right]\end{aligned} \tag{16}$$

and the optimal value β that minimizes it is

$$\tilde{\beta} = \frac{\mathbb{E}_{A\sim P^\theta}\left[L(A)\left(\nabla_\theta \log P^\theta(A)\right)^2\right]}{\mathbb{E}_{A\sim P^\theta}\left[\left(\nabla_\theta \log P^\theta(A)\right)^2\right]} \tag{17}$$

If we approximate the numerator with $\mathbb{E}[L(A)]\mathbb{E}[(\nabla_\theta \log P^\theta(A))^2]$, we obtain that $\tilde{\beta} \approx \mathbb{E}[L(A)]$. By substituting $L(A)$ with the two terms of (9) we get the values of β_1 and β_2 in (13).

We experimentally validate the effectiveness of this choice of β in Section 5.

C. Further experimental details

C.1. Dataset description and models

In this section, we describe the considered synthetic dataset, generated from the system model (1). The latent graph distribution P_A^* is a multivariate Bernoulli distribution of parameters θ_{ij}^*: $P_A^* \equiv P_{\theta^*}(A) = \prod_{ij} \theta_{ij}^{*A_{ij}} (1-\theta_{ij}^*)^{(1-A_{ij})}$. The components of θ^* are all null, except for the edges of the graph depicted in Figure 3 which are set to $3/4$. A heatmap of the adjacency matrix can be found in Figure 4.

Regarding the GNN function f^*, we use the following system model:

$$\begin{cases} y = f_{\psi^*}(A, x) = \tanh\left(\sum_{l=1}^{L} \mathbb{1}[A^l \neq 0] x \psi_l^*\right) \\ A \sim P_{\theta^*}(A) \end{cases} \tag{18}$$

Where $\mathbb{1}[\cdot]$ is the element-wise indicator function: $\mathbb{1}[a] = 1 \iff a$ is true. $x \in \mathbb{R}^{N \text{ x } d_{in}}$ are randomly generated inputs: $x \sim \mathcal{N}(0, \sigma_x^2 \mathbb{I})$. $\psi_l^* \in \mathbb{R}^{d_{out} \text{ x } d_{in}}$ are part of the system model parameters. We summarize the parameters considered in our experiment in Table 1.

The approximating model family (2) used in the experiment is the same as the data-generating process, with all components of parameter vectors θ and ψ being trainable. The squared MMD discrepancy is defined over Rational Quadratic kernel

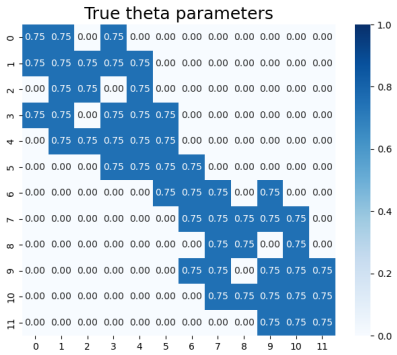

Figure 4: θ^*_{ij} parameters for each edge of the latent adjacency matrix. Each square corresponds to an edge, the number inside is the probability of sampling that edge for each prediction.

θ^*	0.75
σ_x	1.5
N	12
d_{in}	4
d_{out}	1
ψ_1^*	$[-0.2, 0.4, -0.8, 0.6]$
ψ_2^*	$[-0.3, 0.8, 0.2, -0.7]$

Table 1: Table of the parameters used to generate the synthetic dataset.

[Bińkowski et al., 2018]

$$\kappa(y', y'') = \left(1 + \frac{\|y' - y''\|_2^2}{2\,\alpha\,\sigma^2}\right)^{-\alpha}$$

of parameters $\sigma = 0.7$ and $\alpha = 0.02$.

The model is trained using Adam optimizer [Kingma & Ba, 2014] with parameters $\beta_1 = 0.6$, $\beta_2 = 0.95$. Where not specified, the learning rate is set to 0.1 and decreased to 0.01 after 5 epochs. We grouped data points into batches of size 128. Initial values of θ are independently sampled from the $\mathcal{U}(0.25, 0.35)$ uniform distribution.

C.2. Description of the experiment in Section 3

In this experiment, we generate 512 data points using the system model described in Appendix C.1. We construct a model identical to the system model, except that $\theta_{ij} = p$ for all i, j where $\theta^*_{i,j} = 0.75$ and 0 elsewhere. We vary scalar p from 0.5 to 1 with steps of 0.025. Therefore, only the model with $p = 0.75$ is identical to the data-generating model.

For each input x in the dataset, a point prediction is produced by sampling $N_{adj} = 32$ adjacency matrices and computing the median. This approach allows to estimate $\mathcal{L}^{point}$ using the MAE as loss function ℓ, as depicted by the red points in Figure 1, for different values of θ. For comparison purposes, we estimate $\mathcal{L}^{dist}$ using the maximum mean discrepancy as proposed in Section 4.

C.3. Compute resources and open-source software

The paper's experiments were run on a workstation with AMD EPYC 7513 processors and NVIDIA RTX A5000 GPUs; on average, a single model training terminates in a few tens of minutes with a memory usage of about 2GB.

The developed code relies on PyTorch [Paszke et al., 2019] and the following additional open-source libraries: PyTorch Geometric [Fey & Lenssen, 2019], NumPy [Harris et al., 2020] and Matplotlib [Hunter, 2007].