An Empirical Evaluation of Model Completion for Causal Inference

Jiapeng Zhao
University of California, Irvine

Elias Bareinboim
Columbia University

Rina Dechter
University of California, Irvine

Abstract

Model-completion methods learn a full causal generative model consistent with observational data and a given causal graph, and answer interventional queries via probabilistic inference. We empirically compare two approaches presented recently. One approach learns the model using EM, named *EM for Causal Inference (EM4CI)*. The other approach uses neural networks for completion, yielding two neural causal model approaches, MLE–NCM and GAN–NCM. We evaluate these methods on synthetic discrete benchmarks spanning multiple graph families and scales. Results show that EM4CI seems superior on large graphs in terms of accuracy, while NCM-based methods can be competitive on small models but incur substantially higher computational cost.

1 Introduction

Model completion for causal inference was explicitly formulated in the work of (Raichev et al., 2024), who proposed to use the EM scheme for learning. The central idea of model completion is to first learn a full causal generative model that is consistent with the observational distribution and the given causal graph, and then answer interventional queries by performing probabilistic inference on the learned model. This paradigm contrasts with traditional estimand-based approaches, where one must first derive an explicit estimand expression for a causal query and then evaluate it using observational data.

Learning a complete causal model offers several advantages. Once a model consistent with the observational distribution is obtained, causal queries can be answered uniformly through model truncation and probabilistic inference, without deriving query-specific estimands. This can be particularly appealing for complex causal graphs, where estimand expressions may be lengthy, difficult to derive, or computationally expensive to evaluate.(Raichev et al., 2024)

Neural Causal Models (NCMs) (Xia et al., 2021, 2022) adopt a closely related philosophy. Rather than explicitly parameterizing conditional probability tables, NCMs use neural networks to represent causal mechanisms and learn a generative causal model directly from observational data. Causal queries are then evaluated by intervening on the learned neural causal model and performing simulation-based inference. Despite their very different modeling and optimization strategies, both EM4CI and NCMs can be viewed as instances of the broader model-completion paradigm.

In this paper, we empirically compare EM4CI with the two NCM-based methods, MLE-NCM (Xia et al., 2021) and GAN-NCM (Xia et al., 2022). We evaluate these approaches in terms of estimation accuracy for causal-effect queries and training efficiency, across a range of synthetic benchmarks. Our experiments span small and large causal graphs, varying domain sizes, and increasing sample sizes, allowing us to study scalability with respect to both graph structure and data availability. Through this comparison, we aim to better understand the practical strengths and limitations of model-completion methods for causal inference.

Our results show that, EM4CI achieves higher accuracy on large models, while NCM-based methods can be competitive on smaller ones. EM4CI is also typically the fastest method at fewer sample sizes, though its runtime increases at the largest scale and can become comparable to GAN–NCM on some large graphs. Among the neural approaches, MLE–NCM can attain good accuracy in small settings but is often computationally expensive, whereas GAN–NCM scales better but exhibits less stable accuracy across graph structures. Overall, these results highlight clear performance and computational trade-offs among model-completion methods.

2 Background

We consider causal models represented by a directed acyclic graph (DAG) and a structural causal model (SCM) (Pearl, 2009) over observed variables V and latent variables U (e.g., unobserved confounders). Formally, an SCM is defined by a set of structural functions $\mathcal{F} = \{f_V \mid V \in \mathcal{V}\}$ (called mechanisms), where each observed variable $V \in \mathcal{V}$ is determined by $V = f_V(PA_V, U_V)$ with endogenous parents $PA_V \subseteq \mathcal{V}$ and exogenous parents $U_V \subseteq \mathcal{U}$. Together with a distribution $P(U)$ over the latent variables, the SCM induces a **Causal Bayesian Network (CBN)** whose joint distribution factors as: $P(V, U) = \prod_{V_i \in V} P(V_i \mid \mathrm{PA}_i) \cdot \prod_{U_j \in U} P(U_j)$. Interventions are defined via the $do(\cdot)$ operator: under $do(X = x)$, the mechanism for X is replaced by the constant assignment $X = x$, and the resulting *truncated* model defines the interventional distribution $P(Y \mid do(X = x))$. The observational distribution $P(V)$ is obtained by marginalizing out the latent variables U.

Estimand-based Causal Inference A standard workflow is to (i) determine whether a query $P(Y \mid do(X))$ is *identifiable* from the observational distribution $P(V)$ and the causal graph, and if so (ii) derive an estimand—an expression in terms of $P(V)$—and evaluate it using data (e.g., plug-in or ML-based estimators) (Shpitser and Pearl, 2006; Tian, 2002; Jung et al., 2020, 2021). In complex graphs, derived estimands can involve high-dimensional conditionals and large summations, making evaluation expensive and potentially high-variance.

Model Completion Model-completion methods instead aim to learn a *full* causal generative model consistent with the observational distribution and the given graph, and then answer interventional queries by intervening on the learned model and performing probabilistic inference. For identifiable queries, consistency with the observed distribution is sufficient: any completed model that matches $P(V)$ yields the correct $P(Y \mid do(X))$.

2.1 Learning Causal Models Via EM4CI

(Raichev et al., 2024) propose *EM for Causal Inference* (EM4CI), which applies Expectation–Maximization (EM) to fit a latent-variable Causal Bayesian Network (CBN) whose observational distribution matches the data. The E-step performs posterior inference over latent variables given current parameters, and the M-step updates parameters to maximize the expected complete-data log-likelihood. Both training and query answering exploit the CBN structure exact inference is efficient when the graph has bounded treewidth (e.g., via bucket elimination or join-tree methods) (Dechter, 1999, 2013; Darwiche, 2009). EM4CI selects latent variable cardinalities via BIC, trading off fit and model complexity (Darwiche, 2009).

2.2 Neural Causal Models

Neural Causal Models (NCMs) (Xia et al., 2021, 2022) also implement model completion, but represent each causal mechanism using a neural network with an exogenous noise input. Unlike EM4CI, NCMs typically fix simple noise distributions and rely on the neural mechanisms to match $P(V)$. Once trained, causal effects are estimated by intervening on the learned neural model and using simulation-based inference.

We evaluate two NCM training strategies. *MLE–NCM* (Xia et al., 2021) fits the model by maximizing observational likelihood, while *GAN–NCM* (Xia et al., 2022) uses adversarial training to avoid explicit likelihood computation. Both share the same goal as EM4CI: complete a causal generative model from observational samples and answer queries via intervention on the learned model.

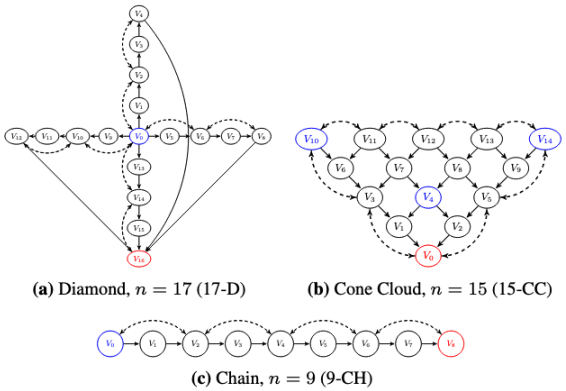
3 Experimental Evaluation

We conduct experiments to compare EM4CI, MLE–NCM, and GAN–NCM in terms of (i) estimation accuracy for causal effect queries and (ii) training time. Since EM4CI already includes significant empirical evaluations on synthetic networks with both small and large graphs, we reuse the models and datasets from EM4CI and run the NCM-based methods on the same graphs and using the same observational data.

3.1 Experimental Setup

Because we reuse the models and datasets from EM4CI, no modifications to the EM4CI implementation are required. We simply reran EM4CI with more trials to obtain reliable performance measures and test its training variance.

To adapt NCM-based methods to the EM4CI benchmarks, we extend the original implementation to support multi-valued discrete variables using categorical latent distributions and one-hot encodings with Gumbel–Softmax relaxation for differentiable training. Observational data are generated from CPT-specified CBNs following the EM4CI setup (Raichev et al., 2024), ensuring identical data-generating processes across methods. For evaluation, we use mean absolute deviation (MAD) between the true and estimated interventional distributions.



(a) Diamond, $n = 17$ (17-D) (b) Cone Cloud, $n = 15$ (15-CC)

(c) Chain, $n = 9$ (9-CH)

Figure 1: 3 large graphs can be scaled with any number of variables

3.2 Benchmarks and Metrics

3.2.1 Benchmarks

We follow the experimental setup of EM4CI (Raichev et al., 2024). Each benchmark instance is generated by first selecting a causal diagram and a target query. Given the diagram, we specify domain sizes for both observed and latent variables and randomly generate conditional probability tables (CPTs) for each variable conditioned on its parents, thereby defining a complete causal Bayesian network (CBN). Observational data are then obtained via forward sampling from the full CBN (Darwiche, 2009), after which the latent variables are discarded, yielding samples from the observed distribution. The ground-truth value of the target query is computed directly on the full CBN using an exact inference algorithm, such as join-tree propagation (Darwiche, 2009; Dechter, 2013).

The causal diagrams are chosen according to two regimes. First, we consider a collection of eight small benchmark graphs(Appendix C). Second, we study three scalable graph families with controllable treewidth: diamond networks (treewidth 5; Figure 1a), cone–cloud networks (treewidth $\mathcal{O}(\sqrt{n})$; Figure 1b) and chain networks (treewidth 3; Figure 1c), abbreviated as CH, D, and CC, respectively.

For the small graphs, the domain size of observed variables is set to $d = 2$, while latent variables have domain size $k = 10$. For the chain, diamond, and cone–cloud models, we use $(d, k) = (4, 10)$. The eight small graphs are evaluated using 100 and 1,000 observational samples, with 10 independent trials per setting, whereas the six large graphs are evaluated using 1,000 and 10,000 samples, with 5 trials per setting (Table 1).

Table 1: Benchmark regimes

Regime	Graphs	Samples	Trials
Small	exp1–8	100,1000	10
Large	6,15CC; 17,65D; 49,99CH	1000,10000	5

3.2.2 Collected Metrics

MAD For a query $P(Y \mid do(X))$, let $p_{x,y} = P^\star(Y = y \mid do(X = x))$ denote the ground-truth interventional probability and $\hat{p}_{x,y}$ its estimate. We define MAD $= \frac{1}{|\mathcal{D}(X)|\,|\mathcal{D}(Y)|} \sum_{x,y} \left|\hat{p}_{x,y} - p_{x,y}\right|$, where the summation ranges over $(x, y) \in \mathcal{D}(X) \times \mathcal{D}(Y)$.

Train Time Total training time required to fit each model on a given dataset, measured in seconds and excluding data generation and query evaluation. Training times are *not* directly comparable across methods due to different hardware usage: EM4CI runs on CPU, whereas GAN–NCM uses GPU acceleration. Nevertheless, the results still indicate relative computational efficiency.

95% Confidence Interval (CI) We report a 95% confidence interval for the mean MAD across independent trials, indicating that, under repeated sampling, we are 95% confident that the *mean MAD* lies within the reported range and reflecting the stability of training across runs. For each setting, we conduct $T \in \{5, 10\}$ independent trials (10 for small graphs and 5 for large graphs), and compute the confidence interval as mean $\pm$ $1.96 \cdot$ SEM, where SEM $= \frac{s}{\sqrt{T}}$ and s is the sample standard deviation of MAD across trials.

3.3 Results

In this section, we first present results on eight small synthetic models, followed by six large graphs. We report complete numerical results for the two classes on benchmarks in Tables 2 and 3, including sample sizes, accuracy via mean absolute deviation (MAD), and training time. Results are further visualized in Figures 2–9 (Appendix A) using bar charts with 95% confidence intervals. Due to the prohibitively high training cost of MLE–NCM at larger scales, it is only evaluated in settings where computation is feasible. Detailed analyses and discussions of these results are provided in the following subsections.

3.3.1 Results on Small Synthetic Models

Table 2 reports the average MAD and training time for EM4CI, GAN–NCM, and MLE–NCM on the eight small graphs, evaluated at sample sizes of $n = 100$ and

Table 2: MAD and Training Time on Small Graphs

		MAD			Time (s)		
Graph	n	EM	GAN	MLE	EM	GAN	MLE
exp1	100	.094	.083	**.010**	**.23**	887	451
exp2	100	.241	**.037**	.055	**.70**	396	553
exp3	100	**.052**	.143	.054	**.41**	700	837
exp4	100	.106	**.103**	.106	**.75**	850	507
exp5	100	.468	**.439**	.501	**.88**	799	595
exp6	100	.280	.157	**.050**	**.24**	837	1843
exp7	100	**.214**	.485	.249	**.42**	670	1542
exp8	100	.126	.056	**.046**	**.46**	513	816
exp1	1000	.093	**.003**	**.003**	**1.46**	627	537
exp2	1000	.127	**.041**	.084	**3.47**	435	486
exp3	1000	**.005**	.043	**.005**	**1.74**	566	616
exp4	1000	**.088**	.122	.124	**5.31**	561	689
exp5	1000	.417	.493	**.301**	**6.27**	493	562
exp6	1000	.126	.022	**.016**	**.30**	637	1134
exp7	1000	**.045**	.458	.102	**3.39**	616	1621
exp8	1000	.153	.051	**.007**	**2.36**	689	594

$n = 1000$ over 10 independent trials. In each trial, we independently sample an observational dataset, train the model, and evaluate the target query, yielding one value for each reported metric.

On small graphs, performance varies across structures and sample sizes. At $n = 100$, no single method dominates: EM4CI, GAN–NCM, and MLE–NCM each achieve the lowest MAD on different graphs, and variance across trials can be substantial. As the sample size increases to $n = 1000$, all methods improve and training variance decreases. MLE–NCM often achieves the lowest or tied-lowest MAD at this scale, but at significantly higher computational cost. EM4CI maintains competitive accuracy with consistently negligible runtime, while GAN–NCM exhibits moderate variability across structures. Overall, small-scale results indicate that neural approaches can be competitive in accuracy when data are sufficient, but with considerable computational overhead.

3.3.2 Results on Large Synthetic Models

Table 3 summarizes the average MAD and training time for six larger graphs evaluated at sample sizes $n = 1000$ and $n = 10000$. At $n = 1000$, EM4CI achieves consistently low MAD across all graph families, remaining robust as graph size increases. GAN–NCM shows substantial degradation on certain structures, particularly chain graphs, while MLE–NCM is generally less accurate than EM4CI and incurs extremely high computational cost. When the sample size increases to $n = 10000$, EM4CI continues to achieve the lowest MAD across most settings. However, its runtime grows significantly with sample size and graph complexity, leading to a crossover effect where GAN–NCM becomes faster on some of the largest graphs, despite remaining less accurate. These results highlight EM4CI's stability and accuracy at moderate scales, and the differing scalability trade-offs of neural approaches at very large sample sizes.

Table 3: MAD and Training Time (s) on Large Graphs

		MAD			Time (s)		
Graph	n	EM	GAN	MLE	EM	GAN	MLE
49-CH	1k	**.0015**	.587	.0085	**103**	447	67378
99-CH	1k	**.0052**	.360	.088	**319**	575	43267
17-D	1k	.042	**.013**	.052	**32**	1730	19905
65-D	1k	**.0088**	.473	.018	**238**	474	69332
6-CC	1k	**.015**	.204	.024	**10**	494	17491
15-CC	1k	**.0085**	.010	.010	**22**	1774	41457
49-CH	10k	**.0044**	.274	–	838	**664**	–
99-CH	10k	**.0034**	.444	–	2596	**1263**	–
17-D	10k	.041	**.017**	–	**246**	2317	–
65-D	10k	**.0039**	.289	–	2110	**896**	–
6-CC	10k	**.013**	.039	–	**63**	1267	–
15-CC	10k	**.0080**	.0093	–	**195**	2313	–

EM4CI was compared against traditional plug-in estimand methods on synthetic problems and real-world Bayesian networks (e.g., Alarm, Barley, and Win95), achieving higher accuracy (Raichev et al., 2024).

4 Conclusion

This paper compares EM4CI with MLE–NCM and GAN–NCM as representative model-completion approaches for causal inference. Across the synthetic benchmarks, EM4CI seems superior in terms of accuracy for the large models while the NCM models seems superior for the small models. EM4CI is also consistently the fastest method at moderate sample sizes; however, its runtime increases substantially at the largest scale considered ($n = 10{,}000$) and can become comparable to or slower than GAN–NCM on some large graphs.

We observe that the NCM-based baselines, MLE–NCM can be competitive in accuracy in some small-scale settings but is often computationally prohibitive, with very large training times. GAN–NCM generally trains faster than MLE–NCM and scales well with sample size, but its accuracy is less consistent for certain graph structures (e.g., chain-like graphs).

An important direction for future work is the evaluation of model-completion methods on real-world benchmarks, including problems with higher-dimensional variables, mixed discrete–continuous domains, and complex data-generating mechanisms.

References

BayesFusion LLC. *QGeNIe Modeler User Manual*. 2022.

I. Beinlich, H. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medical Care*, pages 247–256. Springer-Verlag, Berlin, 1989.

R. Bhattacharya, R. Nabi, and I. Shpitser. Semiparametric inference for causal effects in graphical models with hidden variables. *Journal of Machine Learning Research*, 23:1–76, 2022.

A. Bhattacharyya, S. Gayen, S. Kandasamy, V. Raval, and V. N. Variyam. Efficient interventional distribution learning in the PAC framework. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, pages 7531–7549, 2022.

Y. Chen and A. Darwiche. On the definition and computation of causal treewidth. In *Uncertainty in Artificial Intelligence (UAI)*, 2022.

A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

A. Darwiche. Causal inference using tractable circuits. *CoRR*, 2022.

R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1–2):41–85, 1999.

R. Dechter. *Reasoning with Probabilistic and Deterministic Graphical Models: Exact Algorithms*. Morgan & Claypool Publishers, 2013.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

N. Friedman. The Bayesian structural EM algorithm. In *Uncertainty in Artificial Intelligence (UAI)*, pages 129–138, 1998.

Y. Huang and M. Valtorta. On the completeness of an identifiability algorithm for semi-Markovian models. *Annals of Mathematics and Artificial Intelligence*, 54(4):363–408, 2008.

Y. Jung, J. Tian, and E. Bareinboim. Estimating causal effects using weighting-based estimators. In *AAAI 2020*, pages 10186–10193, 2020.

Y. Jung, J. Tian, and E. Bareinboim. Learning causal effects via weighted empirical risk minimization. In *Advances in Neural Information Processing Systems 33*, 2020.

Y. Jung, J. Tian, and E. Bareinboim. Estimating identifiable causal effects through double machine learning. In *AAAI 2021*, pages 12113–12122, 2021.

Y. Jung, J. Tian, and E. Bareinboim. Estimating identifiable causal effects on Markov equivalence classes through double machine learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

K. Kask, R. Dechter, J. Larrosa, and A. Dechter. Unifying tree decompositions for reasoning in graphical models. *Artificial Intelligence*, 166(1–2):165–193, 2005.

D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

A. V. Kozlov and J. P. Singh. Parallel implementations of probabilistic inference. *IEEE Computer*, pages 33–40, 1996.

K. Kristensen and I. Rasmussen. The use of a Bayesian network in the design of a decision support system for growing malting barley without pesticides. *Computers and Electronics in Agriculture*, 33:197–217, 2002.

S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, pages 191–201, 1995.

R. Mateescu, K. Kask, V. Gogate, and R. Dechter. Join-graph propagation algorithms. *Journal of Artificial Intelligence Research*, 37:279–328, 2010.

Microsoft Inc. win95: An expert system for troubleshooting. 1995.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1989.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

A. Raichev, A. Ihler, J. Tian, and R. Dechter. EM4CI: Causal inference using a model learning approach. 2024. URL: `https://github.com/anniemeow/EM4CI`. Accessed 2024-08-26.

A. Raichev, J. Tian, A. Ihler, and R. Dechter. Estimating causal effects from learned causal networks. *arXiv:2408.14101*, 2024.

I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *AAAI 2006*, page 1219, 2006.

I. Shpitser and J. Pearl. Identification of conditional interventional distributions. In *UAI 2006*, pages 442–449, 2006.

I. Shpitser, T. S. Richardson, and J. M. Robins. An efficient algorithm for computing interventional distributions in latent variable causal models. *CoRR*, abs/1202.3763, 2012.

J. Tian. *Studies in Causal Reasoning and Learning*. PhD Thesis, UCLA, 2002.

J. Tian and J. Pearl. A general identification condition for causal effects. In *AAAI 2002*, pages 567–573, 2002.

S. Tikka. Package `causaleffect`. CRAN, 2022. URL: `https://github.com/santikka/causaleffect/`.

K. Xia, K. Lee, Y. Bengio, and E. Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. In *NeurIPS 34*, pages 10823–10836, 2021.

K. Xia, Y. Pan, and E. Bareinboim. Neural causal models for counterfactual identification and estimation. *CoRR*, abs/2210.00035, 2022.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, 2000.

M. Zaffalon, A. Antonucci, and R. Cabañas. EM-based bounding of unidentifiable queries in str

Raichev, A., Tian, J., Ihler, A., & Dechter, R. (2024). *Estimating causal effects from learned causal networks*. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI 2024)* (pp. 2524–2531). IOS Press.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A Additional Result Visualization

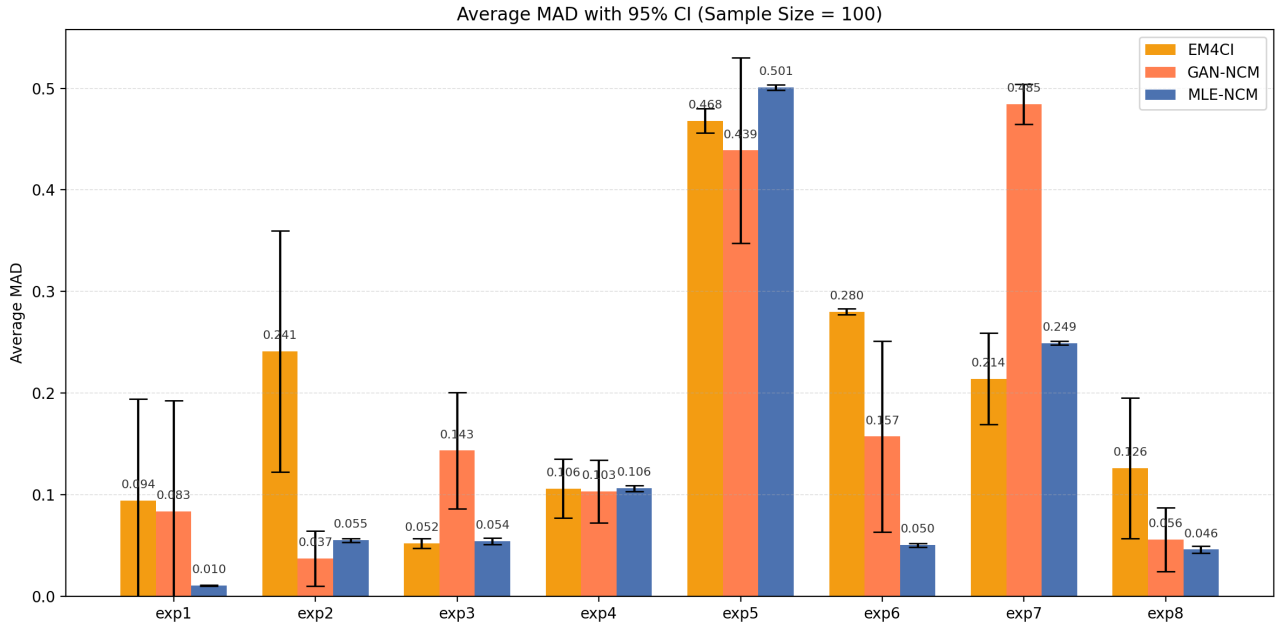


Figure 2: EM4CI achieves the lowest MAD on exp3 and exp7, while GAN-NCM performs best on exp2, exp4, and exp5, and MLE-NCM attains the lowest MAD on exp1, exp6, and exp8. Notably, MLE-NCM exhibits consistently smaller confidence intervals across all graphs.

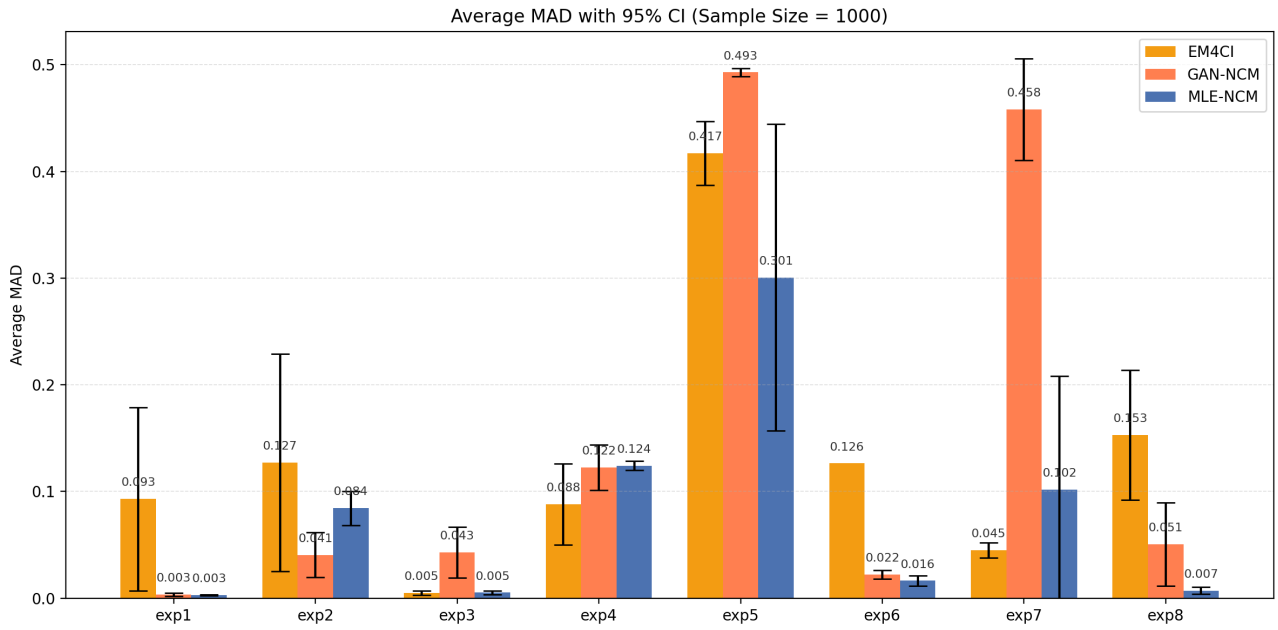


Figure 3: EM4CI achieves the lowest MAD on exp4 and exp7, while GAN-NCM performs best on exp2, and MLE-NCM attains the lowest MAD on exp5, exp6, and exp8. Ties are observed on exp1 (GAN-NCM and MLE-NCM) and exp3 (EM4CI and MLE-NCM).

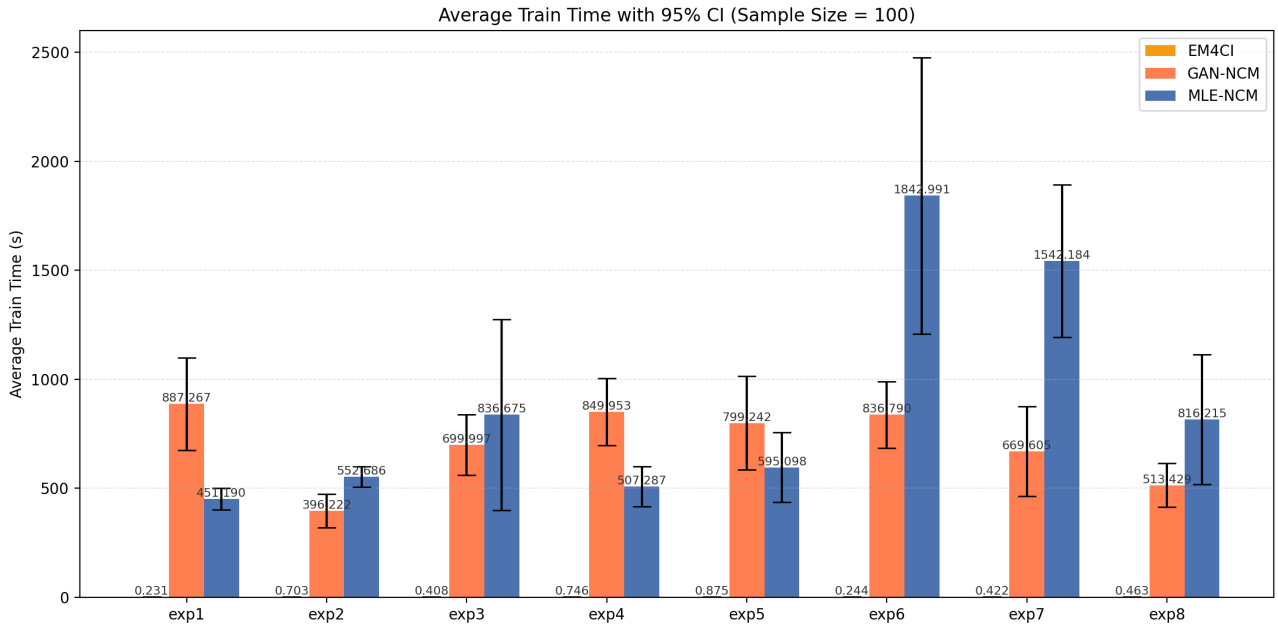


Figure 4: EM4CI completes all runs in under one second, whereas GAN–NCM and MLE-NCM requires 200–1200 seconds.

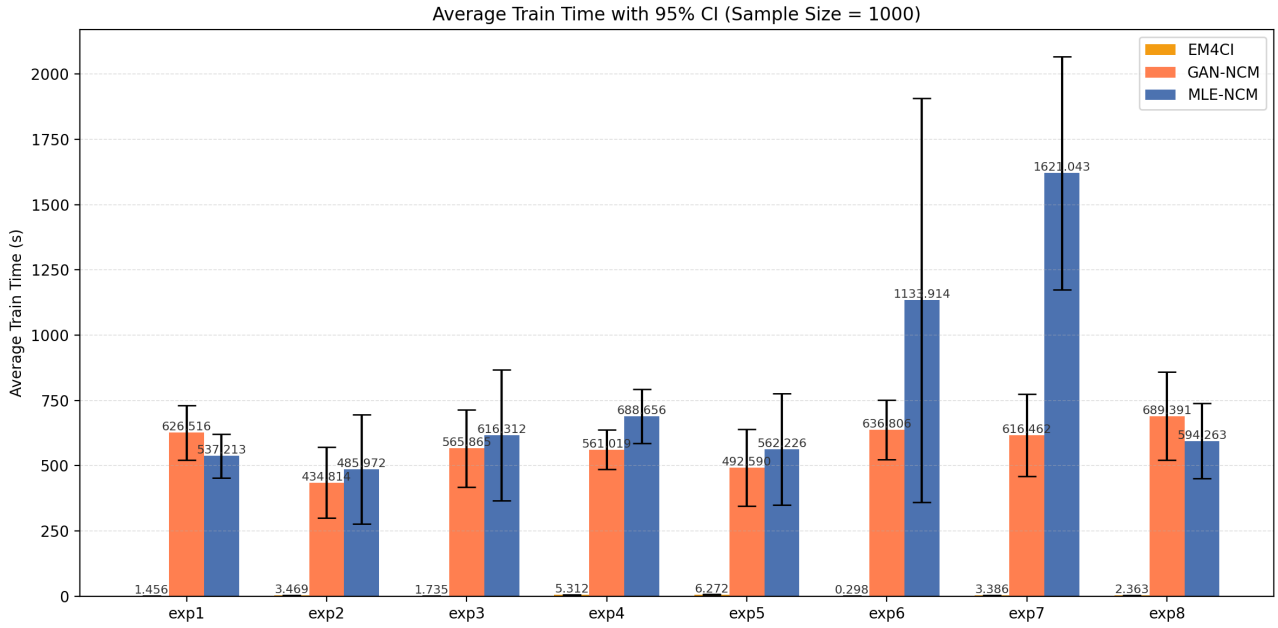


Figure 5: EM4CI remains efficient, finishing within 10 seconds, while GAN–NCM and MLE-NCM ranges from 300 to over 2700 seconds. MLE-NCM runtime shows occasional extreme outliers.

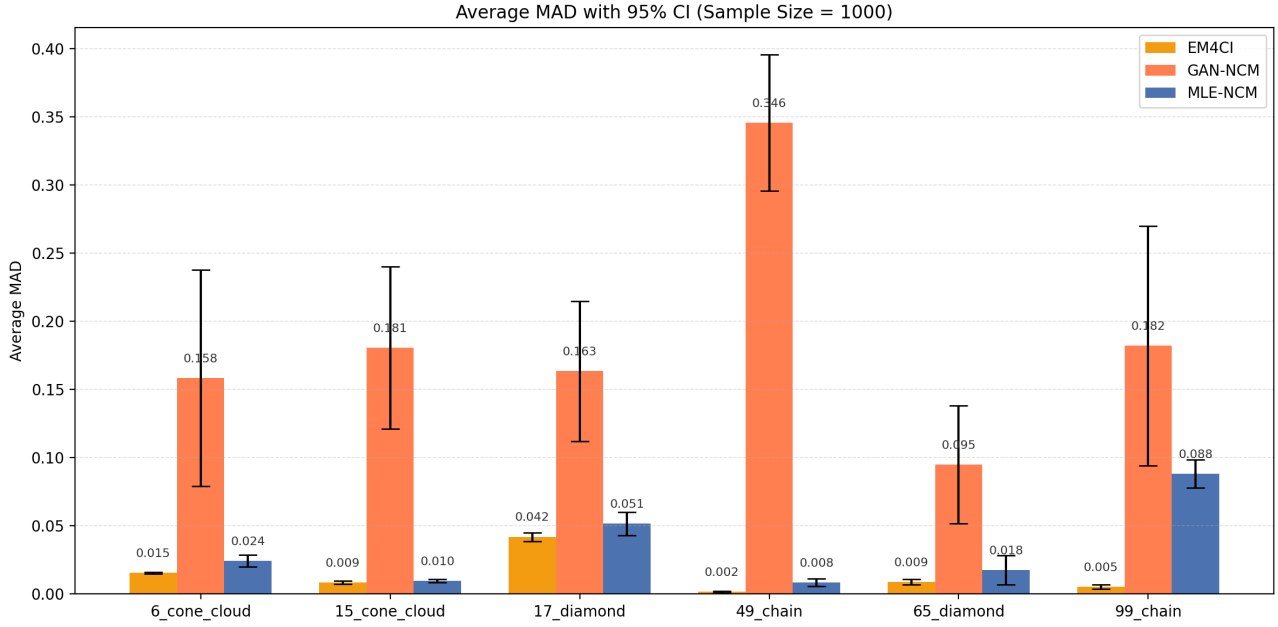


Figure 6: EM4CI achieves consistently low MAD across all graph families. In contrast, GAN–NCM exhibits substantially higher error and variance. MLE–NCM generally improves over GAN–NCM but remains less accurate than EM4CI on all six graphs.

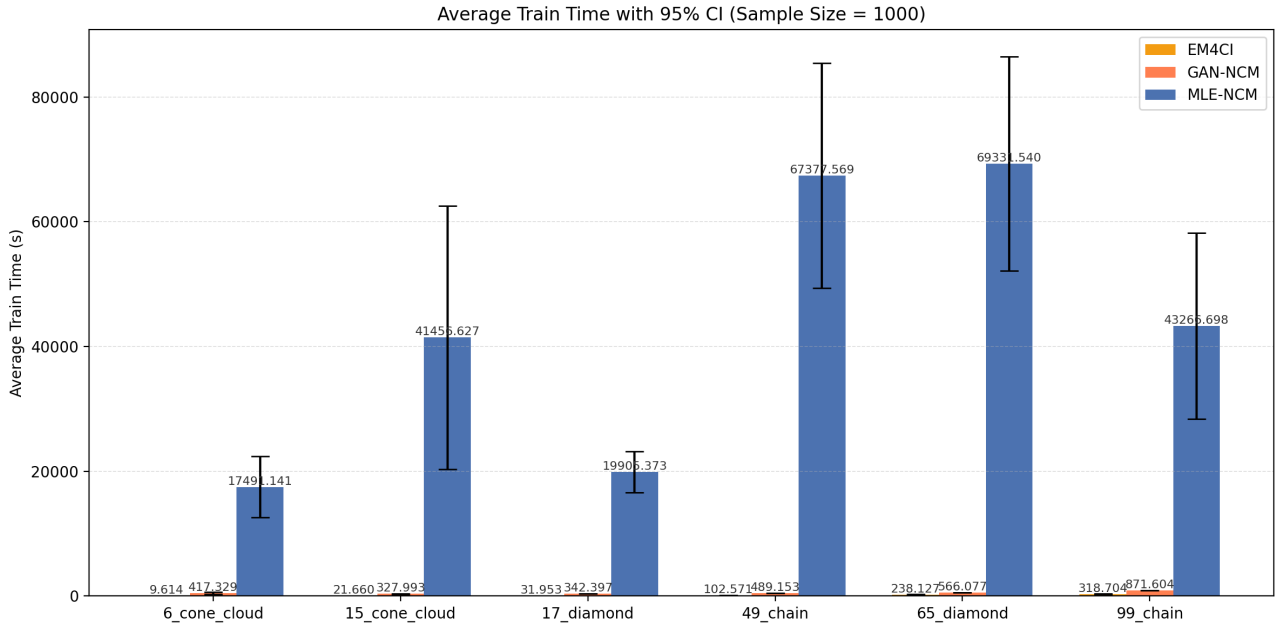


Figure 7: EM4CI is consistently the fastest method, with execution times ranging from approximately 10 to 320 seconds. GAN–NCM requires substantially more time on every graph, while MLE–NCM is the most computationally expensive, with training times reaching tens of thousands of seconds and exhibiting large variance.

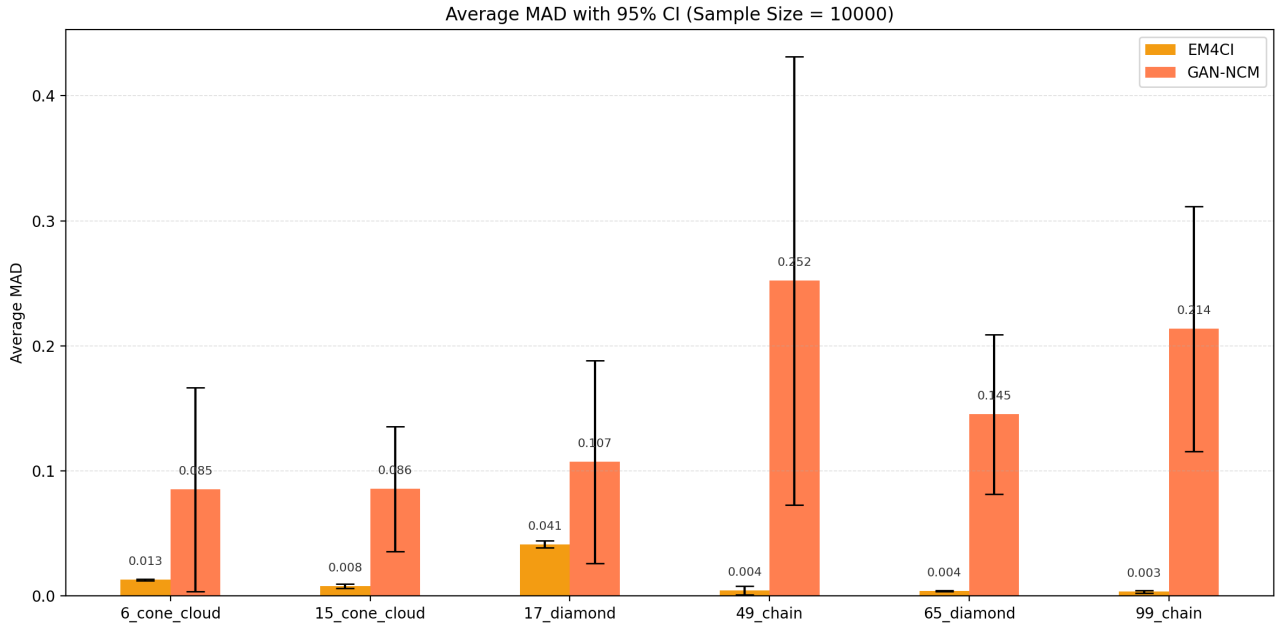


Figure 8: EM4CI consistently achieves the lowest MAD across all graph families, maintaining error below 0.05 even at larger sample sizes. In contrast, GAN–NCM exhibits substantially higher error and variance, particularly on chain-structured graphs, where MAD exceeds 0.20.

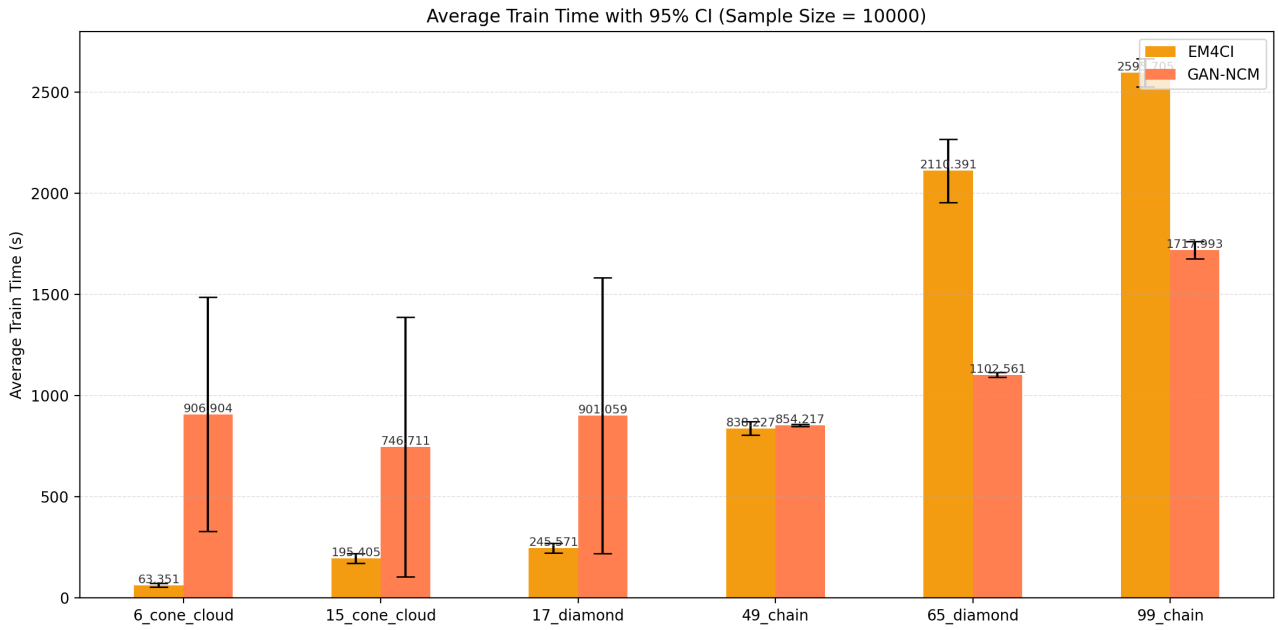


Figure 9: At this scale, EM4CI runtime increases substantially and becomes comparable to or slower than GAN–NCM on several large graphs, particularly the diamond and chain structures.

B Additional Implementation Details for NCM-Based Methods

To evaluate NCM-based approaches on the EM4CI benchmarks, we extended the original NCM implementation to ensure compatibility with the discrete, multi-valued causal models used in our experiments. Below we describe the main adaptations in greater detail. Comprehensive hyperparameter configurations, training settings, and step-by-step instructions for reproducing all experimental results are provided in the public GitHub repository: `https://github.com/JiapengZhao1/NCMs_Discrete.git`.

Support for multi-valued discrete variables The original NCM implementation assumes binary endogenous and exogenous variables. We extend the codebase to support discrete variables with arbitrary finite domain size k. In particular, latent variables are modeled using categorical distributions rather than Bernoulli distributions. Each latent variable takes values in $\{1, \ldots, k\}$ with associated probabilities $\theta_i = P(K = i)$ satisfying $\sum_{i=1}^{k} \theta_i = 1$.

One-hot representations and differentiable relaxation Discrete variables with domain size greater than two are represented using one-hot encodings to enable neural network training. To allow gradient-based optimization through discrete variables, we employ the Gumbel–Softmax relaxation, following the approach suggested in NCMs (Xia et al., 2021). This relaxation provides a differentiable approximation to categorical sampling, enabling backpropagation during training.

CPT-based data generation To ensure compatibility with multi-valued discrete variables, we generate observational data from ground-truth causal models specified by conditional probability tables (CPTs). The CPT parameters are taken from the EM4CI experimental setup (Raichev et al., 2024), ensuring that both EM4CI and NCM-based methods are evaluated on identical data-generating processes.

Evaluation metric We adopt the same metric used in EM4CI to evaluate NCM-based methods to enable a direct and consistent comparison. EM4CI reports performance using the mean absolute deviation (MAD) between the true interventional distribution and the estimated one. More details about the MAD are showed in the following section.

These modifications ensure that NCM-based methods operate under the same data-generating assumptions and

evaluation protocol as EM4CI, allowing for a controlled and fair comparison across methods.

C Graph Figures

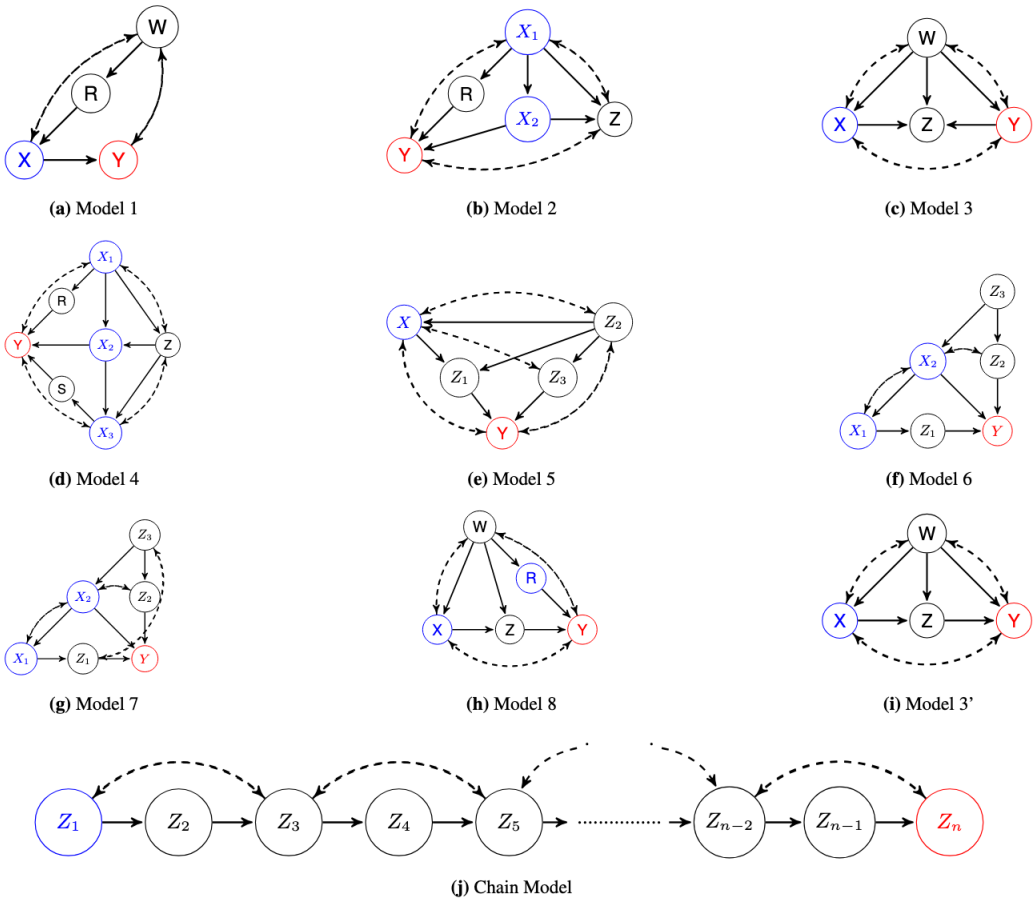


Figure 10: 8 small graphs: Causal diagrams for models used for experiments. Blue variables are intervened on and red variables are the outcome variables corresponding to the query $P(Y|do(X))$.