CAUSAL NEURAL PROBABILISTIC CIRCUITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept Bottleneck Models (CBMs) enhance the interpretability of end-to-end neural networks by introducing a layer of concepts and predicting the class label from the concept predictions. A key property of CBMs is that they support interventions, *i.e.*, domain experts can correct mispredicted concept values at test time to improve the final accuracy. However, typical CBMs apply interventions by overwriting only the corrected concept while leaving other concept predictions unchanged, which ignores causal dependencies among concepts. To address this, we propose the Causal Neural Probabilistic Circuit (CNPC), which combines a neural attribute predictor with a causal probabilistic circuit compiled from a causal graph. This circuit supports exact, tractable causal inference that inherently respects causal dependencies. Under interventions, CNPC models the class distribution based on a Product of Experts (PoE) that fuses the attribute predictor's predictive distribution with the interventional marginals computed by the circuit. We theoretically characterize the compositional interventional error of CNPC *w.r.t.* its modules and identify conditions under which its upper bound is no larger than that of its NPC counterpart. Experiments on four datasets provide evidence that CNPC propagates interventions consistently with the encoded causal structure and achieves competitive intervention efficiency against five baselines.

1 INTRODUCTION

Despite achieving high performance across a range of benchmarks (Deng et al., 2009; Wang et al., 2019), deep learning models are often black-box and lack interpretability (Rudin, 2019), which makes it difficult to understand why a specific decision is made, let alone to interact with the decision process. Concept Bottleneck Models (CBMs) (Koh et al., 2020; Stammer et al., 2021) offer a promising direction by introducing a concept layer that decomposes the decision process into two stages: a neural concept predictor first infers the values of a predefined set of human-understandable concepts from the input, and a label predictor, often a linear layer, then maps the predicted concept values to the final class label. This architecture provides concept-level explanations and, more importantly, allows humans to intervene in the model's decision process by correcting mispredicted concept values. Such interventions are particularly valuable in safety-critical applications such as medical diagnosis (Esteva et al., 2017; De Fauw et al., 2018), as a domain expert can inspect and correct concept values to improve the accuracy and reliability of the final prediction.

Typical CBMs (Koh et al., 2020; Zarlenga et al., 2022; Oikarinen et al., 2023) implement interventions by overwriting only the corrected concept while leaving all other concept predictions unchanged; see Figure 1 (top panel). In practice, however, domain experts often have prior knowledge of causal relations among concepts and may expect an intervention on one concept to update other concepts that causally depend on it. For example, if smoking is known to causally affect lung cancer, intervening on the smoking concept should also change the predicted likelihood of lung cancer. Therefore, CBMs should account for causal dependencies among concepts and propagate interventions through the known causal structure. Such propagation can make model responses more causally faithful under intervention and may also improve intervention efficiency, *i.e.*, achieving higher task accuracy under the same intervention budget.

Realizing causal intervention propagation requires a CBM to model how interventions change the joint distribution over attributes and, in turn, the class distribution. We show that this is challenging

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

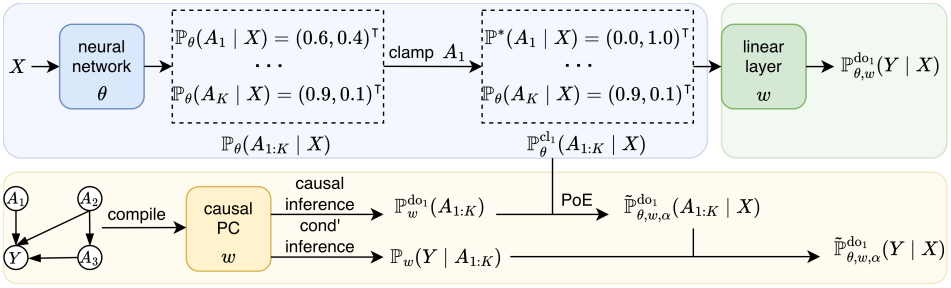


Figure 1: Model architectures and intervention procedures of a CBM and a CNPC. A typical CBM (top-left and top-right modules) intervenes on A_1 by clamping the neural network's prediction for A_1 to the ground-truth distribution, while leaving the predictions for other attributes unchanged. In contrast, CNPC (top-left and bottom modules) combines a neural network with a causal PC compiled from a causal graph, and approximates the interventional class distribution based on a PoE with a balancing weight α that fuses complementary information from the two modules.

for standard CBMs because the interventional attribute[1] distribution cannot, in general, be obtained using the standard CBM modules. To address this challenge, inspired by the recent Neural Probabilistic Circuit (NPC) (Chen et al., 2025; 2026), we propose the Causal Neural Probabilistic Circuit (CNPC). NPC combines a neural attribute predictor with a probabilistic circuit (PC) (Poon & Domingos, 2011; Choi et al., 2020; Vergari et al., 2020), which supports exact, tractable probabilistic inference over attributes and the class label. NPC predicts the class distribution by combining the attribute distribution produced by the neural predictor with the conditional class distribution given attributes computed by the PC. We extend NPC to CNPC by compiling a causal graph over the attributes and class label into the PC, resulting in a causal PC (Darwiche, 2022) that supports both exact, tractable probabilistic inference and causal inference. Without interventions, CNPC follows the same prediction procedure as NPC. Under interventions on one or more attributes, CNPC approximates the interventional attribute distribution using a Product of Experts (PoE) (Hinton, 2002), which combines the input-dependent attribute distribution from the neural predictor with the interventional marginal distribution computed exactly by the causal PC. The resulting distribution is then combined with the conditional class distribution given attributes computed by the causal PC to obtain the class prediction under intervention.

Our theoretical results show that both NPC and CNPC exhibit compositional errors *w.r.t.* the neural predictor and the (causal) PC under interventions, providing a theoretical guarantee that is generally unavailable for standard CBMs. Moreover, under certain conditions, CNPC admits an upper bound on interventional error that is no larger than that of NPC. We evaluate CNPC against five baselines on four datasets—Asia, Sachs, MNISTAdd, and CelebA—under varying numbers of attribute interventions. Compared with NPC, CNPC improves causally-descendant attribute accuracy on average across all four datasets while largely preserving non-descendant accuracy where measurable, indicating that interventions are propagated in a manner consistent with the encoded causal structure. CNPC also consistently achieves higher average task accuracy than NPC, showing that such propagation improves intervention efficiency. Finally, CNPC remains competitive with all five baselines, achieving the highest macro-average task accuracy on Asia, Sachs, and MNISTAdd, as well as the highest macro-average attribute accuracy on Asia and MNISTAdd.

Our contributions are summarized as follows. **(1)** To address NPC's limitation in causal intervention propagation, we propose CNPC, which utilizes a causal PC to estimate the interventional class distribution while propagating attribute interventions through the causal structure. **(2)** We provide theoretical results characterizing the compositional interventional errors of NPC and CNPC, and identify conditions under which CNPC admits an upper bound no larger than that of NPC. **(3)** We provide empirical evidence of the proposed intervention propagation consistent with the encoded causal structure and show that CNPC improves average intervention efficiency over NPC while achieving competitive performance against five baselines.

[1]In the CBM literature, a *concept* is defined as a binary variable indicating the presence or absence of a human-interpretable property (*e.g.*, *is_red_color*, *is_yellow_color*), whereas we use a more general term *attribute*, defined as a categorical variable that takes values from a finite set of such properties (*e.g.*, *color*).

2 Related work

Modeling concept dependencies in CBMs. Standard CBMs (Koh et al., 2020) typically leave non-intervened concept predictions unchanged when selected concepts are corrected. Accounting for concept dependencies can make better use of these corrections by updating the remaining concepts. SCBM (Vandenhirtz et al., 2024) models concept logits jointly with a multivariate Gaussian, while Havasi et al. (2022) capture concept dependencies autoregressively. Concept Realignment (Singhi et al., 2024) propagates corrections through an additional trained network, whereas Causal Intervention (CI) (Zhu et al., 2026) uses gradient-based input updates to exploit dependencies encoded in a fixed concept predictor. These methods exploit statistical correlations among concepts, without guaranteeing that intervention propagation respects human causal knowledge. Recent work (de Felice et al., 2025; Dominici et al., 2025) models *causal* dependencies among concepts and the class variable using a ground-truth or learned causal graph and neural approximations to structural mechanisms. These methods, however, restrict the structural equations to linear forms, which may not adequately capture more complex causal relations. With the same goal of modeling causal dependencies, CNPC instead compiles the causal graph into a PC, which supports exact and tractable causal inference without assuming a specific functional form for the structural equations.

Intervention strategies. Due to the high cost of human interventions, the number of concepts that can be intervened on is often limited. This naturally raises the question of which concepts to select under a fixed intervention budget. Ideally, an intervention strategy should choose the concepts that yield the largest improvement in downstream task performance. Shin et al. (2023); Sheth et al. (2022) study a range of selection criteria (*e.g.*, prediction uncertainty) as well as different selection granularities (*e.g.*, intervening on individual concepts versus concept groups). They find that no single criterion is consistently best across settings, and that intervening on concept groups can be cost-effective when concepts within a group are mutually exclusive. Henceforth, interventions are performed by default on concept groups, *i.e.*, attributes. Beyond these hand-crafted selection criteria, Chauhan et al. (2023); Steinmann et al. (2024) learn policies that adaptively choose the next concept to intervene on. With the causal structure explicitly incorporated, de Felice et al. (2025) select attributes in a top-down order along the causal hierarchy, as intervening on higher-level attributes can potentially influence more downstream attributes. In this work, we follow them and select attributes for intervention by their depth in the causal graph.

3 Preliminaries

3.1 Notation and assumptions

Throughout the paper, random variables, their instantiations, and the instantiation sets are denoted by capital, lowercase, and calligraphic letters, respectively, *e.g.*, $X, x, \mathcal{X}$. CBMs predict the class label solely from the predicted attribute values, which relies on the following assumption:

Assumption 1 (Sufficient attributes (Chen et al., 2026))**.** The class label Y and the input X are conditionally independent given the attributes $A_{1:K}$, *i.e.*, $Y \perp X \mid A_{1:K}$.

In the case of insufficient attributes, one may leverage large language models (LLMs) or vision-language models (VLMs) for automated concept discovery and labeling (Oikarinen et al., 2023; Yüksekgönül et al., 2023; Srivastava et al., 2024), which remains an active research problem in the CBM literature. While this is complementary to our method, we assume that sufficient attributes are annotated, as our main focus is the incorporation of causal dependencies.

We also assume access to the structure[2] of a causal graph $\mathcal{C}$ over the semantically meaningful, observable variables, namely attributes $A_{1:K}$ and the class Y in our setting. Each node in the graph represents a variable, and each directed edge $V_1 \to V_2$ indicates that V_1 is a causal parent (direct cause) of V_2. The availability of the graph structure is common in the causal inference literature (Pearl, 1995; 2009; Tian & Pearl, 2002) and is typically obtained from domain knowledge. If unknown, a plausible graph can instead be learned using existing causal discovery algorithms (Spirtes & Glymour, 1991; Heckerman et al., 1995; Chickering, 2002), which are complementary to our method.

[2]We assume access only to the graph structure, while the specific causal mechanisms can remain unknown.

We assume there are no unobserved confounders among the variables in $\mathcal{C}$. For a random variable $V \in \mathcal{C}$, we use $\mathrm{Pa}_{\mathcal{C}}(V)$ to denote the parent nodes of V in $\mathcal{C}$ and $\mathrm{ND}_{\mathcal{C}}(V)$ to denote the non-descendants of V in $\mathcal{C}$. In this work, we make the following structural assumption on $\mathcal{C}$:

Assumption 2 (Structural assumption on the causal graph)**.** $\mathrm{Pa}_{\mathcal{C}}(Y) \subseteq \{A_{1:K}\} \subseteq \mathrm{ND}_{\mathcal{C}}(Y)$.

It states that every causal parent of Y in $\mathcal{C}$ is an attribute, and every attribute is a non-descendant of Y. This is natural in CBM settings, as it allows attributes to influence the class label, directly or indirectly through other attributes, while excluding causal influence in the reverse direction.

3.2 Neural probabilistic circuits

A Neural Probabilistic Circuit (NPC) (Chen et al., 2026; Chen & Zhao, 2025) consists of two modules. The first is an *attribute predictor*, which is a neural network with K classification heads and parameterized by θ. It takes an input X and predicts K attribute distributions $\{\mathbb{P}_\theta(A_k \mid X)\}_{k=1}^K$. The second is a *probabilistic circuit* (PC) (Choi et al., 2020), specifically a sum-product network (Poon & Domingos, 2011), parameterized by w. It represents the joint distribution $\mathbb{P}_w(Y, A_{1:K})$ while supporting the tractable computation of the conditional distribution $\mathbb{P}_w(Y \mid A_{1:K})$. Leveraging these two modules, NPC predicts the class distribution as:

$$\mathbb{P}_{\theta,w}(Y \mid X) = \sum_{a_{1:K}} \mathbb{P}_w(Y \mid A_{1:K} = a_{1:K}) \cdot \prod_{k=1}^{K} \mathbb{P}_\theta(A_k = a_k \mid X). \tag{1}$$

Although not explicitly described in the original paper, the interventional distribution modeled by NPC can be inferred from the standard intervention procedure used in CBMs. Consider an intervention on attribute A_j, denoted by do_j. NPC would clamp the j-th attribute to its ground-truth distribution, *i.e.*, replacing $\mathbb{P}_\theta(A_j \mid X)$ with $\mathbb{P}_*(A_j \mid X)$ [3] while leaving the remaining predictive distributions unchanged. Consequently, NPC models the interventional class distribution as:

$$\begin{aligned} &\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(Y \mid X) = \sum_{a_{1:K}} \mathbb{P}_w(Y \mid A_{1:K} = a_{1:K}) \cdot \mathbb{P}^{\mathrm{cl}_j}_\theta(A_{1:K} = a_{1:K} \mid X), \\ &\text{where } \mathbb{P}^{\mathrm{cl}_j}_\theta(A_{1:K} = a_{1:K} \mid X) = \mathbb{P}_*(A_j = a_j \mid X) \cdot \prod_{k \neq j} \mathbb{P}_\theta(A_k = a_k \mid X). \end{aligned} \tag{2}$$

This formulation naturally extends to interventions on multiple attributes. Similar to typical CBMs, NPC does not account for causal dependencies among attributes during intervention. One potential barrier is that standard PCs do not support tractable causal inference.

3.3 Causal probabilistic circuits

Traditional approaches to interventional queries reduce the problem to exact inference on the mutilated graph induced by the intervention (*i.e.*, the causal graph with edges into intervened variables removed (Pearl, 1995)) with per-query complexity exponential in treewidth (Dechter, 1999; Koller & Friedman, 2009; Darwiche, 2009). In contrast, Chavira & Darwiche (2007); Darwiche (2022) show that a causal graph can be compiled into a PC that, once built, supports exact causal inference in time linear in circuit size.[4]

In particular, they use the Variable Elimination (VE) algorithm (Koller & Friedman, 2009) to schedule factor operations within the network polynomial (Darwiche, 2003), and construct an arithmetic circuit accordingly. As an example, consider three variables with the graph structure $V_2 \leftarrow V_1 \rightarrow V_3$. Each variable corresponds to a factor that represents its conditional probability distribution (CPD): $\phi_{V_1}(V_1)$, $\phi_{V_2}(V_1, V_2)$, and $\phi_{V_3}(V_1, V_3)$, whose symbolic entries are ϕ_{v_1}, $\phi_{v_2|v_1}$, and $\phi_{v_3|v_1}$, respectively. Each variable is also associated with an evidence factor, $\lambda_{V_i}(V_i)$, with entries given by indicator functions $\lambda_{v_i} = \mathbb{I}(V_i = v_i)$. With these notations, the network polynomial is expressed as:

$$f(\boldsymbol{\phi}, \boldsymbol{\lambda}) = \sum_{v_1, v_2, v_3} \phi_{v_1} \phi_{v_2|v_1} \phi_{v_3|v_1} \lambda_{v_1} \lambda_{v_2} \lambda_{v_3}.$$

[3] In practice, $\mathbb{P}_*(A_j \mid X)$ is typically $\mathbb{I}(A_j = a_j^*(X))$ for some $a_j^*(X) \in \mathcal{A}_j$, which we adopt throughout; relaxing it to stochastic attribute labelings is a natural extension and would not change our qualitative claims.

[4] Throughout this paper, by causal inference, we refer specifically to interventional queries. We clarify that the circuit size itself can be exponential in the worst case, reflecting the inherent hardness of exact inference rather than a limitation of compilation.

Given an elimination order, *e.g.*, $[V_2, V_3, V_1]$, VE schedules the factor operations (multiplication and sum-out), yielding an equivalent expression that is more efficient to evaluate (Dechter, 1999; Zhang & Poole, 1996) (see the full derivation in Appendix B.1):

$$f(\boldsymbol{\phi}, \boldsymbol{\lambda}) = \sum_{v_1} \phi_{v_1} \lambda_{v_1} \cdot \left(\sum_{v_3} \phi_{v_3|v_1} \lambda_{v_3} \right) \cdot \left(\sum_{v_2} \phi_{v_2|v_1} \lambda_{v_2} \right).$$

An arithmetic circuit is then constructed by establishing a sum node for the sum-out operator while establishing a product node for the multiplication operator. For simplicity, suppose all variables are binary and we use v_i and $\bar{v}_i$ to denote $V_i = 1$ and $V_i = 0$. The resulting circuit is illustrated in Figure 2. We refer to this circuit that compiles the causal graph as the *causal PC*. For values of those symbolic entries, one can use gradient descent or EM to seek maximum-likelihood estimates (Darwiche, 2022).

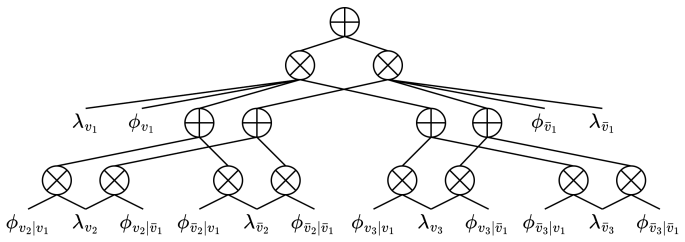


Figure 2: A causal PC that compiles the causal graph $V_2 \leftarrow V_1 \rightarrow V_3$.

As the causal PC encodes the network polynomial, it supports tractable probabilistic inference (Darwiche, 2003). Computing the probability of any joint or marginal event, *e.g.*, $\Pr(v_1, \bar{v}_2)$, only requires setting the indicators whose subscript is compatible with the event to 1, *i.e.*, $\lambda_{v_1} = 1$, $\lambda_{\bar{v}_2} = 1$, $\lambda_{v_3} = 1$, $\lambda_{\bar{v}_3} = 1$, and enabling one forward pass in PC. Conditional inference, hence, requires two PC forward passes. With the compilation of the causal graph, the causal PC also supports tractable causal inference. Computing the interventional probability , *e.g.*, $\Pr^{\text{do}(\bar{v}_2)}(v_1)$, can be done in two steps: (i) Replace the CPD entries of the intervened variable with 1, *i.e.*, $\phi_{v_2|v_1} = 1$, $\phi_{v_2|\bar{v}_1} = 1$, $\phi_{\bar{v}_2|v_1} = 1$, $\phi_{\bar{v}_2|\bar{v}_1} = 1$; (ii) Set the indicators compatible with $(v_1, \bar{v}_2)$ to 1. Following these steps, the causal PC outputs ϕ_{v_1}, *i.e.*, $\Pr(v_1)$, which is the expected result as V_2 has no causal effect on V_1.

4 METHOD

4.1 WHY IS MODELING THE INTERVENTIONAL CLASS DISTRIBUTION HARD

To account for how attribute interventions affect class predictions through causal dependencies, we consider the input-conditioned interventional class distribution. Given an input X and an intervention do_j, which sets A_j to a fixed value according to $\mathbb{P}_*(A_j \mid X)$, the distribution can be written as:

$$\mathbb{P}^{\text{do}_j}(Y \mid X) = \sum_{a_{1:K}} \mathbb{P}(Y \mid A_{1:K} = a_{1:K}) \cdot \mathbb{P}^{\text{do}_j}(A_{1:K} = a_{1:K} \mid X). \tag{3}$$

This expression follows from Assumptions 1, 2, together with the local Markov property and the invariance of causal mechanisms. The full derivation is provided in Appendix C.

The first term in Equation (3) can be evaluated exactly and tractably by a PC or a causal PC; the second term, however, is not directly available from any module. Using Bayes' rule, we rewrite it as:

$$\mathbb{P}^{\text{do}_j}(A_{1:K} = a_{1:K} \mid X) \propto \mathbb{P}^{\text{do}_j}(A_{1:K} = a_{1:K}) \cdot \mathbb{P}^{\text{do}_j}(X \mid A_{1:K} = a_{1:K}).$$

In a simplistic setting where $\text{Pa}_{\mathcal{C}^+}(X) = \{A_{1:K}\}$ [5], $\mathbb{P}^{\text{do}_j}(X \mid A_{1:K}) = \mathbb{P}(X \mid A_{1:K})$ by invariance of causal mechanisms. Consequently, $\mathbb{P}^{\text{do}_j}(A_{1:K} = a_{1:K} \mid X)$ simplifies to:

$$\mathbb{P}^{\text{do}_j}(A_{1:K} = a_{1:K} \mid X) \propto \frac{\mathbb{P}^{\text{do}_j}(A_{1:K} = a_{1:K})}{\mathbb{P}(A_{1:K} = a_{1:K})} \cdot \mathbb{P}(A_{1:K} = a_{1:K} \mid X).$$

Here, the ratio term can be computed via causal and probabilistic inference in the causal PC, while the remaining term can be computed by the attribute predictor. Hence, $\mathbb{P}^{\text{do}_j}(Y \mid X)$ can be obtained from the two modules. Nevertheless, we typically do not know the causal relationships between

[5] Here, $\mathcal{C}^+$ denotes the augmented causal graph obtained by incorporating X into $\mathcal{C}$.

the high-dimensional input X and other variables, and $\mathbb{P}^{\text{do}_j}(X \mid A_{1:K})$ does not coincide with $\mathbb{P}(X \mid A_{1:K})$ in general. This mismatch can arise, for example, if X has additional (possibly unobserved) causal parents beyond $A_{1:K}$, or if some attributes in $A_{1:K}$ are descendants of X. Therefore, modeling $\mathbb{P}^{\text{do}_j}(X \mid A_{1:K})$ would require learning an interventional generative model for X, which is challenging, as many powerful generative models (*e.g.*, diffusion models (Ho et al., 2020) and GANs (Goodfellow et al., 2020)) do not allow access to the generative probabilities but only sampling, and we typically do not have access to X's interventional samples.

4.2 Intervention Propagation: Causal Neural Probabilistic Circuits

In this section, we propose the Causal Neural Probabilistic Circuit (CNPC), which consists of an *attribute predictor* and a *causal PC*; see the top-left and bottom modules in Figure 1. The causal PC is obtained by compiling the causal structure over the attributes and the class label, and its parameters are estimated by maximum likelihood. Following NPC, the attribute predictor is a multi-head neural network that predicts the distributions over the K attributes from the input; it can be trained using the attribute prediction loss, followed by an optional final fine-tuning stage using the class prediction loss while keeping the PC part fixed.

Compared with NPC, the main architectural difference is that CNPC replaces the PC with a causal PC. However, this replacement alone is not sufficient to induce an interventional class distribution that accounts for causal dependencies. We still need to address the issue described in Section 4.1, namely modeling $\mathbb{P}^{\text{do}_j}(A_{1:K} \mid X)$. Since the exact interventional conditional $\mathbb{P}^{\text{do}_j}(A_{1:K} \mid X)$ is unavailable, we approximate it by combining two tractable distributions that capture complementary information: **(i)** the clamped predictive distribution $\mathbb{P}_\theta^{\text{cl}_j}(A_{1:K} \mid X)$ from the neural predictor, which preserves evidence from the input but does not causally propagate the intervention; **(ii)** the interventional marginal $\mathbb{P}_w^{\text{do}_j}(A_{1:K})$ from the causal PC, which correctly propagates the intervention through the causal structure but is not conditioned on the input. To combine the two, we adopt the reverse KL-barycenter approximation (Genest & Zidek, 1986; Heskes, 1997), defined as the solution to the variational problem of minimizing a weighted sum of reverse KL divergences from each source:

$$\tilde{\mathbb{P}}^{\text{do}_j}_{\theta,w,\alpha}(\cdot \mid X) := \arg\min_{q \in \Delta(\mathcal{A}_{1:K})} (1-\alpha)\, D_{\text{KL}}\Big(q \,\Big\|\, \mathbb{P}_\theta^{\text{cl}_j}(\cdot \mid X)\Big) + \alpha\, D_{\text{KL}}\Big(q \,\Big\|\, \mathbb{P}_w^{\text{do}_j}(\cdot)\Big),$$

where $\Delta(\mathcal{A}_{1:K})$ denotes the probability simplex over the joint attribute space, and $\alpha \in [0,1]$ controls the relative influence of the two sources. The unique minimizer is the normalized geometric mean, also known as a product of experts (PoE), given as follows:

$$\tilde{\mathbb{P}}^{\text{do}_j}_{\theta,w,\alpha}(A_{1:K} = a_{1:K} \mid X) = \frac{\Big(\mathbb{P}_\theta^{\text{cl}_j}(A_{1:K} = a_{1:K} \mid X)\Big)^{1-\alpha} \Big(\mathbb{P}_w^{\text{do}_j}(A_{1:K} = a_{1:K})\Big)^{\alpha}}{Z_\alpha(X)}, \tag{4}$$

where $Z_\alpha(X)$ is the normalizing constant.[6]

Empirically, we measure the KL divergence from the exact interventional conditional to the oracle PoE as α varies. The oracle PoE is constructed by replacing the two learned distributions in Equation (4) with the exact observational attribute distribution and the exact interventional marginal, respectively. Figure 3 shows that on the two studied datasets, the divergence reaches nearly zero at an appropriate value of α, indicating that the PoE formulation can closely approximate the exact interventional conditional when its constituent distributions are accurately specified and properly balanced.

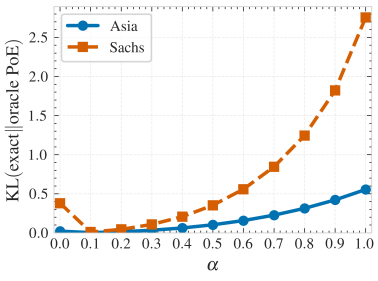


Figure 3: KL divergence from the exact interventional conditional to the oracle PoE on Asia and Sachs.

With this PoE approximation to the interventional conditional, CNPC predicts the interventional class distribution as:

$$\tilde{\mathbb{P}}^{\text{do}_j}_{\theta,w,\alpha}(Y \mid X) = \sum_{a_{1:K}} \mathbb{P}_w(Y \mid A_{1:K} = a_{1:K}) \cdot \tilde{\mathbb{P}}^{\text{do}_j}_{\theta,w,\alpha}(A_{1:K} = a_{1:K} \mid X). \tag{5}$$

Finally, CNPC predicts the attribute labels as $\hat{a}_{1:K} = \arg\max_{a_{1:K} \in \mathcal{A}_{1:K}} \tilde{\mathbb{P}}^{\text{do}_j}_{\theta,w,\alpha}(A_{1:K} = a_{1:K} \mid X)$ and the class label as $\hat{y} = \arg\max_{y \in \mathcal{Y}} \tilde{\mathbb{P}}^{\text{do}_j}_{\theta,w,\alpha}(Y = y \mid X)$. When no intervention is applied, CNPC

[6]The extension to interventions on multiple attributes is straightforward: we simply clamp the intervened attributes to their ground-truth distributions and compute the resulting interventional marginal with causal PC.

predicts the class distribution in the same way as NPC, *i.e.*, via Equation (1). Moreover, setting $\alpha = 0$ recovers NPC as a special case of CNPC.

Remarks. The design aims to make model responses to interventions more consistent with the specified causal structure and potentially improve intervention efficiency, *i.e.*, achieve higher task accuracy under the same intervention budget. We note, however, that causal intervention propagation does not necessarily maximize task accuracy. Methods that propagate corrections based on statistical dependencies may, for example, benefit from updating upstream attributes and achieve larger accuracy gains. Our approach uses the specified causal structure to guide intervention propagation, treating causal faithfulness as an explicit design objective alongside predictive performance.

5 THEORETICAL ANALYSIS

First, analogous to Chen et al. (2026, Theorem 2), Theorem 5 in Appendix E shows that, in the absence of interventions, the prediction errors of both NPC and CNPC admit a compositional bound in terms of the errors of the attribute predictor and the (causal) PC. Consequently, reducing either component error tightens the upper bound on the overall prediction error. Corollaries 3 and 4 further show that a similar compositional error decomposition still holds under interventions.

Corollary 3. *The expected interventional error of NPC is upper-bounded by the sum of the expected interventional error of the attribute predictor and the expected prediction error of the PC:*

$$\mathbb{E}_{X\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{do_j}(Y\mid X)\|\,\mathbb{P}_{\theta,w}^{do_j}(Y\mid X)\right)\right] \le \mathbb{E}_{X\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{do_j}(A_{1:K}\mid X)\|\,\mathbb{P}_\theta^{cl_j}(A_{1:K}\mid X)\right)\right] + \mathbb{E}_{A_{1:K}\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y\mid A_{1:K})\|\,\mathbb{P}_w(Y\mid A_{1:K})\right)\right].$$

Equality holds if for each (x, y), there exists a constant $c(x, y)$ s.t. $\frac{\mathbb{P}_*^{do_j}(a_{1:K}|x)\mathbb{P}_*(y|a_{1:K})}{\mathbb{P}_\theta^{cl_j}(a_{1:K}|x)\mathbb{P}_w(y|a_{1:K})} = c(x, y)$ for all $a_{1:K}$.

Corollary 4. *The expected interventional error of CNPC is upper-bounded by a weighted sum of (i) the expected interventional error of the attribute predictor, (ii) the expected interventional error of the causal PC over attributes, and (iii) the expected prediction error of the causal PC:*

$$\mathbb{E}_{X\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\Big(\mathbb{P}_*^{do_j}(Y\mid X)\,\|\,\tilde{\mathbb{P}}_{\theta,w,\alpha}^{do_j}(Y\mid X)\Big)\right] \le (1-\alpha)\,\mathbb{E}_{X\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\Big(\mathbb{P}_*^{do_j}(A_{1:K}\mid X)\,\|\,\mathbb{P}_\theta^{cl_j}(A_{1:K}\mid X)\Big)\right] + \alpha\,\mathbb{E}_{X\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\Big(\mathbb{P}_*^{do_j}(A_{1:K}\mid X)\,\|\,\mathbb{P}_w^{do_j}(A_{1:K})\Big)\right] + \mathbb{E}_{A_{1:K}\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y\mid A_{1:K})\|\,\mathbb{P}_w(Y\mid A_{1:K})\right)\right].$$

Equality holds if for each (x, y), $\mathbb{P}_\theta^{cl_j}(a_{1:K}\mid x) = \mathbb{P}_w^{do_j}(a_{1:K})$ for all $a_{1:K}$, and there exists a constant $c(x, y)$ s.t. $\frac{\mathbb{P}_*^{do_j}(a_{1:K}|x)\mathbb{P}_*(y|a_{1:K})}{\mathbb{P}_\theta^{cl_j}(a_{1:K}|x)\mathbb{P}_w(y|a_{1:K})} = c(x, y)$ for all $a_{1:K}$.

Both bounds inherit the compositional structure of the observational case, a theoretical guarantee that has not been established for CBMs in general (Chen et al., 2026). Let $\mathbb{B}_{NPC}$ and $\mathbb{B}_{CNPC}$ denote the interventional error bounds in Corollaries 3 and 4, respectively. It follows directly that if

$$\mathbb{E}_{X\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\Big(\mathbb{P}_*^{\mathrm{do}_j}(A_{1:K}\mid X)\,\|\,\mathbb{P}_w^{\mathrm{do}_j}(A_{1:K})\Big)\right] \le \mathbb{E}_{X\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\Big(\mathbb{P}_*^{\mathrm{do}_j}(A_{1:K}\mid X)\,\|\,\mathbb{P}_\theta^{\mathrm{cl}_j}(A_{1:K}\mid X)\Big)\right],$$

then $\mathbb{B}_{CNPC} \le \mathbb{B}_{NPC}$. This result implies that, when the causal PC captures the interventional attribute distribution at least as accurately as the clamped attribute predictor, incorporating it through the PoE cannot worsen the interventional error bound and can lead to a tighter guarantee than NPC.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETTINGS

Datasets. We evaluate on four datasets spanning diverse domains. All datasets are split 80:10:10 into training, validation, and test sets. **(1) Asia** (Lauritzen & Spiegelhalter, 1988) is a Bayesian network for medical diagnosis from bnlearn (Scutari, 2010). We sample 10,000 data points, taking the "dysp" node as the class variable and the remaining nodes as attributes. Following de Felice

Table 1: Accuracy gains of CNPC over NPC, averaged over partial intervention counts ($1 \leq k < K$), in pp. Non-descendant and downstream accuracies exclude directly corrected attributes. Mean attribute accuracy (Attr.) includes all attributes. The non-descendant group is nonempty only at $k = 1$ on Asia, Sachs, and CelebA, and is always empty on MNISTAdd (—).

Dataset	Δ Non-desc.	Δ Downstream	Δ Attr.	Δ Task
Asia	+0.16	+0.54	+0.13	+0.23
Sachs	+0.00	+1.58	+0.63	+0.35
MNISTAdd	—	+0.09	+0.04	+0.09
CelebA	+0.00	+0.48	+0.19	+0.79

et al. (2025), we train an autoencoder to map attribute labels into an embedding, which serves as the input to models. **(2) Sachs** (Sachs et al., 2005) is a Bayesian network for biological signaling from bnlearn, processed identically to Asia. **(3) MNISTAdd** (Manhaeve et al., 2018): Each instance pairs two digit images from MNIST (LeCun et al., 1998), with the digit values as attribute labels and their sum as the class label. We generate 50,000 instances by sampling labels from a predefined causal Bayesian network and then concatenating randomly sampled images. **(4) CelebA** (Liu et al., 2015) contains 202,599 celebrity face images with 40 imbalanced binary attributes. We use the five most balanced attributes and define the class label as a unique combination of attribute labels (yielding 32 classes), and manually annotate the underlying causal graph.

Baselines. We compare CNPC with five representative CBMs: **(1) Vanilla CBM** (Koh et al., 2020) combines a neural concept predictor with a linear label predictor. **(2) CEM** (Zarlenga et al., 2022) represents concepts as high-dimensional embeddings for downstream label prediction. **(3) SCBM** (Vandenhirtz et al., 2024) models concept logits jointly with a multivariate Gaussian and updates non-intervened concepts through conditioning. **(4) C^2BM** (de Felice et al., 2025) incorporates a causal graph into the neural architecture and approximates the structural mechanisms, enabling causal intervention propagation. **(5) NPC** (Chen et al., 2026) combines a neural attribute predictor with a PC but does not explicitly model causal dependencies among attributes.

Implementation. For a fair evaluation, CNPC and all baseline models use multilayer perceptrons (MLPs) as their neural predictors. The attribute intervention order is determined by node depth in the causal graph, defined as the length of the longest directed path from root nodes. For CNPC, the PoE weight α is selected based on validation performance. More details are provided in Appendix D.1.

6.2 Experimental results

We first assess how CNPC propagates interventions compared with NPC, then evaluate its predictive performance against all five baselines. We apply cumulative interventions to $k = 0, \ldots, K$ attributes in the order specified in Section 6.1, where $K = 7, 10, 2, 5$ for Asia, Sachs, MNISTAdd, and CelebA, respectively. All results are averaged over three random seeds, and accuracy differences are expressed in percentage points (pp). Additional ablation studies are provided in Appendix D.3.

Intervention propagation compared with NPC. At each partial intervention level $1 \leq k < K$, we partition the attributes that have *not* been intervened on into descendants of at least one intervened attribute and non-descendants of every intervened attribute. For each method and each partial intervention count, we compute mean accuracy within each group, mean accuracy over all attributes (including directly corrected attributes), and task accuracy. For each metric, we then subtract NPC's accuracy from CNPC's at the same intervention count. Table 1 reports these differences averaged equally over partial intervention counts, excluding counts at which the corresponding group is empty.

Compared with NPC, CNPC achieves identical non-descendant accuracy on Sachs and CelebA and differs by only $+0.16$ pp on Asia. Because NPC leaves non-descendant predictions unchanged and both models have identical initial per-attribute accuracies, these results indicate that CNPC largely preserves non-descendant accuracy under intervention. No non-descendant comparison is available on MNISTAdd because the only uncorrected attribute after the first intervention is downstream. In contrast, downstream accuracy improves on average by 0.54, 1.58, 0.09, and 0.48 pp on Asia, Sachs, MNISTAdd, and CelebA, respectively. The concentration of improvements in downstream attributes, together with stable non-descendant accuracy where measurable, suggests that CNPC propagates interventions in a manner consistent with the encoded causal structure. Across the four

Table 2: Macro-average accuracy (%) over $k = 0, \ldots, K$. Attr.: mean attribute accuracy; Task: task accuracy. The best and second-best means in each column are bold and underlined, respectively, based on unrounded values. Per-count standard deviations are shown in Figures 4 and 6.

	Asia		Sachs		MNISTAdd		CelebA	
Method	Attr.	Task	Attr.	Task	Attr.	Task	Attr.	Task
Vanilla CBM	96.54	80.33	89.97	63.92	98.31	79.71	**93.36**	68.31
CEM	96.49	80.36	88.47	65.55	96.72	93.32	92.58	70.61
SCBM	96.50	80.39	90.42	65.33	97.80	79.77	92.81	<u>71.58</u>
C^2BM	96.52	<u>80.53</u>	**91.27**	67.02	97.79	95.44	92.82	**72.06**
NPC	<u>96.59</u>	80.35	90.28	<u>67.32</u>	<u>98.49</u>	<u>97.06</u>	92.79	70.65
CNPC	**96.69**	**80.55**	<u>90.79</u>	**67.70**	**98.51**	**97.09**	<u>92.91</u>	71.18

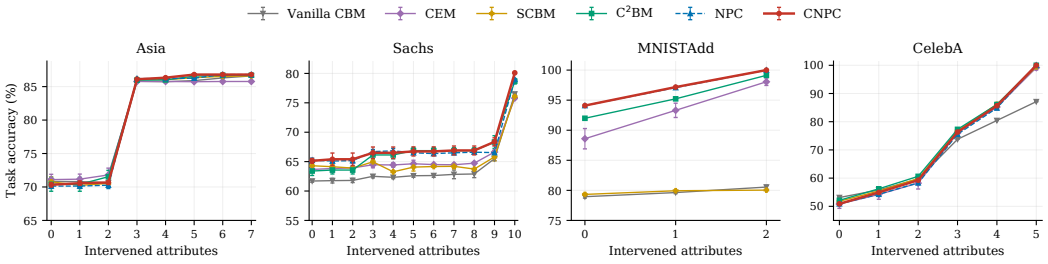


Figure 4: Task accuracy for all six methods across intervention counts. Points show means over three random seeds; error bars indicate one sample standard deviation.

datasets, CNPC improves mean attribute accuracy over NPC by 0.04–0.63 pp and task accuracy by 0.09–0.79 pp, suggesting that the proposed propagation effectively improves intervention efficiency.

Comparison with all baselines. For each method and each intervention count, we compute the mean accuracy over all attributes, including directly corrected attributes, and the task accuracy. We then average each metric equally over $k = 0, \ldots, K$, and report results in Table 2. CNPC achieves the highest macro-average task accuracy on Asia (80.55%), Sachs (67.70%), and MNISTAdd (97.09%). It also leads in attribute accuracy on Asia (96.69%) and MNISTAdd (98.51%), and ranks second on Sachs and CelebA. On CelebA, CNPC ranks third in task accuracy at 71.18%, behind C^2BM (72.06%) and SCBM (71.58%), and ahead of NPC (70.65%). Hence, CNPC consistently exceeds NPC across datasets, while leading task performance on three datasets and remaining competitive on CelebA. Figure 4 and Figure 6 in Appendix D.2 complement these averages by showing task accuracy and mean attribute accuracy at each intervention count. On Asia, CNPC attains the highest task accuracy, including ties, from $k = 4$ onward, although CEM performs better at $k = 1, 2$. On Sachs, C^2BM leads attribute accuracy at every partial intervention count, while CNPC achieves the highest task accuracy at five of the nine partial counts and at full intervention. On MNISTAdd, CNPC leads both metrics at every count, including ties at the endpoints. On CelebA, it improves on NPC in both metrics at all four partial intervention counts, with task accuracy gains of 0.65–0.95 pp, although C^2BM and SCBM retain higher task accuracy at these counts. With all attributes corrected, CNPC reaches 100% task accuracy on both MNISTAdd and CelebA. Overall, these results show that methods designed to propagate interventions generally achieve higher task accuracy, improving intervention efficiency. Although the best-performing method varies across datasets and metrics, CNPC offers competitive intervention efficiency while grounding intervention propagation in the known causal structure.

7 Conclusions

To address NPC's inability to propagate attribute interventions, we propose CNPC, which combines a neural predictor with a PC compiled from a causal graph and approximates the interventional class distribution through a PoE. Our experiments provide evidence that CNPC propagates interventions to downstream attributes in a manner consistent with the encoded causal structure, while achieving competitive attribute and task accuracy across the evaluated baselines. More broadly, CNPC offers a practical approach to incorporating causal information into predictive models and approximating interventional inference within them. We discuss limitations and potential solutions in Appendix A.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing, assist in drafting parts of the experimental results section, and write code for plotting result tables and figures. We did not use generative AI tools for literature review, research ideation or execution, synthetic data creation, or proving mathematical claims. All AI-assisted content was reviewed by the authors: AI-assisted text was manually checked and revised, and code used for plotting results was manually inspected and verified. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The authors are not aware of any ethical concerns raised by this work.

REPRODUCIBILITY STATEMENT

To support reproducibility, we describe the CNPC architecture and interventional inference procedure in Section 4, with further details on causal probabilistic circuits in Appendix B.1. Our assumptions are stated in Section 3.1, and the derivation of the interventional class distribution and proofs of the theoretical results are provided in Appendices C and E, respectively. The datasets, data splits, and experimental protocol are described in Section 6.1, with data generation and preprocessing procedures detailed in Appendix D.1.1. Appendix D.1.2 documents model architectures, training hyperparameters, validation-based selection of the PoE weight, random seeds, computing hardware, and evaluation metrics; dataset-specific intervention orders are listed in Appendix D.1.1. We also provide code in the supplementary material to support reproducibility.

REFERENCES

Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*, pp. 5948–5955. AAAI Press, 2023.

Mark Chavira and Adnan Darwiche. Compiling bayesian networks using variable elimination. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2443–2449, 2007.

Weixin Chen and Han Zhao. Understanding and improving adversarial robustness of neural probabilistic circuits. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Weixin Chen, Simon Yu, Huajie Shao, Lui Sha, and Han Zhao. Neural probabilistic circuits: An overview. In *Eighth Workshop on Tractable Probabilistic Modeling*, 2025.

Weixin Chen, Simon Yu, Huajie Shao, Lui Sha, and Han Zhao. Neural probabilistic circuits: Enabling compositional and interpretable predictions through logical reasoning. *Machine Learning*, 115(9):207, 2026.

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

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf*, pp. 6, 2020.

Adnan Darwiche. A differential approach to inference in bayesian networks. *J. ACM*, 50(3):280–305, 2003.

Adnan Darwiche. *Modeling and reasoning with Bayesian networks*. Cambridge university press, 2009.

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

Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018.

Giovanni de Felice, Arianna Casanova Flores, Francesco De Santis, Silvia Santini, Johannes Schneider, Pietro Barbiero, and Alberto Termine. Causally reliable concept bottleneck models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE Computer Society, 2009.

Gabriele Dominici, Pietro Barbiero, Mateo Espinosa Zarlenga, Alberto Termine, Martin Gjoreski, Giuseppe Marra, and Marc Langheinrich. Causal concept graph models: Beyond causal opacity in deep learning. In *The Thirteenth International Conference on Learning Representations (ICLR)*. OpenReview.net, 2025.

Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nat.*, 542(7639):115–118, 2017.

Christian Genest and James V Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

Robert Gens and Pedro M. Domingos. Learning the structure of sum-product networks. In *ICML*, pp. 873–880, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

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

Tom Heskes. Selecting weighting factors in logarithmic opinion pools. In *Advances in Neural Information Processing Systems 10 (NeurIPS)*, pp. 266–272. The MIT Press, 1997.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 2020.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Steffen L Lauritzen and David J Spiegelhalter. 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.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 3730–3738, 2015.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 3753–3763, 2018.

Tuomas P. Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations (ICLR)*. OpenReview.net, 2023.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Judea Pearl. Causal inference in statistics: An overview. 2009.

Hoifung Poon and Pedro M. Domingos. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 337–346. AUAI Press, 2011.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005.

Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of statistical software*, 35:1–22, 2010.

Ivaxi Sheth, Aamer Abdul Rahman, Laya Rafiee Sevyeri, Mohammad Havaei, and Samira Ebrahimi Kahou. Learning from uncertain concepts via test time interventions. In *Workshop on trustworthy and socially responsible machine learning, NeurIPS*, 2022.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31504–31520. PMLR, 2023.

Nishad Singhi, Jae Myung Kim, Karsten Roth, and Zeynep Akata. Improving intervention efficacy via concept realignment in concept bottleneck models. In *European Conference on Computer Vision (ECCV)*, pp. 422–438. Springer, 2024.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

Divyansh Srivastava, Ge Yan, and Lily Weng. VLG-CBM: training concept bottleneck models with vision-language guidance. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3619–3629. Computer Vision Foundation / IEEE, 2021.

David Steinmann, Wolfgang Stammer, Felix Friedrich, and Kristian Kersting. Learning to intervene on concept bottlenecks. In *Forty-first International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46556–46571. PMLR / OpenReview.net, 2024.

Jin Tian and Judea Pearl. A general identification condition for causal effects. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI)*, pp. 567–573. AAAI Press / The MIT Press, 2002.

Moritz Vandenhirtz, Sonia Laguna, Ricards Marcinkevics, and Julia E. Vogt. Stochastic concept bottleneck models. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Antonio Vergari, YooJung Choi, Robert Peharz, and Guy Van den Broeck. Probabilistic circuits: Representations, inference, learning and applications. *AAAI Tutorial*, 2020.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

Mert Yüksekgönül, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations (ICLR)*. OpenReview.net, 2023.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frédéric Precioso, Stefano Melacci, Adrian Weller, Pietro Lió, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Nevin Lianwen Zhang and David L. Poole. Exploiting causal independence in bayesian network inference. *J. Artif. Intell. Res.*, 5:301–328, 1996.

Han Zhao, Pascal Poupart, and Geoffrey J. Gordon. A unified approach for learning the parameters of sum-product networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 433–441, 2016.

Zhiyu Zhu, Jiayu Zhang, Zhibo Jin, Xinyi Wang, Fang Chen, Przemyslaw Biecek, and Jianlong Zhou. Causal intervention in concept bottleneck models. In *European Conference on Computer Vision (ECCV)*, pp. 172–187. Springer, 2026.

Appendix Contents

A Discussion **14**

B Preliminaries **15**

B.1 Causal probabilistic circuits 15

C Derivation of the interventional class distribution **15**

D Experiments **16**

D.1 Experimental settings 16

D.1.1 Datasets 16

D.1.2 Implementation details 17

D.1.3 Evaluation metrics 18

D.2 More experimental results 19

D.2.1 Intervention propagation at each partial count 19

D.2.2 Mean attribute accuracy 20

D.2.3 Circuit statistics 20

D.2.4 Performance with learned causal graphs 21

D.2.5 Performance under domain shifts 22

D.2.6 Performance with a fixed PoE weight 23

D.3 Ablation studies 25

D.3.1 Effect of the PoE weight 25

E Theoretical analysis **25**

A Discussion

In this section, we discuss the limitations of the current approach and directions for future work.

Following standard CBMs, CNPC assumes access to data annotated with both attribute and class labels. Reliable attribute annotations can be costly, especially when domain expertise is required. Methods that reduce concept supervision using pretrained language or vision-language models (Oikarinen et al., 2023; Yüksekgönül et al., 2023; Srivastava et al., 2024) could therefore be integrated with CNPC to reduce this annotation burden.

By following the design of NPC (Chen et al., 2026), CNPC also inherits its inference bottleneck: the overall predictor explicitly sums over joint attribute assignments, whose number grows exponentially with the number of attributes. This limits scalability despite individual marginal and conditional queries to the underlying PC being linear in circuit size. Improving the scalability of the NPC family is therefore an important direction, for example by exploiting additional structure or using approximate or amortized inference.

Our experiments follow de Felice et al. (2025) and prioritize interventions by node depth in the causal graph. While this provides a consistent evaluation protocol, it may not maximize the benefit of a limited intervention budget. The value of correcting an attribute also depends on its prediction reliability, influence on the class label, and previous corrections. Future work could formulate intervention selection as a budgeted optimization problem that estimates the expected benefit of each correction and, when annotation costs vary, also accounts for its cost.

B PRELIMINARIES

B.1 CAUSAL PROBABILISTIC CIRCUITS

In this section, we elaborate on how VE schedules the factor operations (multiplication and sum-out) within a network polynomial, which in turn determines the structure of the compiled arithmetic circuit.

We continue the example from Section 3.3. That is, we consider three variables with the graph structure $V_2 \leftarrow V_1 \rightarrow V_3$, and apply VE with the variable elimination order $[V_2,\ V_3,\ V_1]$. Table 3 summarizes the VE schedule. At each step, VE collects all current factors that involve the variable to be eliminated, multiplies them, and then sums out that variable. This produces a new factor over the remaining variables, which is carried forward to subsequent steps. Specifically, the new factors produced at different steps are: $\tau_1(V_1) = \sum_{V_2} \phi_{V_2}(V_1, V_2)\ \lambda_{V_2}(V_2)$, $\tau_2(V_1) = \sum_{V_3} \phi_{V_3}(V_1, V_3)\ \lambda_{V_3}(V_3)$, and the network polynomial $f = \sum_{V_1} \phi_{V_1}(V_1)\ \tau_1(V_1)\ \tau_2(V_1)\ \lambda_{V_1}(V_1)$.

Finding the optimal variable elimination order is known to be NP-hard. In our experiments, the variable elimination order is determined using the MinFill heuristic, which greedily selects the variable that adds the fewest new edges to the graph when eliminated.

Subsequently, the compiled arithmetic circuit is obtained by mapping each sum-out operation to a sum node and each multiplication operation to a product node, while leaf nodes correspond to symbolic parameters (the ϕ entries) and indicators (the λ entries).

Table 3: A run of VE for the network polynomial with the elimination order $[V_2,\ V_3,\ V_1]$.

Step	Variable eliminated	Factors used	Variables involved	New factor
1	V_2	$\phi_{V_2}(V_1, V_2),\ \lambda_{V_2}(V_2)$	$V_1,\ V_2$	$\tau_1(V_1)$
2	V_3	$\phi_{V_3}(V_1, V_3),\ \lambda_{V_3}(V_3)$	$V_1,\ V_3$	$\tau_2(V_1)$
3	V_1	$\phi_{V_1}(V_1),\ \tau_1(V_1),\ \tau_2(V_1),\ \lambda_{V_1}(V_1)$	V_1	f

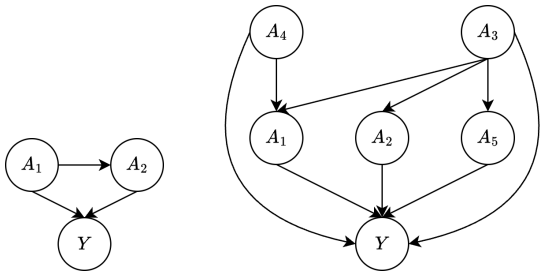
C DERIVATION OF THE INTERVENTIONAL CLASS DISTRIBUTION

Let $a_{1:K}|_{\mathrm{Pa}_{\mathcal{C}}(Y)}$ denote the instantiation of $\mathrm{Pa}_{\mathcal{C}}(Y)$ induced by $a_{1:K}$.

Following Assumption 1, Assumption 2 together with the local Markov property, and the invariance of causal mechanisms, Equation (3) can be derived as follows:

$$
\begin{aligned}
\mathbb{P}^{\mathrm{do}_j}(Y \mid X) &= \sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}(Y \mid A_{1:K} = a_{1:K}, X) \cdot \mathbb{P}^{\mathrm{do}_j}(A_{1:K} = a_{1:K} \mid X), \\
&= \sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}(Y \mid A_{1:K} = a_{1:K}) \cdot \mathbb{P}^{\mathrm{do}_j}(A_{1:K} = a_{1:K} \mid X), \\
&= \sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}(Y \mid \mathrm{Pa}_{\mathcal{C}}(Y) = a_{1:K}|_{\mathrm{Pa}_{\mathcal{C}}(Y)}) \cdot \mathbb{P}^{\mathrm{do}_j}(A_{1:K} = a_{1:K} \mid X), \\
&= \sum_{a_{1:K}} \mathbb{P}(Y \mid \mathrm{Pa}_{\mathcal{C}}(Y) = a_{1:K}|_{\mathrm{Pa}_{\mathcal{C}}(Y)}) \cdot \mathbb{P}^{\mathrm{do}_j}(A_{1:K} = a_{1:K} \mid X), \\
&= \sum_{a_{1:K}} \mathbb{P}(Y \mid A_{1:K} = a_{1:K}) \cdot \mathbb{P}^{\mathrm{do}_j}(A_{1:K} = a_{1:K} \mid X).
\end{aligned}
$$

The first equality follows from the law of total probability. The second equality uses Assumption 1, which states that $Y \perp X \mid A_{1:K}$. The third equality follows from Assumption 2 and the local Markov property: since $\mathrm{Pa}_{\mathcal{C}}(Y) \subseteq \{A_{1:K}\} \subseteq \mathrm{ND}_{\mathcal{C}}(Y)$ and Y is conditionally independent of the non-descendants given $\mathrm{Pa}_{\mathcal{C}}(Y)$, conditioning on all attributes can be reduced to conditioning on the causal parents of Y. The fourth equality uses the invariance of causal mechanisms: the intervention



(a) MNISTAdd (b) CelebA

Figure 5: Causal graphs over attributes $A_{1:K}$ and the class label Y. **Left:** the causal graph for MNISTAdd, with attributes A_1: `first digit` and A_2: `second digit`. **Right:** the causal graph for CelebA, with attributes A_1: `Attractive`, A_2: `Mouth_Slightly_Open`, A_3: `Smiling`, A_4: `Wearing_Lipstick`, and A_5: `High_Cheekbones`.

do_j acts on A_j and does not change the causal mechanism of Y (*i.e.*, the distribution of Y given its causal parents). The last equality again follows from Assumption 2 and the local Markov property.

D Experiments

D.1 Experimental settings

D.1.1 Datasets

Here, we provide further details regarding the data processing procedure for each dataset.

Asia. Following de Felice et al. (2025), we train an autoencoder to map the attribute labels into an embedding, which serves as the input to CBMs. Specifically, this autoencoder comprises two encoder layers and two decoder layers, with a latent dimension of 32. It takes a vector of all attribute labels for a given instance as input and is trained exclusively on the training split using mean squared error (MSE) loss to reconstruct this vector. After training, the encoder is frozen and used to generate embeddings for all three splits. The final instance embedding is constructed by combining 50% of the encoder's output with 50% noise sampled from a standard normal distribution. This noise is injected to ensure the resulting embeddings are non-trivial representations of attributes. Finally, the embeddings are standardized using means and standard deviations computed exclusively from the training split; the same transformation is applied to the validation and test splits. The cumulative intervention order on this dataset is

$$(\texttt{asia}, \texttt{smoke}, \texttt{bronc}, \texttt{tub}, \texttt{lung}, \texttt{either}, \texttt{xray}).$$

Sachs. The Sachs dataset is processed using the identical procedure described for Asia. The cumulative intervention order on this dataset is

$$(\texttt{PKC}, \texttt{Plcg}, \texttt{PKA}, \texttt{PIP3}, \texttt{Raf}, \texttt{Jnk}, \texttt{P38}, \texttt{PIP2}, \texttt{Mek}, \texttt{Erk}).$$

MNISTAdd. The data-generating process for an instance in the MNISTAdd dataset proceeds in three steps. First, the value for attribute A_1 (the first digit) is drawn from a uniform distribution over 0–9. Second, the value for attribute A_2 (the second digit) is drawn from a skewed conditional distribution: with 80% probability, it is assigned the value $(A_1 + 1)\%10$, while the remaining 20% probability is distributed equally among the other possible digits. Finally, the class label is defined as their sum, and the instance image is formed by concatenating randomly sampled MNIST images corresponding to the sampled digit values. The causal graph depicting the relationships between the attributes and the class is illustrated in Figure 5a. The cumulative intervention order on this dataset is

$$(\texttt{first digit}, \texttt{second digit}).$$

CelebA. In this paper, we focus on the five most balanced attributes in CelebA: `Attractive`, `Mouth_Slightly_Open`, `Smiling`, `Wearing_Lipstick`, and `High_Cheekbones`. We define the class label as a unique combination of these binary attributes, resulting in 32 distinct classes. Furthermore, we manually annotate the underlying causal graph over these attributes and the class label, as illustrated in Figure 5b. The cumulative intervention order on this dataset is

(`Smiling`, `Wearing_Lipstick`, `Attractive`, `Mouth_Slightly_Open`, `High_Cheekbones`).

Licenses and terms of use. The Asia and Sachs datasets are generated by sampling from Bayesian networks provided by `bnlearn`, an R package distributed under GPL-2 or GPL-3. The MNISTAdd dataset is constructed from MNIST images, which are distributed under the Creative Commons Attribution-ShareAlike 3.0 license. For CelebA, we use the images and attribute annotations only for non-commercial research purposes, in accordance with the official CelebA terms of use.

D.1.2 Implementation details

To ensure the reliability and reproducibility of our results, we conduct all experiments across three independent trials using the random seeds 42, 52, and 62. All experiments are conducted on NVIDIA RTX A6000 GPUs, each with 48 GB of memory. Unless otherwise stated, for datasets containing image inputs, the images are resized to $224\times224\times3$.

NPC. The attribute predictor of NPC is a multi-task MLP with a shared hidden layer of dimension 128, a ReLU activation, and an independent linear classification head for each attribute. We pretrain this predictor for 100 epochs with a batch size of 256, minimizing the average normalized cross-entropy loss across attributes from Chen et al. (2026) using SGD with a learning rate of 0.01, momentum of 0.9, and weight decay of 4×10^{-5}. Its label predictor is a PC whose structure is learned using LearnSPN (Gens & Domingos, 2013) and whose parameters are optimized using CCCP (Zhao et al., 2016). Following Chen et al. (2026), we then fine-tune the attribute predictor within NPC by minimizing the class negative log-likelihood while keeping the PC fixed. This stage runs for 100 epochs with a batch size of 256 using SGD with an initial learning rate of 0.001, momentum of 0.9, and weight decay of 4×10^{-5}.

CNPC. In principle, CNPC can follow the same training procedure as NPC, except that the causal PC is constructed by compiling the causal graph and estimating its parameters by maximum likelihood. For a controlled comparison in our experiments, CNPC reuses the neural attribute predictor obtained after NPC's final fine-tuning stage and keeps it frozen, without additional fine-tuning within CNPC. Thus, NPC and CNPC differ only in the choice of PC and the inference formula used under intervention. We select α by maximizing overall accuracy—the mean of the accuracies for all K attributes (including corrected attributes) and the class label, with equal weight for each of these $K+1$ variables—on the validation set, independently for each dataset, seed, and intervention count, breaking ties in favor of the smallest α. Across seeds and partial intervention counts ($1 \leq k < K$), the selected values are $\{0, 0.1, 0.2, 0.3, 0.4\}$ for Asia, $\{0.2, 0.3, 0.4\}$ for Sachs, $\{0, 0.2\}$ for MNISTAdd, and $\{0.1, 0.2, 0.3\}$ for CelebA. For full interventions, the saved selections are $\alpha = 0$ for all datasets and seeds.

Vanilla CBM. In the vanilla CBM, the concept predictor is implemented as an MLP with a single hidden layer of dimension 128 and a ReLU activation, and the label predictor as a linear layer. The two predictors are trained independently, as Koh et al. (2020) show that this training paradigm achieves the highest intervention efficiency among the training paradigms they consider. The concept predictor is trained with binary cross-entropy over attribute labels, and the label predictor is trained separately with cross-entropy over class labels using ground-truth concepts. Both stages are trained for 100 epochs with a batch size of 256, using SGD with a learning rate of 0.01, momentum of 0.9, and weight decay of 4e-5.

CEM. In CEM, the concept predictor consists of a linear layer with a ReLU activation, followed by a set of concept blocks that produce concept probabilities and concept embeddings. The resulting high-dimensional concept embeddings are concatenated and passed through a linear layer to produce the final class predictions. We train CEM end-to-end for 100 epochs with a batch size of 256. The training loss is the sum of the concept prediction loss and the class prediction loss, with the concept

loss weight set to 1. We optimize the model using SGD with a learning rate of 0.01, a momentum of 0.9, and a weight decay of 4e-5. We also adopt RandInt, a regularization strategy proposed by Zarlenga et al. (2022), which randomly applies independent concept interventions during training with probability 0.25 to improve intervention efficiency at inference time.

C^2BM and SCBM. To faithfully reproduce the results reported by de Felice et al. (2025), we implement C^2BM and SCBM using the open-source code[7]. For C^2BM, the input is encoded by a one-hidden-layer MLP, and each attribute and the class variable are further modeled by a block that produces embeddings and categorical probabilities. The model is trained end-to-end with the sum of the attribute prediction loss and the class prediction loss, with the attribute loss weight set to 0.8. During training, independent attribute interventions are randomly applied with probability 0.8.

For SCBM, the input is encoded by a one-hidden-layer MLP, after which the model predicts the mean of a multivariate distribution over concept logits and a learned global covariance matrix. The model is trained end-to-end with a loss consisting of a class prediction loss, a Monte Carlo concept prediction loss based on binary cross-entropy, and an ℓ_1 penalty on the off-diagonal entries of the precision matrix, with both the concept loss weight and the precision regularization weight set to 1.

D.1.3 Evaluation metrics

Let $\{(x_i, a_{i,1:K}, y_i)\}_{i=1}^n$ denote the test set, where $a_{i,j}$ and y_i are the ground-truth labels of attribute A_j and the class for instance i. At intervention budget k, let $I_k \subseteq \{1, \dots, K\}$ contain the indices of the first k attributes in the dataset's intervention order. We denote the model's predicted attribute and class labels after these corrections by $\hat{a}_{i,j}^{(k)}$ and $\hat{y}_i^{(k)}$, respectively, with $\hat{a}_{i,j}^{(k)} = a_{i,j}$ for every $j \in I_k$. All predictions are evaluated against the test-set labels.

Let $\mathrm{De}_{\mathcal{C}}(A_\ell)$ denote the strict descendants of A_ℓ in the graph used for evaluation. We partition the uncorrected attributes into downstream and non-descendant index sets:

$$\mathcal{D}_k = \big\{ j \in \{1, \dots, K\} \setminus I_k : A_j \in \textstyle\bigcup_{\ell \in I_k} \mathrm{De}_{\mathcal{C}}(A_\ell) \big\},$$
$$\mathcal{N}_k = \{1, \dots, K\} \setminus (I_k \cup \mathcal{D}_k).$$

Non-descendant attribute accuracy averages accuracy over uncorrected attributes that are not descendants of any intervened attribute, including upstream attributes and those on other branches. *Downstream attribute accuracy* averages accuracy over uncorrected attributes that are descendants of at least one intervened attribute. Using $\mathbb{I}(\cdot)$ for the indicator function, these metrics are

$$\mathrm{Acc}_{\mathrm{ND}}(k) = \frac{1}{n|\mathcal{N}_k|} \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_k} \mathbb{I}\big(\hat{a}_{i,j}^{(k)} = a_{i,j}\big),$$

$$\mathrm{Acc}_{\mathrm{Down}}(k) = \frac{1}{n|\mathcal{D}_k|} \sum_{i=1}^{n} \sum_{j \in \mathcal{D}_k} \mathbb{I}\big(\hat{a}_{i,j}^{(k)} = a_{i,j}\big).$$

Both metrics exclude directly corrected attributes, allowing us to assess changes beyond direct correction. If the corresponding group is empty, its accuracy is undefined and omitted from averages over intervention counts.

Mean attribute accuracy averages accuracy over all K attributes, including directly corrected attributes, while *task accuracy* measures the fraction of correctly predicted class labels:

$$\mathrm{Acc}_{\mathrm{Attr}}(k) = \frac{1}{nK} \sum_{i=1}^{n} \sum_{j=1}^{K} \mathbb{I}\big(\hat{a}_{i,j}^{(k)} = a_{i,j}\big),$$

$$\mathrm{Acc}_{\mathrm{Task}}(k) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\big(\hat{y}_i^{(k)} = y_i\big).$$

All attribute averages give each attribute equal weight, irrespective of its number of categories. Since corrected attributes are assigned their ground-truth values, $\mathrm{Acc}_{\mathrm{Attr}}(K) = 1$ by construction.

[7] https://github.com/gdefe/causally-reliable-cbm

We compute these metrics separately for each seed and report their mean over three seeds as percentages. Macro-average attribute and task accuracies weight $k = 0, \ldots, K$ equally; propagation gains average the CNPC-minus-NPC differences over partial intervention counts $1 \leq k < K$ for which the metric is defined, and are reported in percentage points.

D.2 More experimental results

D.2.1 Intervention propagation at each partial count

Table 4 expands the comparison in Table 1 by reporting absolute accuracies for NPC and CNPC at every partial intervention count $1 \leq k < K$ on each dataset. For all four metrics, we report the mean and sample standard deviation across seeds 42, 52, and 62, without averaging over intervention counts. Non-descendant and downstream accuracies average over the corresponding uncorrected attributes within each seed; mean attribute accuracy includes all attributes, including those directly corrected. We use the same intervention orders and validation-selected PoE weights as in the main experiments. An empty attribute group has undefined accuracy and is marked by a dash: non-descendants remain only at $k = 1$ on Asia, Sachs, and CelebA, and never at a partial intervention count on MNISTAdd. The gains in Table 1 are recovered by subtracting NPC's mean from CNPC's at each count and averaging these differences over counts where the metric is defined, before rounding.

Table 4: Expanded intervention-propagation results (%): mean $\pm$ one sample standard deviation across three seeds for each partial intervention count k. Non-desc. and Downstream exclude directly corrected attributes; Attr. averages over all K attributes; Task measures class prediction accuracy. A dash (—) denotes an empty group.

k	Method	Non-desc.	Downstream	Attr.	Task
Asia ($K = 7$)					
1	NPC	85.10 ± 0.20	96.36 ± 0.15	92.05 ± 0.11	70.13 ± 0.35
1	CNPC	85.26 ± 0.34	96.23 ± 0.06	92.07 ± 0.12	70.57 ± 0.32
2	NPC	—	92.67 ± 0.11	94.77 ± 0.08	70.27 ± 0.50
2	CNPC	—	92.66 ± 0.10	94.76 ± 0.07	70.67 ± 0.23
3	NPC	—	96.67 ± 0.11	98.10 ± 0.06	86.20 ± 0.10
3	CNPC	—	96.49 ± 0.08	98.00 ± 0.04	86.10 ± 0.00
4	NPC	—	95.90 ± 0.15	98.24 ± 0.06	86.20 ± 0.10
4	CNPC	—	96.12 ± 0.02	98.34 ± 0.01	86.33 ± 0.06
5	NPC	—	95.03 ± 0.23	98.58 ± 0.07	86.27 ± 0.12
5	CNPC	—	97.22 ± 0.19	99.20 ± 0.05	86.80 ± 0.00
6	NPC	—	93.27 ± 0.21	99.04 ± 0.03	86.80 ± 0.00
6	CNPC	—	94.43 ± 0.38	99.20 ± 0.05	86.80 ± 0.00
Sachs ($K = 10$)					
1	NPC	80.36 ± 0.25	80.54 ± 0.04	82.43 ± 0.10	65.10 ± 0.17
1	CNPC	80.36 ± 0.25	81.42 ± 0.17	82.96 ± 0.09	65.40 ± 1.06
2	NPC	—	79.72 ± 0.10	83.77 ± 0.08	65.17 ± 0.32
2	CNPC	—	80.37 ± 0.15	84.30 ± 0.12	65.40 ± 1.06
3	NPC	—	79.16 ± 0.08	85.41 ± 0.05	66.70 ± 0.36
3	CNPC	—	80.62 ± 0.10	86.43 ± 0.07	66.50 ± 0.98
4	NPC	—	81.56 ± 0.03	88.94 ± 0.02	66.83 ± 0.40
4	CNPC	—	83.27 ± 0.06	89.96 ± 0.03	66.50 ± 0.98
5	NPC	—	82.17 ± 0.08	91.08 ± 0.04	66.50 ± 0.17
5	CNPC	—	83.71 ± 0.20	91.85 ± 0.10	66.73 ± 0.78
6	NPC	—	82.41 ± 0.16	92.96 ± 0.07	66.37 ± 0.29
6	CNPC	—	83.73 ± 0.06	93.49 ± 0.03	66.73 ± 0.78
7	NPC	—	82.30 ± 0.09	94.69 ± 0.03	66.50 ± 0.10
7	CNPC	—	84.00 ± 0.18	95.20 ± 0.05	66.90 ± 0.75
8	NPC	—	78.58 ± 0.25	95.72 ± 0.05	66.60 ± 0.00
8	CNPC	—	81.13 ± 0.16	96.23 ± 0.03	66.90 ± 0.75
9	NPC	—	74.97 ± 0.55	97.50 ± 0.06	66.53 ± 0.40
9	CNPC	—	77.40 ± 0.10	97.74 ± 0.01	68.37 ± 0.68

Continued on next page

k	Method	Non-desc.	Downstream	Attr.	Task
MNISTAdd ($K = 2$)					
1	NPC	—	97.09 ± 0.08	98.54 ± 0.04	97.09 ± 0.08
1	CNPC	—	97.17 ± 0.03	98.59 ± 0.02	97.17 ± 0.03
CelebA ($K = 5$)					
1	NPC	90.93 ± 0.16	83.67 ± 0.21	88.39 ± 0.15	54.28 ± 0.41
1	CNPC	90.93 ± 0.11	83.91 ± 0.10	88.53 ± 0.07	54.93 ± 0.12
2	NPC	—	83.67 ± 0.21	90.20 ± 0.12	58.32 ± 0.40
2	CNPC	—	84.11 ± 0.12	90.46 ± 0.07	59.26 ± 0.24
3	NPC	—	86.66 ± 0.27	94.66 ± 0.11	75.56 ± 0.49
3	CNPC	—	87.11 ± 0.17	94.84 ± 0.07	76.34 ± 0.27
4	NPC	—	84.87 ± 0.28	96.97 ± 0.06	84.87 ± 0.28
4	CNPC	—	85.66 ± 0.08	97.13 ± 0.02	85.66 ± 0.08

D.2.2 Mean attribute accuracy

Figure 6 reports mean attribute accuracy for all six methods on Asia, Sachs, MNISTAdd, and CelebA at each intervention count, complementing the task accuracy curves in Figure 4. We use the same cumulative intervention order as in Section 6.1 and average accuracy over all attributes, including those directly corrected. Consequently, all methods reach 100% mean attribute accuracy at full intervention; differences at partial intervention counts reflect prediction accuracy on the remaining uncorrected attributes.

CNPC exceeds NPC in mean attribute accuracy at every partial intervention count on Sachs, MNISTAdd, and CelebA. On Asia, it improves on NPC at $k = 1, 4, 5, 6$, with small deficits at $k = 2, 3$. Across all methods, CNPC leads on MNISTAdd at every count, including ties at the endpoints, while C^{2}BM leads at every partial count on Sachs and Vanilla CBM leads on CelebA before full intervention. These curves provide the per-count comparisons underlying the macro-average attribute accuracies in Table 2.

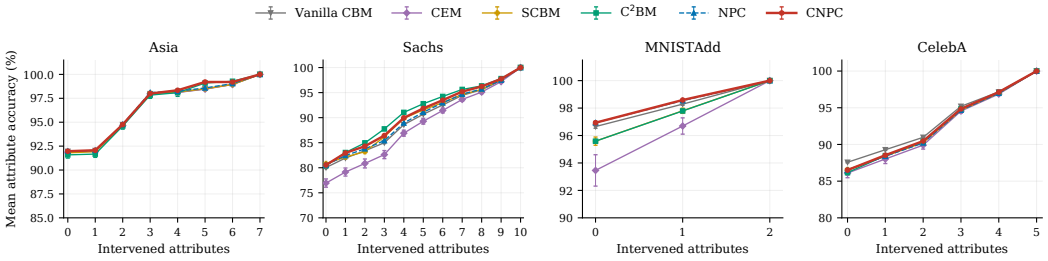


Figure 6: Mean attribute accuracy for all six methods across intervention counts. Accuracy is averaged over all attributes, including directly corrected attributes. Points show means over three random seeds; error bars indicate one sample standard deviation. Both endpoints are included, and axis ranges vary by panel.

D.2.3 Circuit statistics

Table 5 reports graph statistics, compiled circuit sizes, and CPU runtimes for the four datasets. The graph-variable count includes all attributes and the class variable, and $\prod_v |\mathcal{X}_v|$ is the product of their cardinalities. Circuit size refers to the number of edges in the compiled causal PC. Compilation includes circuit construction and traversal setup. The joint-query runtime measures the evaluation of the observational joint probability with all attributes and the class fixed to state zero. This includes setting the categorical leaf values and traversing the circuit; neural prediction and PoE computation are not included.

Circuit size not only depends on graph structure, but also on variable cardinalities and the variable-elimination order. For example, MNISTAdd has fewer graph variables than Asia but a larger com-

Table 5: Graph statistics, circuit sizes, and CPU runtimes. Compilation time is the median of 10 independent compilations; joint-query time is the median of 100 batch-size-one evaluations after 10 warm-up queries. Timings cover the circuit operations described in the text.

Dataset	Graph vars.	$\prod_v \lvert\mathcal{X}_v\rvert$	Graph edges	Circuit size	Compilation (s)	Joint query (s)
Asia	8	2^8 =256	8	1,816	0.0206	0.0181
Sachs	11	3^{11} =177,147	17	16,703	0.2497	0.1548
MNISTAdd	3	$10 \times 10 \times 19$ =1,900	3	10,526	0.0637	0.0772
CelebA	6	$2^5 \times 32$ =1,024	9	8,780	0.0946	0.0868

Table 6: Accuracy gains of CNPC over NPC on Asia with the GES graph, in pp. Differences are averaged equally over partial intervention counts $k = 1, \ldots, 6$, omitting empty groups. Non-descendant accuracy is defined only at $k = 1, 2$, and downstream accuracy only at $k = 2, \ldots, 6$. Both groups exclude directly corrected attributes; Attr. includes all seven attributes.

Dataset	Δ Non-desc.	Δ Downstream	Δ Attr.	Δ Task
Asia (GES)	-0.04	+0.76	+0.15	+0.32

piled circuit. Circuit size does not completely order the runtimes. Among the four datasets, Asia has both the smallest circuit and the shortest compilation and joint-query times, while Sachs has the largest circuit and longest times. However, MNISTAdd has a larger circuit size than CelebA but compiles faster and answers the joint query faster. Since these factors vary jointly across only four datasets, these results do not isolate their individual effects or establish a general scaling law.

D.2.4 Performance with learned causal graphs

The main experiments assume that a causal graph is available as prior knowledge. We now consider the setting in which such expert knowledge is unavailable and a plausible structural prior must be learned from data. On Asia, we apply Greedy Equivalence Search (GES) (Chickering, 2002) with the discrete Bayesian information criterion to the 8,000 training examples, using all seven attributes and the class label, without imposing expert constraints or using validation or test data for discovery. The resulting directed acyclic graph has eight nodes and seven edges. Compared with the original graph, it removes `asia` $\rightarrow$ `tub` and reverses `smoke` $\rightarrow$ `bronc`, giving a structural Hamming distance of two when each reversal counts as one edit. The remaining six edges are unchanged; in particular, `bronc` and `either` remain the parents of the class variable `dysp`, while `asia` becomes isolated and is retained in the experiment.

We share this learned graph across all methods and three random seeds. Sorting attributes by their longest-path depth, with dataset-column order breaking ties, gives the cumulative intervention order

(`asia`, `tub`, `bronc`, `smoke`, `lung`, `either`, `xray`).

For CNPC, we compile the learned structure into a causal PC and fit its parameters on the training split. We reuse the neural attribute predictor obtained after NPC's joint training and select α by validation overall accuracy separately for each seed and intervention count. We also reuse the existing NPC, Vanilla CBM, CEM, and SCBM checkpoints. By contrast, C^{2}BM is retrained with the learned structure because its model architecture depends on that structure. This illustrates a practical benefit of CNPC's compositional architecture: replacing the structural prior only requires recompiling the causal module without retraining the neural attribute predictor.

We use the same evaluation protocol as in Section 6.2, defining descendant groups using the learned graph. Table 6 reports CNPC's gains over NPC at matched partial intervention counts, while Table 7 averages attribute and task accuracy equally over $k = 0, \ldots, 7$. Figure 7 shows both metrics at each count.

Relative to NPC, CNPC improves downstream attribute accuracy by 0.76 pp on average, while non-descendant accuracy changes by only -0.04 pp. Its mean attribute and task accuracy gains over partial intervention counts are 0.15 and 0.32 pp, respectively. The largest downstream gain occurs

Table 7: Macro-average accuracy (%) on Asia with the GES graph, averaged equally over $k = 0, \ldots, 7$ and three seeds. Attr.: mean attribute accuracy; Task: task accuracy. The best and second-best means are bold and underlined, respectively, based on unrounded values.

Method	Attr.	Task
Vanilla CBM	95.89	80.33
CEM	95.85	80.28
SCBM	95.83	80.38
C^2BM	<u>96.02</u>	<u>80.48</u>
NPC	95.95	80.35
CNPC	**96.07**	**80.62**

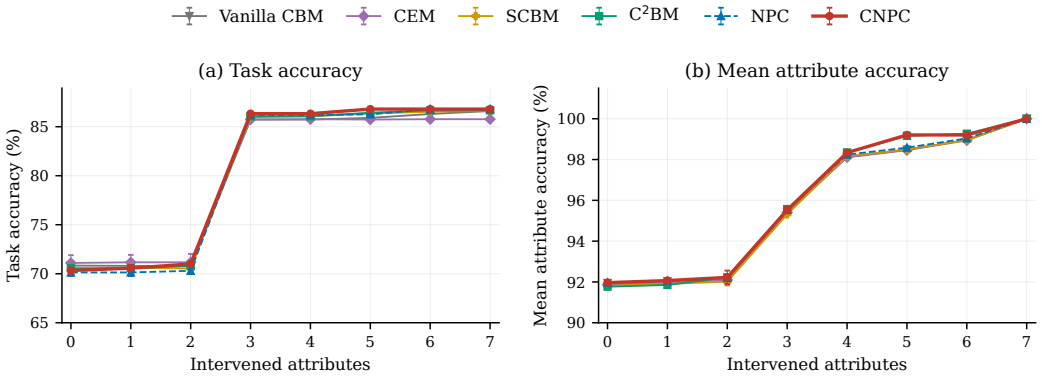


Figure 7: Task accuracy (left) and mean attribute accuracy (right) on Asia using the shared GES intervention order. Points show means over three random seeds; error bars indicate one sample standard deviation. Attribute accuracy includes directly corrected attributes. Both endpoints are included, and axis ranges differ between panels.

at $k = 5$: accuracy on the two remaining attributes, `either` and `xray`, improves by 2.18 pp, and task accuracy increases by 0.53 pp. Because the directly corrected attributes are identical for both methods, these attribute gains arise beyond direct correction. The concentration of gains on descendants, together with the small change on non-descendants, suggests that CNPC's intervention propagation is consistent with the learned structure.

Across all six methods, CNPC attains the highest macro-average attribute accuracy (96.07%) and task accuracy (80.62%), followed by C^2BM (96.02% and 80.48%). The per-count curves show that CNPC achieves the highest mean task accuracy from $k = 3$ onward, including ties with NPC at $k = 6, 7$, whereas CEM leads at $k = 0, 1, 2$. C^2BM has higher mean attribute accuracy than CNPC at $k = 2, 3, 4, 6$, so the macro-average advantage is not uniform across intervention budgets. At full intervention, every method reaches 100% mean attribute accuracy by construction, while CNPC and NPC tie at 86.80% task accuracy. Together with the results using predefined graphs, these findings support CNPC's ability to incorporate a learned structural prior without neural retraining and improve intervention efficiency, with propagation gains respecting the structural prior.

D.2.5 Performance under domain shifts

Section 5 identifies a condition under which incorporating causal evidence through the proposed PoE tightens the interventional error bound: the causal PC's interventional attribute distribution is more accurate in expected KL divergence than the clamped neural attribute predictor. To investigate the practical relevance of this analysis, we study domain shifts, a common setting in which predictions from the neural attribute predictor may be unreliable. To this end, we construct cMNISTAdd, a colored version of MNISTAdd, using the same attribute and class-generating mechanisms described in Appendix D.1.1. Each digit is assigned a fixed color in the training and validation sets, and the digit-to-color mapping is reversed in the test set. Thus, the visual association changes while the

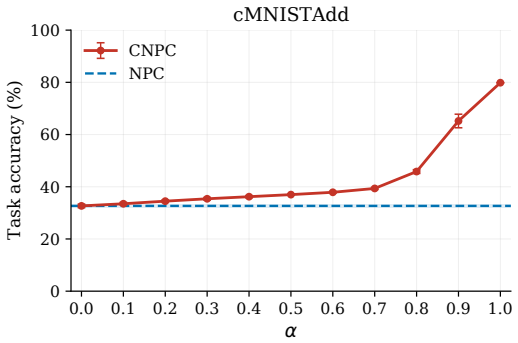


Figure 8: Task accuracy on cMNISTAdd under the reversed digit-to-color mapping, with only the first digit corrected. The solid curve shows CNPC as α varies, and the dashed horizontal line shows NPC. Points and the baseline are means over three random seeds; error bars and the shaded band indicate one sample standard deviation for CNPC and NPC, respectively.

Table 8: Accuracy gains of CNPC with fixed $\alpha = 0.1$ over NPC, averaged over partial intervention counts ($1 \leq k < K$) and three seeds, in pp. Non-descendant and downstream accuracies exclude directly corrected attributes; Attr. includes all attributes. Empty groups are omitted. Non-descendant accuracy is defined only at $k = 1$ on Asia, Sachs, and CelebA, and is undefined on MNISTAdd (—).

Dataset	Δ Non-desc.	Δ Downstream	Δ Attr.	Δ Task
Asia	+0.12	+0.50	+0.14	+0.22
Sachs	+0.00	+0.81	+0.31	+0.63
MNISTAdd	—	+0.05	+0.02	+0.05
CelebA	+0.03	+0.22	+0.09	+0.38

causal relationship between the two digits and their sum remains unchanged. We follow the training procedure in Appendix D.1.2. On all 5,000 test examples, we correct only the first digit A_1 to its ground-truth value and sweep α from 0 to 1 in increments of 0.1.

Figure 8 shows that CNPC matches NPC at $\alpha = 0$ and exceeds it at every tested positive value of α, in each of the three seeds. Mean task accuracy increases monotonically over the tested grid, from NPC's 32.67% to 33.49% at $\alpha = 0.1$, 65.19% at $\alpha = 0.9$, and 79.86% at $\alpha = 1$, yielding a maximum gain of 47.19 pp. At $\alpha = 1$, predictions rely entirely on the causal PC's interventional attribute distribution; the strong performance of this endpoint implies that the preserved attribute relationship provides more useful predictive evidence than the neural predictions under this color shift. The consistently higher task accuracy at every tested positive value of α supports the benefit of incorporating causal evidence under domain shift and is consistent with the error-bound analysis in Section 5.

D.2.6 Performance with a fixed PoE weight

In the main experiments, we select α by maximizing validation overall accuracy, independently for each dataset, seed, and intervention count, as described in Appendix D.1.2. To assess whether CNPC remains competitive with a single shared weight, we fix $\alpha = 0.1$ across all four datasets, three seeds, and positive intervention counts. We reuse the same checkpoints, test splits, baseline results, and cumulative intervention orders as in the main experiments. We follow the evaluation protocol in Section 6.2, averaging propagation gains over partial intervention counts and macro-average accuracies equally over $k = 0, \ldots, K$.

Table 8 shows that fixing α retains positive average propagation gains over NPC. Mean downstream attribute accuracy improves by 0.50, 0.81, 0.05, and 0.22 pp on Asia, Sachs, MNISTAdd, and CelebA, respectively. Non-descendant accuracy changes by only +0.12, 0.00, and +0.03 pp on the three datasets where it is defined, indicating that the larger gains are concentrated on descendants of corrected attributes. At the same partial intervention counts, mean task accuracy improves by 0.22,

Table 9: Macro-average accuracy (%) with fixed $\alpha = 0.1$ for CNPC, averaged equally over $k = 0, \ldots, K$ and three seeds. Attr.: mean attribute accuracy; Task: task accuracy. Baselines are identical to those in Table 2. The best and second-best means are bold and underlined, respectively, based on unrounded values.

	Asia		Sachs		MNISTAdd		CelebA	
Method	Attr.	Task	Attr.	Task	Attr.	Task	Attr.	Task
Vanilla CBM	96.54	80.33	89.97	63.92	98.31	79.71	**93.36**	68.31
CEM	96.49	80.36	88.47	65.55	96.72	93.32	92.58	70.61
SCBM	96.50	80.39	90.42	65.33	97.80	79.77	92.81	<u>71.58</u>
C^2BM	96.52	<u>80.53</u>	**91.27**	67.02	97.79	95.44	92.82	**72.06**
NPC	<u>96.59</u>	80.35	90.28	<u>67.32</u>	<u>98.49</u>	<u>97.06</u>	92.79	70.65
CNPC	**96.70**	**80.54**	<u>90.54</u>	**67.93**	**98.50**	**97.08**	<u>92.85</u>	70.91

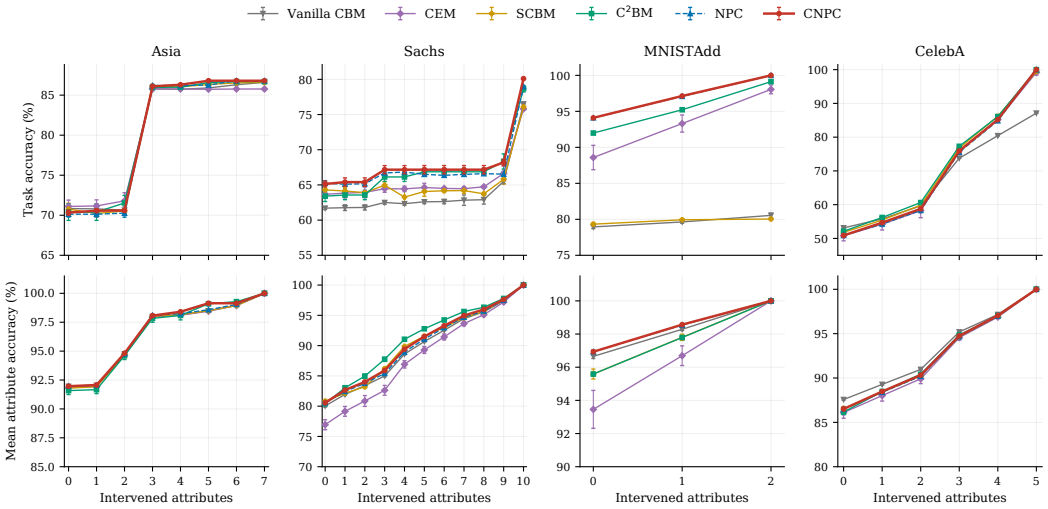


Figure 9: Task accuracy (top row) and mean attribute accuracy (bottom row) across intervention counts with fixed $\alpha = 0.1$ for CNPC and the same five baselines as in the main experiments. Attribute accuracy includes directly corrected attributes. Points show means over three seeds; error bars indicate one sample standard deviation. Both endpoints are included, and axis ranges vary by panel.

0.63, 0.05, and 0.38 pp, respectively. Thus, the positive average task gains persist when the PoE weight is shared across datasets and intervention budgets.

Across all six methods, CNPC achieves the highest macro-average task accuracy on Asia (80.54%), Sachs (67.93%), and MNISTAdd (97.08%), as shown in Table 9. On CelebA, CNPC ranks third at 70.91%, behind C^2BM (72.06%) and SCBM (71.58%), while exceeding NPC (70.65%). It also attains the highest macro-average attribute accuracy on Asia and MNISTAdd, and ranks second on Sachs and CelebA. Relative to the validation-selected configuration, fixing α changes macro-average task accuracy by -0.01, $+0.23$, -0.01, and -0.27 pp on the four datasets, respectively. Figure 9 shows the corresponding results at each intervention count. On Asia, CNPC leads task accuracy from $k = 4$ onward, including ties, but trails NPC by 0.10 pp at $k = 3$. On Sachs, it leads task accuracy at $k = 1, \ldots, 8$ and at full intervention, while C^2BM leads at $k = 9$ and in attribute accuracy at every partial count. On MNISTAdd, CNPC leads both metrics at every count, including ties at the endpoints. On CelebA, it improves on NPC in both metrics at every partial count, with task gains of 0.34–0.45 pp, although C^2BM and SCBM retain higher task accuracy at these counts. These results show that a single shared PoE weight retains CNPC's macro-average attribute and task gains over NPC on all four datasets while remaining competitive with the other baselines, without dataset- or intervention-budget-specific selection of α.

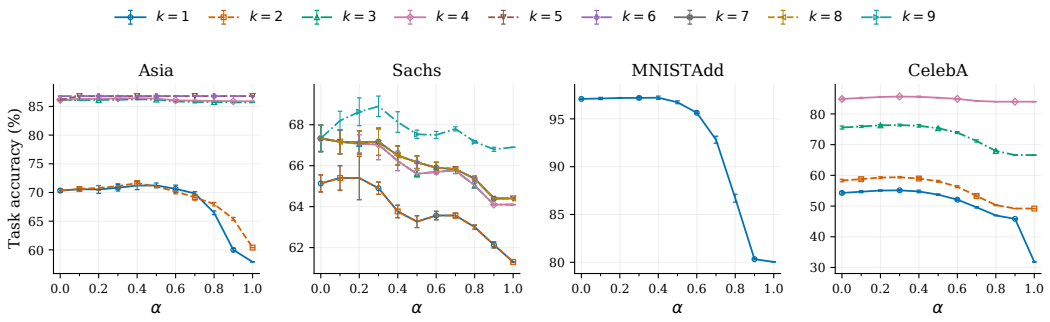


Figure 10: Effect of α on task accuracy of CNPC. Each curve represents a partial intervention count k $(1 \leq k < K)$. Points are means over three random seeds, and error bars show one sample standard deviation. Axis ranges vary by dataset.

D.3 Ablation studies

D.3.1 Effect of the PoE weight

In the main experiments, we select the PoE weight α based on validation performance. Here, we vary α to analyze how balancing the neural predictor's input-dependent evidence against the causal PC's interventional marginal affects task accuracy. To this end, we sweep α from 0 to 1 in increments of 0.1 on all four datasets, evaluating each partial intervention count $1 \leq k < K$. At each (k, α) pair, we report the mean task accuracy and one sample standard deviation over three random seeds. Figure 10 presents one curve per intervention count, yielding six curves for Asia, nine for Sachs, one for MNISTAdd, and four for CelebA.

The effect of α depends on both the dataset and the intervention count. On Asia, mean task accuracy peaks at $\alpha = 0.5$ for $k = 1$ and at $\alpha = 0.4$ for $k = 2, 3, 4$, before declining at larger weights. For $k = 5$, it reaches 86.80% at $\alpha = 0.1$ and remains constant over the remaining tested weights; for $k = 6$, it is 86.80% throughout the sweep. Sachs exhibits a different pattern: the highest means occur at $\alpha = 0.1, 0.2$ for $k = 1, 2$, at $\alpha = 0$ for $k = 3, \ldots, 8$, and at $\alpha = 0.3$ for $k = 9$. In the last case, task accuracy rises from 67.33% at $\alpha = 0$ to 68.90% at $\alpha = 0.3$. On MNISTAdd, the mean peaks at $\alpha = 0.4$, increasing modestly from 97.09% at $\alpha = 0$ to 97.21%, before falling to 80.04% at $\alpha = 1$. On CelebA, all four curves peak at $\alpha = 0.3$, with gains of 0.79–1.07 pp over $\alpha = 0$, and decline at larger weights. These results suggest that incorporating causal evidence generally improves the effectiveness of attribute interventions for class prediction, with the largest gains typically occurring at relatively small values of α (≤ 0.5).

E Theoretical analysis

In this section, we provide more theoretical results and elaborate on the proofs omitted in Section 5.

Theorem 5. *The expected prediction error of NPC (CNPC), measured by KL divergence, is upper-bounded by the sum of the expected prediction error of the attribute predictor and that of the PC (causal PC). Specifically,*

$$\mathbb{E}_{X\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid X) \| \, \mathbb{P}_{\theta,w}(Y \mid X)\right)\right] \leq \mathbb{E}_{X\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(A_{1:K} \mid X) \| \, \mathbb{P}_{\theta}(A_{1:K} \mid X)\right)\right] + \mathbb{E}_{A_{1:K}\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \, \mathbb{P}_{w}(Y \mid A_{1:K})\right)\right].$$

Equality holds if for each (x, y), there exists a constant $c(x, y)$ such that $\frac{\mathbb{P}_*(a_{1:K}|x)\mathbb{P}_*(y|a_{1:K})}{\mathbb{P}_\theta(a_{1:K}|x)\mathbb{P}_w(y|a_{1:K})} = c(x, y)$ for all $a_{1:K}$.

Proof. We proceed in three steps.

Step 1. Show that

$$\mathbb{E}_{X\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid X) \| \, \mathbb{P}_{\theta,w}(Y \mid X)\right)\right] = D_{\mathrm{KL}}\left(\mathbb{P}_*(X, Y) \| \, \mathbb{P}_{\theta,w}(X, Y)\right),$$

where $\mathbb{P}_{\theta,w}(X,Y) = \mathbb{P}_*(X)\,\mathbb{P}_{\theta,w}(Y \mid X)$.

Rewriting $\mathbb{P}_*(X,Y)$ as $\mathbb{P}_*(X)\,\mathbb{P}_*(Y \mid X)$, we can expand the RHS as follows.

$$\begin{aligned}
D_{\mathrm{KL}}\left(\mathbb{P}_*(X,Y) \| \, \mathbb{P}_{\theta,w}(X,Y)\right) &= \sum_{x,y} \mathbb{P}_*(x,y)\ \log \frac{\mathbb{P}_*(y \mid x)}{\mathbb{P}_{\theta,w}(y \mid x)} \\
&= \sum_{x} \mathbb{P}_*(x) \sum_{y} \mathbb{P}_*(y \mid x)\ \log \frac{\mathbb{P}_*(y \mid x)}{\mathbb{P}_{\theta,w}(y \mid x)} \\
&= \sum_{x} \mathbb{P}_*(x)\ D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid x) \| \, \mathbb{P}_{\theta,w}(Y \mid x)\right) \\
&= \mathbb{E}_{X \sim \mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid X) \| \, \mathbb{P}_{\theta,w}(Y \mid X)\right)\right].
\end{aligned}$$

Step 2. Show that

$$D_{\mathrm{KL}}\left(\mathbb{P}_*(X,Y) \| \, \mathbb{P}_{\theta,w}(X,Y)\right) \le D_{\mathrm{KL}}\left(\mathbb{P}_*(X,A_{1:K},Y) \| \, \mathbb{P}_{\theta,w}(X,A_{1:K},Y)\right),$$

where $\mathbb{P}_{\theta,w}(X,A_{1:K},Y) = \mathbb{P}_*(X)\,\mathbb{P}_\theta(A_{1:K} \mid X)\,\mathbb{P}_w(Y \mid A_{1:K})$.

Under Assumption 1, $\mathbb{P}_{\theta,w}(x,a_{1:K},y)$ can be written as $\mathbb{P}_*(x)\,\mathbb{P}_{\theta,w}(a_{1:K},y \mid x)$; thus, the marginal $\sum_{a_{1:K}} \mathbb{P}_{\theta,w}(x,a_{1:K},y)$ equals $\mathbb{P}_{\theta,w}(x,y)$.

Following this, we can expand the LHS as follows.

$$\begin{aligned}
D_{\mathrm{KL}}\left(\mathbb{P}_*(X,Y) \| \, \mathbb{P}_{\theta,w}(X,Y)\right) &= \sum_{x,y} \mathbb{P}_*(x,y)\ \log \frac{\mathbb{P}_*(x,y)}{\mathbb{P}_{\theta,w}(x,y)} \\
&= \sum_{x,y} \left(\sum_{a_{1:K}} \mathbb{P}_*(x,a_{1:K},y) \right)\ \log \frac{\sum_{a_{1:K}} \mathbb{P}_*(x,a_{1:K},y)}{\sum_{a_{1:K}} \mathbb{P}_{\theta,w}(x,a_{1:K},y)} \\
\text{using the log-sum inequality} &\le \sum_{x,y} \sum_{a_{1:K}} \mathbb{P}_*(x,a_{1:K},y)\ \log \frac{\mathbb{P}_*(x,a_{1:K},y)}{\mathbb{P}_{\theta,w}(x,a_{1:K},y)} \\
&= D_{\mathrm{KL}}\left(\mathbb{P}_*(X,A_{1:K},Y) \| \, \mathbb{P}_{\theta,w}(X,A_{1:K},Y)\right).
\end{aligned}$$

Equality holds iff for each (x,y), $\frac{\mathbb{P}_*(x,a_{1:K},y)}{\mathbb{P}_{\theta,w}(x,a_{1:K},y)}$ is some constant $c(x,y)$ for all $a_{1:K}$ with $\mathbb{P}_*(x,a_{1:K},y),\ \mathbb{P}_{\theta,w}(x,a_{1:K},y) > 0$.

Step 3. Decompose $D_{\mathrm{KL}}\left(\mathbb{P}_*(X,A_{1:K},Y) \| \, \mathbb{P}_{\theta,w}(X,A_{1:K},Y)\right)$.

Under Assumption 1, $\mathbb{P}_*(X,A_{1:K},Y)$ can be written as $\mathbb{P}_*(X)\,\mathbb{P}_*(A_{1:K} \mid X)\,\mathbb{P}_*(Y \mid A_{1:K})$.

Following this, we can decompose the target term as follows.

$$\begin{aligned}
D_{\mathrm{KL}}\left(\mathbb{P}_*(X,A_{1:K},Y) \| \, \mathbb{P}_{\theta,w}(X,A_{1:K},Y)\right) &= \sum_{x,a_{1:K},y} \mathbb{P}_*(x,a_{1:K},y)\ \log \frac{\mathbb{P}_*(x,a_{1:K},y)}{\mathbb{P}_{\theta,w}(x,a_{1:K},y)} \\
&= \sum_{x,a_{1:K},y} \mathbb{P}_*(x,a_{1:K},y)\ \log \frac{\mathbb{P}_*(a_{1:K} \mid x)}{\mathbb{P}_\theta(a_{1:K} \mid x)} + \sum_{x,a_{1:K},y} \mathbb{P}_*(x,a_{1:K},y)\ \log \frac{\mathbb{P}_*(y \mid a_{1:K})}{\mathbb{P}_w(y \mid a_{1:K})}.
\end{aligned}$$

We use $(*_1)$ and $(*_2)$ to denote these two terms.

We rewrite the first term.

$$\begin{aligned}
(*_1) &= \sum_{x,a_{1:K}} \mathbb{P}_*(x,a_{1:K})\ \log \frac{\mathbb{P}_*(a_{1:K} \mid x)}{\mathbb{P}_\theta(a_{1:K} \mid x)} \\
&= \sum_{x} \mathbb{P}_*(x) \sum_{a_{1:K}} \mathbb{P}_*(a_{1:K} \mid x) \log \frac{\mathbb{P}_*(a_{1:K} \mid x)}{\mathbb{P}_\theta(a_{1:K} \mid x)} \\
&= \sum_{x} \mathbb{P}_*(x)\ D_{\mathrm{KL}}\left(\mathbb{P}_*(A_{1:K} \mid x) \| \, \mathbb{P}_\theta(A_{1:K} \mid x)\right) \\
&= \mathbb{E}_{X \sim \mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(A_{1:K} \mid X) \| \, \mathbb{P}_\theta(A_{1:K} \mid X)\right)\right].
\end{aligned}$$

We also rewrite the second term.

$$
\begin{aligned}
(*_2) &= \sum_{x,a_{1:K}} \mathbb{P}_*(x, a_{1:K}) \sum_y \mathbb{P}_*(y \mid a_{1:K}) \log \frac{\mathbb{P}_*(y \mid a_{1:K})}{\mathbb{P}_w(y \mid a_{1:K})} \\
&= \sum_{x,a_{1:K}} \mathbb{P}_*(x, a_{1:K}) D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid a_{1:K}) \| \mathbb{P}_w(Y \mid a_{1:K})\right) \\
&= \sum_{a_{1:K}} \mathbb{P}_*(a_{1:K}) D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid a_{1:K}) \| \mathbb{P}_w(Y \mid a_{1:K})\right) \\
&= \mathbb{E}_{A_{1:K}\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \mathbb{P}_w(Y \mid A_{1:K})\right)\right].
\end{aligned}
$$

Summarizing the three steps, we have:

$$
\begin{aligned}
\mathbb{E}_{X\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid X) \| \mathbb{P}_{\theta,w}(Y \mid X)\right)\right] \le\ & \mathbb{E}_{X\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(A_{1:K} \mid X) \| \mathbb{P}_\theta(A_{1:K} \mid X)\right)\right] \\
& + \mathbb{E}_{A_{1:K}\sim\mathbb{P}_*}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \mathbb{P}_w(Y \mid A_{1:K})\right)\right].
\end{aligned}
$$

Equality holds iff for each (x, y), $\frac{\mathbb{P}_*(x,a_{1:K},y)}{\mathbb{P}_{\theta,w}(x,a_{1:K},y)} = \frac{\mathbb{P}_*(a_{1:K}|x)\mathbb{P}_*(y|a_{1:K})}{\mathbb{P}_\theta(a_{1:K}|x)\mathbb{P}_w(y|a_{1:K})}$ is some constant $c(x, y)$ for all $a_{1:K}$ with $\mathbb{P}_*(x, a_{1:K}, y),\ \mathbb{P}_{\theta,w}(x, a_{1:K}, y) > 0$.

□

Corollary 6 (Restatement of Corollary 3)**.** *The expected interventional error of NPC is upper-bounded by the sum of the expected interventional error of the attribute predictor and the expected prediction error of the PC. Specifically,*

$$
\begin{aligned}
\mathbb{E}_{X\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{do_j}(Y \mid X) \| \mathbb{P}_{\theta,w}^{do_j}(Y \mid X)\right)\right] \le\ & \mathbb{E}_{X\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{do_j}(A_{1:K} \mid X) \| \mathbb{P}_\theta^{cl_j}(A_{1:K} \mid X)\right)\right] \\
& + \mathbb{E}_{A_{1:K}\sim\mathbb{P}_*^{do_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \mathbb{P}_w(Y \mid A_{1:K})\right)\right].
\end{aligned}
$$

Equality holds if for each (x, y), there exists a constant $c(x, y)$ such that $\frac{\mathbb{P}_*^{do_j}(a_{1:K}|x)\mathbb{P}_*(y|a_{1:K})}{\mathbb{P}_\theta^{cl_j}(a_{1:K}|x)\mathbb{P}_w(y|a_{1:K})} = c(x, y)$ for all $a_{1:K}$.

Proof. This result is a corollary of Theorem 5. The proof follows the same steps, except that all distributions are taken under the intervention. For completeness, we still spell out the three steps below.

Step 1. Show that

$$
\mathbb{E}_{X\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(Y \mid X) \| \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(Y \mid X)\right)\right] = D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(X, Y) \| \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(X, Y)\right),
$$

where $\mathbb{P}_{\theta,w}^{\mathrm{do}_j}(X, Y) = \mathbb{P}_*^{\mathrm{do}_j}(X)\, \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(Y \mid X)$.

Rewriting $\mathbb{P}_*^{\mathrm{do}_j}(X, Y)$ as $\mathbb{P}_*^{\mathrm{do}_j}(X)\, \mathbb{P}_*^{\mathrm{do}_j}(Y \mid X)$, we can expand the RHS as follows.

$$
\begin{aligned}
D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(X, Y) \| \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(X, Y)\right) &= \sum_{x,y} \mathbb{P}_*^{\mathrm{do}_j}(x, y) \log \frac{\mathbb{P}_*^{\mathrm{do}_j}(y \mid x)}{\mathbb{P}_{\theta,w}^{\mathrm{do}_j}(y \mid x)} \\
&= \sum_x \mathbb{P}_*^{\mathrm{do}_j}(x) \sum_y \mathbb{P}_*^{\mathrm{do}_j}(y \mid x) \log \frac{\mathbb{P}_*^{\mathrm{do}_j}(y \mid x)}{\mathbb{P}_{\theta,w}^{\mathrm{do}_j}(y \mid x)} \\
&= \sum_x \mathbb{P}_*^{\mathrm{do}_j}(x)\, D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(Y \mid x) \| \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(Y \mid x)\right) \\
&= \mathbb{E}_{X\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(Y \mid X) \| \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(Y \mid X)\right)\right].
\end{aligned}
$$

Step 2. Show that

$$
D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(X, Y) \| \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(X, Y)\right) \le D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(X, A_{1:K}, Y) \| \mathbb{P}_{\theta,w}^{\mathrm{do}_j}(X, A_{1:K}, Y)\right),
$$

where $\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(X, A_{1:K}, Y) = \mathbb{P}^{\mathrm{do}_j}_{*}(X)\, \mathbb{P}^{\mathrm{cl}_j}_{\theta}(A_{1:K} \mid X)\, \mathbb{P}^{\mathrm{do}_j}_{w}(Y \mid A_{1:K})$.

Under Assumption 1, $\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x, a_{1:K}, y)$ can be written as $\mathbb{P}^{\mathrm{do}_j}_{*}(x)\ \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(a_{1:K}, y \mid x)$; thus, the marginal $\sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x, a_{1:K}, y)$ equals $\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x, y)$.

Following this, we can expand the LHS as follows.

$$
\begin{aligned}
D_{\mathrm{KL}}\left(\mathbb{P}^{\mathrm{do}_j}_{*}(X,Y) \| \, \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(X,Y)\right) &= \sum_{x,y} \mathbb{P}^{\mathrm{do}_j}_{*}(x,y)\ \log \frac{\mathbb{P}^{\mathrm{do}_j}_{*}(x,y)}{\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x,y)} \\
&= \sum_{x,y} \left(\sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y) \right)\ \log \frac{\sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y)}{\sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x, a_{1:K}, y)} \\
\text{using the log-sum inequality} &\leq \sum_{x,y} \sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y)\ \log \frac{\mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y)}{\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x, a_{1:K}, y)} \\
&= D_{\mathrm{KL}}\left(\mathbb{P}^{\mathrm{do}_j}_{*}(X, A_{1:K}, Y) \| \, \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(X, A_{1:K}, Y)\right).
\end{aligned}
$$

Equality holds iff for each (x, y), $\frac{\mathbb{P}^{\mathrm{do}_j}_{*}(x,a_{1:K},y)}{\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x,a_{1:K},y)}$ is some constant $c(x, y)$ for all $a_{1:K}$ with $\mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y),\ \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x, a_{1:K}, y) > 0$.

Step 3. Decompose $D_{\mathrm{KL}}\left(\mathbb{P}^{\mathrm{do}_j}_{*}(X, A_{1:K}, Y) \| \, \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(X, A_{1:K}, Y)\right)$.

Under Assumption 1, $\mathbb{P}^{\mathrm{do}_j}_{*}(X, A_{1:K}, Y)$ can be written as $\mathbb{P}^{\mathrm{do}_j}_{*}(X)\, \mathbb{P}^{\mathrm{do}_j}_{*}(A_{1:K} \mid X)\, \mathbb{P}^{\mathrm{do}_j}_{*}(Y \mid A_{1:K})$. Following this, we can decompose the target term as follows.

$$
\begin{aligned}
D_{\mathrm{KL}}\left(\mathbb{P}^{\mathrm{do}_j}_{*}(X, A_{1:K}, Y) \| \, \mathbb{P}^{\mathrm{do}_j}_{\theta,w}(X, A_{1:K}, Y)\right) &= \sum_{x,a_{1:K},y} \mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y)\ \log \frac{\mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y)}{\mathbb{P}^{\mathrm{do}_j}_{\theta,w}(x, a_{1:K}, y)} \\
&= \sum_{x,a_{1:K},y} \mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y)\ \log \frac{\mathbb{P}^{\mathrm{do}_j}_{*}(a_{1:K} \mid x)}{\mathbb{P}^{\mathrm{cl}_j}_{\theta}(a_{1:K} \mid x)} + \sum_{x,a_{1:K},y} \mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K}, y)\ \log \frac{\mathbb{P}^{\mathrm{do}_j}_{*}(y \mid a_{1:K})}{\mathbb{P}^{\mathrm{do}_j}_{w}(y \mid a_{1:K})}
\end{aligned}
$$

We use $(*_1)$ and $(*_2)$ to denote these two terms.

We rewrite the first term.

$$
\begin{aligned}
(*_1) &= \sum_{x,a_{1:K}} \mathbb{P}^{\mathrm{do}_j}_{*}(x, a_{1:K})\ \log \frac{\mathbb{P}^{\mathrm{do}_j}_{*}(a_{1:K} \mid x)}{\mathbb{P}^{\mathrm{cl}_j}_{\theta}(a_{1:K} \mid x)} \\
&= \sum_{x} \mathbb{P}^{\mathrm{do}_j}_{*}(x) \sum_{a_{1:K}} \mathbb{P}^{\mathrm{do}_j}_{*}(a_{1:K} \mid x) \log \frac{\mathbb{P}^{\mathrm{do}_j}_{*}(a_{1:K} \mid x)}{\mathbb{P}^{\mathrm{cl}_j}_{\theta}(a_{1:K} \mid x)} \\
&= \sum_{x} \mathbb{P}^{\mathrm{do}_j}_{*}(x)\ D_{\mathrm{KL}}\left(\mathbb{P}^{\mathrm{do}_j}_{*}(A_{1:K} \mid x) \| \, \mathbb{P}^{\mathrm{cl}_j}_{\theta}(A_{1:K} \mid x)\right) \\
&= \mathbb{E}_{X \sim \mathbb{P}^{\mathrm{do}_j}_{*}}\left[D_{\mathrm{KL}}\left(\mathbb{P}^{\mathrm{do}_j}_{*}(A_{1:K} \mid X) \| \, \mathbb{P}^{\mathrm{cl}_j}_{\theta}(A_{1:K} \mid X)\right)\right].
\end{aligned}
$$

By assumption, $\mathrm{Pa}_{\mathcal{C}}(Y) \subseteq \{A_{1:K}\} \subseteq \mathrm{ND}_{\mathcal{C}}(Y)$. Hence, by the local Markov property and the invariance of causal mechanisms, we have:

$$
\begin{aligned}
\mathbb{P}_{*}(Y \mid A_{1:K}) &= \mathbb{P}_{*}(Y \mid \mathrm{Pa}(Y)), \\
\mathbb{P}^{\mathrm{do}_j}_{*}(Y \mid A_{1:K}) &= \mathbb{P}^{\mathrm{do}_j}_{*}(Y \mid \mathrm{Pa}(Y)), \\
\mathbb{P}_{*}(Y \mid \mathrm{Pa}(Y)) &= \mathbb{P}^{\mathrm{do}_j}_{*}(Y \mid \mathrm{Pa}(Y)).
\end{aligned}
$$

Consequently, $\mathbb{P}_*(Y \mid A_{1:K}) = \mathbb{P}_*^{\text{do}_j}(Y \mid A_{1:K})$, and similarly, $\mathbb{P}_w(Y \mid A_{1:K}) = \mathbb{P}_w^{\text{do}_j}(Y \mid A_{1:K})$. Thus, the second term can be written as:

$$
\begin{aligned}
(*_2) &= \sum_{x,a_{1:K}} \mathbb{P}_*^{\text{do}_j}(x, a_{1:K}) \sum_y \mathbb{P}_*^{\text{do}_j}(y \mid a_{1:K}) \, \log \frac{\mathbb{P}_*^{\text{do}_j}(y \mid a_{1:K})}{\mathbb{P}_w^{\text{do}_j}(y \mid a_{1:K})} \\
&= \sum_{x,a_{1:K}} \mathbb{P}_*^{\text{do}_j}(x, a_{1:K}) \sum_y \mathbb{P}_*(y \mid a_{1:K}) \, \log \frac{\mathbb{P}_*(y \mid a_{1:K})}{\mathbb{P}_w(y \mid a_{1:K})} \\
&= \sum_{x,a_{1:K}} \mathbb{P}_*^{\text{do}_j}(x, a_{1:K}) \, D_{\text{KL}}\left(\mathbb{P}_*(Y \mid a_{1:K}) \| \, \mathbb{P}_w(Y \mid a_{1:K})\right) \\
&= \sum_{a_{1:K}} \mathbb{P}_*^{\text{do}_j}(a_{1:K}) \, D_{\text{KL}}\left(\mathbb{P}_*(Y \mid a_{1:K}) \| \, \mathbb{P}_w(Y \mid a_{1:K})\right) \\
&= \mathbb{E}_{A_{1:K} \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \, \mathbb{P}_w(Y \mid A_{1:K})\right) \right]
\end{aligned}
$$

Summarizing the three steps, we have:

$$
\begin{aligned}
\mathbb{E}_{X \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(Y \mid X) \| \, \mathbb{P}_{\theta,w}^{\text{do}_j}(Y \mid X) \right) \right] &\leq \mathbb{E}_{X \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(A_{1:K} \mid X) \| \, \mathbb{P}_\theta^{\text{cl}_j}(A_{1:K} \mid X) \right) \right] \\
&\quad + \mathbb{E}_{A_{1:K} \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \, \mathbb{P}_w(Y \mid A_{1:K}) \right) \right].
\end{aligned}
$$

Equality holds iff for each (x, y), $\frac{\mathbb{P}_*^{\text{do}_j}(x, a_{1:K}, y)}{\mathbb{P}_{\theta,w}^{\text{do}_j}(x, a_{1:K}, y)} = \frac{\mathbb{P}_*^{\text{do}_j}(a_{1:K} \mid x) \mathbb{P}_*^{\text{do}_j}(y \mid a_{1:K})}{\mathbb{P}_\theta^{\text{cl}_j}(a_{1:K} \mid x) \mathbb{P}_w^{\text{do}_j}(y \mid a_{1:K})} = \frac{\mathbb{P}_*^{\text{do}_j}(a_{1:K} \mid x) \mathbb{P}_*(y \mid a_{1:K})}{\mathbb{P}_\theta^{\text{cl}_j}(a_{1:K} \mid x) \mathbb{P}_w(y \mid a_{1:K})}$ is some constant $c(x, y)$ for all $a_{1:K}$ with $\mathbb{P}_*^{\text{do}_j}(x, a_{1:K}, y),\ \mathbb{P}_{\theta,w}^{\text{do}_j}(x, a_{1:K}, y) > 0$.

□

Corollary 7 (Restatement of Corollary 4)**.** *The expected interventional error of CNPC is upper-bounded by a weighted sum of (i) the expected interventional error of the attribute predictor, (ii) the expected interventional error of the causal PC over attributes, and (iii) the expected prediction error of the causal PC. Specifically,*

$$
\begin{aligned}
\mathbb{E}_{X \sim \mathbb{P}_*^{do_j}} \left[D_{\text{KL}}\Big(\mathbb{P}_*^{do_j}(Y \mid X) \parallel \tilde{\mathbb{P}}_{\theta,w,\alpha}^{do_j}(Y \mid X) \Big) \right] &\leq (1 - \alpha) \, \mathbb{E}_{X \sim \mathbb{P}_*^{do_j}} \left[D_{\text{KL}}\Big(\mathbb{P}_*^{do_j}(A_{1:K} \mid X) \parallel \mathbb{P}_\theta^{cl_j}(A_{1:K} \mid X) \Big) \right] \\
&\quad + \alpha \, \mathbb{E}_{X \sim \mathbb{P}_*^{do_j}} \left[D_{\text{KL}}\Big(\mathbb{P}_*^{do_j}(A_{1:K} \mid X) \parallel \mathbb{P}_w^{do_j}(A_{1:K}) \Big) \right] \\
&\quad + \mathbb{E}_{A_{1:K} \sim \mathbb{P}_*^{do_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \, \mathbb{P}_w(Y \mid A_{1:K}) \right) \right].
\end{aligned}
$$

Equality holds if for each (x, y), $\mathbb{P}_\theta^{cl_j}(a_{1:K} \mid x) = \mathbb{P}_w^{do_j}(a_{1:K})$ for all $a_{1:K}$, and there exists a constant $c(x, y)$ such that $\frac{\mathbb{P}_*^{do_j}(a_{1:K} \mid x) \mathbb{P}_*(y \mid a_{1:K})}{\mathbb{P}_\theta^{cl_j}(a_{1:K} \mid x) \mathbb{P}_w(y \mid a_{1:K})} = c(x, y)$ for all $a_{1:K}$.

Proof. Following the three steps outlined in the proof of Corollary 3, we obtain:

$$
\begin{aligned}
\mathbb{E}_{X \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(Y \mid X) \| \, \tilde{\mathbb{P}}_{\theta,w,\alpha}^{\text{do}_j}(Y \mid X) \right) \right] &\leq \mathbb{E}_{X \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(A_{1:K} \mid X) \| \, \tilde{\mathbb{P}}_{\theta,w,\alpha}^{\text{do}_j}(A_{1:K} \mid X) \right) \right] \\
&\quad + \mathbb{E}_{A_{1:K} \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \, \mathbb{P}_w(Y \mid A_{1:K}) \right) \right].
\end{aligned}
$$

Equality holds iff for each (x, y), $\frac{\mathbb{P}_*^{\text{do}_j}(x, a_{1:K}, y)}{\tilde{\mathbb{P}}_{\theta,w,\alpha}^{\text{do}_j}(x, a_{1:K}, y)}$ is some constant $c(x, y)$ for all $a_{1:K}$ with $\mathbb{P}_*^{\text{do}_j}(x, a_{1:K}, y),\ \tilde{\mathbb{P}}_{\theta,w,\alpha}^{\text{do}_j}(x, a_{1:K}, y) > 0$.

Here, we apply an additional step to decompose $\mathbb{E}_{X \sim \mathbb{P}_*^{\text{do}_j}} \left[D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(A_{1:K} \mid X) \| \, \tilde{\mathbb{P}}_{\theta,w,\alpha}^{\text{do}_j}(A_{1:K} \mid X) \right) \right]$.

$$
\begin{aligned}
D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(A_{1:K} \mid X) \| \, \tilde{\mathbb{P}}_{\theta,w,\alpha}^{\text{do}_j}(A_{1:K} \mid X) \right) &= (1 - \alpha) \, D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(A_{1:K} \mid X) \| \, \mathbb{P}_\theta^{\text{cl}_j}(A_{1:K} \mid X) \right) \\
&\quad + \alpha \, D_{\text{KL}}\left(\mathbb{P}_*^{\text{do}_j}(A_{1:K} \mid X) \| \, \mathbb{P}_w^{\text{do}_j}(A_{1:K}) \right) \\
&\quad + \log Z_\alpha(X)
\end{aligned}
$$

By Hölder's inequality, $\log Z_\alpha(X)$ can be bounded as:

$$\begin{aligned}\log Z_\alpha(X) &= \log \sum_{a_{1:K}} \left(\mathbb{P}_\theta^{\mathrm{cl}_j}(A_{1:K} = a_{1:K} \mid X)\right)^{1-\alpha} \left(\mathbb{P}_w^{\mathrm{do}_j}(A_{1:K} = a_{1:K})\right)^{\alpha} \\ &\leq \log \left(\sum_{a_{1:K}} \mathbb{P}_\theta^{\mathrm{cl}_j}(A_{1:K} = a_{1:K} \mid X)\right)^{1-\alpha} \left(\sum_{a_{1:K}} \mathbb{P}_w^{\mathrm{do}_j}(A_{1:K} = a_{1:K})\right)^{\alpha} \\ &= 0\end{aligned}$$

Equality holds when $\mathbb{P}_\theta^{\mathrm{cl}_j}(A_{1:K} = a_{1:K} \mid X) = \mathbb{P}_w^{\mathrm{do}_j}(A_{1:K} = a_{1:K})$ for all $a_{1:K}$.

Summarizing all steps, we have:

$$\begin{aligned}\mathbb{E}_{X\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(Y \mid X) \,\|\, \tilde{\mathbb{P}}_{\theta,w,\alpha}^{\mathrm{do}_j}(Y \mid X)\right)\right] \leq (1-\alpha)\, &\mathbb{E}_{X\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(A_{1:K} \mid X) \,\|\, \mathbb{P}_\theta^{\mathrm{cl}_j}(A_{1:K} \mid X)\right)\right] \\ &+ \alpha\, \mathbb{E}_{X\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*^{\mathrm{do}_j}(A_{1:K} \mid X) \,\|\, \mathbb{P}_w^{\mathrm{do}_j}(A_{1:K})\right)\right] \\ &+ \mathbb{E}_{A_{1:K}\sim\mathbb{P}_*^{\mathrm{do}_j}}\left[D_{\mathrm{KL}}\left(\mathbb{P}_*(Y \mid A_{1:K}) \| \, \mathbb{P}_w(Y \mid A_{1:K})\right)\right].\end{aligned}$$

Equality holds iff for each (x, y), $\mathbb{P}_\theta^{\mathrm{cl}_j}(a_{1:K} \mid x) = \mathbb{P}_w^{\mathrm{do}_j}(a_{1:K})$ for all $a_{1:K}$, and $\frac{\mathbb{P}_*^{\mathrm{do}_j}(x,a_{1:K},y)}{\tilde{\mathbb{P}}_{\theta,w,\alpha}^{\mathrm{do}_j}(x,a_{1:K},y)} = \frac{\mathbb{P}_*^{\mathrm{do}_j}(a_{1:K}|x)\mathbb{P}_*^{\mathrm{do}_j}(y|a_{1:K})}{\tilde{\mathbb{P}}_{\theta,w,\alpha}^{\mathrm{do}_j}(a_{1:K}|x)\mathbb{P}_w^{\mathrm{do}_j}(y|a_{1:K})} = \frac{\mathbb{P}_*^{\mathrm{do}_j}(a_{1:K}|x)\mathbb{P}_*^{\mathrm{do}_j}(y|a_{1:K})}{\mathbb{P}_\theta^{\mathrm{cl}_j}(a_{1:K}|x)\mathbb{P}_w^{\mathrm{do}_j}(y|a_{1:K})} = \frac{\mathbb{P}_*^{\mathrm{do}_j}(a_{1:K}|x)\mathbb{P}_*(y|a_{1:K})}{\mathbb{P}_\theta^{\mathrm{cl}_j}(a_{1:K}|x)\mathbb{P}_w(y|a_{1:K})}$ is some constant $c(x, y)$ for all $a_{1:K}$ with $\mathbb{P}_*^{\mathrm{do}_j}(x, a_{1:K}, y),\ \tilde{\mathbb{P}}_{\theta,w,\alpha}^{\mathrm{do}_j}(x, a_{1:K}, y) > 0$.

□