A Meta-Learning Approach to Causal Inference

Cristian Dragos Manta [1 2] **Philippe Brouillard** [1 2] **Dhanya Sridhar** [1 2]

Abstract

Predicting the effect of unseen interventions is at the heart of many scientific endeavours. While causal discovery is often used to answer these causal questions, it involves learning a full causal model, not tailored to the specific goal of predicting unseen interventions, and operates under stringent assumptions. We introduce a novel method based on meta-learning that predicts interventional effects without explicitly assuming a causal model. Our preliminary results on synthetic data show that it can provide good generalization to unseen interventions, and it even compares favorably to a causal discovery method. Our model-agnostic method opens up many avenues for future exploration, particularly for settings where causal discovery cannot be applied. Our source code is available here.

1. Introduction

Answering causal questions is at the core of many scientific enquiries in various fields such as genomics (Friedman et al., 2000), economics (Heckman, 2008), and the biomedical sciences (Imbens & Rubin, 2015). Scientists want to predict the effect of new interventions on a system. As a concrete example, given gene expression data under interventions such as single-gene knockouts, we could be interested in predicting the effect of an unseen combination of knockouts (Zhang et al., 2023). The standard approach to predicting intervention effects is causal discovery, where a directed acyclic graph that captures the causal mechanisms underlying the system is learned from data. However, if the end goal is predicting the effect of interventions, full causal discovery may not be necessary and incurs significant costs. First, learning DAGs over many variables is both computationally expensive and non-identifiable from limited data (Peters et al., 2017). Second and importantly, many systems in which we want to model perturbations are not well-modeled by standard causal assumptions such as acyclicity (e.g., feedback loops in biology (Tejada-Lapuerta et al., 2025; Freimer et al., 2022)), or the absence of hidden confounders. Consequently, the goal of this work is to model perturbations in systems without the need for causal discovery.

The challenge in modeling perturbations is that, naively, without appropriate inductive biases, a model might fail to capture a key aspect of real-world systems: perturbing one mechanism generally leaves the rest of the system invariant. Models that capture this property adapt much faster to new perturbational regimes during training. Indeed, since they have fewer mechanism parameters to change, they require fewer samples from the new regimes to adapt them. At inference time, this property also allows models to generalize to perturbations never seen during training. Bengio et al. (2019) has leveraged this difference in speed of adaptation to provide a learning signal towards the correct causal structure. While the notion of independently manipulable causal mechanisms is leveraged extensively for causal discovery (Bengio et al., 2019; Perry et al., 2022), here, we adapt this inductive bias to model perturbations in a model-agnostic way.

To develop models of interventions and their effects that generalize to novel interventions, we build on the framework of Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017). Briefly, MAML is an approach to optimizing parameters so that they need to be adapted only minimally to solve a novel task. To achieve this, MAML leverages data from multiple tasks at training time. In this work, we introduce a model that maps interventions to effects, optimizing it in a meta-learning framework so that its parameters require minimal adaptation under novel interventions. Analogous to the multiple tasks used by MAML, we train the model on multiple underlying data-generating processes. The final goal is to obtain a flexible method that will be able to predict the effects of novel interventions given new datasets sampled from new causal models without explicitly learning a DAG. Our contributions are 1) to propose a new optimization-based meta-learning approach that learns to predict the effect of novel interventions, and 2) to compare our method on synthetic data to several baselines, two of which are causal discovery methods.

[1] Université de Montréal [2] Mila - Quebec Artificial Intelligence Institute. Correspondence to: Cristian Dragos Manta <cristian-dragos.manta@mila.quebec>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

2. Background & Related Works

Causal Bayesian Networks. A common class of causal models, which we denote by $\mathcal{M}$, is Causal Bayesian Networks (CBNs). Let $X = (X_1, \dots, X_d) \sim P_X$ be a random vector and $G = (V, E)$ be a directed acyclic graph (DAG). We assume that the distribution P_X is Markov to G, which induces the following factorization:

$$P_X(X) = \prod_{i=1}^{d} P(X_i \mid \mathrm{pa}^G(X_i)) \tag{1}$$

where $\mathrm{pa}^G(X_i)$ are the variables that are the parents of X_i in G. Intervening corresponds to modifying the conditionals of the variable intervened on. Let $I \subseteq X$ be some interventional target. Then, the interventional distribution corresponding to intervening on I is given by:

$$P_X^I(X) = \prod_{i \notin I} P(X_i \mid \mathrm{pa}^G(X_i)) \prod_{i \in I} \tilde{P}(X_i \mid \mathrm{pa}^G(X_i)) \tag{2}$$

where the conditionals P not intervened on are the same as in Eq. 1 and the conditionals $\tilde{P}$ are new ones. This modularity of the mechanisms, which is often called the *independent causal mechanisms principle* (Schölkopf et al., 2021), is at the basis of the idea of using speed of adaptation as a learning signal. One common class of interventions is perfect interventions, where the conditionals are fixed to some values and thus do not depend on their parents.

Related work. Recently, Lorch et al. (2022); Ke et al. (2022); Dhir et al. (2024) proposed amortized approaches to do causal discovery in a supervised manner. Compared to our approach, these methods focus on learning graphs. In other words, if we were to predict the effects of interventions, we would have to first fit a parametric model based on the DAG predicted by their methods. Also, in contrast to the setting we will present, these approaches unrealistically assume that the ground-truth graphs for each dataset are known.

More recently, there is a line of works that aim to predict the effects of interventions without explicitly learning a causal model based on a graph. For example, Lotfollahi et al. (2023); Roohani et al. (2024); Gaudelet et al. (2024) propose methods that are trained on data containing mainly genetic perturbations on single genes in order to predict the effects of novel combinations of perturbations. While the setting is similar to ours, the method we propose is more general and is based on the speed of adaptation.

3. Method

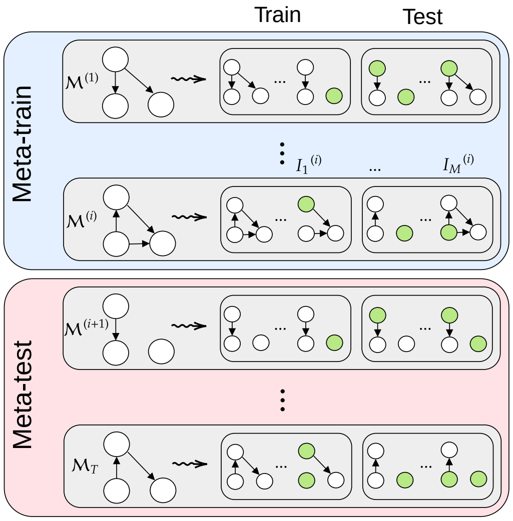

Figure 1. Illustration of our data-generating process.

In this work, we present a novel method based on meta-learning that aims to learn to predict effects of unseen interventions, where the training data consists entirely of observations of causal variables and their interventional targets. Importantly, once trained, our method can predict the effect of unseen interventions without any data in this regime.

3.1. Setting

Our distribution over tasks $p(\mathcal{T})$ is as follows: for each task $\mathcal{T}^{(i)}$, we first draw a causal model $\mathcal{M}^{(i)} \sim p(\mathcal{M})$ and interventional regimes $\mathbf{I}^{(i)} = (I_1^{(i)}, \dots, I_M^{(i)}) \sim p(\mathbf{I})$, where $I_j^{(i)}$ indicates the node targets of the j^{th} interventional regime in task i. The M sampled interventional regimes then allow us to generate a training dataset $\mathcal{D}^{(i)}_{\text{train}}$ and a test dataset $\mathcal{D}^{(i)}_{\text{test}}$ drawn from $\mathcal{M}^{(i)}$. We illustrate this process in Figure 1. We will detail our choices for the priors $p(\mathcal{M})$ and $p(\mathbf{I})$ in section 4.

3.2. Objective

Inspired by Finn et al. (2017) and motivated by maximizing speed of adaptation, our learning objective is:

$$\psi^* \in \arg\min_{\psi} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\text{transfer}}^{\mathcal{D}^{(i)}_{\text{train}}, D^{(i)}_{\text{test}}}(\psi - \alpha \nabla_\psi \mathcal{L}_{\text{fit}}^{D^{(i)}_{\text{train}}}(\psi)), \tag{3}$$

where

$$\mathcal{L}_{\text{fit}}^{\mathcal{D}}(\psi) = \frac{-1}{n} \sum_{i,j,k} \log p(x_j^{(i,k)} | x_{-j}^{(i,k)}, I^{(i)}; \mathcal{D}, \psi); \tag{4}$$

$$\begin{aligned} &\mathcal{L}_{\text{transfer}}^{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}}(\psi) \\ &= \frac{-1}{n} \sum_{i,j,k} \log p(x_{j,\text{test}}^{(i,k)} | x_{-j,\text{test}}^{(i,k)}, I_{\text{test}}^{(i)}; \mathcal{D}_{\text{train}}, \psi). \end{aligned} \tag{5}$$

Algorithm 1 Meta-learning training procedure

```
Require: Randomly initialized neural network parameters $\psi$, learning rates $\alpha, \beta$
for $t = 1, ..., T$ do
    Sample a causal model $\mathcal{M}^{(t)} \sim p(\mathcal{M})$
    Sample interventions $(I_1^{(t)}, ..., I_M^{(t)}) \sim p(\mathbf{I})$
    Construct a training dataset $\mathcal{D}^{(t)}_{\text{train}}$ and a test dataset $\mathcal{D}^{(t)}_{\text{test}}$ from $\mathcal{M}^{(t)}$ and $(I_1^{(t)}, ..., I_M^{(t)})$
end for
while not done training do
    for $t = 1, ..., T$ do
        $\tilde{\psi}^{(t)} \leftarrow \psi - \alpha \nabla_\psi \mathcal{L}_{\text{fit}}^{\mathcal{D}^{(t)}_{\texttt{train}}}(\psi)$
    end for
    $\psi \leftarrow \psi - \beta \nabla_\psi \sum_{t=1}^{T} \mathcal{L}_{\text{transfer}}^{\mathcal{D}^{(t)}_{\texttt{train}}, \mathcal{D}^{(t)}_{\texttt{test}}}(\tilde{\psi}^{(t)})$
end while
```

The index i is over interventional regimes, the index j is over causal variables, and the index k is over samples of a given interventional regime. Furthermore, n denotes the total number of samples. For a given variable x_j, its probability is parameterized by a neural network with parameters ψ that takes as input all the other variables x_{-j} as well as the interventional targets I. The datasets $\mathcal{D}$ or $\mathcal{D}_{\text{train}}$ are considered metadata that the neural network is conditioned on. In $\mathcal{L}_{\text{fit}}^{\mathcal{D}}(\psi)$, the model is penalized for prediction errors on the same samples belonging to the conditioning dataset $\mathcal{D}$. In $\mathcal{L}_{\text{transfer}}^{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}}(\psi)$, the model is penalized for prediction errors on samples coming from the test dataset while using the train dataset $\mathcal{D}_{\text{train}}$ as conditioning information. In practice, in order to optimize Equation 3, we use Algorithm 1.

The downstream application is to predict effects of unseen interventions on a new dataset $\mathcal{D}^*_{\text{test}}$ given a context $\mathcal{D}^*_{\text{train}}$ coming from the same causal model $\mathcal{M}^*$[1]. Thus, after training our model with Algorithm 1, we fine-tune it on $\mathcal{D}^*_{\text{train}}$ by performing a few gradient descent steps on $\mathcal{L}_{\text{fit}}^{\mathcal{D}^*_{\text{train}}}(\psi)$. See section 4 for empirical evidence in favor of this procedure. We call our method **MIP-FT**: "**M**eta-learning for **i**ntervention **p**rediction **f**ine-**t**uned".

3.3. Amortized Model Architecture

In order to make predictions, our model takes as input the data, the interventional targets, and some conditioning metadata (see Appendix D for its explicit form). More precisely, it operates by separately predicting each variable x_j given the rest (x_{-j}), hence the first input is a vector of the values of x_{-j}. The second input is made of the interventional targets I that we want to predict under. The conditioning metadata is a dataset drawn from the same causal model as the data that we are trying to predict, as a form of context. Inspired by Ke et al. (2022); Lorch et al. (2022); Dhir et al. (2024), we first pass the conditioning dataset as input to an alternating attention module (see Figure 5), which extracts implicit information about the causal structure to produce a summary 16-dimensional vector of features. Then, we embed each interventional target into a 16-dimensional vector and add the corresponding embeddings for multi-target interventional embeddings. Note that these parameters are shared across all predicted variables. Finally, we have specialized MLP modules - one for each variable - that take as input the summary metadata features, the intervention embeddings, the values of x_{-j}, in order to predict x_j. In the experiments, we add standard isotropic Gaussian noise to the predictions to obtain a likelihood suitable for the computations of the losses in Equation 4 and Equation 5.

4. Experiments

Our goal is to answer the questions below:

- How does the performance of our meta-learning method compare to different baselines, such as classical causal discovery?
- How does the fine-tuning step affect our model's performance?
- What is the importance of using a meta-learning objective instead of relying on supervised learning?

4.1. Synthetic data

For each task, we first sample a DAG G following the Erdős–Rényi scheme with the same number of edges as the number of variables (d) in expectation, then we sample data from a linear-Gaussian additive noise model as follows:

$$X = WX + N \tag{6}$$

where $N \sim \mathcal{N}(0, 0.1 I_d)$ and $W \in \mathbb{R}^{d \times d}$ are sampled from $\mathcal{N}(0, 1)$ where samples between $[-0.5, 0.5]$ are rejected and resampled. We use the adjacency matrix of G as a mask on W. For each meta-training task $\mathcal{T}^{(i)}$, we present all possible single-node intervention targets and most two-node targets in the training dataset $\mathcal{D}^{(i)}_{\text{train}}$, in addition to the observational setting. We leave out a certain number of unseen two-node interventional regimes for the test dataset $\mathcal{D}^{(i)}_{\text{test}}$. We use perfect interventions where we fix the value of the intervened variables to 2. Note that for our considered setting, the causal model should be identifiable since all variables are intervened upon (Eberhardt et al., 2012).

[1] As a comparison, classical causal discovery methods only perform this step: they are presented with a single dataset (context), and the downstream application is to use the recovered DAG to predict effects of unseen interventions on new samples drawn from the same causal model as the one that was used to generate the context data.

We present here the setting where graphs have $d = 20$ variables (See Appendix C for $d \in \{4, 10\}$). In each training dataset $\mathcal{D}^{(i)}_{\text{train}}$, we include the observational setting, 20 single-node interventions, and $\binom{20}{2} - 10 = 180$ double-node interventions, for a total of 201 regimes. With 20 samples per interventional setting, we have a total of 4020 samples. In each test dataset $\mathcal{D}^{(i)}_{\text{test}}$, we have the remaining 10 unseen two-node interventional regimes, for a total of 200 samples.

We reserve a held-out task $\mathcal{T}^*$, with the same intervention configurations as the previous tasks, but with a new causal model drawn from the same prior, for testing our model (see subsection 4.3).

4.2. Baselines

For several baselines, we do not rely on an amortized approach. Instead, for a given task, we take a DAG and then fit a linear Gaussian model using maximum likelihood estimation (MLE).

- As an upper bound for the performance, we use the ground-truth DAG (`GT`).
- We use the causal discovery methods GIES (`GIES`) (Hauser & Bühlmann, 2012) and IGSP (`IGSP`)(Wang et al., 2017) to learn a DAG. GIES is a score-based method, while IGSP is constraint-based. Both methods support interventional data and assume a linear Gaussian model.
- Finally, as a "trivial" baseline, we also compare against a full DAG (`Full`) that is obtained by sampling a topological ordering and by adding all the possible edges.

4.3. Evaluation Methodology

For our baselines, we only train on the training dataset of the downstream task, $\mathcal{D}^*_{\text{train}}$. We then evaluate the predictions of the learned model on the unseen interventions in $\mathcal{D}^*_{\text{test}}$ and report the mean squared error (`transfer loss`). We aggregate our results over 20 different seeds.

For our amortized learning model, we have three experimental training schemes. The default one (`MIP-FT`) is training according to Algorithm 1 on a set of tasks $\{\mathcal{T}_t\}_{t=1}^{T}$, fine-tuning on the training data $\mathcal{D}^*_{\text{train}}$ of the downstream task $\mathcal{T}^*$, and reporting the mean squared error, $L^{\mathcal{D}^*_{\text{train}}, \mathcal{D}^*_{\text{test}}}_{\text{transfer}}(\psi^*)$, between the model's predictions on $\mathcal{D}^*_{\text{test}}$ and the true samples, given the context $\mathcal{D}^*_{\text{train}}$. The second setting (`MIP`) removes the fine-tuning step. For the last setting (supervised learning, or `SL`), we simply perform one gradient descent step on $L^{\mathcal{D}^{(t)}_{\text{train}}}_{\text{fit}}(\psi)$ for each task $t = 1, ..., T$, omitting the optimization of the transfer objective. However, we still report the transfer loss on $\mathcal{T}^*$ as before. We aggregate our results over 5 different seeds.

4.4. Results

We report our results in Figure 2. We notice that our meta-learning approach with fine-tuning (MIP-FT) performs better than all other methods, except GT and GIES. In particular, we can observe that MIP-FT has better performance than IGSP. The experiments on smaller graph sizes in Appendix C show that the trend of beating more baselines seems to emerge when considering more variables, which is a promising sign for our method regarding scalability.

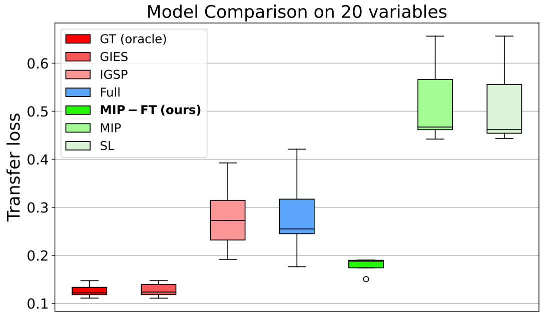

Figure 2. Comparison of the baselines against the different training schemes of our amortized model on datasets over 20 variables. In this setting, MIP-FT outperforms IGSP.

Secondly, we find that, if we remove the fine-tuning step, our model has a much worse performance (when comparing MIP-FT and MIP), a trend that is consistently observed in smaller graphs as shown in Appendix C. This makes sense, since Equation 3 is incentivizing the model to have a low transfer loss once we have already fit to the training dataset of a particular task.

Finally, since MIP and SL have a similar performance, we don't have conclusive evidence to support the benefits of the objective in Equation 3 alone.

5. Conclusion

Our preliminary results are promising: the new method we proposed is better than a causal discovery method in a setting where the ground-truth is a Causal Bayesian Network. For future work, we want to consider more diverse settings, such as larger problems and soft interventions, and also real-world applications such as the prediction of gene perturbations. We also aim to test more deeply the effect of some design choices of our method and study more closely under what conditions our method performs well. We believe this line of research can significantly advance the practical deployment of causal inference methods in complex, real-world domains.

References

Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Dhir, A., Ashman, M., Requeima, J., and van der Wilk, M. A meta-learning approach to bayesian causal discovery. *arXiv preprint arXiv:2412.16577*, 2024.

Eberhardt, F., Glymour, C., and Scheines, R. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. *arXiv preprint arXiv:1207.1389*, 2012.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Freimer, J. W., Shaked, O., Naqvi, S., Sinnott-Armstrong, N., Kathiria, A., Garrido, C. M., Chen, A. F., Cortez, J. T., Greenleaf, W. J., Pritchard, J. K., et al. Systematic discovery and perturbation of regulatory genes in human t cells reveals the architecture of immune networks. *Nature Genetics*, 54(8):1133–1144, 2022.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. Using bayesian networks to analyze expression data. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pp. 127–135, 2000.

Gaudelet, T., Del Vecchio, A., Carrami, E. M., Cudini, J., Kapourani, C.-A., Uhler, C., and Edwards, L. Season combinatorial intervention predictions with salt & peper. *arXiv preprint arXiv:2404.16907*, 2024.

Hauser, A. and Bühlmann, P. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

Heckman, J. J. Econometric causality. *International statistical review*, 76(1):1–27, 2008.

Imbens, G. W. and Rubin, D. B. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

Ke, N. R., Chiappa, S., Wang, J., Goyal, A., Bornschein, J., Rey, M., Weber, T., Botvinic, M., Mozer, M., and Rezende, D. J. Learning to induce causal structure. *arXiv preprint arXiv:2204.04875*, 2022.

Lorch, L., Sussex, S., Rothfuss, J., Krause, A., and Schölkopf, B. Amortized inference for causal structure learning. *Advances in Neural Information Processing Systems*, 35:13104–13118, 2022.

Lotfollahi, M., Klimovskaia Susmelj, A., De Donno, C., Hetzel, L., Ji, Y., Ibarra, I. L., Srivatsan, S. R., Naghipourfar, M., Daza, R. M., Martin, B., et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular systems biology*, 19(6):e11517, 2023.

Perry, R., Von Kügelgen, J., and Schölkopf, B. Causal discovery in heterogeneous environments under the sparse mechanism shift hypothesis. *Advances in Neural Information Processing Systems*, 35:10904–10917, 2022.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Roohani, Y., Huang, K., and Leskovec, J. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, 42(6):927–935, 2024.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Tejada-Lapuerta, A., Bertin, P., Bauer, S., Aliee, H., Bengio, Y., and Theis, F. J. Causal machine learning for single-cell genomics. *Nature Genetics*, pp. 1–12, 2025.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Y., Solus, L., Yang, K., and Uhler, C. Permutation-based causal inference algorithms with interventions. *Advances in Neural Information Processing Systems*, 30, 2017.

Zhang, J., Squires, C., Greenewald, K., Srivastava, A., Shanmugam, K., and Uhler, C. Identifiability guarantees for causal disentanglement from soft interventions. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

A. Acknowledgments

The authors acknowledge Tristan Deleu, Chandler Squires, and Yoshua Bengio for the insightful discussions and for helping us shaping this work. DS acknowledges support from NSERC Discovery Grant RGPIN-2023-04869, and a Canada-CIFAR AI Chair. PB acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). The compute resources required to run the experiments were provided through Mila's cluster.

B. Additional Discussion and Future Work

We have demonstrated that our method, MIP-FT, shows promising generalization capabilities for directly predicting the effects of unseen interventions. When the true data-generating process can be represented by an identifiable DAG, one interesting question is to what extent our model has the capacity to implicitly represent the true DAG. One candidate idea for testing this hypothesis could be to systematically perform hard interventions on different variables and observe how the predicted outputs change. For example, if the predictions for x_j under an intervention on x_i change depending on the value that x_i is set to, then we can conclude that x_i is an ancestor of x_j in the implicit representation of the DAG. However, we need to be very careful about the conclusions drawn using this approach. As our model has no structural inductive biases to learn DAGs, it can simply learn to memorize the effects of all the interventions seen during the fine-tuning step, then perform well on some unseen combinations of targets and poorly on other unseen combinations. So far, we have only tested our model in very limited settings, with access to an overwhelmingly large set of interventions at train time, and testing generalization on only a few unseen two-node combinations.

We leave future experiments as a follow-up work where we will vary more systematically the availability of interventions at train and test time, including introducing interventions affecting more than two nodes at the time. We will also study the generalization on unseen combinations compared to unseen single-node targets and dissect the performance based on the different combinations (instead of simply reporting aggregate statistics), to see whether the conclusions from the current work will be robust. We expect that a model that has implicitly learned the correct DAG will predict well the effects of all possible unseen combinations of interventional targets.

If the conclusions of our work are robust to the experimental settings described above, we could use causal discovery methods to understand what causal model our black box method emulates. However, our work is rather motivated by the settings where the data-generating process does not follow a DAG structure. In such cases, we expect our method to still be able to reliably predict effects of unseen interventions, as opposed to DAG-based causal discovery methods. We will test our aforementioned hypothesis in future work.

C. Additional Experimental Results

We further test our method on a small scale problem ($d = 4$ variables) and on a medium one ($d = 10$). In the $d = 4$ setting, we use 200 samples for each interventional setting. For each training dataset $\mathcal{D}^{(i)}_{\text{train}}$, we include the observational setting, 4 single-node interventional regimes, and $\binom{4}{2} - 1 = 5$ double-node interventional regimes, for a total of 10 regimes and 2000 training samples. For each test dataset $\mathcal{D}^{(i)}_{\text{test}}$, we have exactly one interventional regime: the remaining unseen two-node targets. Hence, $\mathcal{D}^{(i)}_{\text{test}}$ has 200 samples.

In the $d = 10$ setting, we have 50 samples per interventional regime. For each training dataset, we include the observational setting, 10 single-node interventions, and $\binom{10}{2} - 5 = 40$ double-node interventions, for a total of 51 regimes and 2550 training samples. For each test dataset, we have 5 unseen two-node interventional regimes. Hence, $\mathcal{D}^{(i)}_{\text{test}}$ has 250 samples.

Figures 3 and 4 show boxplots for these two settings, while Table 1 shows a numerical summary of the results across all of our experiments.

We notice that the fine-tuning step becomes increasingly important as the number of variables increases, as explained by the higher performance gap between MIP-FT and MIP on 10 variables compared to 4 variables.

Table 1. We show the transfer losses of all the models across all the problem sizes that we tested. We report the average and the standard deviation across 20 runs for the baseline models (GT, GIES, IGSP, Full), and across 5 runs for the different training schemes of our amortized model (MIP-FT, MIP, SL).

d	GT (oracle)	GIES	IGSP	Full	MIP-FT	MIP	SL
4	0.03 ± 0.015	0.03 ± 0.015	0.204 ± 0.37	0.633 ± 0.673	0.11 ± 0.063	0.204 ± 0.124	0.232 ± 0.120
10	0.073 ± 0.009	0.077 ± 0.019	0.183 ± 0.085	0.313 ± 0.148	0.207 ± 0.025	0.627 ± 0.165	0.655 ± 0.162
20	0.126 ± 0.011	0.127 ± 0.011	0.28 ± 0.06	0.279 ± 0.06	0.178 ± 0.015	0.519 ± 0.081	0.514 ± 0.082

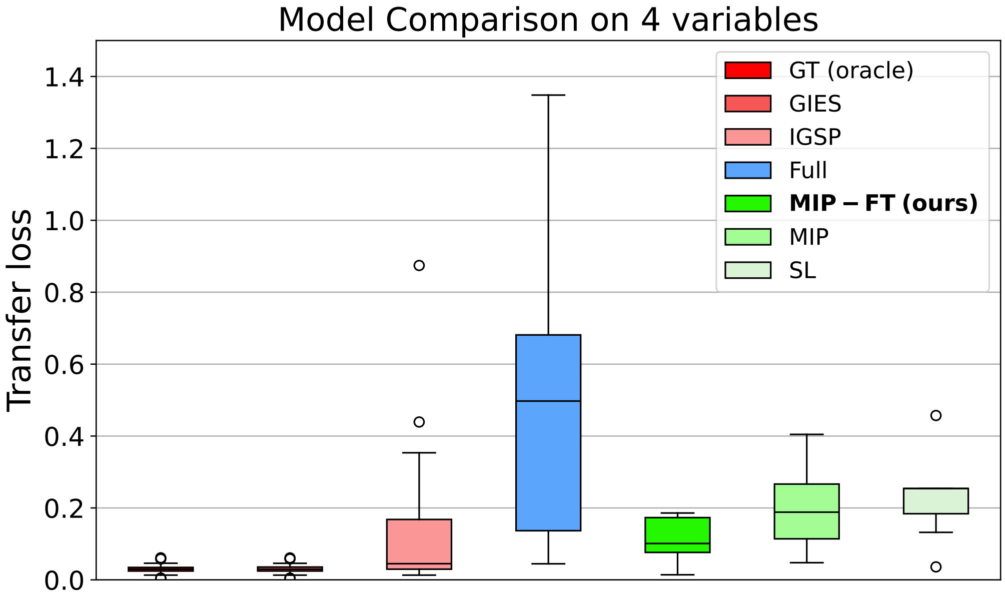

Figure 3. Comparison of the baselines against the different training schemes of our amortized model on datasets over 4 variables.

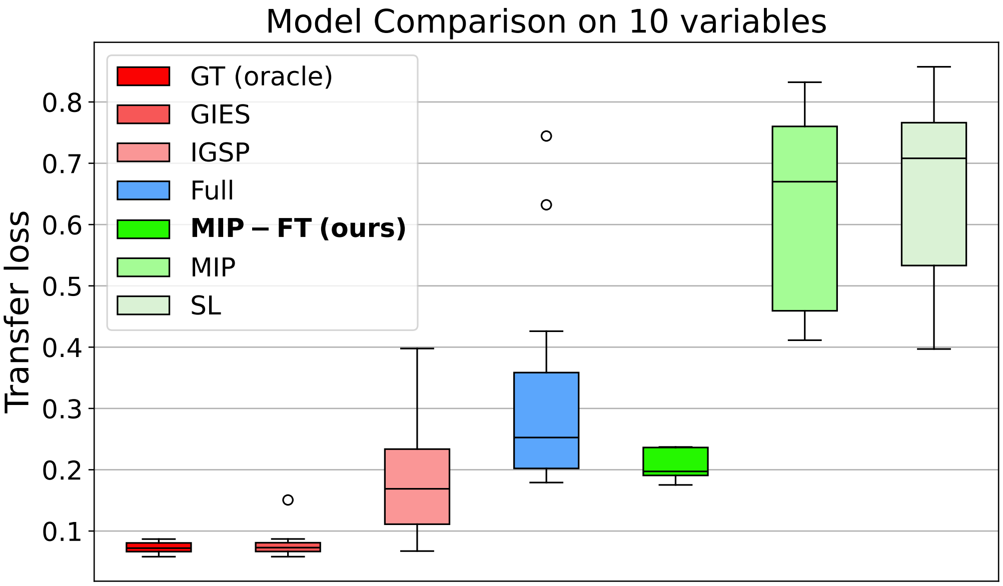

Figure 4. Comparison of the baselines against the different training schemes of our amortized model on datasets over 10 variables.

D. Additional Architectural Details

When used to make predictions on a test dataset $\mathcal{D}_{\text{test}}$ while using the train dataset $\mathcal{D}_{\text{train}}$ belonging to the same causal model as conditioning information (to compute Equation 5), our model can be written as a transformation of the following form:

$$x_{j,\text{test}} = f_j(x_{-j,\text{test}}, E_\Phi(I_{\text{test}}); h_\theta(\mathcal{D}_{\text{train}}), \phi_j) + z \quad \forall j = 1, ..., d. \tag{7}$$

The case where we make predictions on the same dataset as the conditioning one (to compute Equation 4) is a special case, letting $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{test}}$. E_Φ is the interventional embedding function, while h_θ is the alternating attention module. $z \sim \mathcal{N}(0, 1)$ is noise injected so that our model induces a probability distribution over the outputs. f_j contains several MLPs that combine the features provided by the different modules to compute the output. Note that the parameters Φ and θ are shared, while ϕ_j are specific to each causal variable. The set of all parameters together form ψ, which we train in Algorithm 1.

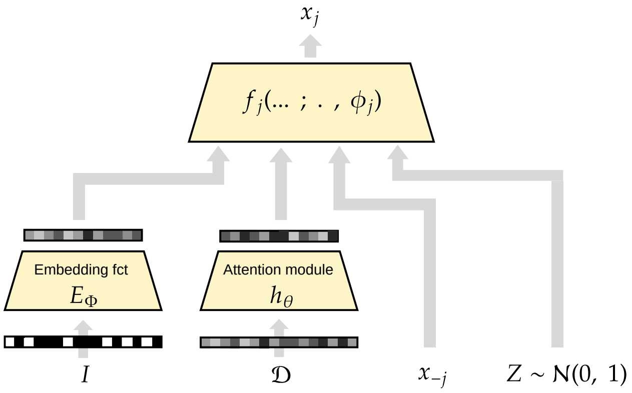

Figure 5. Illustration of the different modules composing our model architecture.

Inspired by Lorch et al. (2022), for the alternating attention module, we obtain the input tokens from $\mathcal{D}$ with position-wise MLPs that convert the values of the causal variables and the intervention targets to 16-dimensional vectors. We pass the input tokens through 8 layers of transformer encoders (Vaswani et al., 2017). Inside each layer, we first perform multi-head attention across the different causal variables for each sample, then we perform multi-head attention across the different samples for each causal variable. The sharing of the parameters of E_Φ across causal variables is motivated by the hypothesis that, if there is an embedding space in which the effects of interventions can be composed "nicely", then the embeddings should be reusable. The sharing of the parameters of h_θ across causal variables is motivated by the observation that, in Lorch et al. (2022), the attention module extracts the key summary features that enable the prediction of the DAG structure underlying $\mathcal{D}$. We use 8 attention heads, a transformer feedforward dimension of 32, a dropout rate of 0.1, and a ReLU activation.

All MLPs have $(32, 32)$ hidden sizes and use the leaky ReLU activation.

E. Further Hyperparameters Details

For GIES, we did a grid search for the sparsity regularization coefficient: $\lambda \in \{1000, 100, 10, 1\}$, including its default value. For IGSP, we use the partial correlation as the independence test, and we did a search for the significance level $\alpha \in \{0.5, 0.1, 0.05, 0.01, 0.005\}$. We keep the hyperparameters that lead to a better performance on a held-out dataset with interventions that were seen at train time.

For all the training schemes of the amortized model, we used the Adam optimizer with a learning rate of 0.00005. We let $T = 500$ meta-training tasks and evaluated on one meta-test task. For MIP-FT, we performed 10 fine-tuning gradient descent steps.