Ψnet: Efficient Causal Modeling at Scale

Florian Peter Busch FLORIAN_PETER.BUSCH@TU-DARMSTADT.DE
Moritz Willig MORITZ.WILLIG@CS.TU-DARMSTADT.DE
Jonas Seng JONAS.SENG@TU-DARMSTADT.DE
Kristian Kersting KERSTING@CS.TU-DARMSTADT.DE
Computer Science Department, TU Darmstadt, Germany
Devendra Singh Dhami D.S.DHAMI@TUE.NL
Department of Mathematics & Computer Science, TU Eindhoven, Netherlands

Editors: J.H.P. Kwisthout & S. Renooij

Abstract

Being a ubiquitous aspect of human cognition, causality has made its way into modern-day machine-learning research. Despite its importance in real-world applications, contemporary research still struggles with high-dimensional causal problems. Leveraging the efficiency of probabilistic circuits, which offer tractable computation of marginal probabilities, we introduce Ψnet, a probabilistic model designed for large-scale causal inference. Ψnet is a type of sum-product network where layering and the einsum operation allow for efficient parallelization. By incorporating interventional data into the learning process, the model can learn the effects of interventions and make predictions based on the specific interventional setting. Overall, Ψnet is a causal probabilistic circuit that efficiently answers causal queries in large-scale problems. We present evaluations conducted on both synthetic data and a substantial real-world dataset, demonstrating Ψnet's ability to capture causal relationships in high-dimensional settings.

Keywords: Probabilistic Circuits; Sum-Product Networks; Causality; Tractable Inference.

1. Introduction

Causal models, in contrast to purely correlational models, provide significantly higher robustness and allow the prediction of the effect of *intervening* in a system without the need to perform the intervention in real life (Schölkopf and von Kügelgen, 2022). However, the vast majority of machine learning (ML) models still rely purely on correlations between variables to make predictions. Disregarding causality diminishes ML models' robustness in out-of-distribution scenarios and can lead them to incorrect inferences and predictions (Schölkopf and von Kügelgen, 2022; Pearl, 2009). By means of correlational training, bound to the first level of the Pearl Causal Hierarchy (PHC; Bareinboim et al. (2022)), they are unable to reason beyond correlational quantities and are unable to predict the effects of interventions or reason counterfactually. Multiple approaches have been proposed to remedy this situation (Pearl, 2019; Shi et al., 2019; Xia et al., 2021). Giving ML models access to interventional (and possibly counterfactual) information lifts them to a higher level of the PHC. Classical neural approaches are often bound to a particular set of queries or –in the case of (tractable) probabilistic models– do not extend to marginal inference (Khemakhem et al., 2021; Melnychuk et al., 2023; Javaloy et al., 2023). Switching to hybrid approaches such as neural causal models (NCM; Xia et al. (2021)) yields more expressive approximators

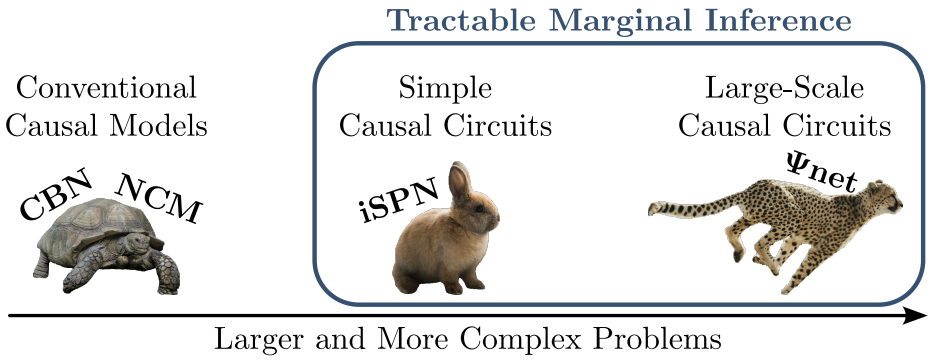

Figure 1: **Moving Causal Models into Large-Scale Domains.** Conventional models struggle with application in larger domains, partly but not exclusively due to non-tractable inference of marginal questions. We introduce Ψnet, a causal probabilistic circuit equipped with optimizations directed at efficiency that allows for quick and large-scale evaluation of causal questions, surpassing simpler causal circuits such as iSPNs.

but still suffers from the same #P-hard time complexity (Eiter and Lukasiewicz, 2002) for exact inference as Bayesian Networks (BN; Pearl (1985, 1995)). Thus, developing efficient algorithms for large data with complex causal structures is an important endeavor.

To mitigate the problem of intractable exact causal inference, recent works (Zečević et al., 2021; Zečević et al., 2023; Cruz and Batista, 2024) have utilized probabilistic circuits that provide exact and tractable inference and marginalization (Choi et al., 2020; Vergari et al., 2021). In this work, we introduce a novel model at the intersection of causality and probabilistic circuits, allowing us to scale causal inference beyond the usual employed toy examples. To this end, we equip einsum networks (Peharz et al., 2020) with the ability to answer interventional queries akin to Zečević et al. (2021). Our model, Ψnet (causal EiNet → Ψnet), leverages the einsum operator to scale exact causal (marginal) inference to more than a thousand variables on a real-world dataset. Fig. 1 shows the motivation for Ψnet.

We demonstrate the abilities of Ψnet on a large synthetic dataset and, more importantly, on the large-scale, real-world "CausalBench" datasets (Zhou et al., 2024). While other models suffer excessive runtime explosions or can only perform approximate inference, Ψnet remains competitive to alternative models in terms of performance while staying in linear time complexity, thus increasing the speed of computing queries by orders of magnitude for high-dimensional problems. Overall, we make the following important contributions:

- We introduce Ψnets, powerful causal models capable of large-scale causal inference.
- We build upon CausalBench, a real-world, large-scale benchmark, to introduce an evaluation for marginal inference of interventional queries.
- We show that Ψnet performs competitively to other models while being much faster.

We proceed as follows: after introducing the required background and notations, we present the construction and learning of our Ψnet model. We then go over our extensive evaluations and present the related work before concluding. We make our code publicly available: https://github.com/olfub/psinet.

2. Background and Notation

Before introducing the Ψnet model, we briefly revisit causal models and probabilistic circuits. We denote random variables by upper-case letters V, sets of random variables in boldface $\mathbf{V}$, values of single and sets of random variables as v and $\mathbf{v}$ respectively, and probabilities of random variables as $P(v)$ or $P(\mathbf{v})$.

2.1. Causal Models

Different approaches to describe causal relationships among a set of variables have been proposed (Rubin, 1974; Pearl, 2009). One outstanding and widely adapted framework for causal modeling is Pearl's formalism of structural causal models (SCM). Due to its wide application, we adapt the notion of SCMs.

Definition 1 (Structural Causal Model) *A structural causal model (SCM) is a tuple* $\mathcal{M} \coloneqq \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P_{\mathbf{U}} \rangle$ *over a set of variables* $\mathbf{X} = \{X_1, \ldots, X_K\}$ *taking values in* $\boldsymbol{\mathcal{X}} = \otimes_{k \in \{1 \ldots K\}} \mathcal{X}_k$ *subject to a strict partial order* $<_{\mathbf{X}}$*, where*

- $\mathbf{V} = \{X_1, \ldots, X_N\} \subseteq \mathbf{X}, N \leq K$ *is the set of endogenous variables,*
- $\mathbf{U} = \mathbf{X} \setminus \mathbf{V} = \{X_{N+1}, \ldots, X_K\}$ *is the set of exogenous variables,*
- $\mathbf{F} = \{f_1, \ldots, f_N\}$ *is the set of deterministic structural equations, i. e.* $V_i \coloneqq f_i(\mathbf{X}')$ *for* $V_i \in \mathbf{V}$ *and* $\mathbf{X}' \subseteq \{X_j \in \mathbf{X} | X_j <_{\mathbf{X}} V_i\}$*,*
- $P_{\mathbf{U}}$ *is the probability distribution over the exogenous variables* $\mathbf{U}$*.*

The set of structural equations $\mathbf{F}$ induces a directed graph $G(\mathcal{M})$ which by definition is acyclic due to $<_{\mathbf{X}}$. The exogenous variables $\mathbf{U}$ are unobserved and treated as noise. Thus, $\mathbf{U}$ is commonly assumed to follow a distribution of mutually independent noise terms. The noise distributions in conjunction with the structural assignments in $\mathbf{F}$ induce the probability distribution $P_{\mathbf{V}}^{\mathcal{M}}$ over the set of endogenous variables $\mathbf{V}$.

Since the structural equations have a causal interpretation in SCMs, interventions can be made that result in a new SCM. In general, an intervention on a variable $V_i \in \mathbf{V}$ replaces the structural equation $f_i \in \mathbf{F}$ determining the value of V_i by some other function f_i^*. Replacing a function f_i with another function f_i^* changes the induced joint probability distribution over the set of endogenous variables, often referred to as the interventional distribution. Commonly, one distinguishes between *soft interventions* and *perfect interventions*. While soft interventions describe a non-deterministic intervention (e.g., replacing a square function with a sinus function), perfect interventions set the intervened variable to a constant value (Bongers et al., 2021). Perfect interventions are commonly written in terms of the *do*-operator (Pearl, 2009) as $do(V_i = v_i)$, i.e., V_i is set to v_i. Since an intervention $do(V_i = v_i)$ removes all dependencies between V_i and its parents, it induces an SCM $\mathcal{M}_{do(V_i = v_i)}$ with a different associated causal graph $G(\mathcal{M}_{do(V_i = v_i)})$. The experimental section considers problems with both perfect and soft interventions.

2.2. The Challenge of Tractable Inference

Given a fully defined SCM, computing probabilities for *full evidence queries* of the form $P(V_1 = v_1, V_2 = v_2, \ldots, V_n = v_n)$ is straightforward and can be done by following the factorization of the induced distribution. Since the factorization can be read off the directed acyclic graph (DAG) induced by the SCM, models closely resembling the SCM structure such as Causal Bayesian Networks (CBNs) (Heckerman et al., 1995; Pearl, 1995; Neapolitan et al., 2004; Acharya et al., 2018) are a natural choice here.

However, in many situations, only a subset of variables is of interest. For example, a physician might want to know the probability of a patient having a disease, regardless of the realization of certain (maybe even unobserved) variables. In such cases, one is interested in the *marginal probability* of a variable. In contrast to complete evidence queries, exact marginalization in (C)BNs and SCMs has exponential complexity in general (Park and Darwiche, 2004). Besides marginals, computing *conditional probabilities* plays a crucial role in real-world applications as they allow us to make predictions even if only partial evidence is available. From Bayes rule stating that $P(V_1 = v_1 | V_2 = v_2) = \frac{p(V_1=v_1, V_2=v_2)}{P(V_2=v_2)}$, it follows that computing the exact *conditional probability* is also not tractable in general for (C)BNs and SCMs as it involves computing a marginal distribution.

Given that marginal and conditional inference over (interventional) distributions is intractable in CBNs and SCMs, we aim for an alternative representation of (interventional) distributions that allows tractable inference. Probabilistic circuits enable tractable computation of marginals and conditionals in purely probabilistic (i.e., non-causal) settings. Thus, besides the probabilistic setting, we aim to utilize PCs to perform tractable marginal and conditional inference in large-scale causal problems as well.

2.3. Causal Probabilistic Circuits

A probabilistic circuit (PC) (Choi et al., 2020; Peharz et al., 2020) is a computational graph encoding a distribution over a set of random variables $\mathbf{X}$. It is defined as a tuple $(\mathcal{G}, \phi)$ where $\mathcal{G} = (V, E)$ is a rooted, directed acyclic graph and $\phi : V \to 2^{\mathbf{X}}$ is the *scope* function assigning a subset of random variables to each node in $\mathcal{G}$. For each internal node N of $\mathcal{G}$, the scope is defined as the union of scopes of its children, i.e., $\phi(\mathsf{N}) = \cup_{\mathsf{N}' \in \mathrm{ch}(\mathsf{N})} \phi(\mathsf{N}')$. Each leaf node L computes a distribution/density over its scope $\phi(\mathsf{L})$. All internal nodes are either sum nodes S or product nodes P where each sum node computes a convex combination of its children, i.e., $\mathsf{S} = \sum_{\mathsf{N} \in \mathrm{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}} \mathsf{N}$, and each product computes a product of its children, i.e., $\mathsf{P} = \prod_{\mathsf{N} \in \mathrm{ch}(\mathsf{P})} \mathsf{N}$. For a more thorough introduction to PCs, refer to Choi et al. (2020).

This work considers Sum-Product Networks (SPNs). SPNs are *decomposable* and *smooth* PCs (Poon and Domingos, 2011; Peharz et al., 2020). A PC is decomposable if for each product node $\mathsf{P} \in V$ it holds that $\phi(\mathsf{N}) \cap \phi(\mathsf{N}') = \emptyset$ for $\mathsf{N}, \mathsf{N}' \in \mathrm{ch}(\mathsf{P})$, $\mathsf{N} \neq \mathsf{N}'$, i.e., the children of a product node P have non-overlapping scopes. Decomposability allows marginalization in linear time of the circuit size (Peharz et al., 2015). A PC is said to be smooth if for each sum node $\mathsf{S} \in V$ it holds that $\phi(\mathsf{N}) = \phi(\mathsf{N}')$ for $\mathsf{N}, \mathsf{N}' \in \mathrm{ch}(\mathsf{S})$, i.e., all children of a sum node S have the same scope. While smoothness has no computational implications, it ensures that an SPN represents a valid probability distribution.

While SPNs are purely probabilistic models, let us now turn to the intersection of causality and probabilistic circuits to achieve tractable inference for causal queries (Cruz

and Batista, 2024). Previous works introduced methods to transform SCMs and CBNs to PCs (Darwiche, 2022; Papantonis and Belle, 2020). However, information can be lost in the transformation from CBNs to PCs, and the transformation from SCMs to PCs requires variable elimination (Zhang et al., 1994), which is generally inefficient (Chavira and Darwiche, 2007). A model closer to our approach is the *interventional Sum-Product Network* (iSPN, Zečević et al. (2021)). An iSPN is a joint model of a SPN and a neural network. The neural network uses the (mutilated) causal graph as its input to predict SPN weights. The so-parameterized SPN can then be queried in the same manner as conventional SPNs. By conditioning the weights on the causal graph, the SPN is able to learn the difference between different types of interventions and can thus be used to compute causal queries.

Although iSPNs take an important step towards efficient causal inference, they employ a standard SPN structure, which is still a bottleneck for large datasets. The inference algorithm of SPNs has to traverse all nodes which prohibits scaling iSPNs to datasets with thousands of variables (as we will show in our experimental evaluation). Therefore, we aim to leverage the benefits of the einsum architecture (Peharz et al., 2020) to propose Ψnet, a causal circuit for large-scale causal inference.

3. Ψnet: A Large-Scale Causal Model

Approximating causal quantities remains a challenging task, especially in problems consisting of a large number of variables. As the typically employed causal models –such as CBNs and SCMs and derived models– suffer from #P-hard complexity, we investigate and discuss the practical challenges of scaling up models of the more efficient class of PCs to large numbers of variables in causal settings. Scaling beyond scenarios with hundreds of variables also brings classical PCs to their practical limits and requires us to select particularly efficient models that still obtain results within practically feasible run times.

Einsum networks (Peharz et al., 2020). A promising class of candidate models is *Einsum networks* (in short *EiNets*; Peharz et al. (2020)), which represent SPNs by a set of layers building on top of each other, where succeeding layers make use of the einsum operation to aggregate the leaf probabilities towards the next layers efficiently. More precisely, the leaf densities are represented using exponential families and form the input layer. The succeeding layers are so-called *einsum layers* that contain both sum and product nodes as used in conventional SPNs. Say that K is the number of densities computed for each sum node, and L is the number of sum nodes in a layer. Then, the k^{th} value of the l^{th} sum node with a single product node as a child can be calculated as $\mathbf{S}_{lk} = \mathbf{W}_{lkij}\mathbf{N}_{li}\mathbf{N}'_{lj}$, where $\mathbf{N}$ and $\mathbf{N}'$ are $L \times K$ matrices, representing the left and right children of the product nodes, and $\mathbf{W}$ is a weight tensor with $L \times K \times K \times K$ dimensions. This computation can be performed using the efficient einsum operation. Given such sum nodes with only one child each, another sum node can be used on top, thereby combining the outputs of multiple product nodes with sum nodes in two steps. The computation in this so-called *mixing layer* can also be conducted using an einsum operation. The use of the efficient einsum operation, the layer-parallelism, together with a large number of repetitions (i.e., multiple sum nodes with the same scope), results in a highly efficient large-scale SPN.

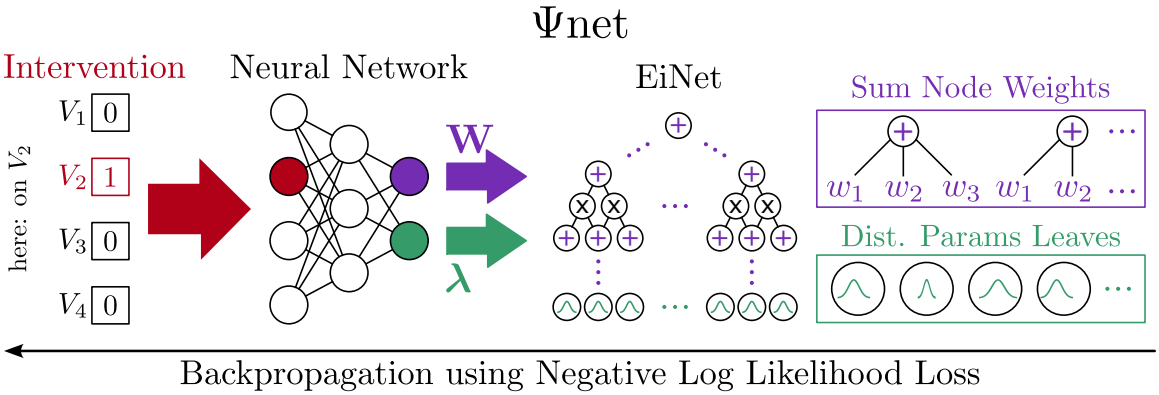

Figure 2: **Architecture of Ψnet.** Given a causal setting (i.e., a specific intervention), a neural network sets all the parameters of an EiNet model (a probabilistic circuit utilizing the einsum operation). The so-parameterized EiNet can be queried to answer causal queries in a tractable manner.

Construction of Ψnet. In this paper, we introduce an extension of EiNets to the causal domain. To be precise, the Ψnet model consists of a neural network and an EiNet component. The EiNet is a regular EiNet, parameterized by the sum weights $\mathbf{W}$ and the leaf distributions $\mathcal{N}(\cdot|\boldsymbol{\lambda})$.[1] Given the probabilities in the leaves, probabilities of queries can be calculated by making a pass through the EiNet einsum and mixing layers. Since product nodes are not parametrized, the EiNet parameters are given by the sum node weights and the parameters of the leaf distributions, i.e., as $\psi = \{\mathbf{W}, \boldsymbol{\lambda}\}$. Ψnet, instead of learning fixed ψ directly, makes use of a neural network component f_θ parametrized by θ to set these weights depending on the causal setting $\mathcal{C}$. The causal setting should consist of information describing how the SCM of the interventional setting differs from the observational SCM (thus, the difference can also be 0). We arrive at the following definition:

Definition 2 (Ψnet) *A Ψnet is a tuple $\Psi = (g_\psi, f_\theta)$, where g is an EiNet parameterized by $\psi = \{\mathbf{W}, \boldsymbol{\lambda}\}$, such that $\mathbf{W}$ and $\boldsymbol{\lambda}$ are the sum and leaf node parameters of g, respectively. Parameters ψ are assigned via a universal function approximator f, such that $f : \mathcal{C} \to \psi$, where $\mathcal{C}$ is the intervention information of the causal setting.*

The Ψnet model is shown in Figure 2. There are two parts to computing a causal query. Firstly, the causal setting $\mathcal{C}$ needs to be specified. We deliberately opted for a general definition as different types of causal settings can be used depending on the application. Generally, the causal setting contains all the causal elements in a query, such as the intervention considered. For example, if we allow interventions on any variable of our set of variables $\mathbf{X}$, we require $|\mathbf{X}|$ input variables to indicate which variables are intervened upon. If there is only one kind of intervention that is considered for each variable (such is the case in the CausalBench experiment, see Section 4.2), this input suffices to describe the causal setting, and we have $\mathcal{C} \in \mathbb{R}^{|\mathbf{X}|}$. If it should be specified to which value an intervention changes a variable and only interventions on a single variable are considered at

1. We use Gaussian distributions in the leaf nodes as computation is simple and as mixture of Gaussian distributions are very expressive. Other distributions could just as easily be used.

a time, our causal setting requires an additional value containing the intervention value $\mathcal{C} \in \mathbb{R}^{|\mathbf{X}|+1}$. For multiple interventions, it can be expanded even further to specify the new values for all intervened variables $\mathcal{C} \in \mathbb{R}^{2|\mathbf{X}|}$. One type of causal setting is also used in the iSPN paper (Zečević et al., 2021) where the (mutilated) causal graph is input into the neural network. In this case, the neural network part of the iSPN falls in line with the Ψnet definition, however iSPN does not use an EiNet for computing probabilities.

Given a causal setting, the neural network f then takes such a setting as input (e.g., observational, $do(V_2)$, ...) and outputs the EiNet parameters $\mathbf{W}$ and $\boldsymbol{\lambda}$. Note that for conventional SPNs/EiNets, these parameters are only changed during training but remain fixed once a model is trained. By conditioning them on the causal setting, we introduce variability in the model such that parameters and, thereby, probability distributions can vary in different interventional settings. Also note that once a causal setting has been passed through the neural network and the EiNet parameters are set, the resulting EiNet is a conventional EiNet model and can be queried as such. For example, the queries $P(\mathbf{X} = \mathbf{x}|do(X_1 = 1))$, $P(X_2 = 0|do(X_1 = 1), X_3 = 0)$, $\arg\max_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}|do(X_1 = 1))$ can all be computed with only one pass of $do(X_1 = 1)$ through the neural network. The neural network part needs only be executed again when queries for a different causal setting should be computed.

Training Ψnet. Ψnets can be trained end-to-end using a standard negative log-likelihood loss (akin to training iSPNs or conditional SPNs (Shao et al., 2020)). In contrast to traditional SPNs, we do not update the SPN parameters $\psi = \{\mathbf{W}, \boldsymbol{\lambda}\}$ during training but instead continue backpropagating through the neural network and update its weights θ. For each causal setting $\mathcal{C}$, the SPN parameters ψ are set by the neural network f_θ and, therefore, only the parameters θ of this network need to be updated and learned. The neural network takes interventional data as its inputs to make the EiNet approximate the interventional distribution. In order to be able to correlate interventional signals of the causal setting $\mathcal{C}$ with the required change in EiNet parameters ψ, we require interventional data to be present during Ψnet training.

Adapting to large-scale problems. A particular challenge of our paper lies in the practical application of Ψnets to real-world problems. When working with Ψnets, there are a large number of hyperparameters to deal with. Choosing bad ones can hinder successful learning, which can become a particularly prominent problem, given the large-scale nature of our datasets. As we will highlight in our experimental section on CausalBench, simply optimizing for training loss can also considerably hinder the model's performance. For instance, using deep neural networks improves training loss but leads to overfitting, resulting in reduced evaluation performance. Choosing shallow, single-layer neural networks combined with a large depth of the EiNet component proved to be the most important factor in obtaining the best results for our experiments.

4. Experiments

In order to properly illustrate the capabilities of the Ψnet model, we start by generating large synthetic datasets and investigating the model performance. We then follow up with an extensive evaluation on a large-scale real-world causal benchmark called *CausalBench*

(Zhou et al., 2024). We support our evaluation using two baseline models, a conventional iSPN (Zečević et al., 2021) and a neural causal model (NCM) (Xia et al., 2021).

4.1. Baseline Models

Ψnet can answer full evidence, marginal, and conditional queries in interventional and observational settings. In our experimental evaluation, we compare Ψnet with iSPNs (Zečević et al., 2021) and neural causal models (NCMs) (Xia et al., 2021). Since Ψnet and iSPNs follow a similar approach, they are directly comparable. In contrast, NCMs require some adaptations to serve as a proper baseline which we discuss below. Another alternative approach to learning (interventional) distributions is interventional normalizing flows (Melnychuk et al., 2023), where a dedicated model is trained for each intervention-target variable pair available. However, as this approach fundamentally differs from ours, a proper comparison to our approach is not trivial. Thus, we do not include it in our evaluation.

The Neural Causal Model: A good Baseline? NCMs train a model given a causal graph and observational data. For each variable, a neural network is trained on its parents and the applicable exogenous variables. In order to allow for a fair comparison between NCM and Ψnets, we had to extend NCMs in multiple aspects. We support training NCMs on interventional data by adjusting the loss function to not include any loss on the intervened variable. We add continuous leaves using normalizing flows instead of standard neural networks as function approximators. Lastly, one experiment in CausalBench (more information in Subsection 4.2) uses interventions on a variable where the precise effect of the intervention is unknown. Here, the distributions of the *soft interventions* themselves need to be learned as well. NCMs do not allow for this directly. Thus, we introduce binary instrumental variables for all variables. Each is connected to cause one regular variable and indicates the presence/absence of an intervention. Despite changing the model in these ways, the NCM still has significant downsides. Firstly, it does not support the direct computation of marginal probabilities. Instead, marginalization in NCMs is naïvly performed by first sampling and then averaging values of a large number of samples. Secondly, a causal graph is required to train the actual NCM. Learning a robust causal graph for high-dimensional real-world problems is known to be a particularly challenging task (Kitson et al., 2023).

4.2. CausalBench - An Extended Benchmark

CausalBench (Zhou et al., 2024) was introduced as a benchmarking tool for causal discovery algorithms. It consists of two datasets on biological cell activations (continuous variables) and includes both observational and interventional data. Within the data, interventions indicate a *knockdown* of a particular gene. The causal effects of individual knockdowns can vary in strength. Hence, we regard interventional data as data resulting from *soft* interventions. In the original paper, the authors compare different methods to discover (causal) graphs using multiple evaluations. These evaluations include biological evaluations, where the graphs are evaluated based on known gene interactions. In the statistical evaluation, any potential path $A \to \cdots \to B$ from variable A to B in a causal graph is evaluated by comparing the marginal interventional distribution $P(B|do(A))$ of the effect variable with the marginal observational distribution $P(B)$ of the effect variable. If these distributions significantly differ, it is assumed that a directed path from A to B exists.

We argue that this should be taken one step further and, instead of merely detecting *that* the observational and interventional distributions differ, we consider *how* the distributions differ. In our evaluation, we investigate the effect of interventions on each variable by considering their marginal probability distributions. For any variables A and B, we consider the interventional (observational) marginal distribution $P(B|do(A))$ (observational: $P(B)$) predicted by a trained model, compute its most probable explanation (MPE) and compare it to the interventional (observational) mean of the test data. The MPE is defined as $\text{argmax}_B P(B|do(A))$ in the interventional case and $\text{argmax}_B P(B)$ in the observational case. Since we parameterize the leaves of our Ψnet and iSPN with Gaussian distributions, the MPE corresponds to the mean in the leaves. Note that this comparison reflects how well a model predicts the actual causal effect of A on B instead of merely predicting whether a connection exists at all. By breaking the comparison down to single values per pair of variables A and B, we enable a full evaluation on all pairs (i.e., over a million comparisons).

4.3. Setup

In the following, we briefly introduce the experimental setup for this paper. We include further information on hyperparameters and technical details in the appendix.

Synthetic Problems. To conduct an experiment under controlled conditions, we randomly generate causal graphs consisting of a fixed number of nodes and edges. We then sample 10,000 data points from the unintervened 'observational' case and a further 10,000 interventional samples for each variable. We run experiments over 5 seeds and evaluate the following configurations of nodes/edges: 5/5, 10/12, 15/20, 20/30, 50/100, 100/250. Model performance is recorded after 0.5, 1, 2, 4, and 5 hours of training. We train the NCM models using the ground-truth causal graph.

CausalBench. For our large-scale real-world evaluation, we use the train and test sets of the K562 and RPE1 datasets provided within the CausalBench repository.[2] All models are trained on observational and interventional data and with a time budget of 8 hours for training. We conducted a preliminary hyperparameter search for the Ψnet model to arrive at an optimized Ψnet model. For the NCM, we use binary instrumental leaves to allow for soft interventions and implement all continuous variables using normalizing flows. In our evaluation, we compare the ground truth marginal probabilities of observational and interventional distributions with the MPE predictions of Ψnet and iSPN. For NCM we compare with the naïve marginal probabilities obtained from sampling. Just like for the ground truth, the NCM calculates the mean over samples for the evaluation.

4.4. Synthetic Experiments

Figure 3 shows the errors between model prediction and ground truth. The error is shown for each node/edge pair and after different points in time. We see that Ψnet and iSPN outperform NCM, especially on smaller problems where NCM performs worse than on larger problems. This might be due to the problem creation, where the smaller problems are more likely to include variables with a higher number of inputs (causes), which might be more difficult to learn for the NCM since the functions that need to be learned are more

2. https://github.com/causalbench/causalbench

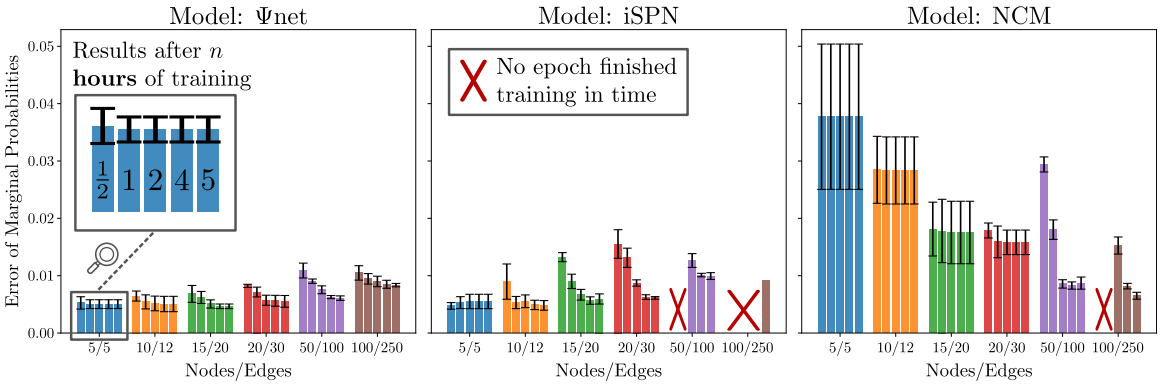

Figure 3: **Average Error on Synthetic Data.** On the y-axis, the average error between the true marginal distribution and the model prediction is plotted. Each x-axis consists of 6 sets of bars, indicating the number of nodes/edges the model has been trained on. Each adjacent block of bars in the same color for the same node/edge pair has results for different times of training. Ψnet and iSPN outperform NCM and Ψnet appears to converge towards better results more quickly than iSPN for larger problems (best viewed in color).

complex. While both Ψnet and iSPN perform well, we can see that Ψnet starts off better than iSPN on the larger problems. For 50 and 100 nodes, the iSPN failed to finish a single training epoch within the first time thresholds. Here, only a single seed managed to finish one epoch after 5 hours for the 100/250 problem (as the time for finishing one epoch of this problem size is almost exactly 5 hours)[3]. This indicates that the iSPN is as expressive as the Ψnet, however learning is much more difficult and time consuming.

This is emphasized by the inference times shown in Table 1. For Ψnet and iSPN, we looked at all marginal probabilities for all possible interventions (plus one observational setting), i.e., for n variables, we run $(n+1)n$ queries. This is the time recorded in the table. For NCM, we only run $n+1$ queries as all marginal probabilities here are based on samples that are identical within an interventional setting. To allow for a fair comparison, we multiplied the NCM time by the number of variables involved (the resulting values are entered in the NCM* row in the table). Thus, the relation between inference time represents the relation between the speed of inference. Ψnet is much faster than iSPN, improving upon the iSPN by orders of magnitude for the larger problems. Having adjusted for single queries, we see that NCM is far slower than both causal circuits.

4.5. CausalBench

The CausalBench benchmark, composed of two datasets with 1158 and 651 variables, is a large-scale, real-world benchmark with true interventional data. Contrary to assumptions often made in theoretical causality research, the interventions are not perfect but instead soft interventions; representing a knockdown of the intervened gene. Learning a causal graph or model of a problem of this size is considerably difficult with current methods.

3. If all 5 seeds would be considered nevertheless, mean and variance of this configuration would compute as 0.0088 ± 0.00065.

graph	5/5	10/12	15/20	20/30	50/100	100/250
N/E	*mean±std*	*mean±std*	*mean±std*	*mean±std*	*mean±std*	*mean±std*
Ψnet	0.39 ± 0.00	1.91 ± 0.08	4.08 ± 0.03	8.68 ± 0.08	61.24 ± 0.64	278.00 ± 2.28
iSPN	0.59 ± 0.01	4.56 ± 0.05	15.24 ± 0.16	35.78 ± 0.45	565.68 ± 9.96	4580.71 ± 26.46
NCM*	1.48 ± 0.12	12.06 ± 0.44	40.96 ± 1.89	98.84 ± 2.15	1579.39 ± 42.11	12820.70 ± 70.23

Table 1: **Inference Times (Seconds).** Time taken to compute the marginal probability for each variable under any intervention. Ψnet is clearly much faster than the other models. The sample-based approach by NCM is by far the slowest since it has to traverse an entire causal graph. Note that the values specified for NCM* illustrate the projected times when computing the same amount of queries as the other models (more information in the text).

	K562				**RPE1**			
	Ψnet	Ψnet*	iSPN	NCM	Ψnet	Ψnet*	iSPN	NCM
MSE	0.0942	0.0791	0.0615	1.0165	0.1131	0.0617	0.0627	0.1438
Time (s)	548.36	534.67	2595.26	48778.51	168.54	165.35	907.19	15325.26

Table 2: **CausalBench Results.** The mean squared error shows the average error when predicting the most likely value. The time to conduct this evaluation is shown in the second row. Despite the Ψnet configuration that was optimized for loss achieving the best loss values, we see a simpler neural network architecture (Ψnet*), more similar to the iSPN, performing well. Both variants of Ψnet are significantly faster than the competing methods.

For the larger K562 dataset, we conducted an extensive hyperparameter search to optimize our Ψnet performance. We applied the same parameters to the RPE1 dataset. The results of these experiments are shown in Table 2. The so-trained Ψnets (no star) perform worse than the iSPN models in the evaluation. For this reason, we also trained additional Ψnet models, which are closer to the iSPN architecture. To reduce overfitting, we increased the batch size to 64 and used only a single hidden layer with a small number of neurons instead of the larger neural network architecture employed before. The new Ψnet* models reach roughly the same error margins as the iSPN while still being considerably faster.

A comparison of the errors reveals several things. For one, optimizing for loss does not appear to be sufficient. Instead, a validation set should be used to assess the model performance on the metrics of interest. The iSPN error is on the same level as Ψnet* while the NCM performs much worse, especially in the K562 dataset. While better results when training for a much longer amount of time are plausible, the additional overhead of the graph discovery part required for the NCM indicates that much more time would be needed, which is a crucial disadvantage when considering even larger problems.

Most importantly, we observe strong differences in the inference time of our models. Here, Ψnet is far superior to both competing models, beating especially NCM by orders of magnitude. Therefore, while Ψnet does not provide the smallest errors here, its errors are still comparable to that of the best method while being much faster to run. In applications where inference speed is crucial, the small trade-off in accuracy for the significant speed-up in inference time is acceptable.

5. Related Work

Arithmetic circuits (Darwiche, 2003) in general are a precursor to SPNs and have also been considered for a preliminary approach to causal inference (Darwiche, 2022). There are also other tractable probabilistic models next to SPNs (Kisa et al., 2014; Van den Broeck et al., 2019). While Causal Bayesian Networks are powerful and can be transformed into SPNs and back (Zhao et al., 2015), this transformation from SPNs generally leads to degenerate[4] Bayesian Networks incapable of subsequent causal inference (Papantonis and Belle, 2020), seemingly contradicting the possibility of causal SPNs. This seeming contradiction is addressed in Zečević et al. (2021) in which a model class extension for SPNs is introduced that basically conditions the model on the do-operator. More recently, Papantonis and Belle (2023) introduced an algorithm to transform SPNs into Bayesian Networks which simplifies the calculation of interventional queries. Considering not only interventional but also counterfactual questions, Han et al. (2023) highlighted that calculating counterfactuals using circuits is not any more complex than interventional or observational questions and Huber et al. (2023) consider counterfactuals in circuits by investigating partial identifiability. A compositional perspective on inference in probabilistic circuits allows inferring interventional distributions (Wang and Kwiatkowska, 2023). Recently, normalizing flows have also been used for causality (Khemakhem et al., 2021; Javaloy et al., 2023; Melnychuk et al., 2023) but lack the capability to perform marginal inference.

Shao et al. (2020) introduced *conditional SPN* where a NN is used to set the SPN parameters depending on some set of conditional variables that are input in the NN. Building on top of this work, Zečević et al. (2021) introduced *interventional SPN* (iSPNs) which use the same idea of a combined NN and SPN architecture to allow for interventional queries and thus inspired this paper. Recently, SPN's were combined with characteristic circuits (Yu et al., 2023), to develop the first causal model for mixed domain consisting of heterogeneous data (Poonia et al., 2024) thereby highlighting the significance of this research direction.

6. Conclusion

We introduce Ψnet, a causal probabilistic circuit capable of tackling tractable marginal inference in high-dimensional real-world settings. Our experimental evaluation demonstrated Ψnets competitive performance on synthetic data, and, –using a new evaluation on the large-scale real-world CausalBench dataset– competitive results while being *significantly faster* than competing methods. When scaling up to even larger problems or when short inference time is crucial, Ψnet might be the only existing model able to tackle such problems.

Future work involves improving the model performance even further and making finding the right model parameters easier. An important bottleneck of applying causal models on a large scale remains to be a lack of real-world datasets next to CausalBench.

Limitations. In this paper, we only considered interventions on single variables. Intervening on multiple variables at the same time can follow the same procedure. As our experiments on CausalBench showed, it is not sufficient to transfer hyperparameters from one well-performing dataset (K562) to another (RPE1) in the hope of preserving high accuracy. We hope that future work can make Ψnets more robust to the choice of hyper-parameters.

4. A bipartite graph in which the actual variables of interest are not connected is called degenerate.

Acknowledgments

This work is supported by the Hessian Ministry of Higher Education, Research, Science and the Arts (HMWK; "The Third Wave of AI"). Further, this work benefited from the National High-Performance Computing project for Computational Engineering Sciences (NHR4CES). The authors acknowledge the support of the German Science Foundation (DFG) project "Causality, Argumentation, and Machine Learning" (CAML2, KE 1686/3-2) of the SPP 1999 "Robust Argumentation Machines" (RATIO). This work was partly funded by the Collaboration Lab "AI in Construction" (AICO) of the TU Darmstadt and HOCHTIEF. The Eindhoven University of Technology authors received support from their Department of Mathematics and Computer Science and the Eindhoven Artificial Intelligence Systems Institute.

References

J. Acharya, A. Bhattacharyya, C. Daskalakis, and S. Kandasamy. Learning and testing causal models with interventions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

E. Bareinboim, J. D. Correa, D. Ibeling, and T. Icard. On pearl's hierarchy and the foundations of causal inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*. 2022.

S. Bongers, P. Forré, J. Peters, and J. M. Mooij. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 2021.

M. Chavira and A. Darwiche. Compiling bayesian networks using variable elimination. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

Y. Choi, A. Vergari, and G. Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. 2020. URL http://starai.cs.ucla.edu/papers/ProbCirc20.pdf.

D. Cruz and J. Batista. Causality and tractable probabilistic models. *Frontiers in Computer Science*, 2024.

A. Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM (JACM)*, 2003.

A. Darwiche. Causal inference using tractable circuits. *arXiv preprint arXiv:2202.02891*, 2022.

T. Eiter and T. Lukasiewicz. Complexity results for structure-based causality. *Journal of Artificial Intelligence (AIJ)*, 2002.

Y. Han, Y. Chen, and A. Darwiche. On the complexity of counterfactual reasoning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 1995.

D. Huber, Y. Chen, A. Antonucci, A. Darwiche, and M. Zaffalon. Tractable bounding of counterfactual queries by knowledge compilation. In *The 6th Workshop on Tractable Probabilistic Modeling @ UAI 2023*, 2023.

A. Javaloy, P. Sánchez-Martín, and I. Valera. Causal normalizing flows: from theory to practice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

I. Khemakhem, R. Monti, R. Leech, and A. Hyvarinen. Causal autoregressive flows. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

D. Kisa, G. Van den Broeck, A. Choi, and A. Darwiche. Probabilistic sentential decision diagrams. In *Principles of Knowledge Representation and Reasoning (KR)*, 2014.

N. K. Kitson, A. C. Constantinou, Z. Guo, Y. Liu, and K. Chobtham. A survey of bayesian network structure learning. *Artificial Intelligence Review*, 2023.

I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

V. Melnychuk, D. Frauen, and S. Feuerriegel. Normalizing flows for interventional density estimation. In *International Conference on Machine Learning (ICML)*, 2023.

R. E. Neapolitan et al. *Learning Bayesian Networks*. Pearson Prentice Hall Upper Saddle River, 2004.

G. Papantonis and V. Belle. Transparency in sum-product network decompilation. In *European Conference on Artificial Intelligence (ECAI)*, 2023.

I. Papantonis and V. Belle. Interventions and counterfactuals in tractable probabilistic models: Limitations of contemporary transformations. *arXiv preprint arXiv:2001.10905*, 2020.

J. D. Park and A. Darwiche. Complexity results and approximation strategies for map explanations. *Journal of Artificial Intelligence Research (JAIR)*, 2004.

J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Cognitive Science Society (CogSci)*, 1985.

J. Pearl. From bayesian networks to causal networks. In *Mathematical Models for Handling Partial Knowledge in Artificial Intelligence*. 1995.

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

J. Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 2019.

R. Peharz, S. Tschiatschek, F. Pernkopf, and P. Domingos. On theoretical properties of sum-product networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

R. Peharz, S. Lang, A. Vergari, K. Stelzner, A. Molina, M. Trapp, G. Van Den Broeck, K. Kersting, and Z. Ghahramani. Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *International Conference on Machine Learning (ICML)*, 2020.

H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011.

H. Poonia, M. Willig, Z. Yu, M. Zečević, K. Kersting, and D. S. Dhami. χSPN: Characteristic interventional sum-product networks for causal inference in hybrid domains. In *Uncertainty in Artificial Intelligence (UAI)*, 2024.

A. Reisach, C. Seiler, and S. Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 1974.

B. Schölkopf and J. von Kügelgen. From statistical to causal learning. In *International Congress of Mathematicians (ICM)*, 2022.

X. Shao, A. Molina, A. Vergari, K. Stelzner, R. Peharz, T. Liebig, and K. Kersting. Conditional sum-product networks: Imposing structure on deep probabilistic architectures. In *International Conference on Probabilistic Graphical Models (PGM)*, 2020.

C. Shi, D. Blei, and V. Veitch. Adapting neural networks for the estimation of treatment effects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

G. Van den Broeck, N. Di Mauro, and A. Vergari. Tractable probabilistic models: Representations, algorithms, learning, and applications. *Tutorial at UAI*, 2019.

A. Vergari, Y. Choi, A. Liu, S. Teso, and G. Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

B. Wang and M. Kwiatkowska. Compositional probabilistic and causal inference using tractable circuit models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

K. Xia, K.-Z. Lee, Y. Bengio, and E. Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Z. Yu, M. Trapp, and K. Kersting. Characteristic circuits. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

M. Zečević, D. S. Dhami, and K. Kersting. Not all causal inference is the same. *Transactions on Machine Learning Research (TMLR)*, 2023.

M. Zečević, D. Dhami, A. Karanam, S. Natarajan, and K. Kersting. Interventional sum-product networks: Causal inference with tractable probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

N. L. Zhang, R. Qi, and D. Poole. A computational theory of decision networks. *International Journal of Approximate Reasoning (IJAR)*, 1994.

H. Zhao, M. Melibari, and P. Poupart. On the relationship between sum-product networks and bayesian networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Y. Zhou, X. Wu, B. Huang, J. Wu, L. Feng, and K. C. Tan. Causalbench: A comprehensive benchmark for causal learning capability of large language models. *arXiv preprint arXiv:2210.17283*, 2024.

Appendix A. Experimental Setup

In this appendix, we give full details on the experimental setup, expanding on the information given in the main paper.

Training Details. We train all models using the Adam optimizer (Kingma and Ba, 2017) and use exponential learning rate decay with 0.99 for Ψnet[5] and iSPN and cosine annealing with warm restarts (Loshchilov and Hutter, 2017) for NCM, where we employ a cycle length of 50 and a minimum learning rate of 1e-4 for all experiments. The learning rate used for Ψnet and iSPN is 2e-4 for the synthetic experiments and 1e-3 for CausalBench. Ψnet and iSPN are trained using a single-layer neural network with 128 (Ψnet) or 75 (iSPN) neurons for the synthetic problems. For CausalBench, switching to a three layer (width 1024, 2048, 512) NN for Ψnet improves training loss, but decreases final evaluation metrics. Ψnet* and iSPN use only 75 neurons for CausalBench. For synthetic datasets a batch size of 128 (NCM: 50,000) is used and models are trained over 100 (NCM: 1000) epochs (using early stopping if models converge faster). The batch size for CausalBench is 64 for Ψnet*, 32 for Ψnet and iSPN, and 1024 for NCM. Due to the different way training works for NCM, a larger batch size might be beneficial but is not possible due to memory constraints. The depth of the Ψnet is set to be the maximum depth available per dataset when splitting scope into individual variables at the leaves. This results in a depth of $\lceil \log_2(|\text{Variables}|) \rceil$. All models were built using Pytorch and trained on Nvidia DGX-clusters with A-100 GPUs with up to 80GB.

Synthetic Problems. To conduct an experiment under controlled conditions, we randomly generated the causal graphs and edge weights, consisting of a fixed number of nodes and edges. We then sample 10,000 data points from the unintervened 'observational' case and a further 10,000 interventional samples for each variable. We run experiments over 5 different seeds, each, and evaluate on the following configurations of nodes/edges: 5/5, 10/12, 15/20, 20/30, 50/100, 100/250. Model performance is recorded after 0.5, 1, 2, 4, and 5 hours of training (using the model of the last epoch finished within the given amount of time). We train the NCM models using the ground-truth causal graph.

CausalBench. For our large-scale real-world evaluation, we use the train and test sets of the K562 and RPE1 datasets provided within the CausalBench repository.[6] All models are trained on observational and interventional data and with a time budget of 8 hours for training. We conducted a preliminary hyperparameter search for the Ψnet model, varying depth of the EiNet model, the neural network architecture, batch size, learning rate, and the number of sum nodes in the EiNet to arrive at an optimized Ψnet model. A neural network architecture with three hidden layers was used for Ψnet due to the hyperparameter search. For the NCM, we use instrumental leaves, as described in Section 4, to allow the model to estimate the intervention value of the soft intervention. Instrumental variables, indicating the presence of an intervention, are binary, while all others are implemented as continuous variables using normalizing flows. We use the *Sortnregress* (Reisach et al., 2021) algorithm results as an input for the graph structure for the NCM.[7] For evaluation, we compare the

5. If not specified otherwise, the training details from Ψnet also apply to Ψnet*.
6. https://github.com/causalbench/causalbench
7. The Sortnregress algorithm only uses observational data, while the NCM is also trained with interventional data. The time that the Sortnregress algorithm takes to return the causal graph is not part of the 8-hour time budget. The 8-hour budget is fully used to train the NCM after being given the causal

	K562				RPE1			
	Ψnet	Ψnet*	iSPN	NCM	Ψnet	Ψnet*	iSPN	NCM
MSE	0.0942	0.0791	0.1093	0.8448	0.1131	0.0617	0.0573	0.1563
Time (s)	543.32	551.73	2583.98	50109.32	160.04	168.77	831.04	15313.94

Table 3: **CausalBench Results; 24 Hours of Training.** The mean squared error shows the average error when predicting the most likely value. The time to conduct this evaluation is shown in the second row.

ground truth marginal probabilities of observational and interventional distributions with the MPE predictions of Ψnet and iSPN. For NCM we compare with the naïve marginal probabilities obtained from sampling. We only use the same number of samples as the batch size here. Additional samples could be generated at the cost of runtime. Just like for the ground truth, the NCM calculates the mean over samples for the evaluation. We again implement early stopping, which stops training when the loss does not improve further.

Appendix B. Further Experimental Results

We show results when training for 24 hours instead of 8 in Table 3. Both Ψnet and Ψnet* results remain the same as early stopping prevents training more than 8 hours. While NCM makes some improvements for K562, it still remain much worse in MSE than the all the models based on SPNs.

graph. Therefore, the actual time required to train the Sortnregress + NCM pipeline is actually above 8 hours.