GNN-Guided Block Selection in Gibbs MCMC

Anonymous Author(s)
Affiliation
Address
email

Abstract

Exact inference in large Bayesian Networks (BNs) is computationally intractable, limiting its practical application. Markov Chain Monte Carlo (MCMC) methods like Gibbs sampling offer a scalable alternative but can be arbitrarily slowed by highly coupled variables— addressable by jointly sampling some variables as a block. We propose an automated block detection method to amortise inference time: training a Graph Neural Network (GNN) to propose blocks directly from the BN structure. We further introduce a novel coupling heuristic based on the Markov chain's spectral gap, which we show can be more robust than existing heuristics. Our GNN, trained on a dataset of small, randomly generated BNs, generalizes well to larger networks, accelerating MCMC sample efficiency in our experiments.

1 Introduction

Bayesian Networks (BNs) are a class of probabilistic graphical models that represent complex multivariable distributions through local conditional dependencies. They provide an inerptretable framework for probabilistic modeling, supporting arbitrary posterior inference queries given observed evidence. BNs can be constructed from human expert knowledge, structure learning algorithms, or even elicited from large language models. However, exact inference is computationally intractable for larger networks. Approximate inference methods like **Markov Chain Monte Carlo (MCMC)** offer a practical alternative for achieving acceptable accuracy within reasonable computation time.

Gibbs sampling is a common MCMC method for BNs due to its simple computation of local updates and 100% acceptance ratio. However, highly coupled variables can severely slow convergence and mixing. **Blocked Gibbs sampling** addresses this by grouping coupled variables into **blocks** that are jointly sampled, breaking out of sticky Markov chain states.

While practitioners can specify blocks a priori using domain knowledge, automatic blocking methods are essential for general applicability. Venugopal and Gogate (2013) present a robust algorithm for dynamically proposing blocks during MCMC based on information revealed by initial samples. However, this approach faces a fundamental catch-22: effective blocks require sufficient posterior samples of the full range of Markov chain modal states to observe coupling, yet collecting this data is slowed by the couplings themselves.

Circumventing this issue, we train a **Graph Neural Network (GNN)** Kipf and Welling (2017) to propose effective blocks directly from BN structure before MCMC starts; this can be used in tandem with dynamic block refinement. Our GNN is trained offline on a diverse dataset of randomly generated BNs, amortising the computational cost of block identification. We further introduce an alternative coupling heuristic based on the Markov chain's **spectral gap**, which demonstrates superior performance on adversarially coupled BNs compared to existing measures.

Yoon et al. (2019) trained GNNs to directly do inference, though with fixed approximation error; our method fully integrates into MCMC and can be used with existing techniques. GNN-proposed blocks can still be refined further dynamically refined using MCMC samples (Venugopal and Gogate,

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2013). Learned variable sampling distributions allow proposing larger blocks (Wang et al., 2018), non-uniform variable/block selection rates speed up convergence, and of course parallelisation across chains and variables with non-intersecting Markov blankets is still effective (Gonzalez et al., 2011).

2 Guiding Block Selection in Gibbs MCMC with GNNs

In order to propose good blocks, we first must investigate coupling: Section 2.1 derives a good heuristic metric to detect coupling between pairs of variables. We use this in the algorithm of Section 2.2 to propose a block partition $\mathbb{B} = \{B_i\}_{i=1}^k$ to cover the variables of the BN: $\{X_j\}_{X_j \in \mathcal{V}} = \bigsqcup B_i$ (though in some cases overlapping blocks are better). Finally in Section 2.3, we show how we train a GNN to predict this coupling metric; put together we can propose blocks for an arbitrary input BN.

2.1 Deriving the spectral gap coupling heuristic

For a finite discrete-time Markov chain (S, T, μ_0) of state space S, transition matrix $T_{ij} = P(Z_{t+1} = s_j | Z_t = s_i)$ and initial distribution $Z_0 \sim \mu_0$, convergence in **total variation distance (TVD)** to the stationary distribution π is governed by its spectral gap Levin and Peres (2017), per the bound $||\mu_0^T T^n - \pi^T||_{TV} = O(\lambda_2^n)$, where $\lambda_2 \neq 1$ is the second-largest eigenvalue. The spectral gap $\gamma = 1 - \lambda_2$ thus provides a natural basis for block selection. Unfortunately, computing the spectral gap for multivariable chains is impractical except in special cases (see e.g. Chimisov et al. (2018)), requiring enumeration of an exponentially large product state space.

An important simplification that Venugopal and Gogate (2013) make is to consider only variable pairs. For two variables X, Y in a BN, the joint posterior $\pi(X, Y)$ can be feasibly estimated in a short MCMC run as the $|\mathcal{X}| \times |\mathcal{Y}|$ matrix P. The hellinger distance of P vs its independent assumption rank one approximation, $Q = p_X p_Y^T$, where e.g. $(p_X)_x = \sum_{y \in Y} P_{xy}$, is a good metric of coupling as if the indepedence approximation is exact, X and Y would seem to be not at all coupled. Their proposed heuristic is thus $\text{HD}(P, Q) := \frac{1}{\sqrt{2}} \sqrt{\sum_{i,j} (\sqrt{P_{ij}} - \sqrt{Q_{ij}})^2}$.

We develop a spectral gap based pairwise score by constructing an averaged subset transition matrix $T^{(X,Y)} : |\mathcal{X}| \times |\mathcal{Y}| \to |\mathcal{X}| \times |\mathcal{Y}|$ that models marginal dynamics for variables X, Y under standard Gibbs sampling. This decomposes as $T^{(X,Y)} = \frac{1}{2} T_X^{(X,Y)} + \frac{1}{2} T_Y^{(X,Y)}$, where $T_Y^{(X,Y)}$ represents sampling Y with X fixed. Denoting their joint Markov Blanket as $\text{MB}(X, Y) = \text{MB}$, the transition probabilities $T_Y^{(X,Y)} : (x_0, y_0) \to (x_0, y')$ are derived by averaging over the distribution of MB, which we approximate as $\text{MB} \sim \pi(\text{MB} | x_0, y_0)$ (as we perform Gibbs sampling over the whole BN)

$$P_{\text{Gibbs}}(Y | X = x_0, Y = y_0, \text{MB}) \quad \propto \quad P(Y | Pa(Y)) \prod_{Z : Y \in Pa(Z)} P(Z | Pa(Z)) \tag{1}$$

$$P_{\text{Gibbs}}(Y | X = x_0, Y = y_0) \approx \int \pi(\text{MB} | x_0, y_0) P_{\text{Gibbs}}(y | \text{MB}, x_0) d\text{MB} \tag{2}$$

Since $\pi(\text{MB} | x_0, y_0)$ is intractable, we approximate by averaging over $y_0 \sim \pi(Y = y | X = x) = \int \pi(\text{MB} | x) P(y | \text{MB}, x) d\text{MB}$ to get uniform probabilities $\tilde{T}_Y^{(X,Y)} : (x_0, *) \to (x_0, y')$. This represents a best-case mixing scenario where sampling y' jumps straight to the true posterior $\pi(Y | X = x_0)$, so is a lower bound for mixing time of the actual Gibbs chain.

$$E_{Y_0 \sim \pi(Y | x_0)} \left[P_{\text{Gibbs}}(Y | X = x_0, Y = y_0) \right] \approx \tag{3}$$

$$\int \left[\int \pi(\text{MB} | x_0, y_0) P_{\text{Gibbs}}(y | \text{MB}, x_0) d\text{MB} \right] \pi(y_0 | x_0) dy_0 \tag{4}$$

$$\int \left[\int \pi(\text{MB} | x_0, y_0) \pi(y_0 | x_0) dy_0 \right] P_{\text{Gibbs}}(y | \text{MB}, x_0) d\text{MB} = \tag{5}$$

$$\int \pi(\text{MB} | x_0) P_{\text{Gibbs}}(y | \text{MB}, x_0) d\text{MB} = \pi(Y | X = x_0) \tag{6}$$

This allows us to derive an approximate matrix $\tilde{T}^{(X,Y)}$ and compute its spectral gap using only the posterior $\pi(X, Y)$: the same information required for the Hellinger distance heuristic. While these heuristics perform similarly on aggregate across random BNs, in certain adversarially constructed BNs (e.g. see A.2) spectral gap substantially outperforms Hellinger distance for block proposals.

2.2 Adapting the greedy pairwise heuristic blocking algorithm

We adapt the greedy block merging algorithm from Venugopal and Gogate (2013). Given a pairwise distance function $\rho(X, Y)$ between variables in the BN, where larger values indicate stronger coupling, we construct a block partition $\mathbb{B}$ of a BN of n variables. Starting with n singleton blocks, we iteratively merge the pair of blocks with maximum inter-block score sum $\sum_{X \in B_i, Y \in B_j} \rho(X, Y)$ (or mean/max). We compare replacing hellinger distance based ρ_{HD} with spectral gap We explore replacing the hellinger distance based ρ_{HD} with one from spectral gap ρ_{spectral}. For simplicity we constrain merged block size to a uniform cap rather than via more precise computational cost of Gibbs sampling each block's joint distribution — otherwise larger blocks are generally better.

2.3 Producing block proposals with GNNs

GNNs provide a natural architecture to analyse an input BN so as to produce block proposals. Rather than learning to propose blocks directly, we supervise train the GNN to produce either ρ_{HD} or ρ_{spectral} with a final bilinear layer between node embeddings. For ease of computing pairwise posteriors for these ground truth heuristics, we limit our training dataset to randomly generated BNs of 10-40 nodes and average degree 1.7. Despite this constraint, our GNNs generalise effectively to larger networks, with spectral gap showing slight performance advantages over Hellinger distance: see Figure 1.

Loopy Belief Propagation (LBP) Koller and Friedman (2009) inspires our architecture through its effective graph-based message passing for computing single variable marginal posteriors. LBP operates on **factor graphs**, where factors generalize **Conditional Probability Tables (CPTs)** $P(X|Pa(X))$ by removing normalisation constraints. We convert BNs to factor graphs by replacing each CPT hyperedge $P(X|Pa(X) = \{Y_i\}_{i=1}^{\deg(X)})$ with a **factor node** F_X. Variable nodes $X, Y_1, ..., Y_{\deg(X)}$ connect bidirectionally to F_X, enabling distinct message passing phases.

We employ 2 relational graph convolution layers Schlichtkrull et al. (2018) with GRU units Cho et al. (2014) across 2 message passing rounds. While deeper networks with additional rounds would likely improve performance, scaling proved challenging; future work could explore alternative architectures like graph attention networks Veličković et al. (2018).

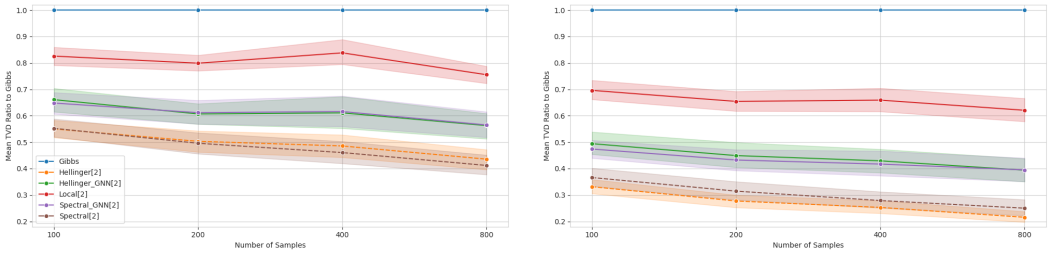
We trained on 22,732 discrete BNs using an 80-20 train-validation split. CPT encoding constrains our architecture: we flatten each CPT as an initial node feature vector of size $N_d^{N_p+1}$ (max domain size $N_d = 5$, max number of parents $N_p = 6$) for factor nodes as well as a factor indicating one-hot flag, while variable nodes are zero intialised. Edge types encode direction (to/from factor) and position (0 for child, $1, ..., N_p$ for parents), determining relational graph convolution parameters. Following LBP's un-normalised factors, we handle evidence by disconnecting observed variables from their factors and contracting those CPTs to conforming entries, which become unnoramlised. To encourage GNN comprehension across all factors, we further add multiplicative noise to all CPTs making them all unnormalised - this improves performance.

Randomly generated BNs were filtered to fit the max number of parents constraint, with CPTs deliberately extremised with higher occurence of entries in [0.99, 1.0] to encourage highly coupled variables and hence the presence of some higher spectral gap / hellinger distance scores. Additional training details appear in Appendix A.1.

3 Experiments

We evaluate our method's practical utility, particularly generalisation performance on larger BNs, where exact inference is intractable.

We test our GNN's block proposal performance on 100 BNs of 85-115 nodes each, measuring mean TVD between MCMC 200-sample predicted and "ground truth" (7,000-sample) single variable marginals. To account for stochasticity in highly coupled networks, we average results across 25



(a) Maximum block-size 2

(b) Maximum block-size 4

Figure 1: GNN block proposals on 100 small BNs (10-40 nodes) test-set achieve lower mean TVDs than Gibbs and the control of random local blocking, across a range of MCMC sample sizes. Mean and 95% confidence intervals shown.

independent runs per BN. Figure 2 shows that both GNN-predicted spectral gap and Hellinger distance blocking substantially outperform random local blocking across maximum block sizes (2 and 4 shown). Spectral gap shows only very modest improvements over Hellinger distance.

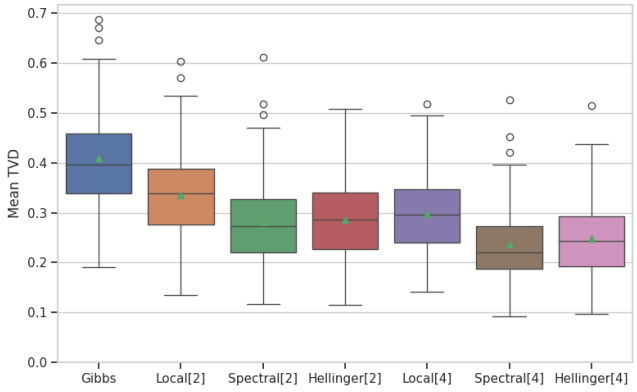


Figure 2: 200 sample MCMC runs of various maximum block-sizes, on 100 large BNs (85-115 nodes). GNN guided block proposals generalise well to larger BNs where exact inference is infeasible, outperforming the random local blocking control; spectral slightly outpeforms Hellinger. Means shown as green triangles.

4 Conclusion

We presented a method to accelerate MCMC inference in Bayesian Networks using Graph Neural Networks to propose variable blocks for joint sampling. This amortised approach circumvents the catch-22 of dynamic blocking methods that require samples to identify the couplings that slow sampling. Our experiments demonstrate that GNN-proposed blocks, trained on spectral gap or Hellinger distance heuristics, substantially accelerate posterior convergence on large networks.

Future work could explore more expressive architectures like Graph Attention Networks with additional layers, to better capture global dependencies. A more ambitious direction would be to replace heuristic supervision with direct block proposal mechanisms trained via reinforcement learning on convergence speed, perhaps even with active exploration of sampling moves. Further analysis is warranted to characterize the topological or parametric properties of BNs for which our proposed spectral gap heuristic offers the greatest advantage over hellinger distance.

References

Chimisov, C., Latuszynski, K., and Roberts, G. (2018). Adapting the gibbs sampler. *arXiv preprint arXiv:1801.09299*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Gonzalez, J., Low, Y., Gretton, A., and Guestrin, C. (2011). Parallel gibbs sampling: From colored fields to thin junction trees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 324–332. JMLR Workshop and Conference Proceedings.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Levin, D. A. and Peres, Y. (2017). *Markov chains and mixing times*, volume 107. American Mathematical Soc.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks.

Venugopal, D. and Gogate, V. (2013). Dynamic blocking and collapsing for gibbs sampling. *arXiv preprint arXiv:1309.6870*.

Wang, T., Wu, Y., Moore, D., and Russell, S. J. (2018). Meta-learning mcmc proposals. *Advances in neural information processing systems*, 31.

Yoon, K., Liao, R., Xiong, Y., Zhang, L., Fetaya, E., Urtasun, R., Zemel, R., and Pitkow, X. (2019). Inference in probabilistic graphical models by graph neural networks. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 868–875. IEEE.

A Technical Appendices and Supplementary Material

Technical appendices with additional results, figures, graphs and proofs may be submitted with the paper submission before the full submission deadline (see above), or as a separate PDF in the ZIP file below before the supplementary material deadline. There is no page limit for the technical appendices.

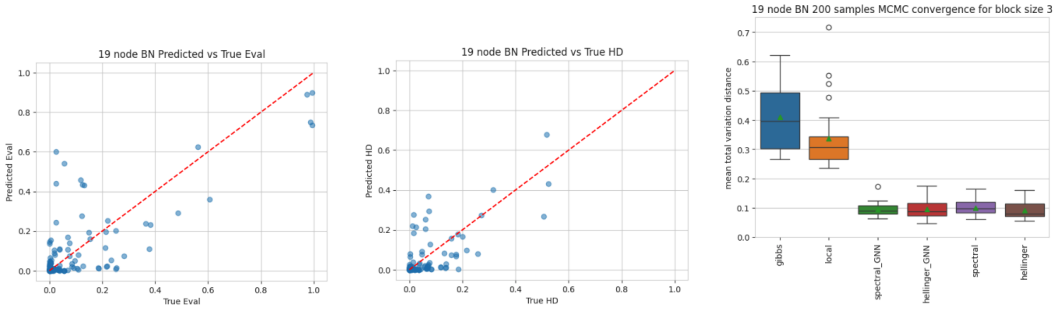
A.1 GNN training and examples

We trained both the spectral gap and hellinger distance predicting GNNs on a single A600 for 20 epochs each. We used pyAgrum's random bayesian network generator, which produces diverse graphs, of varying degree (of given average, 1.7), varying CPTs (which we further extremised to promote coupling), varying domain size up to a given cap (we constrained to $N_d = 5$). We rejection sample out BNs satisfying the max number of parents constraint ($N_p = 6$), and finally randomly choose a number of evidence variables and their observed assignments, of between 1 and 20% of the BN's variables.

For spectral gap, we trained the GNN to predict the second largest eigenvalue of the approximate Markov chain subset transition kernel of a pair of variables in the BN. For our randomly generated BNs, this is quite an unbalanced distribution, with high concentration of values close to 0, same for hellinger distance - however a few pairs of nodes have higher values corresponding to a higher degree of coupling. In order to encourage the GNN to learn not just to predict 0 distance uniformly, we actually employ mean cubed loss rather than mean squared, to more sharply penalise large errors.

A.2 Example Adversarial BN requiring spectral gap analysis

We constructed by hand a small BN of three nodes X, Y, Z and conditional structure $Y \to X;\ Y \to Z$ where both X, Y and Y, Z are desirable blocking candidates. However assuming a max block-size of 2 where we only do a partition style blocking, the spectral gap and hellinger distance metrics disagree on which is preferential to block. Y is a root node with uniform distribution $P(Y = i) = 0.25$ for $i \in \{0, 1, 2, 3\}$, and the CPTs of $P(X|Y)$ and $P(Z|Y)$, shown in Figures 5b and 5c, both ensure that the marginal probabilities $P(X)$ and $P(Z)$ are likewise uniform. The far more

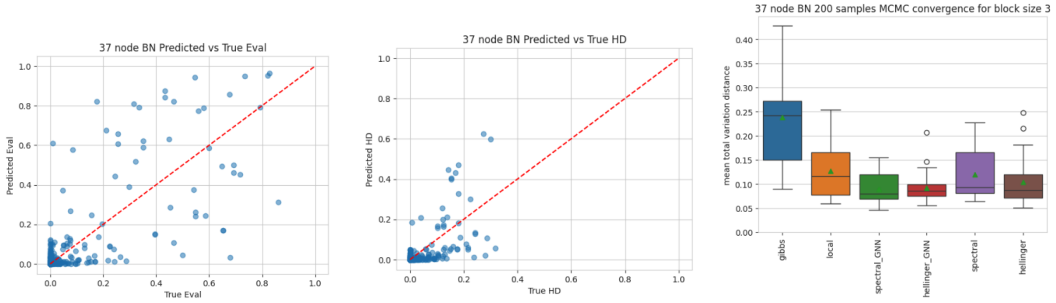


(a) GNN 2nd largest eigenvalue λ_2 predictions. Corr-coeff 0.79

(b) GNN Hellinger distance predictions. Corr-coeff 0.72

(c) MCMC convergence for different blocking methods

Figure 3: GNN predictive accuracy of pairwise variable distance metircs, and MCMC convergence performance, shown for a single validation set BN of 19 nodes.



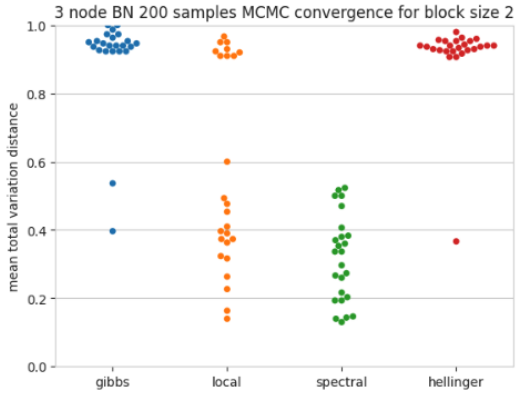
(a) GNN 2nd largest eigenvalue λ_2 predictions. Corr-coeff 0.77

(b) GNN Hellinger distance predictions. Corr-coeff 0.65

(c) MCMC convergence for different blocking methods

Figure 4: GNN predictive accuracy of pairwise variable distance metircs, and MCMC convergence performance for a real life BN of 37 nodes: `https://www.bnlearn.com/bnrepository/discrete-medium.html#alarm`, with evidence VENTALV: 0, HYPOVOLEMIA: 1, INSUFFANESTH: 0, HRBP: 1

extreme, 2x2-modal concentration of probabilities in $P(X|Y)$ actually lead to far slower convergence of the Markov chain, as evidenced in Figure 5a, unlike the more weakly 4-modally concentrated coupling in $P(Z|Y)$. Hellinger distance scores are however insensitive to this large discrepancy, indeed giving preference to blocking Z, Y over X, Y: $\text{HD}(X, Y) = 0.532$ and $\text{HD}(Y, Z) = 0.543$. The eigenvalues of the approximate transition matrices $\tilde{T}^{(Y,Z)}$ has 2nd largest eigenvalue 0.8652 so spectral gap 0.1348 as opposed to $\tilde{T}^{(X,Y)}$ which has $\lambda_2 = 0.9994$ and so a far smaller spectral gap of 0.0006, so spectral gap based blocking correctly blocks X, Y. In this case random local blocking blocks correctly half of the time.



(a) MCMC convergence for max block-size 2

	X			
Y	0	1	2	3
0	0.0001	0.4999	0.0001	0.4999
1	0.4999	0.0001	0.4999	0.0001
2	0.0001	0.4999	0.0001	0.4999
3	0.4999	0.0001	0.4999	0.0001

(b) X Given Y CPT

	Z			
Y	0	1	2	3
0	0.0455	0.9082	0.0455	0.0009
1	0.0009	0.0455	0.9082	0.0455
2	0.0455	0.0009	0.0455	0.9082
3	0.9082	0.0455	0.0009	0.0455

(c) Z Given Y CPT

Figure 5: Adversarial 3 node example BN $Y \to X;\ Y \to Z$ has high degrees of coupling, but magnitudes more so between X, Y than Z, Y. The spectral gap heuristic correctly identifies this unlike hellinger distance, and so exhibits far better MCMC convergence. Shown here is the