Learning to induce causal structure

Anonymous Authors[1]

Abstract

The fundamental challenge in causal induction is to infer the underlying graph structure given observational and/or interventional data. Most existing causal induction algorithms operate by generating candidate graphs and evaluating them using either score-based methods (including continuous optimization) or independence tests. In our work, we instead treat the inference process as a black box and design a neural network architecture that learns the mapping from *both observational and interventional data* to graph structures via supervised training on synthetic graphs. The learned model generalizes to new synthetic graphs, is robust to train-test distribution shifts, and achieves state-of-the-art performance on naturalistic graphs for low sample complexity.

1. Introduction

The problem of discovering the causal relationships that govern a system through observing its behavior, either passively (*observational data*) or by manipulating some of its variables (*interventional data*), lies at the core of many scientific disciplines, including medicine, biology, and economics. By using the graphical formalism of causal Bayesian networks (CBNs) (Koller & Friedman, 2009; Pearl, 2009), this problem can be framed as inducing the graph structure that best represents the relationships. Most approaches to causal structure induction are based on an unsupervised learning paradigm in which the structure is directly inferred from the system observations, either by ranking different structures according to some metrics (score-based approaches) or by determining the presence of an edge between pairs of variables using conditional independence tests (constraint-based approaches) (Drton & Maathuis, 2017; Glymour et al., 2019; Heinze-Deml et al., 2018a;b) (see Fig. 1(a)). The unsupervised paradigm poses however some challenges: score-based approaches are burdened with the high computational cost of having to explicitly consider all possible

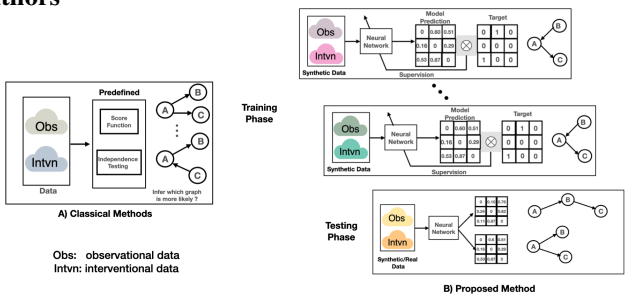

Figure 1. (A). Standard unsupervised approach to causal structure induction: Algorithms use a predefined scoring metric or statistical independence tests to select the best candidate structures. (B). Our supervised approach to causal structure induction (CSIvA): A model is presented with data and structures as training pairs and learns a mapping between them.

structures and with the difficulty of devising metrics that can balance goodness of fit with constraints for differentiating causal from purely statistical relationships (e.g. sparsity of the structure or simplicity of the generation mechanism); constraint-based methods are sensitive to failure of independence tests and require faithfulness, a property that does not hold in many real-world scenarios (Koski & Noble, 2012; Mabrouk et al., 2014).

In this work, we propose a supervised learning paradigm in which a model is first trained on synthetic data generated using different CBNs to learn a mapping from data to graph structures and then used to induce the structures underlying datasets of interest (see Fig. 1(b)). The model is a novel variant of a transformer neural network that receives as input a dataset consisting of *observational and interventional* samples corresponding to the same CBN and outputs a prediction of the CBN graph structure. The mapping from the dataset to the underlying structure is achieved through an attention mechanism which alternates between attending to different variables in the graph and to different samples from a variable. The output is produced by a decoder mechanism that operates as an autoregressive generative model on the inferred structure. Our approach can be viewed as a form of meta-learning, as the model learns about the relationship between datasets and structures underlying them. Supervised learning methods based on observational data have been shown to be feasible by Lopez-Paz et al. (2015a;b) and Li et al. (2020). By allowing the use of *both* observational *and interventional* data, our method enables greater flexibility.

A requirement of a supervised approach would seem to be

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SCIS workshop.

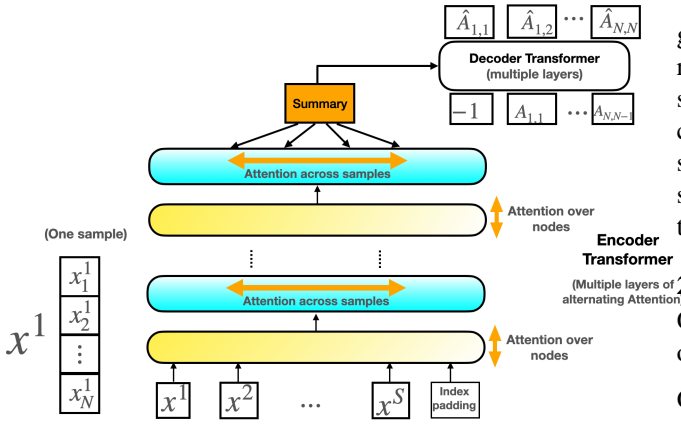

Figure 2. Our model architecture and the structure of the input and output at training time. The input is a dataset $\mathcal{D} = \{x^s := (x_1^s, \ldots, x_N^s)^\intercal\}_{s=1}^S$ of S samples from a CBN and its adjacency matrix A. The output is a prediction $\hat{A}$ of A. Even though the model receives a set of observations $\mathcal{D}$ at each gradient update, this is a single-example SGD approach because each update has only a single target A. The attention in a transformer normally only operates over different columns. We instead also take attention over the different rows, in alternating layers.

that the distributions of the training and test data match or highly overlap. Obtaining real-world training data with a known causal structure that matches test data from multiple domains is extremely challenging. We show that meta-learning enables the model to generalize well to data from naturalistic CBNs even if trained on synthetic data with relatively few assumptions. We show that our model can learn a mapping from datasets to structures and outperforms unsupervised approaches on classic benchmarks such as the Sachs (Sachs et al., 2005) and Asia (Lauritzen & Spiegelhalter, 1988) datasets, despite never directly being trained on such data. Our contributions can be summarized as follows:

- We tackle causal structure induction with a supervised approach (CSIvA) that maps datasets composed of *both observational and interventional* samples to structures.
- We introduce a variant of a transformer architecture whose attention mechanism is structured to discover relationships among variables across samples.
- We show that our proposed method generalizes to novel structures, whether or not training and test distributions match. Most importantly, training on synthetic data transfers effectively to naturalistic CBNs.

2. Causal Structure Induction via Attention (CSIvA)

Our approach is to treat causal structure induction as a supervised learning problem, by training a neural network to learn to map *observational and interventional* data to the graph structure of the underlying CBN. Obtaining diverse, real-world, data with known causal relationships in amounts sufficient for supervised training is not feasible. The key contribution of this work is to introduce a method that uses synthetic data generated from CBNs with different graph structures and CPDs that is robust to shifts between the training and test data distributions.

2.1. Supervised approach

Our approach is to learn a distribution of graphs conditioned on observational and interventional data as follows.

Our model defines a distribution $\hat{t}(\mathcal{G} \mid \mathcal{D}; \Theta)$ over graphs conditioned on observational and *interventional* data and parametrized by Θ. Specifically, $\hat{t}(A \mid \mathcal{D}; \Theta)$ has the following auto-regressive form: $\hat{t}(A \mid \mathcal{D}; \Theta) = \prod_{l=1}^{N^2} \sigma(A_l; \hat{A}_l = f_\Theta(A_{1,\ldots,(l-1)}, \mathcal{D}))$, where $\sigma(\cdot; \rho)$ is the Bernoulli distribution with parameter ρ, which is a function f_Θ built from an encoder-decoder architecture explained in Section 2.2 taking as input previous elements of the adjacency matrix A (represented here as an array of N^2 elements) and $\mathcal{D}$. It is trained via maximum likelihood estimation (MLE), i.e $\Theta^* = \text{argmin}_\Theta \mathcal{L}(\Theta)$, where $\mathcal{L}(\Theta) = -\mathbb{E}_{(\mathcal{G},\mathcal{D})\sim t}[\ln \hat{t}(\mathcal{G} \mid \mathcal{D}; \Theta)]$, which corresponds to the usual cross-entropy (CE) loss for the Bernoulli distribution. Training is achieved using a stochastic gradient descent (SGD) approach in which each gradient update is performed using a pair $(\mathcal{D}^i, A^i)$. The data-sampling distribution $t(\mathcal{G}, \mathcal{D})$ and the MLE objective uniquely determine the target distribution learned by the model. In the infinite capacity case, $\hat{t}(\cdot \mid \mathcal{D}; \Theta^*) = t(\cdot \mid \mathcal{D})$. To see this, it suffices to note that the MLE objective $\mathcal{L}(\Theta)$ can be written as $\mathcal{L}(\Theta) = \mathbb{E}_{\mathcal{D}\sim t}[\text{KL}(\hat{t}(\cdot \mid \mathcal{D}; \Theta); t(\cdot \mid \mathcal{D}))] + c$, where KL is the Kullback-Leibler divergence and c is a constant. In the finite-capacity case, the distribution defined by the model $\hat{t}(\cdot \mid \mathcal{D}; \Theta^*)$ is only an approximation of $t(\cdot \mid \mathcal{D})$.

2.2. Model architecture

The function f_Θ defining the model's probabilities is built using two transformer networks. It is formed by an encoder transformer and by a decoder transformer (which we refer to as "encoder" and "decoder" for short). At training time, the encoder receives as input dataset $\mathcal{D}^i$ and outputs a representation that summarizes the relationship between nodes in $\mathcal{G}^i$. The decoder then recursively outputs predictions of the elements of the adjacency matrix A^i using as input the elements previously predicted and the encoder output. This is shown in Fig. 2 (where with omitted index i, as in the remainder of the section). At test time we obtain deterministic predictions of the adjacency matrix elements by taking the argmax of the Bernoulli distribution and use those as inputs to the decoder.

2.2.1. ENCODER

Our encoder is structured as an $(N+1)\times(S+1)$ lattice. The $N \times S$ part of the lattice formed by the first N rows and first S columns receives a dataset $\mathcal{D} = \{(x_1^s, \ldots, x_N^s)^\mathsf{T}\}_{s=1}^S$. This is unlike standard transformers which typically receive as input a single data sample (e.g., a sequence of words in neural machine translation applications) rather than a set of data samples. Row $N+1$ of the lattice is used to specify whether each data sample is observational, through value -1, or *interventional*, through integer value in $\{1, \ldots, N\}$ to indicate the intervened node.

The goal of the encoder is to infer causal relationships between nodes by examining the set of samples. The transformer performs this inference in multiple stages, each represented by one transformer layer, such that each layer yields a $(N+1) \times (S+1)$ lattice of representations. The transformer is designed to deposit its summary representation of the causal structure in column $S+1$.

Embedding of the input. Each data-sample element x_n^s is embedded into a vector of dimensionality H. Half of this vector is allocated to embed the value x_n^s itself, while the other half is allocated to embed the unique identity for the node X_n. The value embedding is obtained by passing x_n^s, whether discrete or continuous, through an MLP[1] encoder specific to node X_n. We use a node-specific embedding because the values of each node may have very different interpretations and meanings. The node identity embedding is obtained using a standard 1D transformer positional embedding over node indices. For column $S+1$ of the input, the value embedding is a vector of zeros.

Alternating attention. Traditional transformers discover relationships among the elements of a data sample arranged in a one-dimensional sequence. With our two-dimensional lattice, the transformer could operate over the entire lattice at once to discover relationships among both nodes and samples. Given an encoding that indicates position n, s in the lattice, the model can in principle discover stable relationships among nodes over samples. However, the inductive bias to encourage the model to leverage the lattice structure is weak. Additionally, the model is invariant to sample ordering, which is desirable because the samples are *iid*. Therefore, we arrange our transformer in alternating layers. In the first layer of the pair, attention operates across all nodes of a single sample $(x_1^s, \ldots, x_N^s)^\mathsf{T}$ to encode the relationships among two or more nodes. In the second layer of the pair, attention operates across all samples for a given node $(x_n^1, \ldots, x_n^S)$ to encode information about the distribution of node values. Alternating attention in transformers was also done in Kossen et al. (2021).

[1]Using an MLP for a discrete variable is a slightly inefficient implementation of a node value embedding, but it ensures that the architecture is general.

Encoder summary. The encoder produces a *summary* vector e_n^{sum} with H elements for each node X_n, which captures essential information about the node's behavior and its interactions with other nodes. The summary representation is formed independently for each node and involves combining information across the S samples (the columns of the lattice). This is achieved with a method often used with transformers that involves a weighted average based on how informative each sample is. The weighting is obtained using the embeddings in column $S+1$ to form queries, and embeddings in columns $1, \ldots, S$ to provide keys and values, and then using standard key-value attention.

2.2.2. DECODER

The decoder uses the summary information from the encoder to generate a prediction of the adjacency matrix A of the underlying $\mathcal{G}$. It operates sequentially, at each step producing a binary output indicating the prediction $\hat{A}_{k,l}$ of $A_{k,l}$, proceeding row by row. The decoder is an autoregressive transformer, meaning that each prediction $\hat{A}_{kl}$ is obtained based on all elements of A previously predicted, as well as the summary produced by the encoder. Our method does not enforce acyclicity. Although this could in principle yield cycles in the graph, in practice we observed strong performance regardless. Nevertheless, one could likely improve the results e.g. by using post-processing (Lippe et al., 2021) or by extending the method with an accept-reject algorithm (Castelletti & Mascaro, 2022; Li et al., 2022).

Auxiliary loss. We found that autoregressive decoding of the flattened $N \times N$ adjacency matrix is too difficult for the decoder to learn alone. To provide additional inductive bias to facilitate learning of causal graphs, we added the auxiliary task of predicting the parents $A_{n,:}$ and children $A_{:,n}$ of node X_n from the encoder summary, e_n^{sum}. This is achieved using an MLP to learn a mapping f_n, such that $f_n(e_n^{\text{sum}}) = (\hat{A}_{n,:}, \hat{A}_{:,n}^\mathsf{T})$. While this prediction is redundant with the operation of the decoder, it short circuits the autoregressive decoder and provides a strong training signal to support proper training.

3. Experiments

We report on a series of experiments of increasing challenge to our supervised approach to causal structure induction. First, we examined whether CSIvA generalizes well on synthetic data for which the training and test distributions are identical (Section 3.1). This experiment tests whether the model can learn to map from a dataset to a structure. Second, we examined generalization to an out-of-distribution (OOD) test distribution, and we determined hyperparameters of the synthetic data generating process that are most robust to OOD testing (Section E.2). Third, we trained CSIvA using the hyperparameters from our second experiment, and evaluated it on a different type of OOD test distribution from two naturalistic CBNs (Section 3.2). This last experiment

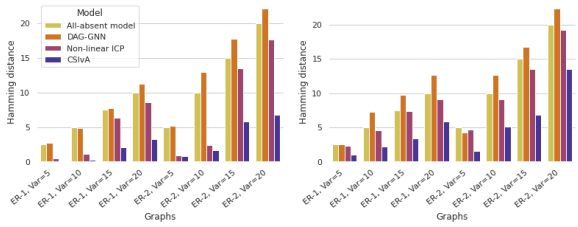

Figure 3. Hamming distance $\mathcal{H}$ between predicted and ground-truth adjacency matrices on the continuous (left) and MLP (right) data, compared to DAG-GNN (Yu et al., 2019) and non-linear ICP (Heinze-Deml et al., 2018b), averaged over 128 sampled graphs. CSIvA significantly outperforms all other baselines.

is the most important test of our hypothesis that causal structure of synthetic datasets can be a useful proxy for discovering causal structure in realistic settings.

3.1. In-distribution experiments

In this set of experiments, we investigated whether CSIvA can learn to map from data to structures in the case in which the training and test distributions are identical. In this setting, our supervised approach has an advantage over unsupervised ones, as it can learn about the training distribution and leverage this knowledge during testing. We examined the performance on data with increasing order of difficulty, starting with linear (continuous data), before moving to non-linear cases (MLP and Dirichlet data).

Continuous data. Results on continuous data are presented in Figure 3(a). CSIvA achieves Hamming distance $\mathcal{H} < 7$ on all evaluated graphs (up to size 20). Similar to previous findings (Yu et al., 2019; Ke et al., 2020a), larger and denser graphs are more challenging to learn. Non-linear ICP achieves fairly good performance on smaller graphs ($N \leq 10$), however, the performance drops quickly as size of graphs increases ($N > 10$). Non-linear ICP also required a modification[2] to the dataset wherein multiple samples were collected from the same modified graph after a point intervention (20 samples per intervention), while other methods only sampled once per intervention.

MLP data. Results on MLP data are shown in Figure 3(b). Our model significantly outperforms non-linear ICP and DAG-GNN. Differences become more apparent with larger graph sizes ($N \geq 10$) and denser graphs (ER-2 vs ER-1), as these graphs are more challenging to learn.

Dirichlet data. Due to the limitation of space, results of experiments on the Dirichlet data is discussed in Appendix section E.1.3.

[2]Without this modification, the method achieved near chance performance.

	Sachs	Asia	Child
Number of nodes	11	8	20
All-absent Baseline	17	8	25
GES Chickering (2002)	19	4	33
DAG-notears Zheng et al. (2018)	22	14	23
DAG-GNN Yu et al. (2019)	13	8	20
GES Hauser & Bühlmann (2012)	16	11	31
ICP Peters et al. (2016)	17	8	27*
Non-linear ICP Heinze-Deml et al. (2018b)	16	8	23*
DAG-EQ Li et al. (2020)	16	-	-
CSIvA (MLP data)	**6**	**3**	**11**
CSIvA (Dirichlet data)	**5**	**3**	**10**

Table 1. **Results on Sachs and Asia data:** Hamming distance $\mathcal{H}$ between predicted and ground-truth adjacency matrices. *To maintain computational tractability, the size of parental sets considered was limited to 3.

3.2. Sim-to-real experiments

In this final set of experiments, we evaluated CSIvA's ability to generalize from being trained on MLP and Dirichlet data to being evaluated on the widely used Sachs (Sachs et al., 2005) and Asia (Lauritzen & Spiegelhalter, 1988) CBNs from the BnLearn repository, which have $N = 11$ and $N = 8$ nodes respectively. We followed the established protocol from Ke et al. (2020a); Lippe et al. (2021); Scherrer et al. (2021) where we sampled observational and interventional data from the CBNs provided by the repository. These experiments are the most important test of our hypothesis that causal structure of synthetic datasets can be a useful proxy for discovering causal structure in realistic settings.

We emphasize that all hyperparameters for the MLP and Dirichlet data generation and for the learning procedure were chosen without using the Sachs and Asia data; only after the architecture and parameters were finalized was the model tested on these benchmarks. Furthermore, to keep the setup simple, we trained on data sampled from a single set of hyperparameters instead of a broader mixture. Findings in Section E.2 suggest that ER-2 graphs with $\alpha = 0.25$ work well overall and hence were chosen.

We report the results in Table 1. We compare to a range of baselines from Heinze-Deml et al. (2018b); Yu et al. (2019); Gamella & Heinze-Deml (2020) and others. Note that we do not compare to the method in Ke et al. (2020a), as this method needs at least $500,000$ data samples (which is more than 300 times the amount required by our method). CSIvA trained on both the MLP data and on the Dirichlet data significantly outperforms all other methods on both the Asia and the Sachs dataset. This serves as strong evidence that our model can learn to induce causal structures in the more realistic real-world CBNs, while only trained on synthetic data.

References

Ahmed, O., Träuble, F., Goyal, A., Neitz, A., Bengio, Y., Schölkopf, B., Wüthrich, M., and Bauer, S. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. *arXiv preprint arXiv:2010.04296*, 2020.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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.

Budhathoki, K. and Vreeken, J. Causal inference by stochastic complexity. *arXiv:1702.06776*, 2017.

Castelletti, F. and Mascaro, A. Bcdag: An r package for bayesian structure and causal learning of gaussian dags. *arXiv preprint arXiv:2201.12003*, 2022.

Chickering, D. M. Optimal structure identification with greedy search. *Journal of machine learning research*, 3 (Nov):507–554, 2002.

Cooper, G. F. and Yoo, C. Causal Discovery from a Mixture of Experimental and Observational Data. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pp. 116–125, San Francisco, CA, USA, 1999.

Cowell, R. G., Dawid, A. P., Lauritzen, S., and Spiegelhalter, D. J. *Probabilistic Networks and Expert Systems, Exact Computational Methods for Bayesian Networks*. Springer-Verlag, 2007.

Daniusis, P., Janzing, D., Mooij, J., Zscheischler, J., Steudel, B., Zhang, K., and Schölkopf, B. Inferring deterministic causal relations. *arXiv preprint arXiv:1203.3475*, 2012.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Douglas, L., Zarov, I., Gourgoulias, K., Lucas, C., Hart, C., Baker, A., Sahani, M., Perov, Y., and Johri, S. A universal marginalizer for amortized inference in generative models. *arXiv preprint arXiv:1711.00695*, 2017.

Drton, M. and Maathuis, M. H. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4(1):365–393, 2017.

Eaton, D. and Murphy, K. Bayesian structure learning using dynamic programming and MCMC. In *Uncertainty in Artificial Intelligence*, pp. 101–108, 2007.

Gamella, J. L. and Heinze-Deml, C. Active invariant causal prediction: Experiment selection through stability. *arXiv preprint arXiv:2006.05690*, 2020.

Ghassami, A., Salehkaleybar, S., Kiyavash, N., and Zhang, K. Learning causal structures using regression invariance. In *Advances in Neural Information Processing Systems*, pp. 3011–3021, 2017.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019.

Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., and Sebag, M. Causal generative neural networks. *arXiv preprint arXiv:1711.08936*, 2017.

Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., and Sebag, M. Learning functional causal models with generative neural networks. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pp. 39–80. Springer, 2018.

Goyal, A., Didolkar, A., Ke, N. R., Blundell, C., Beaudoin, P., Heess, N., Mozer, M. C., and Bengio, Y. Neural production systems. *Advances in Neural Information Processing Systems*, 34, 2021a.

Goyal, A., Didolkar, A., Lamb, A., Badola, K., Ke, N. R., Rahaman, N., Binas, J., Blundell, C., Mozer, M., and Bengio, Y. Coordination among neural modules through a shared global workspace. *arXiv preprint arXiv:2103.01197*, 2021b.

Goyal, A., Friesen, A. L., Banino, A., Weber, T., Ke, N. R., Badia, A. P., Guez, A., Mirza, M., Konyushkova, K., Valko, M., et al. Retrieval-augmented reinforcement learning. *arXiv preprint arXiv:2202.08417*, 2022.

Guyon, I. Cause-effect pairs kaggle competition, 2013. *URL https://www. kaggle. com/c/cause-effect-pairs*, pp. 165, 2013.

Guyon, I. Chalearn fast causation coefficient challenge, 2014. *URL https://www. codalab. org/competitions/1381*, pp. 165, 2014.

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.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heckerman, D., Geiger, D., and Chickering, D. M. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

Heinze-Deml, C., Maathuis, M. H., and Meinshausen, N. Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391, 2018a.

Heinze-Deml, C., Peters, J., and Meinshausen, N. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018b.

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pp. 689–696, 2009.

Ivanov, O., Figurnov, M., and Vetrov, D. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021.

Kalainathan, D., Goudet, O., Guyon, I., Lopez-Paz, D., and Sebag, M. Sam: Structural agnostic model, causal discovery and penalized adversarial learning. *arXiv preprint arXiv:1803.04929*, 2018.

Ke, N. R., Bilaniuk, O., Goyal, A., Bauer, S., Schölkopf, B., Mozer, M. C., Larochelle, H., Pal, C., and Bengio, Y. Dependency structure discovery from interventions. 2020a.

Ke, N. R., Wang, J., Mitrovic, J., Szummer, M., Rezende, D. J., et al. Amortized learning of neural causal representations. *arXiv preprint arXiv:2008.09301*, 2020b.

Ke, N. R., Didolkar, A. R., Mittal, S., Goyal, A., Lajoie, G., Bauer, S., Rezende, D. J., Mozer, M. C., Bengio, Y., and Pal, C. Systematic evaluation of causal discovery in visual model based reinforcement learning. 2021.

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

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

Koski, T. and Noble, J. A review of Bayesian networks and structure learning. *Mathematica Applicanda*, 40, 2012.

Kossen, J., Band, N., Lyle, C., Gomez, A. N., Rainforth, T., and Gal, Y. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. Gradient-based neural dag learning. *arXiv preprint arXiv:1906.02226*, 2019.

Lauritzen, S. L. and Spiegelhalter, D. J. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

Li, H., Xiao, Q., and Tian, J. Supervised whole dag causal discovery. *arXiv preprint arXiv:2006.04697*, 2020.

Li, Y., Akbar, S., and Oliva, J. B. Flow models for arbitrary conditional likelihoods. *arXiv preprint arXiv=1909.06319*, 2019.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Lago, A. D., Hubert, T., Choy, P., d'Autume, C. d. M., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Cherepanov, A., Molloy, J., Mankowitz, D. J., Robson, E. S., Kohli, P., de Freitas, N., Kavukcuoglu, K., and Vinyals, O. Competition-level code generation with alphacode. *arXiv preprint arXiv:2203.07814*, 2022.

Lippe, P., Cohen, T., and Gavves, E. Efficient neural causal discovery without acyclicity constraints. *arXiv preprint arXiv:2107.10483*, 2021.

Lopez-Paz, D., Muandet, K., and Recht, B. The randomized causation coefficient. *J. Mach. Learn. Res.*, 16:2901–2907, 2015a.

Lopez-Paz, D., Muandet, K., Schölkopf, B., and Tolstikhin, I. Towards a learning theory of cause-effect inference. In *International Conference on Machine Learning*, pp. 1452–1461, 2015b.

Mabrouk, A., Gonzales, C., Jabet-Chevalier, K., and Chojnacki, E. An efficient Bayesian network structure learning algorithm in the presence of deterministic relations. *Frontiers in Artificial Intelligence and Applications*, 263: 567–572, 2014.

Mitrovic, J., Sejdinovic, D., and Teh, Y. W. Causal inference via kernel deviance measures. In *Advances in Neural Information Processing Systems*, pp. 6986–6994, 2018.

Monti, R. P., Zhang, K., and Hyvarinen, A. Causal discovery with general non-linear relationships using non-linear ica. *arXiv preprint arXiv:1904.09096*, 2019.

Mooij, J. M., Magliacane, S., and Claassen, T. Joint causal inference from multiple contexts. *arXiv preprint arXiv:1611.10351*, 2016.

Müller, S., Hollmann, N., Arango, S. P., Grabocka, J., and Hutter, F. Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

Pearl, J. *Causality*. Cambridge university press, 2009.

Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B. Identifiability of causal graphs using functional models. In *Proceedings of the 27th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 589–598, 2011.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005.

Scherrer, N., Bilaniuk, O., Annadani, Y., Goyal, A., Schwab, P., Schölkopf, B., Mozer, M. C., Bengio, Y., Bauer, S., and Ke, N. R. Learning neural causal models with active interventions. *arXiv preprint arXiv:2109.02429*, 2021.

Shimizu, S., Hoyer, P. O., Hyvärinen, A., and Kerminen, A. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct): 2003–2030, 2006.

Spirtes, P., Glymour, C. N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., and Richardson, T. *Causation, prediction, and search*. MIT press, 2000.

Sun, X., Janzing, D., Schölkopf, B., and Fukumizu, K. A kernel-based causal learning algorithm. In *Proceedings of the 24th international conference on Machine learning*, pp. 855–862. ACM, 2007.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, J. X., King, M., Porcel, N., Kurth-Nelson, Z., Zhu, T., Deck, C., Choy, P., Cassin, M., Reynolds, M., Song, F., et al. Alchemy: A structured task distribution for meta-reinforcement learning. *arXiv preprint arXiv:2102.02926*, 2021.

Yoon, J., Jordon, J., and Van Der Schaar, M. Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*, 2018.

Yu, Y., Chen, J., Gao, T., and Yu, M. Dag-gnn: Dag structure learning with graph neural networks. *arXiv preprint arXiv:1904.10098*, 2019.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*, 2012.

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, pp. 9472–9483, 2018.

Zhu, S. and Chen, Z. Causal discovery with reinforcement learning. *arXiv preprint arXiv:1906.04477*, 2019.

A. Appendix.

The appendix is organized as follows, Section D describes the transformers architecture, Section E discusses detailed results for experiments in Section **??** and Section E.2 discusses results for further ablation studies on how the amount of interventions in the data impacts the performance of our model.

B. Background

In this section we give some background on causal Bayesian networks (CBNs) and on transformer neural networks, which form the main ingredients of our approach (see Appendix D for more details).

Causal Bayesian networks. A *Bayesian network* (Cowell et al., 2007; Koller & Friedman, 2009; Pearl, 1988; 2009) is a pair $\mathcal{M} = \langle \mathcal{G}, p \rangle$, where $\mathcal{G}$ is a *directed acyclic graph* (DAG) whose nodes $X_1, \dots, X_N$ represent random variables and edges express statistical dependencies among them, and where p is a joint distribution over all nodes that factorizes into the product of the conditional probability distributions (CPDs) of each node X_n given its *parents* $\text{pa}(X_n)$ (namely all nodes with an edge onto X_n), i.e. $p(X_1, \dots, X_N) = \prod_{n=1}^{N} p(X_n \,|\, \text{pa}(X_n))$. The structure of $\mathcal{G}$ can be represented by an adjacency matrix A, defined by setting the (k, l) entry, $A_{k,l}$, to 1 if there is an edge from X_l to X_k and to 0 otherwise. Therefore, the n-th row of A, denoted by $A_{n,:}$, indicates the parents of X_n while the n-th column, denoted by $A_{:,n}$, indicates the *children* of X_n. A BN $\mathcal{M}$ can be given causal semantic by interpreting an edge between two nodes as expressing causal rather than statistical dependence. For the experiments, we consider datasets whose elements are *observational data samples*, namely samples from $p(X_1, \dots, X_N)$, and *interventional data samples*, namely samples from $p_{\text{do}(X_{n'}=x)}(X_1, \dots, X_N) = \prod_{n=1, n\neq n'}^{N} p(X_n \,|\, \text{pa}(X_n)) \delta_{X_{n'}=x}$, where $\delta_{X_{n'}=x}$ is a delta function, corresponding to an *atomic intervention* on variable $X_{n'}$ that forces the variable to take on value x. Two adjacency matrices A^i and A^j can be compared using the *Hamming distance* ($\mathcal{H}$), defined as the norm of the difference between them, $\mathcal{H} = |A^i - A^j|_1$.

Transformer neural network. A transformer (Devlin et al., 2018; Vaswani et al., 2017) is a neural network equipped with layers of self-attention mechanisms that make them suited to modelling structured data. In traditional applications, attention is used to account for the sequential nature of the data, e.g. a sentence is treated as a stream of words. In our case, each input of the transformer is a dataset of observational or interventional samples corresponding to the same CBN. Attention is thus used to account for the structure induced by the CBN graph structure and by having different samples from the same node. Transformers are permutation invariant with respect to the positions of the input elements, which ensures that the graph structure prediction does not depend on the node and sample position.

C. Related work

Methods for inferring causal graphs can broadly be categorized into score-based (continuous optimization methods included), constraint-based, and asymmetry-based methods. Score-based methods search through the space of possible candidate graphs, usually DAGs, and ranks them based on some scoring function (Chickering, 2002; Cooper & Yoo, 1999; Goudet et al., 2017; Hauser & Bühlmann, 2012; Heckerman et al., 1995; Tsamardinos et al., 2006; Zhu & Chen, 2019). Recently, Zheng et al. (2018); Yu et al. (2019); Lachapelle et al. (2019) framed the structure search as a continuous optimization problem. There also exist score-based methods that use a mix of continuous and discrete optimization (Bengio et al., 2019; Ke et al., 2020a; Lippe et al., 2021; Scherrer et al., 2021). Constraint-based methods (Monti et al., 2019; Spirtes et al., 2000; Sun et al., 2007; Zhang et al., 2012; Zhu & Chen, 2019) infer the DAG by analyzing conditional independencies in the data. Eaton & Murphy (2007) use dynamic programming techniques. Asymmetry-based methods (Shimizu et al., 2006; Hoyer et al., 2009; Peters et al., 2011; Daniusis et al., 2012; Budhathoki & Vreeken, 2017; Mitrovic et al., 2018) assume asymmetry between cause and effect in the data and use this to estimate the causal structure. Peters et al. (2016); Ghassami et al. (2017); Rojas-Carulla et al. (2018); Heinze-Deml et al. (2018a) exploit invariance across environments to infer causal structure. Mooij et al. (2016) propose a modelling framework that leverages existing methods.

Several learning-based methods have been proposed (Bengio et al., 2019; Goudet et al., 2018; Guyon, 2013; 2014; Kalainathan et al., 2018; Ke et al., 2020a;b; Lopez-Paz et al., 2015b). These works are mainly concerned with learning only part of the causal induction pipeline, such as the scoring function. Hence, are significantly different from our work, which uses an end-to-end supervised learning approach to learn to map from datasets to graphs. Neural network methods equipped with learned masks exist in the literature (Douglas et al., 2017; Goyal et al., 2021a; Ivanov et al., 2018; Li et al., 2019; Yoon et al., 2018), but only a few have been adapted to causal inference. Several transformer models (Goyal et al., 2022; Kossen

et al., 2021; Müller et al., 2021) have been proposed for learning to map from datasets to targets. However, none has been applied it to causal discovery. Few supervised learning approaches have been proposed, one framing the task as a kernel mean embedding classification problem (Lopez-Paz et al., 2015a;b) and one operating directly on covariance matrices (Li et al., 2020). These models accept observational data only, and because causal identifiability requires *both observational and interventional* data, our model is in principle more powerful (see Table **??**).

D. Transformer Neural Networks

The transformer architecture, introduced in Vaswani et al. (2017), is a multi-layer neural network architecture using stacked self-attention and point-wise, fully connected, layers. The classic transformer architecture has an encoder and a decoder, but the encoder and decoder do not necessarily have to be used together.

Scaled dot-product attention. The attention mechanism lies at the core of the transformer architecture. The transformer architecture uses a special form of attention, called the scaled dot-product attention. The attention mechanism allows the model to flexibility learn to weigh the inputs depending on the context. The input to the QKV attention consists of a set of queries, keys and value vectors. The queries and keys have the same dimensionality of d_k, and values often have a different dimensionality of d_v. Transformers compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values. In practice, transformers compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. The matrix of outputs is computed as: $\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$.

Encoder. The encoder is responsible for processing and summarizing the information in the inputs. The encoder is composed of a stack of N identical layers, where each layer has two sub-layers. The first sub-layer consists of a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. Transformers employ a residual connection (He et al., 2016) around each of the two sub-layers, followed by layer normalization (Ba et al., 2016). That is, the output of each sub-layer is $\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$, where $\mathrm{Sublayer}(x)$ is the function implemented by the sub-layer itself.

Decoder. The decoder is responsible for transforming the information summarized by the encoder into the outputs. The decoder also composes of a stack of N identical layers, with a small difference in the decoder transformer layer. In addition to the two sub-layers in each encoder layer, a decoder layer consists of a third sub-layer. The third sub-layer performs a multi-head attention over the output of the encoder stack. Similar to the encoder, transformers employ residual connections around each of the sub-layers, followed by layer normalization. Transformers also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i.

E. Detailed results.

Detailed results for experiments in Section 3 are described in the tables below.

Hyperparameters. For all of our experiments (unless otherwise stated) our model was trained on $I = 15,000$ pairs $\{(\mathcal{D}^i, A^i)\}_{i=1}^{I}$, where each dataset $\mathcal{D}^i$ contained $S = 1500$ observational and interventional samples. For experiments on discrete data, a data-sample element x^s could take values in $\{1, 2, 3\}$. Details of the data generating process can be found in Section **??**. For evaluation in Sections 3.1 and E.2, our model was tested on $I' = 128$ (different for the training) pairs $\{(\mathcal{D}^{i'}, A^{i'})\}_{i'=1}^{I'}$, where each dataset $\mathcal{D}^{i'}$ contained $S = 1500$ observational and *interventional* samples. For the Sachs and Asia benchmarks, our model was still tested on $I' = 128$ (different for the training) pairs $\{(\mathcal{D}^{i'}, A^{i'})\}_{i'=1}^{I'}$, however, $A^{i'} = A^{j'}$ since there is only a single adjacency matrix in each one of the benchmarks. For each experimental setting, we present test results averaging performance over the 128 datasets and 3 random seeds and up to size 20 graphs. The model was trained for $500,000$ iterations using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of $1\mathrm{e}{-4}$.

We parameterized our architecture such that inputs to the encoder were embedded into 128-dimensional vectors. The encoder transformer had 12 layers and 8 attention-heads per layer. The final attention step for summarization had 8 attention heads. The decoder was a smaller transformer with only 4 layers and 8 attention heads per layer. Discrete inputs were

encoded using an embedding layer before passing into our model.

Comparisons to baselines. In Section 3.1, we compare CSIvA to two strong baselines in the literature, namely non-linear ICP (Heinze-Deml et al., 2018b) and DAG-GNN (Yu et al., 2019). Non-linear ICP can handle both observational and *interventional* data, while DAG-GNN can only use observational data. These two baselines are unsupervised methods, i.e. they are not tuned to a particular training dataset but instead rely on a general-purpose algorithm. We also compared to an all-absent model corresponding to a zero adjacency matrix, which acts as a sanity check baseline. We also considered other methods (Chickering, 2002; Hauser & Bühlmann, 2012; Zhang et al., 2012; Gamella & Heinze-Deml, 2020), but only presented a comparison with non-linear ICP and DAG-GNN as these have shown to be strong performing models in other works (Ke et al., 2020a; Lippe et al., 2021; Scherrer et al., 2021). For Section 3.2, we also compared to additional baselines from Chickering (2002); Hauser & Bühlmann (2012); Zheng et al. (2018); Gamella & Heinze-Deml (2020); Li et al. (2020). Note that methods from Chickering (2002); Zheng et al. (2018); Yu et al. (2019); Li et al. (2020) take observational data only. DAG-GNN outputs several candidate graphs based on different scores, such as evidence lower bound or negative log likelihood, we chose the best result to compare to our model. Note that non-linear ICP does not work on discrete data, i.e. on the MLP and Dirichlet data, therefore a small amount of Gaussian noise $\mathcal{N}(0, 0.1)$ was added to this data in order for the method to run.

E.1. In-distribution experiments

E.1.1. Results on continuous data

Results for comparions between our model CSIvA and baselines non-linear ICP (Heinze-Deml et al., 2018b) and DAG-GNN (Yu et al., 2019) are shown in Table 2. Both non-linear ICP and our model CSIvAperform well on the data. Both are significantly better compared to DAG-GNN (Yu et al., 2019), which only takes observational data.

	ER = 1				**ER = 2**			
	Var = 5	**Var = 10**	**Var = 15**	**Var = 20**	**Var = 5**	**Var = 10**	**Var = 15**	**Var = 20**
Abs*	2.50	5.00	7.50	10.00	5.00	10.00	15.00	20.00
(Yu et al., 2019)	2.71	4.76	7.71	11.32	5.20	8.81	17.81	22.21
(Heinze-Deml et al., 2018b)	0.47	1.10	6.3	8.6	0.90	2.41	13.52	17.71
Our Model	**0.12** $\pm$ 0.03	**0.35** $\pm$ 0.05	**2.10** $\pm$ 0.07	**3.21** $\pm$ 0.07	**0.81** $\pm$ 0.05	**1.73** $\pm$ 0.04	**5.62** $\pm$ 0.19	**6.86** $\pm$ 0.21

Table 2. **Results on Continuous data**. Hamming distance $\mathcal{H}$ for learned and ground-truth edges on synthetic graphs, compared to other methods, averaged over 128 sampled graphs. The number of variables varies from 5 to 20, expected degree = 1 or 2, and the value of variables are drawn from $\mathcal{N}(0, 0.1)$. Note that for (Heinze-Deml et al., 2018b), the method required nodes to be causally ordered, and 20 repeated samples taken per intervention, as interventions were continuously valued. "Abs" baselines are All-Absent baselines, which is an baseline model that outputs all zero edges for the adjacency matrix.

E.1.2. Results on MLP data

Results for comparisons between our model CSIvA and baselines non-linear ICP (Heinze-Deml et al., 2018b) and DAG-GNN (Yu et al., 2019) on MLP data are shown in Table 3. MLP data is non-linear and hence more challenging compared to the continuous linear data. Our model CSIvA significantly outperforms non-linear ICP and DAG-GNN. The difference becomes more apparent as the graph size grows larger and more dense.

We visualized samples that our model generated on the test data. The samples are shown in Figure 4 and Figure 5. The samples are randomly chosen, each subplot is a sample from a distinct test data. The edges in the graph are shown in 3 colors, they each represent the following: (a) Green edges indicate that our model has generated the correct edge. (b) A red edge indicates a missing edge, that is our model did not generate the edge, which exist in the groundtruth graph. (c) A blue edge indicates a redundant edge, such that our model generated an edge that does not exist in the groundtruth graph. As shown in Figure 4 and 5, our model is able to generate the correct graph almost all of the times.

E.1.3. Results on Dirichlet data.

The Dirichlet data requires setting the values of the parameter α. Hence, we run two sets of experiments on this data.

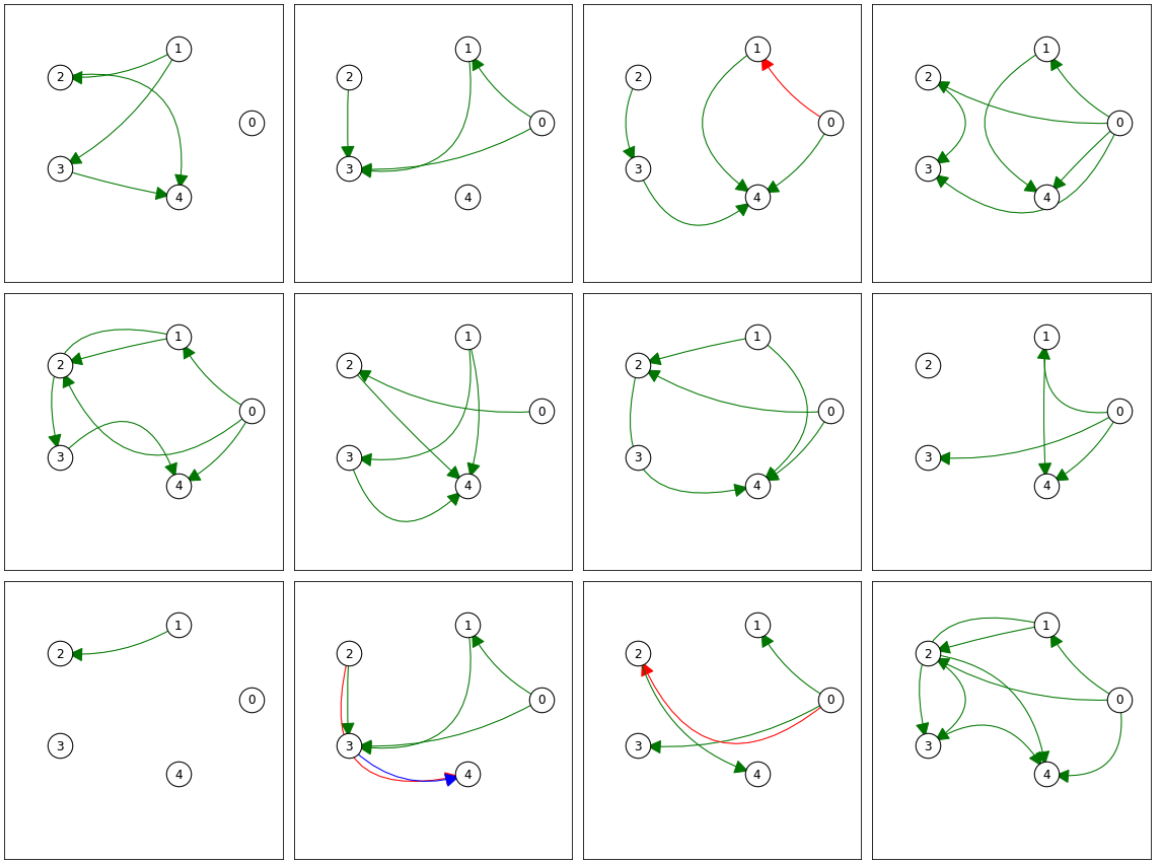

Figure 4. This figures visualizes samples that our model generated on test data. The model was trained and tested on MLP data of size 5 with **ER-**2 graphs. The samples are randomly chosen. The green edges indicate that our model has generated the correct edges; red edges indicate edges that our model had missed; and blue edges are the ones that our model generated, which were not in the groundtruth graph. As shown above, our model is able to generate the correct graph almost all of the times, while only occasionally generating 1 or 2 incorrect edges in a graph.

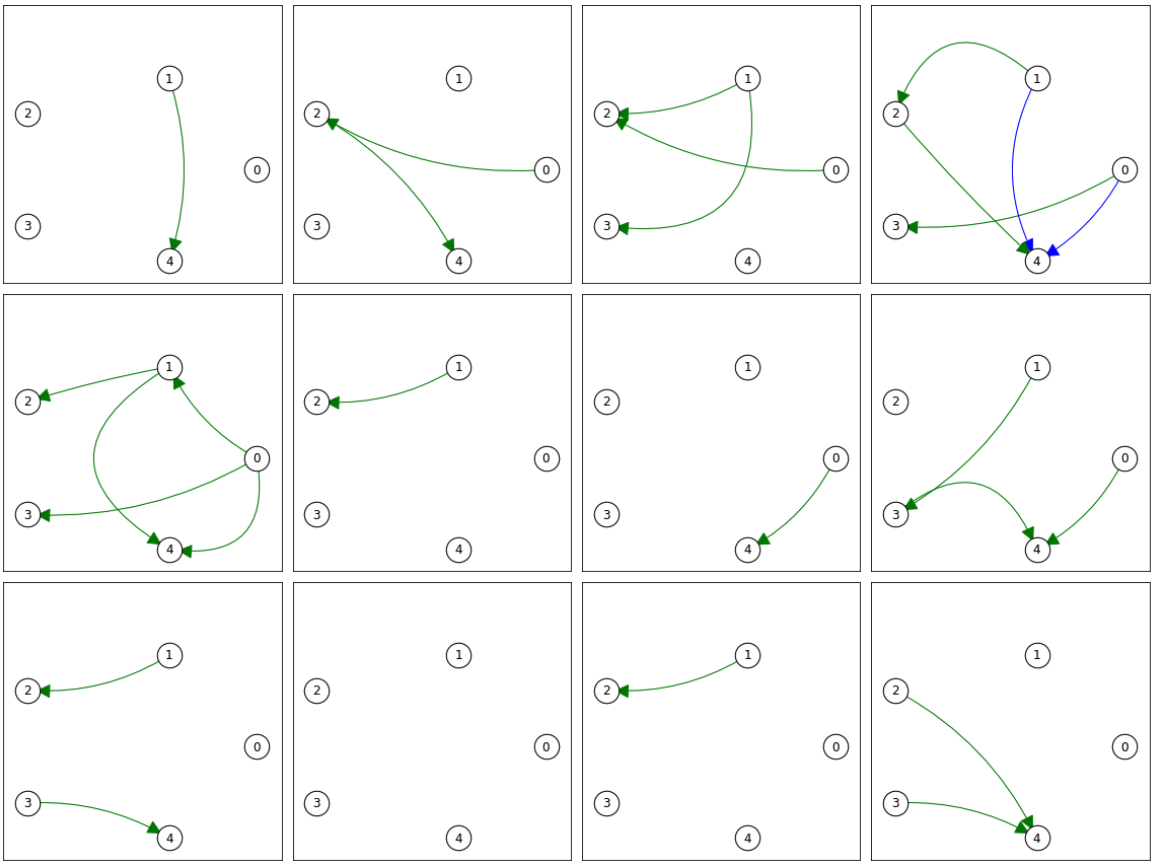

Figure 5. This figures visualizes samples that our model generated on test data. The model was trained and tested on MLP data of size 5 with **ER-**1 graphs. The samples are randomly chosen. The green edges indicate that our model has generated the correct edges; red edges indicate edges that our model had missed; and blue edges are the ones that our model generated, which were not in the groundtruth graph. As shown above, our model is able to generate the correct graph almost all of the times, while only occasionally generating 1 or 2 incorrect edges in a graph.

	ER = 1				ER = 2			
	Var = 5	**Var = 10**	**Var = 15**	**Var = 20**	**Var = 5**	**Var = 10**	**Var = 15**	**Var = 20**
Abs*	2.50	5.00	7.50	10.00	5.00	10.00	15.00	20.00
(Yu et al., 2019)	2.52	7.30	9.74	12.72	4.33	12.78	16.73	22.33
(Heinze-Deml et al., 2018b)	2.43	4.62	7.42	9.05	4.76	9.12	13.52	19.25
CSIvA	**0.98** $\pm$ 0.16	**2.25** $\pm$ 0.17	**3.38** $\pm$ 0.12	**5.92** $\pm$ 0.19	**1.51** $\pm$ 0.47	**5.12** $\pm$ 0.26	**6.82** $\pm$ 0.23	**13.50** $\pm$ 0.35

Table 3. **Results on MLP data**. Hamming distance $\mathcal{H}$ for learned and ground-truth edges on synthetic graphs, compared to other methods, averaged over 128 sampled graphs ($\pm$ standard deviation). The number of variables varies from 5 to 20, expected degree = 1 or 2, and the dimensionality of the variables are fixed to 3. We compared to the strongest baseline model that uses observational data (Yu et al., 2019) and also the strongest that uses *interventional* data (Heinze-Deml et al., 2018b). Note that for (Heinze-Deml et al., 2018b), the method required nodes to be causally ordered, and Gaussian noise $\mathcal{N}(0, 0.1)$ to be added. "Abs" baselines are All-Absent baselines, which is an baseline model that outputs all zero edges for the adjacency matrix.

In the first set of experiments, we investigated how different values of α impact learning in CSIvA. As shown in Table 5 in the appendix, CSIvA performs well on all data with $\alpha \leq 0.5$, achieving $\mathcal{H} < 2.5$ in all cases. CSIvA still performs well when $\alpha = 1.0$, achieving $\mathcal{H} < 5$ on size 10 graphs. Learning with $\alpha > 1$ is more challenging. This is not surprising, as $\alpha > 1$ tends to generate more uniform distributions, which are not informative of the causal relationship between nodes.

In the second set of experiments, we compared CSIvA to non-linear ICP and DAG-GNN. To limit the number of experiments to run, we set $\alpha = 1.0$, as this gives the least amount of prior information to CSIvA. As shown in Figure 6, our model significantly outperforms non-linear ICP and DAG-GNN. Our model achieves $\mathcal{H} < 5$ on size 10 graphs, almost half of the error rate compared to non-linear ICP and DAG-GNN, both achieving a significantly higher Hamming distance ($\mathcal{H} = 9.3$ and $\mathcal{H} = 9.5$ respectively) on larger and denser graphs. Refer to Table 5 for complete sets of results.

Results for comparisons between our model CSIvA and baselines non-linear ICP (Heinze-Deml et al., 2018b) and DAG-GNN (Yu et al., 2019) on Dirichlet data are shown in Table 4. MLP data is non-linear and hence more challenging compared to the continuous linear data. Our model CSIvA significantly outperforms non-linear ICP and DAG-GNN. The difference becomes more apparent as the graph size grows larger and more dense.

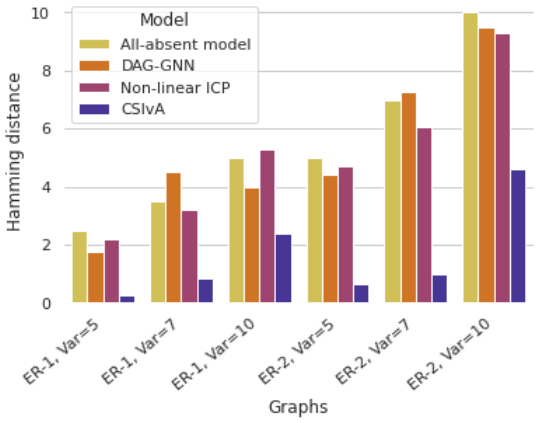

Figure 6. **Results on Dirichlet data**. Hamming distance $\mathcal{H}$ between predicted and ground-truth adjacency matrices on Dirichlet data, averaged over 128 sampled graphs.

	ER = 1			ER = 2		
	Var = 5	**Var = 7**	**Var = 10**	**Var = 5**	**Var = 7**	**Var = 10**
All-absent Model	2.5	3.5	5.0	5.0	7.0	10.0
(Yu et al., 2019)	1.75	4.5	4.0	4.5	7.25	9.50
(Heinze-Deml et al., 2018b)	2.2	3.2	5.3	4.7	6.1	9.3
CSIvA	**0.26** $\pm$ 0.05	**0.83** $\pm$ 0.06	**2.37** $\pm$ 0.07	**0.65** $\pm$ 0.05	**0.97** $\pm$ 0.06	**4.59** $\pm$ 0.08

Table 4. **Results on Dirichlet data.** Hamming distance $\mathcal{H}$ for learned and ground-truth edges on synthetic graphs, compared to other methods, averaged over 128 sampled graphs ($\pm$ standard deviation). The number of variables varies from 5 to 10, expected degree = 1 or 2, the dimensionality of the variables are fixed to 3, and the α is fixed to 1.0. We compare to the strongest causal-induction methods that uses observational data (Yu et al., 2019) and the strongest that uses *interventional* data (Heinze-Deml et al., 2018b).

We also compare how different α values of Dirichlet data impacts learning for our model. Our model performs well on all graphs where $\alpha \leq 0.5$, and the performance starts to degard as $\alpha = 1.0$. When $\alpha = 5.0$, our model is almost performing

similarly to the All-absent model (outputting all zero edges). This is to be expected, as larger alpha values is less informative of the causal relationships between variables.

	ER = 1				ER = 2			
	Var = 5	**Var = 10**	**Var = 15**	**Var = 20**	**Var = 5**	**Var = 10**	**Var = 15**	**Var = 20**
$\alpha = 0.1$	0.18± 0.03	0.72± 0.04	1.31± 0.04	2.45 ± 0.04	0.39± 0.04	1.27± 0.07	1.98± 0.12	4.09± 0.04
$\alpha = 0.25$	0.14± 0.03	0.77± 0.05	1.62± 0.05	3.51 ± 0.05	0.29± 0.04	1.27± 0.07	3.04± 0.20	6.41± 0.12
$\alpha = 0.5$	0.14± 0.04	0.94± 0.05	4.26± 0.07	7.35± 0.04	0.41 ± 0.03	2.11± 0.06	8.25± 0.07	15.54± 0.10
$\alpha = 1.0$	0.26± 0.05	2.37± 0.07	4.90± 0.05	10.10± 0.07	0.68± 0.03	4.32± 0.07	10.24± 0.07	21.81± 0.07
$\alpha = 5.0$	1.27± 0.12	4.9± 0.05	14.73± 0.11	19.49± 0.05	3.21± 0.05	9.99 ± 0.03	24.19± 0.05	37.03± 0.24
Abs*	2.5	5.0	7.5	10.0	5.0	10.0	15.0	20.0

Table 5. **Results on Dirichlet data.** Hamming distance $\mathcal{H}$ (lower is better) for learned and ground-truth edges on synthetic graphs, averaged over 128 sampled graphs. Our model accomplished a hamming distance of less than 2.5 for Dirichlet data with $\alpha <= 0.5$. "Abs" baselines are All-Absent baselines, which is an baseline model that outputs all zero edges for the adjacency matrix.

E.2. Out-of-distribution experiments

In this set of experiments, we evaluated CSIvA's ability to generalize to aspects of the data generating distribution that are often unknown, namely graph density and parameters of the CPDs, such as the α values of the Dirichlet distribution. Hence, these experiments investigate how well CSIvA generalizes when graph sparsity and alpha values for the Dirichlet distribution of the training data differ from the test data.

Varying graph density. We evaluated how well our model performs when trained and tested on CBNs with varying graph density on MLP and $\alpha = 1$ Dirichlet data. We fixed the number of nodes to $N = 7$, with variables able to take on discrete values in $\{1, 2, 3\}$. The graphs in training and test datasets can take ER degree $\in \{1, 2, 3\}$. Results are shown in Table 7 for the MLP data and Table 8 for the Dirichlet data.

For the MLP data, models trained on ER-2 graph generalizes the best. For Dirichlet data, there isn't one value of graph density that consistently generalizes best across graphs with different densities. Nevertheless, ER-2 graphs give a balanced trade-off and generalizes well across graphs with different sparsity.

Train		$\alpha = 0.1$	$\alpha = 0.25$	$\alpha = 0.5$
	$\alpha = 0.1$	0.31	0.33	0.52
Test	$\alpha = 0.25$	0.72	0.40	0.41
	$\alpha = 0.5$	1.8	0.71	0.35

Table 6. **Results on varying α values for Dirichlet data:** Hamming distance $\mathcal{H}$ between predicted and ground-truth adjacency matrices.

Varying α. We next trained and evaluated on data generated from Dirichlet distributions with $\alpha \in \{0.1, 0.25, 0.5\}$. Results for ER-1 graphs with $N = 7$ are found in Table 6. There isn't a value of α that performs consistently well across different values of α for the test data. Nevertheless, $\alpha = 0.25$ is a balanced trade-off and generalizes well across test data with $0.1 \leq \alpha \leq 0.5$.

Train		ER-1	ER-2	ER-3
	ER-1	1.2	0.9	1.3
Test	ER-2	3.3	1.8	2.1
	ER-3	5.0	2.8	2.8

Table 7. **Results on varying graph density for MLP data:** Hamming distance $\mathcal{H}$ between predicted and ground-truth adjacency matrices.

Train		ER-1	ER-2	ER-3
	ER-1	0.19	0.21	0.28
Test		0.86	0.29	0.25
	ER-3	1.61	0.60	0.23

Table 8. **Results on graph sparsity for Dirichlet data ($\alpha = 1$):** Hamming distance $\mathcal{H}$ between predicted and ground-truth adjacency matrices.

Amount of samples. We evaluated CSIvA on different amount of samples $(100, 200, 500, 1000, 1500)$ per CBNs. Results for Dirichlet data sampled from $N = 10$ graphs are shown in Figure 7. We can see that the model performance improves as it observes up to 1000 samples for ER-1 graphs, whereas having 1500 samples gives slightly better results compared to 1000 samples for ER-2 graphs.

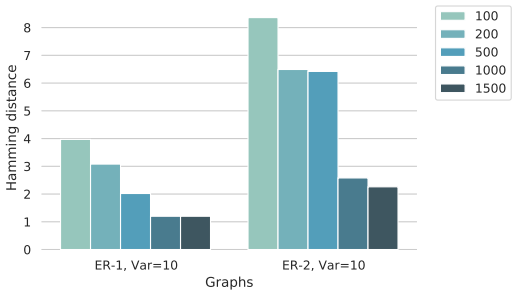

Figure 7. **Results on varying amount of samples**. Hamming distance $\mathcal{H}$ between predicted and ground-truth adjacency matrices for synthetic data. Results for CSIvA trained on Dirichlet data with $N = 10$ and $\alpha = 0.5$ with different numbers of samples per CBNs. The model performance increases as the sample size increases.

Amount of interventions Our previous experiments were all performed using a fixed amount of interventions (80%) in the training and test sets. To investigate how changing the proportion of interventional data impacts the performance of our model, we train the model with varying amounts of interventions in the training set and evaluate it using different amount of interventions during test time.

To be specific, during training, the model is trained on data with varying amount of interventions, which is randomly sampled from the uniform distribution over the set $\{0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. During test time, we evaluate the model on different amount of interventions, and we report the performance in hamming distance. We trained the model on Dirichlet data with 15 nodes and $\alpha = 0.25$. The model is trained and tested on 1000 samples per dateset. The results are found in Figure 8.

As shown in Figure 8, our model's performance is worst if it only receives observational data $(0\%$ interventions), and the performance of our model improves as the amount of interventional data increases. This is a clear indication that our model is able to extract information from interventional data for predicting the graph structure.

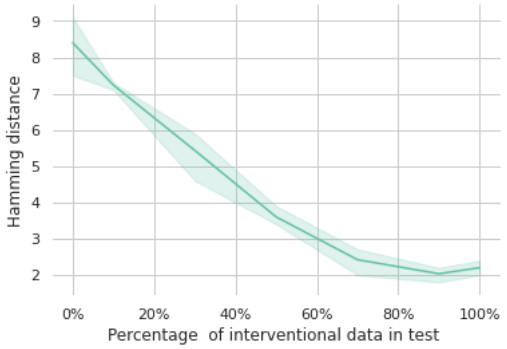

Figure 8. **Results on varying amount of interventions in data.** Hamming distance $\mathcal{H}$ for learned and ground-truth edges on synthetic Dirichlet graphs ($Var = 15$ and $\alpha = 0.25$). Pure observational data $(0\%$ interventions) performs the worst, which is to be expect, since intervention data is need for causal identifiability. The performance of our model improves as it observes more interventions, this suggest that our model is able to extract useful information from interventions in order to predict the causal structure.

F. Discussion

In this paper, we have presented a novel approach towards causal graph structure inference. Our method is based on learning from synthetic data in order to obtain a strong learning signal (in the form of explicit supervision), using a novel transformer-based architecture which directly analyzes the data and computes a distribution of candidate graphs. We demonstrated that even though only trained on synthetic data, our model generalizes to out-of-distribution.

Our method is based on transformers with self-attention, which scales quadratically with the length of the inputs, making it challenging to scale to larger graphs. However, methods such as Jaegle et al. (2021); Goyal et al. (2021b) enable transformers to scale linearly with the number of inputs (and outputs), one extension is to incorporate them into our framework. Another direction of future work would be to use the proposed framework for learning causal structure from raw visual data. This could be useful, e.g. in an RL setting in which an RL agent interacts with the environment via observing low level pixel data (Ahmed et al., 2020; Ke et al., 2021; Wang et al., 2021).

As causal inference is applied to important real-world domains, such as social, economical and medical sciences, and can serve as a basis for decision making, it is crucial to perform a thorough experimental validation of causal models. In particular, performance on finite amounts of data as well as out-of-distribution data must be carefully assessed.