LEARNING CAUSAL VARIABLE ORDERINGS WITH POINTER NETWORKS FOR CAUSAL STRUCTURE DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal structure learning from observational data requires searching over a large combinatorial space of possible directed acyclic graphs (DAGs), making accurate and efficient structure learning challenging. Existing causal discovery methods based on continuous optimization and reinforcement learning either optimize graph structures directly or explore variable orderings through sequential decision making. We propose CaPo, a neural causal discovery framework that formulates causal variable ordering as a learnable permutation to guide constrained graph search. A Pointer Network generates candidate variable orderings, while an ordering-induced mask restricts admissible edge directions according to the learned ordering, thereby reducing the search space over candidate graph structures. The key novelty is to learn variable orderings directly from observational data and optimize them according to downstream causal graph quality, rather than relying on fixed or explicitly searched orderings. The ordering policy is optimized through reinforcement learning using a graph-level causal structure reward, and a DAG-constrained graph learner estimates the corresponding causal structure under the learned ordering. Experiments on Bayesian network benchmarks of varying sizes show that CaPo achieves competitive or lower Structural Hamming Distance (SHD) and higher F1-scores than state-of-the-art causal discovery baselines across multiple datasets. CaPo also requires less computational time than several compared methods, demonstrating efficient causal structure learning across networks of different sizes.

1 INTRODUCTION

Causal discovery aims to identify the underlying causal relationships among variables from observational data. In contrast to conventional predictive learning, which primarily captures statistical associations, causal discovery seeks to recover directed dependencies that can provide insights into the mechanisms underlying observed data. Reliable causal structures can support downstream tasks such as intervention analysis, scientific discovery, and decision-making, which has been demonstrated in domains including healthcare, biology, economics, and environmental science (Stein et al., 2025).

Despite its importance, causal structure learning remains a challenging combinatorial problem. Given d variables, the number of possible DAG structures increases super-exponentially with d, making exhaustive structure enumeration impractical. Consequently, existing approaches have investigated constraint-based search, score-based optimization, continuous formulations, and learning-based search strategies to avoid explicit enumeration of the entire graph space (Spirtes et al., 2000; Chickering, 2002; Chickering et al., 2004). Constraint-based methods infer graph structure from conditional independence relationships, but their performance can be sensitive to the accuracy and computational cost of conditional independence tests. Score-based methods formulate structure learning as an optimization problem over DAGs, but still face the combinatorial complexity of searching over a large space of candidate graph structures. Continuous formulations have further enabled gradient-based optimization of causal structures by replacing discrete graph search with differentiable objectives (Zheng et al., 2018; Lachapelle et al., 2020; Bello et al., 2022). However, these methods still optimize a high-dimensional continuous parameterization of the adjacency structure and subsequently derive a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

discrete graph from the optimized parameters, making the optimization increasingly costly as the number of variables grows.

Another alternative perspective is to consider the ordering of variables rather than searching directly over all possible graph structures. Every DAG admits at least one topological ordering, and a valid ordering constrains the directions of edges that can be considered between variables. Therefore, learning an informative ordering can provide a structured way of reducing the space of admissible causal structures. Previous research has demonstrated the potential of ordering-based causal discovery and has investigated score-based, heuristic, reinforcement-learning, and other search strategies for identifying useful permutations of variables (Teyssier & Koller, 2005; Raskutti & Uhler, 2018; Wang et al., 2021; Yang et al., 2023; Xu et al., 2024). These studies indicate that variable ordering can play an important role in the efficiency and quality of causal structure learning.

However, learning an effective variable ordering remains challenging. The space of possible permutations is factorial in the number of variables, and evaluating a candidate ordering may require an additional causal structure estimation step. As a result, repeatedly exploring candidate permutations is computationally expensive. Moreover, existing ordering strategies generally focus on search procedures, ordering representations, or task-specific optimization mechanisms, rather than treating the ordering itself as a reusable neural permutation-learning problem. This motivates a question: *can a permutation-specific neural architecture be used to learn variable orderings directly from observational data and use these orderings as structural constraints for subsequent causal graph estimation?*

In this work, we investigate a neural approach to causal node ordering in which variable permutations are learned from observational data and then used to guide causal structure estimation. The learned ordering is converted into an ordering-induced constraint on admissible edge directions, after which a causal structure learning procedure estimates the corresponding graph. A reinforcement-learning model is used to optimize the ordering according to the quality of the resulting causal structure. The main contributions of this work are:

1. **Learnable Neural Variable Ordering.** We formulate causal variable ordering as a learnable permutation problem and employ a Pointer Network to explicitly generate variable orderings directly from observational data. This provides a permutation-specific neural mechanism for learning causal variable orderings, distinguishing the proposed approach from methods that treat ordering primarily as a generic sequential decision process.
2. **Ordering-Guided Causal Structure Learning.** We introduce an ordering-guided causal discovery framework in which the learned permutation is transformed into an ordering-induced mask that restricts admissible edge directions before DAG estimation. This creates an explicit interface between neural permutation generation and constrained graph learning, allowing the learned ordering to directly shape the subsequent causal structure search.
3. **Graph-Quality-Driven Ordering Optimization.** We train the Pointer Network using the REINFORCE policy-gradient algorithm, where the quality of the causal graph obtained under a generated ordering serves as the graph-level reward. This directly connects permutation learning to downstream causal structure quality, so that the ordering policy is optimized toward permutations that yield higher-quality causal graph estimates.

2 Related Work

Causal structure learning has traditionally been approached through constraint-based, score-based, and hybrid formulations. Constraint-based methods, such as the PC algorithm and its extensions including FCI, infer causal structures by exploiting conditional independence relationships in observed data (Spirtes et al., 2000; 1995). Score-based methods, such as Greedy Equivalence Search (GES) and Hill-Climbing, instead search over DAG structures by optimizing statistical criteria such as the Bayesian Information Criterion (BIC) (Chickering, 2002; Chickering et al., 2004). Hybrid approaches, including Max–Min Hill-Climbing (MMHC), combine constraint-based and score-based strategies to improve search efficiency (Tsamardinos et al., 2006). However, these approaches ultimately rely on searching over a combinatorial space of candidate DAG structures, which becomes increasingly challenging as the number of variables grows.

Continuous optimization methods avoid explicit discrete graph search by representing causal structure through continuous parameters and optimizing differentiable objectives. NOTEARS introduced a smooth characterization of acyclicity that enables gradient-based DAG learning (Zheng et al., 2018). Subsequent methods extended this formulation to nonlinear and neural settings, including DAG-GNN and GraN-DAG (Yu et al., 2019; Lachapelle et al., 2020). Learning Sparse Nonparametric DAGs further extended continuous optimization to general nonlinear and semiparametric models (Zheng et al., 2020). GOLEM introduced a likelihood-based formulation for DAG learning under different noise assumptions, while DAGMA proposed a log-determinant characterization of acyclicity with improved numerical properties for continuous DAG optimization (Ng et al., 2020; Bello et al., 2022). These methods demonstrate that differentiable optimization can substantially simplify DAG structure learning by replacing discrete graph search with gradient-based optimization. However, because the graph structure is optimized directly, these formulations do not inherently exploit a learned global variable ordering to exclude directionally incompatible edges before graph estimation.

An alternative line of research uses variable orderings to impose structure on the causal search space. Classical ordering-based methods search over permutations using score-based and heuristic strategies (Teyssier & Koller, 2005; Raskutti & Uhler, 2018). Since every DAG admits at least one topological ordering, a valid ordering restricts the directions of admissible edges and can therefore provide a structured representation of the causal search space. The challenge is then shifted from direct DAG search to identifying an informative variable ordering, whose search space remains factorial in the number of variables. Recent methods have focused on making this ordering-based search more efficient. QWO develops an efficient procedure for permutation-based causal discovery in linear Gaussian models, reducing the computational cost of constructing the optimal DAG associated with a candidate permutation (Shahverdikondori et al., 2024). CaPS introduces an ordering-based framework for causal discovery with mixed linear and nonlinear relationships and separates ordering identification from subsequent parent selection and pruning (Xu et al., 2024). FLOP further investigates efficient discrete search over variable orders and parent sets and demonstrates competitive performance with continuous causal discovery methods in linear settings (Wienöbst et al., 2026). These approaches improve the efficiency of ordering-based causal discovery, but primarily rely on discrete search or ordering-based graph construction rather than learning a parameterized neural model specifically to generate causal variable permutations.

Reinforcement learning provides another approach for exploring discrete causal search spaces. Zhu et al. formulate causal discovery as a reinforcement learning problem in which an agent searches for high-scoring causal structures (Zhu et al., 2020). Direct graph construction through sequential edge-level decisions, however, leads to a large combinatorial action space. CORL addresses this challenge by reformulating causal discovery as an ordering search problem, where causal orders are generated through a multi-step decision process and an encoder–decoder policy is optimized using reward mechanisms defined for candidate orderings (Wang et al., 2021). However, the ordering search itself can be computationally expensive, and the learned ordering is subsequently passed to a separate variable-selection stage for constructing the final causal graph. This separates ordering optimization from the subsequent graph construction process. RCL-OG adopts another ordering-based reinforcement learning strategy by representing candidate ordering states with an order graph and learning transitions between ordering states using deep reinforcement learning (Yang et al., 2023). This formulation requires additional modeling of the ordering state-transition space and estimation of the distribution over candidate orderings. Moreover, the ordering is optimized within the ordering-search process rather than through direct feedback from the quality of the causal graph constructed under the resulting ordering. These approaches therefore introduce challenges related to computational cost, additional ordering-state modeling, and the separation between ordering optimization and downstream graph estimation.

Neural permutation models provide a complementary mechanism for learning variable orderings. Pointer Networks, for example, were originally introduced to generate permutations of input elements using attention-based pointer mechanisms (Vinyals et al., 2015). Such architectures provide a natural parameterized representation for permutation generation and can avoid explicit enumeration of the permutation space. However, their use as a dedicated neural mechanism for learning causal variable orderings, with the resulting ordering directly imposed as a structural constraint during downstream DAG estimation, remains less explored. CaPo builds on this perspective by using a Pointer Network as the ordering policy and optimizing the generated permutations according to downstream causal graph quality. The learned ordering is explicitly converted into a structural mask that restricts admissible

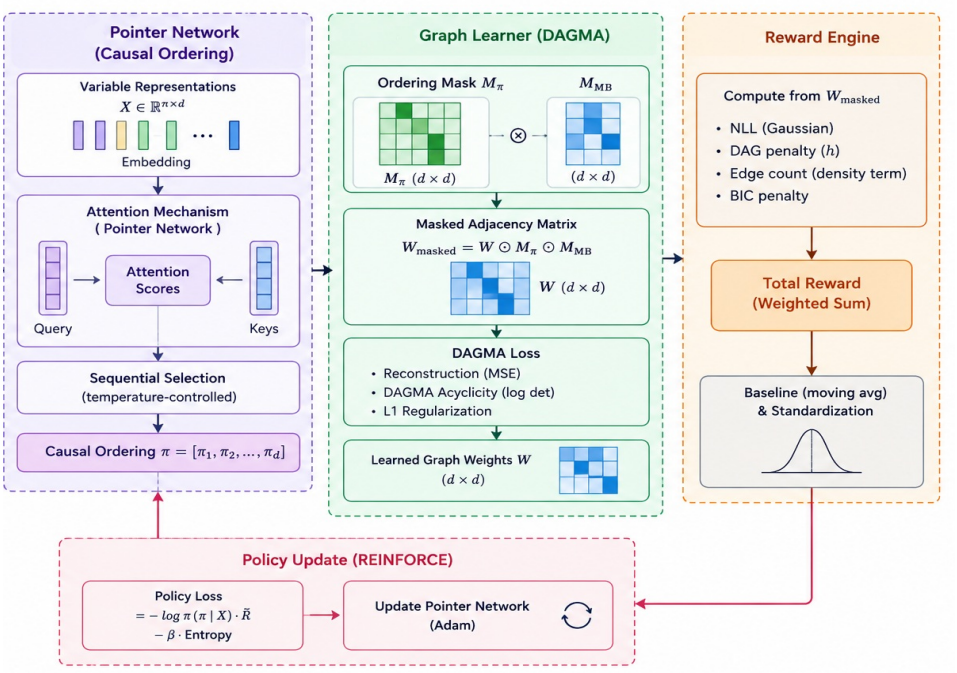


Figure 1: Overview of the proposed CaPo framework.

edge directions during graph estimation, thereby coupling permutation learning with subsequent causal structure learning rather than treating ordering identification and graph estimation as separate search procedures.

3 METHODOLOGY

The proposed framework, CaPo, formulates causal structure discovery as a learning framework in which a causal variable ordering is learned from observational data and subsequently used to constrain DAG structure learning. Given an observational dataset $X \in \mathbb{R}^{n \times d}$, CaPo learns a permutation $\pi = (\pi_1, \ldots, \pi_d)$ of the d variables using a Pointer Network. The learned permutation defines the relative order of the variables and induces a structural constraint on the subsequent graph learning stage: if variable i precedes variable j in π, the direction $i \to j$ is admissible, whereas the reverse direction $j \to i$ is excluded. Thus, the learned ordering restricts the set of candidate edge directions considered during DAG estimation.

As an initial preprocessing step, the variables are standardized to place them on comparable scales for the graph reconstruction objective. A correlation-based screening mask is then constructed to remove variable pairs with weak and statistically insignificant linear association. This mask serves only as a structural screening mechanism and is combined with the ordering constraint during graph learning.

The Pointer Network receives the variable representations and sequentially selects unvisited variables to generate a permutation π. At each step, attention scores are computed over the remaining variables, and temperature-controlled categorical sampling is used to maintain exploration during training. Once the Pointer Network generates a permutation, it is converted into an ordering mask M_π. The mask permits an edge $i \to j$ only when i precedes j in the generated ordering. The ordering mask is then combined with the correlation-based screening mask to define the admissible connections for the graph learner.

Under these constraints, the graph learner estimates a weighted adjacency matrix W. The graph objective consists of a reconstruction loss, a DAG acyclicity constraint, and an ℓ_1 sparsity term. The resulting graph objective is converted into a scalar graph-level reward that evaluates the quality of the

Algorithm 1 Training Procedure for CaPo: Reinforcement Learning-Based Causal Ordering

Require: Observational data $\mathbf{X}$, training epochs T
Ensure: Estimated causal graph G
1: Standardize $\mathbf{X}$ and compute a screening mask $\mathbf{M}$
2: Initialize Pointer Network $\mathcal{P}_\theta$ and graph learner W
3: **for** $t = 1, \ldots, T$ **do**
4: Generate a permutation $\pi \sim \mathcal{P}_\theta(\pi|\mathbf{X})$
5: Construct an ordering mask $\mathbf{O}_\pi$ allowing edges only from earlier to later variables
6: Optimize W using the ordering and screening constraints
7: Compute graph reward $R = -\mathcal{L}_{graph}(W)$
8: Update θ using REINFORCE
9: **end for**
10: Threshold the learned adjacency matrix W
11: **return** estimated causal graph G

causal structure obtained under the generated ordering. The overall interaction between permutation generation, structural constraints, graph estimation, and reward feedback is illustrated in Figure 1.

The Pointer Network is trained using the REINFORCE policy-gradient algorithm. For each sampled permutation, the graph learner estimates the corresponding causal structure and computes the graph-level reward based on reconstruction quality, acyclicity, and graph complexity. The normalized reward is used to update the parameters of the ordering policy, while an entropy term encourages exploration of alternative permutations. This feedback mechanism enables the ordering policy to favor permutations that lead to higher-quality causal graph estimates. The complete training and graph extraction procedure is summarized in Algorithm 1. After training, the learned edge weights are thresholded to obtain the final binary DAG.

4 RESULTS

We evaluate CaPo on Bayesian Network benchmarks from the Bayesian Network Repository(Scutari, 2022), covering a range of domains, graph sizes, and structural complexities. The evaluated networks are Asia, Csuite, Sachs, Child, Insurance, Alarm, Water, Barley, Hailfinder, Hepar, Pathfinder, Andes, Diabetes, Pigs, and Link. The corresponding network structures are used as ground truth for post-hoc causal structure evaluation. For all datasets, approximately 5,000 observational samples were used. Before being used in the experiments, the generated observational data were discretized into discrete variables. The same observational datasets were used across different random seeds, while the seeds affected only the model initialization and optimization process. Each experiment was repeated using six different random seeds, and the reported results are the mean performance across the six runs. Ground-truth structural metrics were computed exclusively for post-hoc evaluation and were not used during training or model selection.

CaPo is independently applied to each dataset using the causal ordering framework. The Pointer Network generates a candidate variable ordering, which is converted into an ordering-induced structural mask, after which the graph learner estimates the causal structure under DAG and sparsity constraints. The ground-truth graphs are used only for final evaluation. Table 1 reports the resulting Precision, Recall, F1-score, and SHD. CaPo achieves F1-scores of 0.7692, 0.8000, and 0.7040 on Asia, Csuite, and Sachs, respectively, with corresponding SHD values of 3, 3, and 8. On Child, CaPo obtains an F1-score of 0.5780 and an SHD of 20, while the framework is also evaluated on substantially larger networks containing up to 724 variables.

Tables 2–4 compare CaPo with state-of-the-art causal discovery methods using average SHD, where lower values indicate better structural recovery. The benchmark datasets cover small, medium, and large networks, enabling evaluation across a broad range of graph sizes and complexities. Overall, CaPo achieves the lowest or tied-lowest SHD on most of the evaluated benchmarks and maintains competitive performance across networks of substantially different sizes. Compared with conventional causal discovery methods, including GES, DirectLiNGAM, NOTEARS, PC, DAG-GNN, DAGMA, and GOLEM, CaPo generally achieves lower SHD, with the improvements becoming particularly evident on several medium- and large-scale networks. For example, on representative small and

Table 1: Causal discovery performance across Bayesian network benchmark datasets. Values are reported as mean ± standard deviation.

Dataset	Nodes	Precision	Recall	F1	SHD
Asia	8	1.000 ± 0.0000	0.6250 ± 0.0000	0.7692 ± 0.0000	3 ± 0
Csuite	9	0.7273 ± 0.1179	0.8889 ± 0.1257	0.8000 ± 0.1235	3 ± 1
Sachs	11	0.8960 ± 0.0092	0.5798 ± 0.0059	0.7040 ± 0.0072	8 ± 1
Child	20	0.5903 ± 0.0175	0.5671 ± 0.0290	0.5780 ± 0.0187	20 ± 0
Insurance	27	0.6562 ± 0.0395	0.4038 ± 0.0508	0.5000 ± 0.0441	42 ± 2
Water	32	0.7179 ± 0.0373	0.4142 ± 0.0166	0.5253 ± 0.0226	48 ± 1
Alarm	37	0.4595 ± 0.0372	0.4359 ± 0.0337	0.4474 ± 0.0341	42 ± 3
Barley	48	0.6250 ± 0.0198	0.1786 ± 0.0350	0.2778 ± 0.0251	78 ± 2
Hailfinder	56	0.6333 ± 0.0288	0.2879 ± 0.0309	0.3958 ± 0.0298	58 ± 1
Hepar	70	0.7377 ± 0.0157	0.3659 ± 0.0000	0.4892 ± 0.0018	94 ± 1
Pathfinder	109	0.7085 ± 0.0769	0.0482 ± 0.0194	0.0891 ± 0.0317	190 ± 1
Andes	223	0.4082 ± 0.0281	0.3872 ± 0.0179	0.3973 ± 0.0224	280 ± 4
Diabetes	413	0.3728 ± 0.0098	0.0761 ± 0.0027	0.1264 ± 0.0042	602 ± 5
Pigs	441	0.3114 ± 0.0186	0.5533 ± 0.0206	0.3983 ± 0.0203	168 ± 0
Link	724	0.3387 ± 0.0071	0.1049 ± 0.0075	0.1601 ± 0.0095	1100 ± 2

Table 2: SHD comparison across small-size benchmark datasets. Lower is better.

Dataset	GES	DL	NT	PC	CORL	DAG-GNN	DAGMA	GOLEM	RCL-OG	QWO	CaPS	FLOP	**CaPo**
Asia	7	7	7	9	6	8	7	9	6	8	8	9	**3**
Csuite	5	7	5	8	4	6	3	6	4	9	9	12	**3**
Sachs	10	14	10	17	12	14	14	14	17	10	10	17	**8**
Child	62	34	43	51	36	31	35	35	29	30	25	58	**20**
Insurance	82	49	59	72	45	44	46	48	47	59	51	61	**42**
Mean	33.2	22.2	24.8	31.4	20.6	20.6	21.0	22.4	20.6	23.2	20.6	31.4	**15.2**

medium networks, CaPo achieves SHD values of 3, 8, and 20, compared with the corresponding values of 7, 10, and 43 for NOTEARS, 9, 14, and 34 for DirectLiNGAM, and 7, 14, and 35 for DAGMA. CaPo also compares favorably with recent ordering-based approaches, including RCL-OG, QWO, CaPS, and FLOP, achieving lower SHD on most of the reported benchmarks; for example, on the same representative networks, CaPo obtains 3, 8, and 20 compared with 6, 17, and 29 for RCL-OG, 8, 10, and 30 for QWO, 8, 10, and 25 for CaPS, and 9, 17, and 58 for FLOP, respectively. On larger networks, CaPo continues to obtain competitive or lower SHD than both conventional and ordering-based methods, indicating that the learned global variable ordering remains effective as network size increases. Overall, these results support the effectiveness of learning a global variable ordering and incorporating it as a structural constraint for causal discovery across networks with different sizes and complexities.

Finally, computational efficiency is evaluated using the execution-time comparison in Figure 2. On the benchmarks with directly comparable results, CaPo generally requires less execution time than the compared reinforcement learning methods. For example, on Sachs, CaPo requires approximately 368.5 seconds, compared with 1228 seconds for CORL and 1583.73 seconds for RCL-OG. On Hailfinder, CaPo requires approximately 558 seconds, compared with 2802.69 seconds for RCL-OG and 467 seconds for DAGMA-MLP. The runtime difference becomes particularly pronounced on larger networks. For example, on Pathfinder and Andes, CaPo requires approximately 945 and 2029 seconds, respectively, compared with 5897.99 and 19840.08 seconds for RCL-OG. Overall, Tables 1–4 and Figure 2 show that CaPo achieves competitive causal structure recovery together with substantially lower execution time than several compared approaches on the benchmarks with available runtime measurements.

Table 3: SHD comparison across medium-size benchmark datasets. Lower is better.

Dataset	GES	DL	NT	PC	CORL	DAG-GNN	DAGMA	GOLEM	RCL-OG	QWO	CaPS	FLOP	**CaPo**
Water	48	51	67	60	53	55	48	52	67	67	66	75	**48**
Alarm	106	44	50	83	48	45	46	45	56	70	47	74	**42**
Barley	95	90	95	100	90	92	88	101	86	127	84	115	**78**
Hailfinder	75	61	59	75	70	68	67	90	67	91	66	161	**58**
Hepar	120	114	112	125	118	116	115	119	125	121	123	130	**94**
Mean	88.8	72.0	76.6	88.6	75.8	75.2	72.8	81.4	80.2	95.2	77.2	111.0	**64.0**

Table 4: SHD comparison across larger benchmark datasets. Lower is better.

Dataset	GES	DL	NT	PC	CORL	DAG-GNN	DAGMA	GOLEM	RCL-OG	QWO	CaPS	FLOP	**CaPo**
Pathfinder	230	199	270	280	220	230	197	243	199	225	195	594	**190**
Andes	310	297	297	320	305	382	315	320	320	336	338	450	**280**
Pigs	300	266	173	290	240	250	230	240	260	310	280	400	**168**
Diabetes	760	748	686	800	720	730	710	740	605	700	620	900	**602**
Link	1250	1208	1150	1300	1200	1220	1180	1210	1126	1250	1125	1500	**1100**
Mean	570.0	543.6	515.2	598.0	537.0	562.4	526.4	550.6	502.0	564.2	511.6	768.8	**468.0**

5 Discussion

5.1 Convergence and Ordering Baselines

To evaluate the optimization dynamics and training stability of the proposed approach, Figure 4 presents the post-hoc F1-score trajectories across four representative benchmark spanning different network sizes: Asia (8 variables), Sachs (11 variables), Child (20 variables), and Hepar (70 variables). The proposed ordering mechanism (**CaPo**) is compared with three alternative ordering strategies: Random Ordering, Transformer Ordering, and Metaheuristic Ordering. The F1-score was computed only for post-hoc evaluation and was not used for reward computation, model updates, or any training or model-selection decisions. Across the evaluated datasets, the alternative ordering strategies exhibit greater variability and more pronounced oscillations, whereas CaPo generally maintains a more stable trajectory toward improved structural recovery.

In contrast, **CaPo** demonstrates stable convergence behavior. As training progresses, the proposed method achieves higher peak F1-scores while approaching a stable steady state without severe performance degradation. This separation indicates that integrating the Pointer Network with graph-level reinforcement learning can provide a more stable ordering search trajectory.

5.2 Reward Formulation Comparison

We evaluated five alternative reward formulations to identify an effective optimization signal for the neural ordering policy. The Bayesian Information Criterion (BIC) is a model-selection criterion that balances data fit against model complexity, thereby favoring models that explain the data well without unnecessary structural complexity. We also consider a Soft-BIC variant that provides a continuous formulation of the BIC-based objective for optimization. The primary **Graph-Quality Reward** jointly accounts for data fit, graph sparsity, and acyclicity and is defined as

$$R = -\lambda_{\text{fit}} L_{\text{fit}} - \lambda_{\text{sparse}} \|W\|_1 - \lambda_{\text{DAG}} h(W), \tag{1}$$

where L_{fit} denotes the graph reconstruction loss, $\|W\|_1$ promotes sparsity, and $h(W)$ is the acyclicity constraint. As shown in Figure 5, the **Graph-Quality Reward** achieves the lowest SHD across the evaluated dataset groups. On the large-scale Andes dataset, it obtains an SHD of **280**, compared with 289 for **BIC Only**, 287 for **BIC + DAG Penalty**, 283 for **BIC + Sparsity**, and 291 for **Soft-BIC + DAG**. These results suggest that jointly incorporating data fit, sparsity, and acyclicity provides a useful optimization signal for learning neural variable orderings.

To further investigate whether optimization of the selected reward translates into improved causal discovery performance during training, Figure 6 tracks the reward trajectory together with the ground-truth F1-score. Although the ground-truth graph is not available to the reinforcement learning agent during optimization, the reward and F1-score exhibit a consistent positive trend throughout training. This alignment provides empirical evidence that the **Graph-Quality Reward** can serve as a surrogate

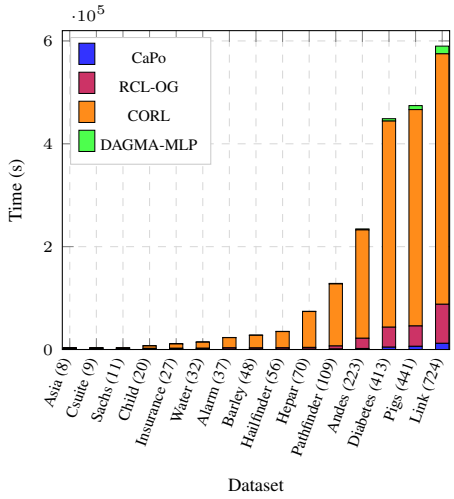


Figure 2: Stacked execution time of CaPo, RCL-OG, CORL, and DAGMA-MLP across benchmark datasets, ordered by the number of nodes. Each bar represents one dataset, with different colors indicating the execution time of the four methods.

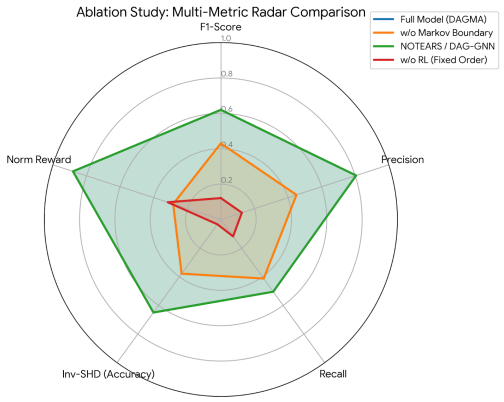


Figure 3: Multi-metric radar chart illustrating the ablation study results. All metrics are normalized such that a larger polygon coverage area indicates greater overall performance across the evaluated metrics.

for causal structure quality and can guide the neural permutation generator toward more accurate causal orderings.

5.3 Component Contributions

As illustrated in Figure 3, our ablation study evaluates the contributions of RL-based topological ordering search, Markov Boundary pre-filtering, and the continuous acyclicity regularizer. Removing the RL module (`w/o RL`) reduces the F_1-score from 0.6200 to 0.1212 and increases the SHD from 10.51 to 29.00. Similarly, omitting the Markov Boundary filter results in a Precision value of 0.4504.

Substituting the DAGMA acyclicity penalty with alternative regularizers such as `NOTEARS` and `DAG-GNN` yields the same reported F_1 value of 0.6200 as the **Full Model**. These results suggest that the ordering architecture can operate with different continuous acyclicity regularizers, while the RL ordering mechanism and structural screening play important roles in the evaluated ablations.

6 Conclusion

We presented CaPo, a neural causal discovery framework that learns global variable orderings using a Pointer Network and incorporates the learned ordering as a structural constraint for DAG learning. By combining the ordering-induced constraint with data-driven screening, CaPo reduces the space of admissible edge configurations while enabling the subsequent graph learner to estimate the causal structure. The graph quality is further used as a learning signal to optimize the ordering policy through REINFORCE, coupling variable ordering with downstream causal structure recovery.

Experiments on Bayesian network benchmarks with different numbers of variables demonstrate that CaPo achieves competitive structural recovery while providing favorable computational efficiency compared with several existing approaches. These results indicate that learned variable orderings can serve as an effective structural constraint for causal discovery.

Future work will investigate more informative and stable reward estimation for ordering learning, including reward formulations that better capture structural recovery quality and provide less noisy feedback to the Pointer Network. Further directions include improving scalability to larger causal graphs, exploring alternative ordering objectives, and evaluating the framework on more complex and real-world causal discovery settings.

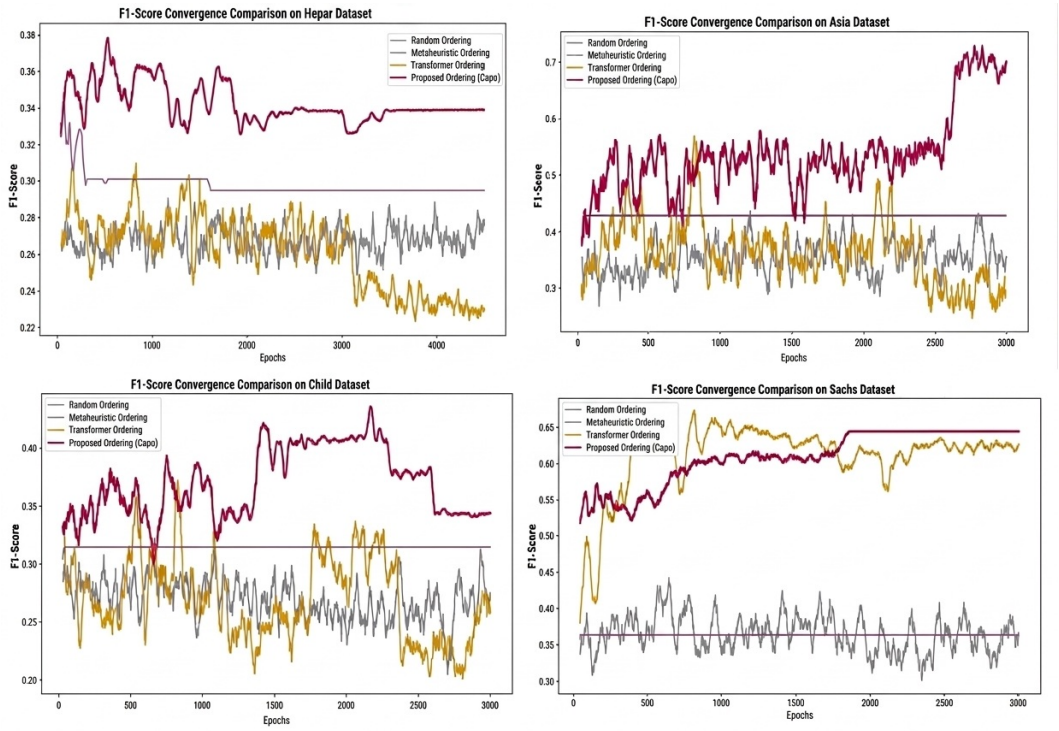


Figure 4: F1-score convergence comparison across various benchmark datasets (Hepar, Asia, Child, and Sachs). The proposed ordering method (**CaPo**) exhibits stable convergence, whereas baseline strategies (Random, Transformer, and Metaheuristic orderings) show greater fluctuations during training.

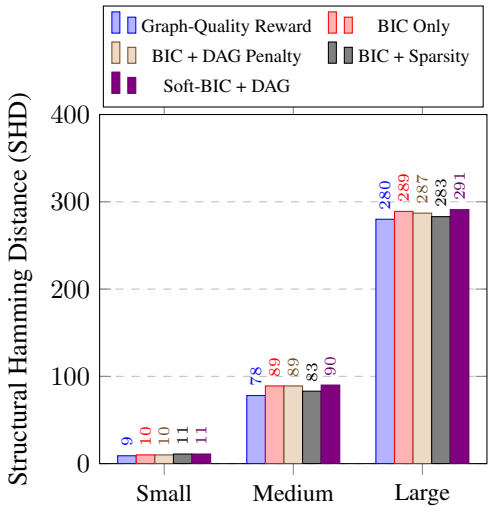


Figure 5: SHD comparison of five reward functions across small-, medium-, and large-size datasets. Lower SHD indicates better causal structure recovery.

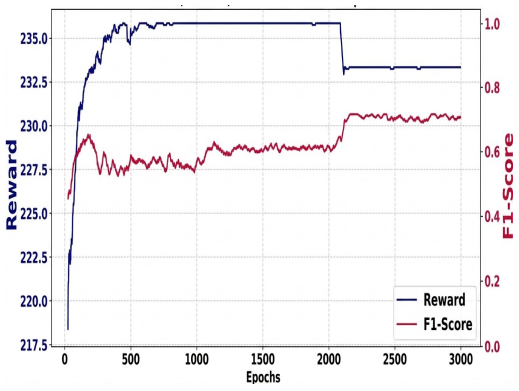


Figure 6: Trajectory of the continuous graph reward alongside the ground-truth F1-score across training epochs. The alignment between reward optimization and F1-score improvement illustrates the relationship between the unsupervised graph-quality reward and causal structure recovery.

AI USE STATEMENT

This work used artificial intelligence tools only for assistance with language editing, organization, and presentation. The authors remain responsible for the scientific content, methodology, experiments, analysis, and conclusions.

Reproducibility Statement

The paper provides descriptions of the proposed neural ordering framework, reinforcement learning procedure, graph-learning objective, experimental datasets, evaluation metrics, and ablation studies. The implementation and experimental code of the proposed framework, CaPo, are publicly available at https://github.com/CausalPointerNetwork/CaPo.

References

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=8rZYMpFUgK.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002. URL https://jmlr.org/papers/v3/chickering02b.html.

David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004. URL https://jmlr.org/papers/v5/chickering04a.html.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Alexandre Lacoste. Gradient-based neural dag learning. In *International Conference on Learning Representations*, 2020. URL https://mlanthology.org/iclr/2020/lachapelle2020iclr-gradientbased/.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17943–17954, 2020. URL https://dl.acm.org/doi/10.5555/3495724.3497230.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graph models based on sparsest permutations. *Statistical Science*, 33(4):498–514, 2018. doi: 10.1002/sta4.183. URL https://doi.org/10.1002/sta4.183.

Marco Scutari. Bayesian network repository, 2022. URL https://www.bnlearn.com/bnrepository/.

Mohammad Shahverdikondori, Ehsan Mokhtarian, and Negar Kiyavash. Qwo: Speeding up permutation-based causal discovery in ligams. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-1778. URL https://doi.org/10.52202/079017-1778.

Peter Spirtes, Christopher Meek, and Thomas S. Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 499–506. Morgan Kaufmann, 1995. URL https://dl.acm.org/doi/10.5555/2074158.2074215.

Peter Spirtes, Clark N. Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2000. doi: 10.7551/mitpress/1754.001.0001. URL https://doi.org/10.7551/mitpress/1754.001.0001.

Gideon Stein, Maha Shadaydeh, Jan Blunk, Niklas Penzel, and Joachim Denzler. Causalrivers: Scaling up benchmarking of causal discovery for real-world time-series. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wmV4cIbgl6.

Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, 2005. URL https://mlanthology.org/uai/2005/teyssier2005uai-ordering/.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006. doi: 10.1007/s10994-006-6889-7. URL https://doi.org/10.1007/s10994-006-6889-7.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2692–2700, 2015. URL https://research.google/pubs/pointer-networks/.

Xiaoqiang Wang, Yali Du, Shengyu Zhu, Liangjun Ke, Zhitang Chen, Jianye Hao, and Jun Wang. Ordering-based causal discovery with reinforcement learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 3566–3573, 2021. doi: 10.24963/ijcai.2021/491. URL https://doi.org/10.24963/ijcai.2021/491.

Marcel Wienöbst, Leonard Henckel, and Sebastian Weichwald. Embracing discrete search: A reasonable approach to causal structure learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/59db52d2abe2c60ec722e50e2852baad-Abstract-Conference.html.

Zhuopeng Xu, Yujie Li, Cheng Liu, and Ning Gui. Ordering-based causal discovery for linear and nonlinear relations. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-0141. URL https://doi.org/10.52202/079017-0141.

Dezhi Yang, Guoxian Yu, Jun Wang, Zhengtian Wu, and Maozu Guo. Reinforcement causal structure learning on order graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10737–10744, 2023. doi: 10.1609/aaai.v37i9.26274. URL https://ojs.aaai.org/index.php/AAAI/article/view/26274.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/yu19a.html.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e347c51419ffb23ca3fd5050202f9c3d-Abstract.html.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Learning sparse nonparametric dags. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 3414–3425. PMLR, 2020. URL https://proceedings.mlr.press/v108/zheng20a.html.

Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1g2skStPB.

A Theoretical Analysis

A.1 Accuracy Analysis

The main structural restriction in CaPo is induced by the learned variable ordering. Let π denote a permutation of the d variables and let M_π denote the corresponding ordering mask. An edge $i \to j$ is admissible only when

$$\pi(i) < \pi(j).$$

Let $\mathcal{G}_\pi$ denote the set of DAGs whose directed edges are consistent with this ordering constraint.

Consider the ideal case in which the learned ordering π is a valid topological ordering of the true causal DAG G^*. By definition of a topological ordering, every true causal edge $i \to j$ satisfies

$$\pi(i) < \pi(j).$$

Consequently,

$$G^* \in \mathcal{G}_\pi.$$

Therefore, when the learned ordering is a valid topological ordering of the true DAG, the ordering constraint does not exclude the true causal structure from the feasible solution space. At the same time, for every pair of distinct variables (i, j), at most one of the two possible directions $i \to j$ and

$j \to i$ is admissible. Thus, the ordering constraint eliminates directionally inconsistent candidates before the graph-learning stage.

This property provides a structural inductive bias for graph estimation. In particular, the ordering preserves the true DAG when the learned ordering is topologically valid, while restricting the graph learner to structures that are consistent with the generated ordering. The additional screening mask can further remove candidate connections that do not satisfy the data-driven screening criterion. Hence, the combined constraints reduce the effective space of candidate causal structures while retaining the true structure under the ideal ordering condition.

The effect of these constraints also depends on how strictly the ordering and other structural restrictions are imposed. More restrictive constraints can reduce the number of candidate structures considered by the graph learner, but overly restrictive constraints may also exclude true causal edges when the learned ordering or screening criterion is incorrect.

Moreover, the global ordering can influence the optimization of the downstream graph learner. Without an ordering constraint, the graph learner may search over a larger space of possible graph structures, which can contain undesirable local solutions. By restricting the search to structures consistent with a global ordering, CaPo may provide a more structured search space and reduce the likelihood of converging to such undesirable solutions. This effect is not guaranteed, however, because an incorrect ordering can instead restrict the feasible space in an unfavorable way.

The resulting reduction in the candidate structure space can therefore influence structural recovery when the learned ordering is sufficiently informative. However, the ordering constraint alone does not guarantee improvements in evaluation metrics such as SHD or F1-score, since these metrics also depend on the quality of the learned ordering, the screening mask, and the downstream graph estimation procedure. The empirical effect is therefore evaluated in Table 1.

A.2 Computational Efficiency Analysis

The computational design of CaPo has three main components: (i) compact permutation generation, (ii) ordering-based restriction of the candidate graph space, and (iii) complete-order evaluation in which the downstream graph learner is applied after a complete permutation has been generated.

Compact permutation generation. Let d denote the number of variables and h the embedding dimension. CaPo employs a compact Pointer Network specifically designed for permutation generation. The implemented ordering module consists of an input embedding layer followed by query, key, and scalar pointer-score transformations. Unlike a full Transformer or recurrent sequence model, the ordering module contains no multi-head self-attention blocks, recurrent layers, or stacked sequence-processing blocks.

At each step t, the Pointer Network selects one variable from the remaining $d - t + 1$ variables. Therefore, generating one complete permutation requires exactly

$$\sum_{t=1}^{d} 1 = d$$

variable-selection steps. At step t, a pointer score is computed for each unselected variable. Consequently, the total number of pointer scores evaluated for one complete permutation is

$$\sum_{t=1}^{d} (d - t + 1) = \frac{d(d+1)}{2} = \mathcal{O}(d^2).$$

For the implemented architecture, let n denote the number of observations. The input embedding maps each variable representation from dimension n to dimension h. Together with the query, key, and scalar pointer transformations, the number of trainable parameters in the ordering module is

$$\begin{aligned} P_{\mathrm{PN}} &= (nh + h) + (h^2 + h) + (h^2 + h) + (h + 1) \\ &= nh + 2h^2 + 4h + 1. \end{aligned}$$

Thus, the ordering module uses a compact set of linear transformations rather than the multi-head attention, feed-forward blocks, or recurrent gating mechanisms of larger sequence architectures. This characterizes the lightweight architectural design of the CaPo ordering module.

Reduction in candidate edge directions. Without an ordering constraint, every unordered pair of variables $\{i, j\}$ can potentially give rise to two directed edges, $i \to j$ and $j \to i$. Since there are

$$\binom{d}{2} = \frac{d(d-1)}{2}$$

unordered variable pairs, the maximum number of directed candidate edges is

$$2\binom{d}{2} = d(d-1) = \mathcal{O}(d^2).$$

For a fixed ordering π, exactly one direction is compatible with the ordering for each pair:

$$i \to j \quad \text{if} \quad \pi(i) < \pi(j).$$

Hence, the ordering mask reduces the maximum number of admissible directed candidate edges to

$$\binom{d}{2} = \frac{d(d-1)}{2}.$$

Therefore, the maximum number of directionally admissible candidate edges is reduced by a factor of two:

$$\frac{d(d-1)/2}{d(d-1)} = \frac{1}{2}.$$

This reduction does not change the asymptotic order from $\mathcal{O}(d^2)$ to a lower polynomial order. Rather, it removes one of the two possible directions for every variable pair before graph estimation, thereby reducing the effective directed candidate space by a constant factor.

Additional reduction from the screening mask. Let M_{screen} denote the data-driven screening mask and define the number of surviving candidate edges under ordering π as

$$m(\pi) = \sum_{i \neq j} \mathbb{I}\left[M_{\text{screen}}(i, j) = 1 \ \wedge \ \pi(i) < \pi(j)\right].$$

Since the ordering allows at most one direction per variable pair,

$$m(\pi) \leq \frac{d(d-1)}{2}.$$

Consequently, the number of edge configurations that remain admissible for graph estimation is reduced from the unconstrained $d(d-1)$ directed possibilities to at most $m(\pi)$ candidates. This represents a reduction in the effective candidate-edge space; the corresponding wall-clock reduction depends on how the screening mask is implemented in the graph-learning procedure.

If the screening mask retains a fraction ρ of the unordered variable pairs, then

$$m(\pi) \approx \rho \frac{d(d-1)}{2}, \qquad 0 \leq \rho \leq 1.$$

Relative to the unconstrained directed candidate space, the corresponding fraction of surviving candidates is approximately

$$\frac{m(\pi)}{d(d-1)} \approx \frac{\rho}{2}.$$

Thus, when $\rho < 1$, the combination of ordering and screening further reduces the effective candidate-edge space beyond the reduction induced by ordering alone.

Complete-order evaluation. CaPo generates the complete variable permutation before invoking the downstream graph-learning stage. Thus, if one graph-learning evaluation is performed for each sampled complete permutation, the number of downstream graph evaluations is

$$N_{\text{CaPo}} = 1.$$

For a generic sequential ordering procedure, suppose that the downstream graph is evaluated after k intermediate ordering states, where

$$1 \leq k \leq d.$$

Then,

$$N_{\text{seq}} = k, \qquad N_{\text{CaPo}} = 1.$$

Consequently, the ratio of downstream graph evaluations is

$$\frac{N_{\text{CaPo}}}{N_{\text{seq}}} = \frac{1}{k}.$$

In the special case where a graph evaluation is performed after every one of the d ordering decisions, this becomes

$$\frac{N_{\text{CaPo}}}{N_{\text{seq}}} = \frac{1}{d}.$$

This is a conditional result based on the evaluation strategy and does not constitute a universal runtime bound for sequential ordering-based methods.

Structure-aware ordering optimization. Let $G(\pi)$ denote the graph estimated under ordering π and let $L_{\text{structure}}(G(\pi))$ denote the structure-learning loss associated with the graph obtained under the sampled ordering. The ordering policy is optimized according to the graph-level objective

$$\max_{\theta} \mathbb{E}_{\pi \sim p_\theta(\pi|X)} \left[R(\pi) \right], \qquad R(\pi) = -L_{\text{structure}}(G(\pi)).$$

Thus, the ordering policy receives feedback from the quality of the downstream graph rather than from an independently defined ordering objective.

During training, entropy regularization encourages exploration of alternative permutations:

$$J(\theta) = \mathbb{E}_{\pi \sim p_\theta(\pi|X)}[R(\pi)] + \beta \mathcal{H}(p_\theta),$$

where $\beta \geq 0$ controls the strength of the entropy term. This regularization encourages the policy to maintain probability mass over alternative orderings during training. It does not provide a guarantee of global optimality or elimination of local optima.

Overall computational implication. The above analysis identifies three structural sources of computational efficiency in CaPo. First, the ordering policy generates a complete permutation using d sequential selection decisions and a compact pointer-based architecture. Second, a fixed ordering eliminates one of the two possible directions for every variable pair, reducing the maximum number of admissible directed candidates from $d(d-1)$ to $d(d-1)/2$. Third, the screening mask can further reduce the effective candidate-edge space to $m(\pi)$.

Moreover, because CaPo evaluates the downstream graph after generating a complete permutation, one sampled ordering requires one graph-learning evaluation under the stated evaluation model. These results establish reductions in the candidate search space and, conditionally, in the number of downstream graph evaluations. They do not by themselves imply a universal wall-clock complexity bound, since the latter depends on the implementation of the graph learner and the compared methods. Therefore, the practical runtime of CaPo is evaluated empirically in Figure 2.