ACTIVA: Amortized Causal Effect Estimation via Variational Autoencoders

Anonymous authors
Paper under double-blind review

Abstract

Predicting post-intervention distributions from observational data is central to many scientific and decision-making problems, but remains challenging due to causal ambiguity, restrictive modeling assumptions, and the lack of amortization across tasks. We introduce ACTIVA, a transformer-based conditional variational autoencoder for amortized estimation of full interventional distributions from observational data and intervention queries. ACTIVA learns a conditional latent prior that supports zero-shot inference by amortizing causal knowledge across diverse training tasks. We provide a consistency result showing that, under idealized conditions, ACTIVA's learning objective targets a mixture over the interventional distributions of causal models that are observationally compatible with the input. Empirically, on synthetic datasets and biologically realistic gene-expression simulations, ACTIVA substantially outperforms a correlational baseline, reduces spurious non-descendant effects, and achieves competitive performance relative to strong amortized baselines. Our results show that ACTIVA is a promising approach for estimating interventional distributions from observational data.

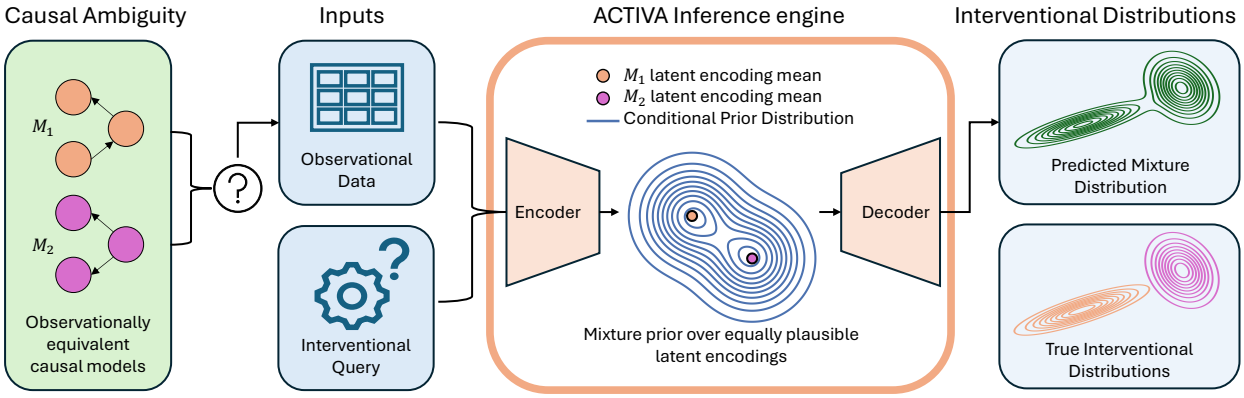

Figure 1: Causal models and their corresponding interventional distributions are generally ambiguous from observational data alone. Given observational data and an intervention query, ACTIVA encodes this ambiguity via a prior that represents the mixture over ambiguous models in the latent space. Through ACTIVA's decoder, the latent prior distribution is then transformed into a mixture over the interventional distributions of the ambiguous causal models.

1 Introduction

Estimating what would happen under an intervention from observational data alone is a fundamental goal in fields such as healthcare, economics, and finance (Shi & Norgeot, 2022; Panizza & Presbitero, 2014; Kumar et al., 2023). These predictions are essential for decision-making, since in many real-world settings

interventions are costly, risky, or impossible to test directly. Yet observational data by itself is often insufficient to determine causal effects uniquely: different causal mechanisms can generate the same observed data while implying different outcomes under intervention (Bareinboim et al., 2022). This fundamental ambiguity makes causal effect estimation from observations alone both important and challenging.

To address these ambiguities, current approaches often introduce additional assumptions, for instance by assuming access to the causal graph (Sánchez-Martin et al., 2022; Sanchez et al., 2022; Chao et al., 2023; Poinsot et al., 2024). An alternative approach to address these ambiguities is to make them explicit in the form of uncertainties. For example, providing an interval or distribution of plausible causal effects has been shown to be feasible and useful in practice (Maathuis et al., 2009; Balazadeh et al., 2025). Such a prediction allows for a clearer picture of what range of effects to expect. Although having a distribution over causal effects is a promising direction, it only represents uncertainty over a specific point-estimate. A natural extension is to look at the overall distributional shift under intervention of the causally relevant variables directly. In addition to answering cause-effect point estimates, such a distribution can provide further task-relevant insights, such as expressing ambiguity via multi-modal distributions while still providing point estimates about specific effects.

A further practical complication is that, in many real-world applications, rather than knowing the fully specified intervention, one often observes only an intervention label together with the causal variable(s) that the intervention targets. For example, we may want to estimate the effect of administering a blood-pressure medication on patient outcomes without precise information about, e.g., the exact dosage or timing. In such a setting, many current causal inference approaches are not directly applicable, as they assume fully specified interventions.

Adding to that, causal inference methods often require extensive computation for each new problem instance. Recent work shows that across a diverse set of causal inference tasks, these computations can be amortized (Löwe et al., 2022; Lorch et al., 2022; Scetbon et al., 2024; Sauter et al., 2024b; Mahajan et al., 2024; Annadani et al., 2025; Robertson et al., 2025; Balazadeh et al., 2025; Ma et al., 2025; Dhir et al., 2025). That means that a model is trained upfront — with significant computational effort — to solve the task, such that inference is computationally cheap by reusing learned knowledge. These results suggest a way towards foundation models for causal inference.

In this paper, we propose ACTIVA, a conditional variational autoencoder (CVAE) model for amortized causal inference from observational data and intervention labels. For inference, we condition our prior on the observational data and the intervention query of interest to obtain a latent distribution corresponding to plausible causal models. The decoder transforms the prior distribution into a mixture over interventional distributions that are plausible under the observational data. In empirical evaluations, we validate that ACTIVA can successfully recover post-interventional distributions and point estimates at inference time, even on novel instances with implicit uncertainty. In contrast to approaches that first recover or assume a causal graph, ACTIVA directly amortizes the prediction of post-interventional distributions, using the decoder to map latent representations of causal uncertainty into full distributional predictions. In summary, our contributions are as follows:

- We provide a CVAE model (ACTIVA) designed for amortized post-interventional distribution estimation in regimes with partially specified interventions from observational data.
- We provide a consistency result showing that, under idealized conditions, ACTIVA's learning objective targets a mixture over the interventional distributions of observationally equivalent causal models.
- We implement ACTIVA and show empirically that the theory-guided architecture learns useful post-interventional predictors in finite-sample settings: it substantially improves over a correlational baseline, reduces spurious non-descendant effects, and remains competitive with strong amortized baselines.

Overall, our work highlights the significance of amortized causal inference as a tool to overcome traditional hurdles in distributional causal effect estimation. The code reposi-

tory is available at `https://anonymous.4open.science/r/Amortized_Interventional_Distribution_Estimation-0B6D/README.md`; our data and trained models at `https://osf.io/5vebr/overview?view_only=486a76013aa340e59da1c5f81cfa04cf`.

The remainder of the paper is organized as follows: We start by introducing the most important background concepts and notation (Section 2). We then define ACTIVA (Section 3) and provide a theoretical characterization (Section 4). We then go into detail on how we implemented ACTIVA in practice (Section 5) and our experimental setup (Section 6). In Section 7 we provide our empirical results. Finally, we discuss the related work (Section 8) and provide a conclusion to our paper (Section 9).

2 Background and Notation

Conditional β-VAEs. Variational Autoencoders (VAEs) are generative models that define the joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\gamma(\boldsymbol{x}|\boldsymbol{z})p_\eta(\boldsymbol{z})$, where $\boldsymbol{z}$ are latent variables governing the data $\boldsymbol{x}$ and $\theta = \{\gamma, \eta\}$ are the parameters determining the data generation. The marginal data likelihood $p_\theta(\boldsymbol{x})$ is optimized using the evidence lower bound (ELBO) (Kingma & Welling, 2013):

$$\log p_\theta(\boldsymbol{x}) \geq \mathop{\mathbb{E}}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} [\log p_\gamma(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\eta(\boldsymbol{z})), \tag{1}$$

where ϕ parametrizes the encoding distribution q. The ELBO has a reconstruction term that ensures the model accurately reconstructs the data from the latent representation, and a Kullback-Leibler divergence (KL) term that regularizes the encoding distribution $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ to approximate some prior $p_\eta(\boldsymbol{z})$.

Conditional VAEs (CVAEs) (Sohn et al., 2015) extend this framework to allow for conditional generation given some auxiliary information $\boldsymbol{c}$. Here, we adopt a condition-dependent prior variant in which the auxiliary information affects generation only through the latent variable, i.e., $\boldsymbol{x} \perp\!\!\!\perp \boldsymbol{c} \mid \boldsymbol{z}$, yielding the generative process:

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{c}) = p_\gamma(\boldsymbol{x}|\boldsymbol{z})p_\eta(\boldsymbol{z}|\boldsymbol{c}). \tag{2}$$

In β-VAEs (Higgins et al., 2017), the KL term is scaled by a hyperparameter β, leading to a modified ELBO as follows:

$$\mathop{\mathbb{E}}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})} [\log p_\gamma(\boldsymbol{x}|\boldsymbol{z})] - \beta\, \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c}) \| p_\eta(\boldsymbol{z}|\boldsymbol{c})). \tag{3}$$

The choice of β governs the trade-off between accurately reconstructing the data and disentangling the latent representations. While $\beta > 1$ emphasizes disentanglement, in our setting, we use $\beta < 1$ to prioritize accurate reconstruction.

Causal Models. In this paper, we employ the notion of structural causal models (SCM) for describing causal data generating processes. For a detailed definition, we refer the reader to Bareinboim et al. (2022).

We treat a causal model $\mathcal{M}$ as a generative process over d variables $\boldsymbol{V} = \{V_1, ..., V_d\}$, denoting an assignment of these variables as $\boldsymbol{v} = \{v_1, ..., v_d\}$. The model is causal in the sense that each variable is generated as a function of its direct causes and an exogenous noise term. Specifically, any variable $V_j \in \boldsymbol{V}$ is determined by its direct causes with $V_j \leftarrow f_j(Pa_{V_j}, U_j)$, where f_j is an arbitrary function of the direct causes Pa_{V_j} (representing parent variables of V_j) and a noise term U_j. The causal graph $G^{\mathcal{M}}$ of model $\mathcal{M}$ encodes all parent relations via directed edges between the corresponding variables. Each model $\mathcal{M}$ induces a joint distribution $p_{\mathcal{M}}(\boldsymbol{V})$, called the observational distribution, and we denote a dataset of N i.i.d. samples from this distribution as $\boldsymbol{D}^{\mathcal{M}} := \boldsymbol{v}_{1:N}^{\mathcal{M}}$, where $\boldsymbol{v}_n^{\mathcal{M}} \sim p_{\mathcal{M}}(\boldsymbol{V})$ for $1 \leq n \leq N$. We refer to the class of models $[\mathcal{M}^o] := \{\mathcal{M} : p_{\mathcal{M}}(\boldsymbol{V}) = p_{\mathcal{M}^o}(\boldsymbol{V})\}$ as observationally equivalent models defined by some representative model $\mathcal{M}^o$.

In a causal model, performing a so-called intervention $do(V = v)$ manipulates $\mathcal{M}$ such that the target variable V is forced to take on the value v, regardless of V's causes. Such an intervention results in an intervened model that we denote as $\mathcal{M}_{do(V=v)}$ or $\mathcal{M}_{do(V)}$ when v is clear from the context. $\mathcal{M}_{do(V)}$ induces a joint distribution $p_{\mathcal{M}}(\boldsymbol{V} \mid do(V))$ that we call the interventional or post-interventional distribution. Similarly to the observational case, we denote a dataset of N i.i.d. samples from this distribution as $\boldsymbol{D}^{\mathcal{M}_{do(V)}} := \boldsymbol{v}_{1:N}^{\mathcal{M}_{do(V)}}$,

where $\boldsymbol{v}_n^{\mathcal{M}_{do(V)}} \sim p_{\mathcal{M}}(\boldsymbol{V} \mid do(V))$. Although we use this definition of interventions for clarity, our approach does not rely on the full specification of the intervention presented.

The purpose of this work is to estimate the interventional distribution $p_{\mathcal{M}}(\boldsymbol{V} \mid do(V))$ from an observational dataset $\boldsymbol{D}^{\mathcal{M}}$. In general, identifying this distribution is not possible without additional assumptions or interventional data, not least because of the ambiguities introduced by observational equivalence (Bareinboim et al., 2022). This fundamental property of causality naturally extends to our work, which we address in the theoretical analysis section of this work.

3 ACTIVA

In this section, we outline a β-CVAE specification for amortized post-interventional distribution estimation called ACTIVA. In our setup, a dataset of observational samples and a query intervention serve as a condition for the following generative process:

$$p_\theta(\boldsymbol{V}, \boldsymbol{z} | \boldsymbol{D}^{\mathcal{M}}, do(V)) = p_\gamma(\boldsymbol{V}|\boldsymbol{z}) p_\eta(\boldsymbol{z}|\boldsymbol{D}^{\mathcal{M}}, do(V)) \tag{4}$$

In this process, the observational data $\boldsymbol{D}^{\mathcal{M}}$ together with an intervention query of interest $do(V)$ is mapped to a distribution over latent points $\boldsymbol{z}$ via some prior p_η. For each of these latents, the decoder distribution p_γ then parametrizes a distribution over the data variables $\boldsymbol{V}$, leading to a full joint distribution p_θ over latents and data variables.

Intuitively, the latent prior captures uncertainty over the causal model: the same observational data may be compatible with multiple data-generating causal models, each implying potentially different interventional distributions. As a result, the predicted post-interventional distribution can be seen as an estimation of the true distribution capturing the uncertainty about the causal model. In Section 4, we formalize this intuition as a consistency result: We characterize the target of ACTIVA's learning objective under idealized conditions.

We assume a training set $\mathbf{D}^{tr} = [\boldsymbol{D}_j^{\mathcal{M}_{do(V)}}, \boldsymbol{D}_j^{\mathcal{M}}, do(V)_j]_{j=1}^J$ with J post-interventional datasets $\boldsymbol{D}^{\mathcal{M}_{do(V)}}$, J observational datasets $\boldsymbol{D}^{\mathcal{M}}$, and J queries $do(V)$. In our setting, a sampled training task $\mathbf{D}_j^{tr} \sim p_{tr}(\mathbf{D}^{tr}) = \int p_{tr}(\mathbf{D}^{tr} \mid \mathcal{M}) p_{tr}(\mathcal{M}) d\mathcal{M}$ is an element from the training data distribution according to a pre-defined distribution $p_{tr}(\mathcal{M})$ of causal models. This predefined distribution can be interpreted as a simulator over the class of causal models which are expected at inference time.

To find the parameters γ, η, ϕ, we use the conditional ELBO formulation for likelihood maximization as in Equation 3 using the conditional posterior $q_\phi(\boldsymbol{z}|\mathbf{D}_j^{tr})$. Here, $\boldsymbol{D}_j^{\mathcal{M}_{do(V)}}$ plays the role of the input to be reconstructed and $[\boldsymbol{D}_j^{\mathcal{M}}, do(V)_j]$ as the auxiliary information, mapping to $\boldsymbol{x}$ and $\boldsymbol{c}$ in Equation 3, respectively. Our overall learning objective is:

$$\mathcal{L}(\gamma, \eta, \phi; \mathbf{D}_j^{tr}) = \mathop{\mathbb{E}}_{q_\phi(\boldsymbol{z}|\mathbf{D}_j^{tr})} \left[\log p_\gamma(\boldsymbol{D}_j^{\mathcal{M}_{do(V)}} | \boldsymbol{z}) \right] - \beta \, \mathrm{KL}(q_\phi(\boldsymbol{z}|\mathbf{D}_j^{tr}) \parallel p_\eta(\boldsymbol{z}|\boldsymbol{D}_j^{\mathcal{M}}, do(V)_j)). \tag{5}$$

Because the interventional samples are independent samples, the likelihood factorizes as $p_\gamma\left(\boldsymbol{D}_j^{\mathcal{M}_{do(V)}} \mid \boldsymbol{z}\right) = p_\gamma\left(\boldsymbol{v}_{1:N;j}^{\mathcal{M}_{do(V)}} \mid \boldsymbol{z}\right) = \Pi_{i=1}^N p_\gamma\left(\boldsymbol{v}_{i;j}^{\mathcal{M}_{do(V)}} \mid \boldsymbol{z}\right)$. Furthermore, we use the reparameterization trick (Kingma & Welling, 2013) for gradient estimation, allowing backpropagation through the stochastic sampling process of q_ϕ.

Importantly, this setup does not require the intervention to be fully specified functionally; it suffices to know an intervention label together with the variables targeted by $do(V)_j$, allowing application to partially defined intervention setting.

Our goal is to amortize causal inference by training ACTIVA over a distribution of tasks that resemble those expected during test time as well as possible. More explicitly, we expect the tasks during test time to come from similar families of causal models and have the same interventions as during training. Accordingly, optimizing the amortized objective amounts to maximizing the expectation of ELBO over the distribution

of the training datasets.

$$\max_{\gamma,\eta,\phi} \quad \mathop{\mathbb{E}}_{\mathbf{D}_j^{tr}\sim p_{tr}(\mathbf{D}^{tr})} \mathcal{L}(\gamma,\eta,\phi;\mathbf{D}_j^{tr}) \tag{6}$$

This objective optimizes the model to learn inference procedures across datasets generated by causal models drawn from $p_{tr}(\mathcal{M})$, beyond specific datasets observed in the training. The broader the training distribution of the datasets, the better the model can generalize to new datasets during test-time inference (Montagna et al., 2024).

4 Theoretical Analysis

In this section, we characterize the population-level estimand of ACTIVA's learning objective. Under idealized expressivity and infinite samples assumptions, we show that ACTIVA targets the post-interventional distribution obtained by averaging over causal models that are observationally compatible with the input dataset. The result is formulated at the level of the simulator distribution $p_{tr}(\mathcal{M})$ and therefore concerns unseen inference tasks drawn from the same task-generating process. When the observational equivalence class contains a single model, this target reduces to the unique post-interventional distribution. Thus, the analysis does not provide a finite-sample guarantee or claim identification of the true causal model from observational data alone; instead, it makes precise the distribution toward which the population objective directs optimization.

Assumption 1. In the infinite sample limit, the observational dataset $\boldsymbol{D}^{\mathcal{M}^o}$ identifies the class of observationally equivalent causal models $[\mathcal{M}^o]$.

This assumption only requires that infinitely many observational samples determine which models in the support of $p_{tr}(\mathcal{M})$ remain observationally equivalent to the input. In particular, in the infinite-sample limit, the observational dataset $\boldsymbol{D}^{\mathcal{M}^o}$ acts as a sufficient statistic for the equivalence class $[\mathcal{M}^o]$, so any function conditioned on $\boldsymbol{D}^{\mathcal{M}^o}$, including the learned prior $p_\eta(z|\boldsymbol{D}^{\mathcal{M}^o}, do(V))$, depends on the data only through $[\mathcal{M}^o]$. This justifies denoting conditioning on the dataset and the equivalence class interchangeably throughout the analysis below.

Assumption 2. There exist globally optimal parameters (γ^*,η^*,ϕ^*) of Equation 6 such that for every example $\mathbf{D}^{tr} = [\boldsymbol{D}^{\mathcal{M}_{do(V)}}, \boldsymbol{D}^{\mathcal{M}}, do(V)]$ in the support of $p_{tr}(\mathbf{D}^{tr})$,

$$\int p_{\gamma^*}(\boldsymbol{V} \mid \boldsymbol{z})\, q_{\phi^*}(\boldsymbol{z} \mid \mathbf{D}^{tr})\, d\boldsymbol{z} = p_{\mathcal{M}}(\boldsymbol{V} \mid do(V)). \tag{7}$$

In other words, at the population optimum, the encoder-decoder pair recovers the correct post-interventional distribution for every task in the support of the simulator. This assumption is well-grounded in our training setting: the encoder observes samples drawn directly from the true interventional distribution, which is therefore measured directly from the training data without requiring graphical criteria or additional assumptions (Pearl, 2009; Bareinboim et al., 2022). What this assumption does require however, is the joint closure of two standard gaps in amortized variational inference (Cremer et al., 2018): an approximation gap, requiring sufficient per-task expressiveness of the model, and an amortization gap, requiring a single shared parameterization to attain these per-task optima across all tasks. This assumption should therefore be read as a population-level idealization.

We next define the population-level aggregated posterior distribution over all simulator tasks that share the same observational equivalence class and intervention query:

$$\bar{q}_\phi(\boldsymbol{z} \mid [\mathcal{M}^o], do(V)) := \int q_\phi(\boldsymbol{z} \mid \mathbf{D}^{tr})\, p_{tr}(\mathbf{D}^{tr} \mid [\mathcal{M}^o], do(V))\, d\mathbf{D}^{tr}. \tag{8}$$

That is, $\bar{q}_\phi$ averages the encoder outputs over all tasks from the simulator consistent with the queried observational equivalence class $[\mathcal{M}^o]$ and that correspond to the same intervention query.

Assumption 3. The conditional prior family $p_\eta(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}}, do(V))$ is expressive enough to represent the aggregated posterior defined in Equation 8.

Following the standard ELBO decomposition for amortized variational models (Hoffman & Johnson, 2016), the expected KL term in Equation 6, restricted to tasks with observational class $[\mathcal{M}^o]$ and query $do(V)$, contributes to the optimization of the prior only through

$$\mathrm{KL}\Big(\bar{q}_{\phi^*}(\boldsymbol{z} \mid [\mathcal{M}^o], do(V)) \,\Big\|\, p_\eta(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V))\Big), \tag{9}$$

up to terms that do not depend on η. A derivation is given in Appendix A.

Proposition 1. Under Assumptions 1–3, the optimal conditional prior for an input $\big(\boldsymbol{D}^{\mathcal{M}^o}, do(V)\big)$ satisfies

$$p_{\eta^*}(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = \bar{q}_{\phi^*}(\boldsymbol{z} \mid [\mathcal{M}^o], do(V)). \tag{10}$$

Proof. The dependence of the objective on the prior parameters η is given by a KL term in Equation 9. This KL divergence is always non-negative and equals zero if and only if its two arguments coincide almost everywhere. Since Assumption 3 guarantees that the prior family can represent the aggregated posterior, the optimum is attained exactly at $p_{\eta^*}(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = \bar{q}_{\phi^*}(\boldsymbol{z} \mid [\mathcal{M}^o], do(V))$. □

Proposition 1 shows that, at the population optimum, the learned prior does not select a single latent code. Instead, it matches the average encoder distribution over all simulator tasks that are observationally compatible with the input and share the same intervention query. An important structural feature of this formulation deserves emphasis: during training, the encoder $q_\phi(\boldsymbol{z} \mid \mathbf{D}^{tr})$ observes both the observational and interventional samples, whereas at inference, only the prior $p_\eta(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}}, do(V))$ receives the observational data and the intervention identifier alone. Thus, training concerns a strictly more informative setting than inference. The role of Proposition 1 is precisely to bridge this gap: it shows that the learned prior, optimized to match the aggregated encoder distribution, absorbs the information that the encoder extracted from interventional data during training and makes it available at inference time. We now characterize the resulting predictive distribution.

Proposition 2. Under Assumptions 1–3, the ACTIVA predictive distribution for an input $\big(\boldsymbol{D}^{\mathcal{M}^o}, do(V)\big)$ is given by

$$p_{\theta^*}(\boldsymbol{V} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = \int p_{\mathcal{M}}(\boldsymbol{V} \mid do(V))\, p_{tr}(\mathcal{M} \mid [\mathcal{M}^o], do(V))\, d\mathcal{M}. \tag{11}$$

Proof sketch. Starting from the generative model in Equation 4, we first marginalize out the latent variable $\boldsymbol{z}$. Substituting the result from Proposition 1 into the marginal shows that the predictive distribution is an average, with respect to $p_{tr}(\mathbf{D}^{tr} \mid [\mathcal{M}^o], do(V))$, over the task-wise decoder marginals induced by $q_{\phi^*}(\boldsymbol{z} \mid \mathbf{D}^{tr})$. Assumption 2 identifies each such decoder marginal with the true post-interventional distribution of the model that generated the corresponding task. Therefore, the overall prediction is exactly the simulator-weighted mixture of post-interventional distributions over models in the observational equivalence class $[\mathcal{M}^o]$. The full proof is given in Appendix B.

Proposition 2 characterizes the estimand of ACTIVA's learning objective: under idealized conditions, the objective targets a mixture over all post-interventional distributions that are observationally compatible with the input dataset and supported by the simulator distribution. The mixture weights are given explicitly by the conditional simulator distribution $p_{tr}(\mathcal{M} \mid [\mathcal{M}^o], do(V))$, which measures how much probability mass the training distribution assigns to each model among those that remain observationally compatible with the input and correspond to the queried intervention. Thus, the uncertainty represented by ACTIVA is not arbitrary but is induced jointly by observational ambiguity and by the simulator distribution $p_{tr}(\mathcal{M})$.

Corollary 1. If the observational equivalence class of the input collapses to a single model, i.e., $[\mathcal{M}^o] = \{\mathcal{M}^o\}$, then the queried effect is identifiable within the support of $p_{tr}(\mathcal{M})$, and Proposition 2 reduces to

$$p_{\theta^*}(\boldsymbol{V} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = p_{\mathcal{M}^o}(\boldsymbol{V} \mid do(V)). \tag{12}$$

Proof. If $[\mathcal{M}^o] = \{\mathcal{M}^o\}$, then the conditional simulator distribution $p_{tr}(\mathcal{M} \mid [\mathcal{M}^o], do(V))$ places all its mass on $\mathcal{M}^o$. Hence, the integral in Equation 11 collapses to $p_{\mathcal{M}^o}(\boldsymbol{V} \mid do(V))$. □

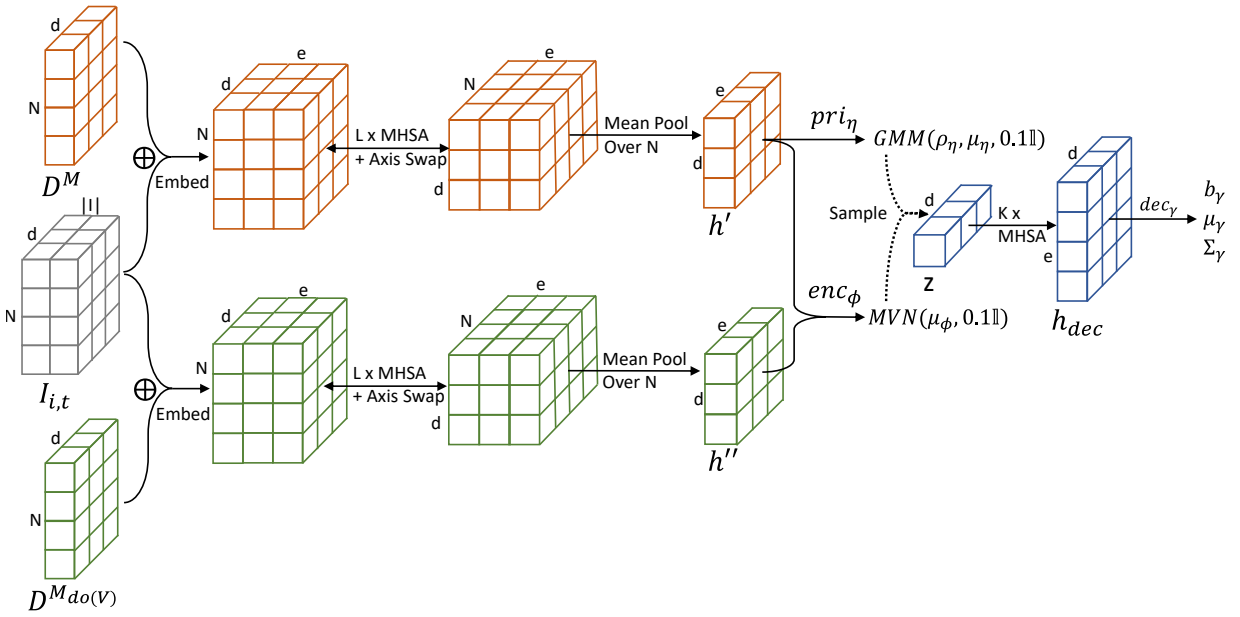

Figure 2: Overview of the proposed model architecture. The orange blocks represent the observational stream, which processes observational data alongside the intervention representation to parameterize both the prior and the posterior. The green blocks represent the interventional stream, which processes interventional samples exclusively during training to inform the posterior. Finally, the blue blocks represent the decoder stream. In a brief walkthrough: inputs are embedded, processed via alternating MHSA blocks, and mean-pooled over the sample dimension to yield intermediate representations (h' and h''). These representations parameterize the GMM prior and MVN posterior. A latent variable z is then sampled and passed through the transformer-based decoder to predict the final post-interventional mixture parameters (b_γ, μ_γ, Σ_γ).

Corollary 1 shows that in the identifiable case, ACTIVA recovers the unique post-interventional distribution rather than a non-trivial mixture.

Overall, this result provides a characterization of our training target and frames ACTIVA as an amortized estimator of post-interventional distributions under causal ambiguity. In identifiable settings, the objective targets the queried interventional distribution exactly; otherwise, it targets a principled mixture over the post-interventional distributions of observationally equivalent models supported by the simulator. Whether finite-sample, finite-capacity implementations approximate this target well is an empirical question, which we address in Section 7.

5 Model Architecture

This section outlines the neural architecture of our proposed model, detailing how we jointly encode observational data and targeted interventions into a permutation-equivariant latent space. We subsequently describe the specific parameterizations of our prior and posterior distributions, as well as the transformer-based decoder used to generate post-interventional predictions. Figure 2 provides an overview of the model architecture.

Representing Interventions. To represent interventions, we create a matrix representation. We consider $i \in \{1, \ldots, |I|\}$ the index of a possible intervention value v_i, where $|I|$ is the number of possible values, and a selector $\boldsymbol{t} \in \{0, 1\}^d$, with d the number of variables, indicating the intervention target(s). By not encoding the intervention value directly, but rather the index of the intervention, we remove the need to have a full specification of the intervention. This allows us to model interventional distributions based on the index of the interventions, as long as they can be attributed to interventional training data and have specified targets.

We construct the intervention representation $\boldsymbol{I}_{i,t}$ representing $do(\boldsymbol{V_t} = v_i)$ as follows. We perform a one-hot encoding of i creating a vector $\boldsymbol{i_{oh}} \in \mathbb{R}^{|I|}$ and repeat it d times, resulting in a matrix $\boldsymbol{i_{rep}} \in \mathbb{R}^{d\times|I|}$. We then apply $\boldsymbol{t}$ as a mask to this matrix, effectively zeroing out the rows that correspond to non-intervened variables. Finally, we repeat the intervention representation N times to match the number of input samples and obtain $\boldsymbol{I}_{i,t} \in \mathbb{R}^{N\times d\times|I|}$. This construction ensures that the intervention-relevant information for each variable is provided as local information alongside the variable itself and maintains permutation equivariance.

Embedding Network. We encode the input data and intervention according to an embedding network $h_\alpha(\cdot)$ based on the extension of non-parametric encoders (Kossen et al., 2021; Lorch et al., 2022), where α are the parameters of the network. For various causal tasks, this architecture has been successfully used to encode datasets into a vector containing causally relevant information (Lorch et al., 2022; Scetbon et al., 2024; Annadani et al., 2025; Balazadeh et al., 2025; Robertson et al., 2025; Ma et al., 2025). In particular, given some data $\boldsymbol{D} \in \mathbb{R}^{N\times d}$ of N samples, an intervention index i and a binary target vector $\boldsymbol{t}$, we append $\boldsymbol{I}_{i,t}$ to the dataset, resulting in an augmented dataset $\boldsymbol{D}_{it} \in \mathbb{R}^{N\times d\times|I|+1}$. Then we apply L blocks of multi-head self-attention (MHSA) that alternate in attending over d features and N samples. After the transformer blocks, we average over the sample axis to obtain an embedding $\boldsymbol{h} \in \mathbb{R}^{d\times e}$, where e indicates the embedding dimension, thus ensuring that the embedding is permutation invariant with respect to the sample dimension and equivariant with respect to the feature dimension.

Prior and Posterior. Guided by our theoretical findings, we choose the conditional prior to be a Gaussian Mixture Model (GMM) with d dimensions and C components. The component means $\boldsymbol{\mu}_\eta \in \mathbb{R}^{C\times d}$ and component weights $\boldsymbol{\rho}_\eta \in \mathbb{R}^C$ are determined by a learnable prior function $pri_\eta(h_{\alpha'}(\boldsymbol{D}^{\mathcal{M}}, \boldsymbol{I}_{i,t}))$ applied to the observational data. Specifically, pri_η are two multi-layer perceptrons (MLP) that we apply to the embedding $\boldsymbol{h}$ for each feature independently to compute the means and mixture weights. The outputs parameterize our prior distribution $p_\eta(\boldsymbol{z}|\boldsymbol{D}^{\mathcal{M}}, do(V)) = \sum_{c=1}^{C} \rho_{\eta,c}\mathcal{MVN}(\boldsymbol{z}; \boldsymbol{\mu_{\eta,c}}, 0.1\mathbb{1}_d)$, where $\mathbb{1}_d$ is the d-dimensional identity matrix. We set the prior to have a fixed variance of 0.1 and no covariance for model simplicity, and enforce $\sum_{c=1}^{C} \rho_{\eta,c} = 1$ via a softmax layer.

Furthermore, we choose the posterior distribution to be a d-dimensional multivariate normal distribution (MVN) with a constant, diagonal covariance matrix whose means are determined by a learnable encoder function $\boldsymbol{\mu}_\phi = enc_\phi(h_{\alpha''}(\boldsymbol{D}^{\mathcal{M}_{do(V)}}, \boldsymbol{I}_{i,t}), h_{\alpha'}(\boldsymbol{D}^{\mathcal{M}}, \boldsymbol{I}_{i,t}))$, where $h_{\alpha'}$ is the observational data network used in the prior, and $h_{\alpha''}$ is the network encoding interventional data with different parameters. We concatenate the resulting embeddings $\boldsymbol{h}'$ and $\boldsymbol{h}''$ along the embedding dimensions and again apply a two-layer MLP to each feature independently to compute the mean. This fully parametrizes our encoding distribution $q_\phi(\boldsymbol{z}|\boldsymbol{D}^{\mathcal{M}_{do(V)}}, \boldsymbol{D}^{\mathcal{M}}, do(V)) = \mathcal{MVN}(\boldsymbol{z}; \boldsymbol{\mu}_\phi, 0.1\mathbb{1}_d)$, with diagonal covariance matrix, matching the prior setup.

Decoder. We model the decoder $dec_\gamma(\boldsymbol{z})$ as a transformer that outputs the parameters of a Gaussian mixture, enabling a closed-form expression for the estimated interventional distributions. We process $\boldsymbol{z} \in \mathbb{R}^d$ via K standard transformer blocks (Vaswani et al., 2017), resulting in the embedding $\boldsymbol{h}_{dec} \in \mathbb{R}^{d\times e}$. We feed $\boldsymbol{h}_{dec}$ through a row-wise linear layer to predict the means $\boldsymbol{\mu}_\gamma \in \mathbb{R}^{B\times d}$, where B is the number of decoder mixture components, maintaining permutation equivariance regarding the variable ordering. To predict the covariances $\boldsymbol{\Sigma}_\gamma \in \mathbb{R}^{B\times d\times d}$ in a permutation equivariant manner, we first apply a row-wise linear layer to compute $\boldsymbol{u} \in \mathbb{R}^{B\times d\times e}$ from $\boldsymbol{h}_{dec}$, and then compute the covariances via $\boldsymbol{\Sigma}_\gamma = \boldsymbol{u} \cdot \boldsymbol{u}$ similarly to (Lorch et al., 2022), and add a small constant ϵ to the diagonals for computational stability. Lastly, we estimate the decoder component weights $\boldsymbol{b}_\gamma \in \mathbb{R}^B$ of the mixture by first summing $\boldsymbol{h}_{dec}$ along the d feature dimensions and then passing the pooled representation through a linear layer and a softmax layer.

Permutation Equivariance. Importantly, we maintain permutation equivariance of the predicted mixture with respect to variable ordering. This is crucial because variable labels in causal problems are arbitrary: graphs that differ only by a relabeling represent the same underlying structure, and permutation equivariance forces the model to treat them consistently. As a result, the model can share statistical strength across all labelings of the same graph, reducing the effective graph hypothesis space by collapsing relabelings of the same underlying graph into a single case. Encoding this symmetry has been shown to improve statistical efficiency, scaling, and predictive performance in causal discovery (Lorch et al., 2022; Li et al., 2020).

Practical Approximation of the Theoretical Model. The theory in Section 4 guides the design of our finite-capacity implementation. In the population setting, ACTIVA predicts post-interventional distributions by averaging over simulator-supported causal models that are compatible with the observational input. We implement this idea by using a multi-modal conditional prior over latent representations and a decoder that maps latent samples to post-interventional mixture distributions. Since the idealized construction cannot be implemented exactly, we make several tractable approximations: finite observational samples replace population-level equivalence classes, a finite GMM prior approximates the induced latent mixture, and a Gaussian posterior with fixed diagonal covariance replaces the idealized posterior. These approximations are chosen to preserve the central interpretation of the theory while enabling efficient training and inference. Empirically, the results in Section 7 show that this theory-guided approximation is effective in practice: even when implemented through tractable finite approximations, ACTIVA learns predictive distributions that reflect post-interventional structure rather than merely observational correlations.

6 Experimental Setup

This section outlines the experimental setup designed to assess our approach for predicting post-interventional distributions. We first detail the synthetic and semi-synthetic datasets, followed by a description of the baseline models, the specific evaluation protocol, and our implementation details for training and inference.

6.1 Data

We evaluate the performance of the proposed method on three types of datasets: one biologically plausible type and two synthetic types. The synthetic datasets allow us to systematically analyze how well our amortized approach recovers linear causal effects under different noise assumptions. The semi-synthetic SERGIO datasets bring our method closer to real-world conditions by simulating biologically realistic gene-expression data under interventions. Further details on dataset generation are provided in Appendix C.

For the synthetic data, we generate samples from linear additive causal models with either Gaussian or Beta noise, in both 2-variable and 8-variable settings. We apply interventions on each variable and obtain four datasets in total: *Gauss 2*, *Beta 2*, *Gauss 8*, and *Beta 8*. These configurations enable a controlled comparison between scenarios that are classically non-identifiable and those that are identifiable from observational data (Peters et al., 2017).

For the semi-synthetic data, we use the SERGIO simulator to generate gene expression data that closely resembles real single-cell gene expression patterns (Dibaeinia & Sinha, 2020). We simulate single-variable interventions corresponding to gene knockouts on each variable and construct datasets with 8 variables, as well as an out-of-distribution (OOD) evaluation dataset with 11 variables and perturbation hyperparameters.

6.2 Baselines

Gaussian Conditional. This baseline fits a multivariate Gaussian to the observational samples and uses the resulting conditional distribution $p(\boldsymbol{V} \setminus \boldsymbol{V_t} \mid \boldsymbol{V_t} = v_i)$ as a surrogate for the interventional distribution $p(\boldsymbol{V} \setminus \boldsymbol{V_t} \mid (\boldsymbol{V_t} = v_i))$. This baseline serves as a weak baseline on estimation accuracy that exploits all the correlation structure present in the data but ignores the causal directionality.

Linear Causal. This baseline assumes access to the true causal graph and fits a linear SCM by ordinary least-squares regression of each variable on its parents (Bollen, 1989). For an intervention $\mathrm{do}(\boldsymbol{V_t} = v_i)$, it computes $p(\boldsymbol{V} \setminus \boldsymbol{V_t} \mid \mathrm{do}(\boldsymbol{V_t} = v_i))$ analytically by graph surgery (Pearl, 2009), replacing the mechanism for $\boldsymbol{V_t}$ with v_i and propagating the resulting moments through the fitted linear system. Under correct specification, this baseline asymptotically recovers the exact interventional mean regardless of the Gaussianity of the noise. However, the full post-intervention distribution is Gaussian only under Gaussian noise; otherwise, the Gaussian prediction is misspecified, even if its mean is correct (Peters et al., 2017). This baseline serves as an asymptotic oracle estimator of the post-interventional distributions for the Gaussian datasets, and a privileged-information estimator for the other datasets.

MACE-TNP extension. This model is our extension of MACE-TNP (Dhir et al., 2025), a recent strong approach for post-interventional distribution prediction. We extend the original model formulation to predict the joint post-interventional distribution instead of variable-specific marginals, matching our problem specification. This transformer-based neural process (TNP) uses an encoder to process observational context data and an interventional query, similar to our approach. The main high-level difference from our method is that it requires the exact specification of the intervention value during inference, constituting a slightly privileged information setting compared to ACTIVA, which only needs the identifier. The details of this model can be found in Appendix D.

6.3 Evaluation Protocol

We evaluate our model and baselines based on the following marginals. **All** includes all variables $\boldsymbol{V}$ including $\boldsymbol{V_t}$ (the intervened variable itself). This subset informs us about the overall predicted post-interventional distribution quality. **All**$\backslash \boldsymbol{V_t}$ excludes the intervention variable. This allows for a separate evaluation for scenarios where the exact specification of the intervention is known, and hence the prediction of the intervened variable is not of interest. **Descendants** retains only variables that are descendants of $\boldsymbol{V_t}$ in the causal graph i.e. $\boldsymbol{V_t}$ is a direct or indirect cause, allowing for investigation of the prediction fit of true causal effects. $\boldsymbol{V_t}$ is not included, and descendants are identified via breadth-first search on the ground-truth adjacency matrix; **Non-Descendants** retains all variables that are not descendants of $\boldsymbol{V_t}$ (excluding $\boldsymbol{V_t}$), that is, variables for which the true causal effect is zero by definition. Our evaluation metrics are as follows:

Average Treatment Effect (ATE). We define the ATE of intervention $do(\boldsymbol{V_t} = v_i)$ on $\boldsymbol{V}$ as

$$\text{ATE} \;:=\; \mathbb{E}\big[\boldsymbol{V} \mid do(\boldsymbol{V_t} = v_i)\big] - \mathbb{E}\big[\boldsymbol{V} \mid do(\emptyset)\big], \tag{13}$$

where the second expectation is taken under the observational distribution. We estimate the ground-truth ATE from held-out interventional samples and measure accuracy via mean absolute error (MAE) between the predicted and ground-truth ATEs. Because true ATEs are exactly zero for non-descendants, the non-descendant-MAE quantifies the rate of spurious predictions.

Negative Log-Likelihood (NLL). We assess the calibration of the predicted interventional distribution by computing the average negative log-likelihood of the held-out interventional samples under the predicted density. NLL rewards both accurate means and well-calibrated variances.

Energy distance (ERG). To assess distributional fidelity in the sample space, we compute the energy distance (Székely & Rizzo, 2013) between the set of predicted samples $\hat{\boldsymbol{v}}_{1:N}^{\mathcal{M}_{do(\boldsymbol{v_t})}}$ and the held-out interventional samples $\boldsymbol{v}_{1:N}^{\mathcal{M}_{do(\boldsymbol{v_t})}}$. The energy distance is zero if and only if the two distributions are identical and is sensitive to differences in any moment of the distribution, providing a complementary view to NLL.

6.4 Training and Inference

We train our model on the *Beta*, *Gaussian*, and *SERGIO* datasets described above. For the 8 variable problems, we train our models for 1000 epochs; for the two variable problems and SERGIO we train 5000 epochs. We evaluate the NLL on the validation set every 25 and 100 steps, respectively, and retain the best performing model as our trained model. For all training runs, we used 20 mixture components for the prior GMM, 4 samples from the posterior to estimate the KL term during training, and 5 samples from the prior to create the mixture of the decoder during training. Architecturally, we used 4 blocks for the encoder and the decoder, 4 attention heads in each attention module, a hidden dimension of 256 for all modules, and an embedding dimension of 64. Regarding the other hyperparameters, we used a β of 0.5 and a learning rate of .0005. All modules and training scripts were implemented with Jax (Bradbury et al., 2018) and Flax (Heek et al., 2024). Experiments were run on the DAS6 computing cluster (Bal et al., 2016).

To infer the post-interventional distribution from our trained model, we follow the following procedure. First, the observational data and interventional query are provided to the prior as input, resulting in the prior distribution over latents. Since the prior is a GMM, obtaining closed-form solutions for interventional distributions at inference is intractable. We instead approximate the distribution by sampling 10 latent

variables. We then decode each of these latent samples and form a uniform mixture of the resulting decoded distributions. Since we set the number of decoder components to 2 throughout our models, the resulting GMM has 20 components, matching our MACE-TNP extension. In the case of sample-based analysis, we sample from the predicted decoder distribution.

7 Experiments

We evaluate ACTIVA along three complementary axes. First, we compare predicted interventional samples of all models and baselines qualitatively to identify characteristic failure modes. Second, we quantify distributional fit using negative log-likelihood (NLL) and energy distance across different causal marginals. Third, we assess whether these distributional predictions translate into accurate downstream treatment-effect estimates via the ATE.

7.1 Qualitative Analysis

We begin with a qualitative comparison of the predicted interventional distributions to better understand the behavior of each method. For four randomly selected test SCMs, we visualize samples from the predicted interventional distributions for all possible interventions. Figure 3 shows the results for the Beta 8 dataset[1]; the corresponding plots for the remaining synthetic datasets are provided in Appendix E.

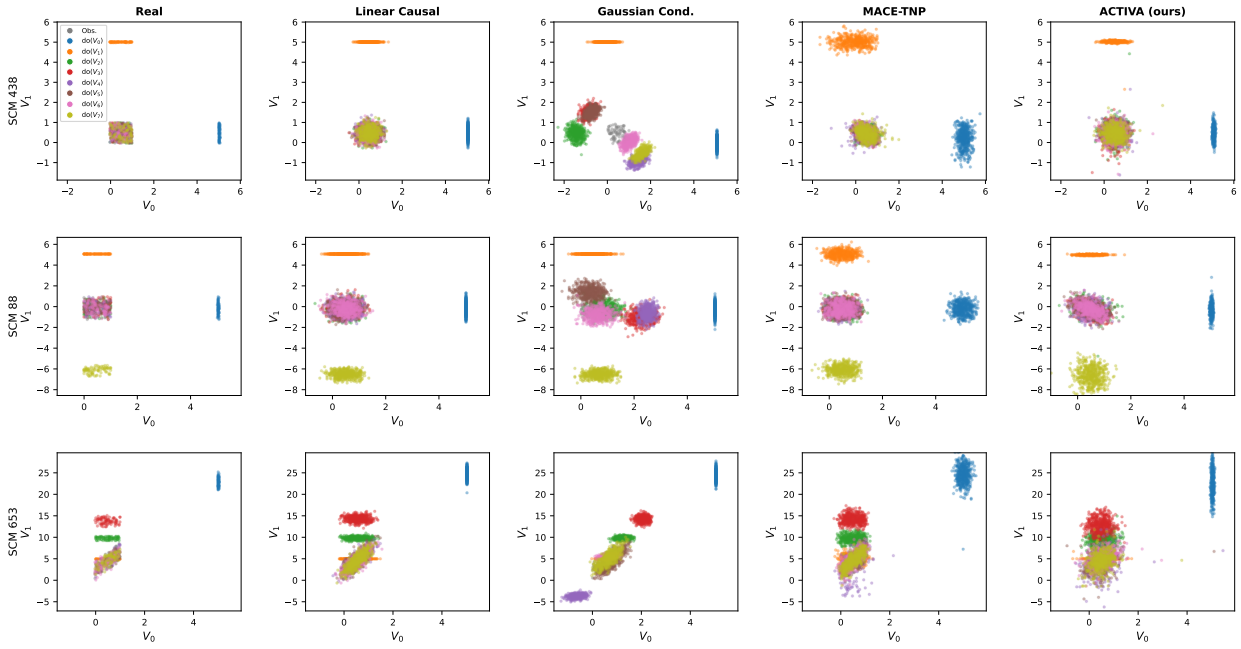

Figure 3: Samples from the observational and interventional distributions of the ground-truth model (first column), the baselines (columns 2-4), and our model (last column) for the *Beta 8* data. Each row corresponds to a randomly sampled SCM from the test data. The colors indicate which variables have been intervened on ($do(V_i = 5)$). All samples are projected on the V_0 vs. V_1 plane. The closer the distributions are to those in the first column, the more accurate the prediction is.

The Linear Causal baseline reproduces the main interventional shifts when its modeling assumptions hold, as expected, given access to the ground-truth graph. In contrast, the Gaussian Conditional baseline often predicts visible distributional changes even when no causal effect is present, illustrating the failure of a purely correlational predictor in this setting.

[1]The grey points are the ground-truth observational samples from the data, repeated in every plot for visualization.

Table 1: Average inference performance of our trained model and the baseline on held-out test according to various marginals explained in Section 6.3. The *Data* column indicates which dataset has been used with how many variables. We report the average energy distance and average negative log likelihood over the causal models in the corresponding test sets. The Linear Causal approach has privileged information (PI) by having access to the causal graph and additionally is an asymptotic oracle (O) for the Gaussian datasets. Lower is better for all metrics.

		All		**All\V_t**		**Desc.**		**Non-Desc.**	
Data	**Approach**	ERG ↓	NLL ↓	ERG ↓	NLL ↓	ERG ↓	NLL ↓	ERG ↓	NLL ↓
Gauss 2	Linear Causal (O)	—	—	**0.09**	**0.90**	**0.32**	**1.39**	**0.02**	**0.73**
	ACTIVA (ours)	**0.14**	**-1.57**	*0.14*	**0.90**	0.48	*1.41*	*0.03*	*0.78*
	MACE-TNP	*0.16*	-0.73	0.17	*1.09*	*0.47*	**1.39**	0.06	0.99
	Gaussian Cond.	—	—	1.16	8.55	0.33	**1.39**	1.43	11.00
Beta 2	Linear Causal (PI)	—	—	0.16	0.82	0.31	1.44	**0.01**	*0.19*
	ACTIVA (ours)	**0.02**	**-2.27**	**0.02**	**0.22**	**0.03**	**0.24**	**0.01**	**0.18**
	MACE-TNP	*0.07*	*-1.59*	*0.07*	*0.52*	*0.13*	*0.82*	**0.01**	0.22
	Gaussian Cond.	—	—	2.00	37.8	0.31	1.43	3.69	74.14
Gauss 8	Linear Causal (O)	—	—	**0.34**	**5.82**	**0.53**	**3.46**	**0.08**	**4.08**
	ACTIVA (ours)	*1.84*	**6.65**	1.84	9.12	2.75	5.52	0.46	5.85
	MACE-TNP	**1.58**	*8.79*	*1.57*	*8.57*	*2.53*	*5.18*	*0.38*	*5.48*
	Gaussian Cond.	—	—	4.83	38.07	2.74	6.94	4.03	36.00
Beta 8	Linear Causal (PI)	—	—	**0.34**	**2.85**	**0.58**	**3.02**	**0.05**	**1.11**
	ACTIVA (ours)	*1.43*	**3.06**	1.43	5.53	2.28	4.60	0.27	2.75
	MACE-TNP	**1.03**	*4.25*	*1.02*	*4.47*	*1.68*	*3.77*	*0.18*	*2.11*
	Gaussian Cond.	—	—	5.21	78.04	3.04	11.51	4.28	75.52
SERGIO	Linear Causal (PI)	—	—	**1.07**	14.07	*1.88*	10.80	**0.78**	12.60
	ACTIVA (ours)	**1.14**	**6.70**	*1.14*	**9.23**	**1.86**	**3.94**	0.87	**8.99**
	MACE-TNP	*1.20*	*10.93*	1.17	*10.36*	2.31	*7.88*	*0.80*	*9.31*
	Gaussian Cond.	—	—	2.81	28.85	1.97	15.12	2.67	26.92
OOD	Linear Causal (PI)	—	—	**1.07**	*18.55*	**2.55**	*16.96*	**0.73**	**15.96**
	ACTIVA (ours)	2.97	15.50	*2.96*	**17.97**	7.63	**11.43**	*1.87*	*16.48*
	Gaussian Cond.	—	—	8.22	90.87	*3.00*	23.29	8.20	88.56

Comparing ACTIVA and MACE-TNP to the ground-truth samples, both methods recover the mean shifts induced by the interventions. Qualitatively, MACE-TNP sometimes matches the spread of the non-intervened variables more closely, whereas ACTIVA often yields tighter predictions for the intervened coordinates in the examples. Overall, these plots support the following quantitative results: ACTIVA captures the main interventional shifts while avoiding many of the spurious effects produced by the correlational baseline.

7.2 Interventional Distribution Estimation

We next quantify how well ACTIVA captures the post-interventional distributions observed qualitatively above. For each test SCM, we run inference with the model trained on the corresponding training split and evaluate the resulting distributions using NLL and energy distance. For the Linear Causal and Gaussian Conditional baselines, we omit the *All* condition because these methods do not directly predict the intervened dimensions. Bold and italic numbers indicate the best and second-best results per condition, respectively. Evaluation of MACE-TNP on the OOD SERGIO dataset is not possible because our implementation requires the same input dimensionality during training and inference.

Table 1 shows first that ACTIVA consistently improves over the Gaussian Conditional baseline on NLL and, most importantly, strongly reduces spurious effects on non-descendants. This is the clearest evidence

that ACTIVA learns predominantly causal rather than merely correlational structure: variables whose true causal effect is zero are assigned substantially smaller distribution shifts than under the conditional Gaussian baseline.

On the two-variable datasets, ACTIVA achieves very low errors overall. The strongest result appears on *Beta 2*, where ACTIVA outperforms both MACE-TNP and the Linear Causal baseline, which has privileged information, across the reported metrics, consistent with the advantage of a flexible mixture decoder in a non-Gaussian setting. On *Gauss 2*, ACTIVA remains close to the asymptotic oracle Linear Causal baseline, indicating that the theoretical non-identifiability of the Gaussian noise setting seems to play a minor role in the practical estimation quality. Furthermore, ACTIVA improves over MACE-TNP on *Gauss 2*, although not uniformly across every condition and metric.

On the larger synthetic datasets *Gauss 8* and *Beta 8*, the results reveal complementary strengths across amortized methods. ACTIVA remains clearly superior to the correlational baseline, but MACE-TNP attains lower energy distance and lower NLL on several 8-variable synthetic subsets. This suggests that, for larger synthetic graphs, MACE-TNP can better match sample-space geometry, whereas ACTIVA is still competitive but no longer uniformly best.

The semi-synthetic SERGIO results highlight a different strength of ACTIVA. Here, ACTIVA achieves the best NLL in all reported marginals and remains competitive in energy distance, indicating particularly strong density estimation on biologically plausible data. In the OOD SERGIO setting, ACTIVA still substantially outperforms the Gaussian Conditional baseline, especially in NLL, but performance degrades relative to the in-distribution setting, most notably on descendant energy distance. We therefore interpret the OOD result as evidence of partial robustness rather than uniformly strong generalization.

Comparing ACTIVA and MACE-TNP reveals complementary trade-offs. ACTIVA tends to provide better-calibrated joint densities when the intervened variables are included in the target distribution, as reflected by its stronger NLL in the *All* condition. In contrast, MACE-TNP is often stronger in energy distance on the larger synthetic problems when the target variable is not of interest. Taken together, these results suggest that ACTIVA is particularly attractive when the goal is to estimate the full post-intervention density from observational data and a weakly specified intervention query, rather than only a marginal over non-intervened variables.

Overall, the results in Table 1 show that ACTIVA successfully leverages causal information at inference time. In a single forward pass, it produces competitive interventional distributions, substantially improves over a correlational baseline, and remains especially strong on small graphs and semi-synthetic data, given only observational data and a weakly specified intervention query.

7.3 Downstream Task Performance

We finally assess whether the distributional predictions of ACTIVA translate into accurate downstream effect estimates. This is especially of interest, as we are interested in the ability to use our methods to derive causal conclusions; even when distributional aspects like covariances and multi-modality are not of interest. To this end, Figure 4 reports the mean absolute error of the ATE for the full set of non-intervened variables, for descendants only, and for non-descendants only.

Figure 4 suggests the same qualitative pattern as Table 1. ACTIVA clearly improves over the Gaussian Conditional baseline, especially on non-descendants, indicating that it captures not only causal effects but also the absence of effects. Thus, the gains in Section 7.2 are not limited to density modeling alone; they carry over to a standard causal estimand.

Relative to MACE-TNP, ACTIVA is broadly competitive rather than uniformly superior. The clearest advantage appears on *Beta 2*, whereas MACE-TNP is stronger on *Beta 8*, and the remaining datasets are comparatively close. This suggests that the rich distributional modeling of ACTIVA does not come at the expense of average-effect estimation, but it also does not automatically translate into uniformly better ATE performance compared to MACE-TNP.

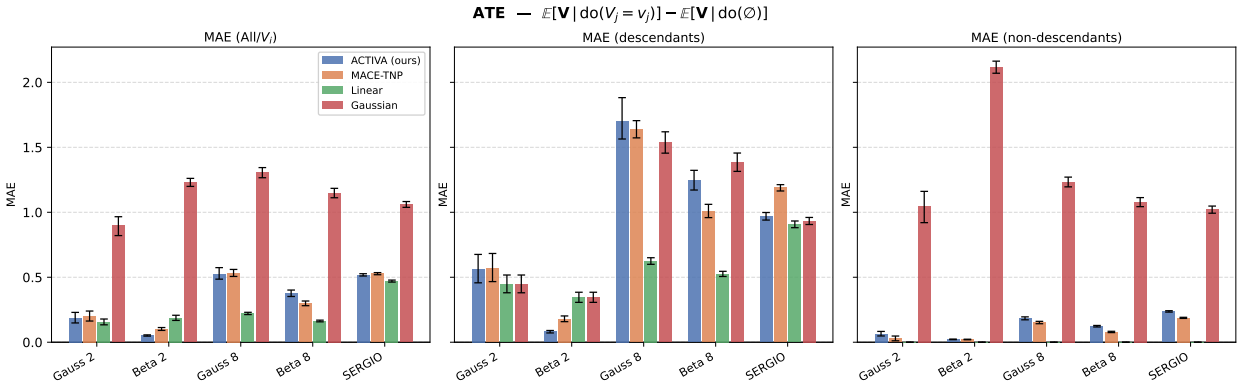

Figure 4: Mean absolute error of the ATE of the predicted distributions w.r.t. the ground-truth sample ATE. The left panel shows the error on the full joint distribution (excluding the intervened variable), the middle panel shows the error for the causal descendants, and the right panel for the causal non-descendants of the intervention target. The different approaches and baselines are color-coded. Lower is better.

Oracle-level performance on small graphs and semi-synthetic data. ACTIVA is also competitive with the privileged-information Linear Causal baseline on the smaller problems and on SERGIO. Especially for the *Beta 2* dataset, these results indicate that predicting the post-interventional distribution as a GMM helps with the downstream mean prediction. However, for larger graphs, this advantage over simple Gaussian prediction is quickly offset when the underlying causal graph is known.

These results are consistent with ACTIVA exploiting causal rather than merely correlational structures from observational data and weakly specified intervention queries. Furthermore, this benefit remains visible even when the final goal is a standard treatment-effect estimate rather than the full interventional distribution.

8 Related Work

Deep learning approaches to causal inference have advanced the modeling of observational, interventional, and counterfactual distributions, yet, most still require either known or learnable causal graphs, relatively strong structural assumptions, or dataset-specific fitting at test time. This holds for VAE-based models (Yang et al., 2021; Qi & Yu, 2023), diffusion generative models (Sanchez et al., 2022; Chao et al., 2023), graph-neural and flow-based approaches (Zečevi et al., 2021; Sánchez-Martin et al., 2022; Poinsot et al., 2024), and adversarial network approaches (Rahman & Kocaoglu, 2024).

More recently, a line of work has begun to amortize causal inference across datasets. Foundation-model and meta-learning approaches train a single model across many simulated tasks and then perform zero-shot or in-context inference on new datasets (Balazadeh et al., 2025; Robertson et al., 2025; Bynum et al., 2025; Ma et al., 2025). These methods substantially reduce per-dataset computation, but they primarily target causal-effect summaries or conditional interventional quantities in identifiable settings, rather than the full joint post-intervention distribution over all variables.

The closest recent work to ours is MACE-TNP, which meta-learns interventional distributions directly from observational data under causal-graph uncertainty (Dhir et al., 2025). Relative to MACE-TNP, ACTIVA differs in two main ways. First, ACTIVA is designed to predict the full joint post-intervention distribution rather than the interventional distribution of a designated outcome variable. Second, ACTIVA is tailored to weakly specified interventions, conditioning on intervention identifiers and targets rather than requiring a fully specified intervention value at inference time.

Our work is also related to approaches that amortize inference over SCMs, including zero-shot and meta-learned SCM inference methods (Löwe et al., 2022; Nilforoshan et al., 2023; Sauter et al., 2024b; Mahajan et al., 2024). These methods aim to recover or simulate causal models themselves, whereas our goal is differ-

ent: we seek direct estimation of post-intervention distributions from observational data and an intervention query without requiring explicit graph recovery at inference time.

Finally, some methods focus on specialized causal-estimation settings, such as treatment-outcome estimation, limited overlap, or marginal interventional supervision, without targeting the full joint interventional distribution across all variables (Louizos et al., 2017; Vowels et al., 2021; Wu & Fukumizu, 2023; Vanderschueren et al., 2023; Garrido et al., 2024). In contrast, ACTIVA is aimed at the amortized estimation of the full post-interventional distribution from observational inputs and weakly specified intervention queries, while explicitly representing ambiguity through a mixture distribution induced by observationally compatible causal models.

9 Conclusion

We introduced ACTIVA, a conditional variational autoencoder for amortized estimation of post-intervention distributions from observational data and weakly specified intervention queries. Unlike classical causal-effect estimators that return only summary quantities, ACTIVA is designed to predict the full joint distribution after intervention, thereby capturing both downstream effect estimates and richer forms of causal uncertainty, such as multi-modality. Our theoretical analysis provides a consistency result: under idealized conditions, ACTIVA's learning objective targets a mixture over the interventional distributions of models that remain observationally compatible with the input, making explicit how observational ambiguity is reflected in the estimand.

Empirically, ACTIVA demonstrates that this theory-guided amortized approach is effective in finite models. Across synthetic datasets and biologically realistic gene-expression simulations, ACTIVA substantially outperforms a purely correlational baseline, reduces spurious effects on non-descendants, and achieves competitive performance relative to strong amortized baselines. Its distributional predictions also translate into competitive downstream ATE estimates, showing that the learned post-interventional distributions capture causal structure in a way that remains useful for standard causal-effect summaries. Taken together, these findings position ACTIVA as a promising approach when the goal is zero-shot estimation of full post-interventional distributions from observational data and partially specified interventions, without requiring explicit causal model recovery at inference time.

More broadly, we view this work as a step toward amortized causal inference systems that represent structural ambiguity explicitly in the predicted post-intervention distribution, rather than collapsing it into a single point estimate. Important directions for future work include tightening the connection between the population-level theory and finite-sample implementations, improving performance on larger and more weakly matched out-of-distribution settings, and evaluating the approach on real-world interventional problems where weakly specified interventions arise naturally.

References

Yashas Annadani, Panagiotis Tigas, Stefan Bauer, and Adam Foster. Amortized Active Causal Induction with Deep Reinforcement Learning. *Advances in Neural Information Processing Systems*, 37:44216–44239, 1 2025. URL `https://arxiv.org/abs/2405.16718v1`.

Henri Bal, Dick Epema, Cees de Laat, Rob van Nieuwpoort, John Romein, Frank Seinstra, Cees Snoek, and Harry Wijshoff. A Medium-Scale Distributed System for Computer Science Research: Infrastructure for the Long Term. *Computer*, 49(5):54–63, 5 2016. ISSN 0018-9162. doi: 10.1109/MC.2016.127.

Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Benson Li, Junwei Ma, Jesse C Cresswell, and Rahul G Krishnan. CausalPFN: Amortized Causal Effect Estimation via In-Context Learning. *arXiv preprint*, 2025. URL `https://github.com/vdblm/CausalPFN`.

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. On Pearl's Hierarchy and the Foundations of Causal Inference. *Probabilistic and Causal Inference*, pp. 507–556, 2 2022. doi: 10.1145/3501714.3501743. URL `https://dl.acm.org/doi/10.1145/3501714.3501743`.

Kenneth A Bollen. *Structural Equations with Latent Variables*. John Wiley \& Sons, 1989.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Lucius E J Bynum, Aahlad Puli, Diego Herrero-Quevedo, Nhi Nguyen, Carlos Fernandez-Granda, Kyunghyun Cho, and Rajesh Ranganath. Black Box Causal Inference: Effect Estimation via Meta Prediction. *arXiv preprint*, 2025.

Patrick Chao, Patrick Blöbaum, Shiva Prasad, and Kasiviswanathan Amazon. Interventional and Counterfactual Inference with Diffusion Models. *arXiv preprint*, 2023. URL https://github.com/patrickrchao/.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference Suboptimality in Variational Autoencoders. 2018.

Anish Dhir, Cristiana Diaconu, Valentinian Mihai Lungu, James Requeima, Richard E Turner, and Mark van der Wilk. Estimating Interventional Distributions with Uncertain Causal Graphs through Meta-Learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=IQlcfc40Ja.

Payam Dibaeinia and Saurabh Sinha. SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks. *Cell Systems*, 11(3):252–271, 9 2020. ISSN 2405-4712. doi: 10.1016/J.CELS.2020.08.003.

P. Erdós and A. Rényi. On random graphs. *Publicationes Mathematicae*, pp. 290–297, 1959.

Sergio Garrido, Elke Kirschbaum, Armin Kekic, and Atalanti Mastakouri. Estimating Joint interventional distributions from marginal interventional data. *arXiv preprint*, 9 2024. URL http://arxiv.org/abs/2409.01794.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024. URL http://github.com/google/flax.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International conference on learning representations*, 2 2017.

Matthew D Hoffman and Matthew Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference NIPS*, 2016.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013. doi: 10.61603/ceas.v2i1.33. URL https://arxiv.org/abs/1312.6114v11.

Jannik Kossen, Neil Band, Clare Lyle, Aidan N. Gomez, Thomas Rainforth, and Yarin Gal. Self-Attention Between Datapoints: Going Beyond Individual Input-Output Pairs in Deep Learning. *Advances in Neural Information Processing Systems*, 34:28742–28756, 12 2021.

Satyam Kumar, Yelleti Vivek, Vadlamani Ravi, and Indranil Bose. Causal Inference for Banking Finance and Insurance A Survey. *arXiv preprint*, 7 2023. URL https://arxiv.org/abs/2307.16427v1.

Hebi Li, Qi Xiao, and Jin Tian. Supervised Whole DAG Causal Discovery. *arXiv preprint*, 2020.

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. DiBS: Differentiable Bayesian Structure Learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 12 2021. URL https://github.com/larslorch/dibs.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized Inference for Causal Structure Learning. In S Koyejo, S Mohamed, A Agarwal, D Belgrave, K Cho, and A Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 13104–13118. Curran Associates, Inc., 2022.

Christos Louizos, Uri Shalit, Joris M. Mooij, David Sontag, Richard Zemel, and Max Welling. Causal Effect Inference with Deep Latent-Variable Models. *Advances in Neural Information Processing Systems*, 30, 2017.

Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized Causal Discovery: Learning to Infer Causal Graphs from Time-Series Data. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 509–525. PMLR, 8 2022. URL https://proceedings.mlr.press/v177/lowe22a.html.

Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation Models for Causal Inference via Prior-Data Fitted Networks. *arXiv preprint*, 2025. URL https://github.com/yccm/CausalFM.

Marloes H Maathuis, Markus Kalisch, Peter Bühlmann, and Eth Zürich. Estimating High-Dimensional Intervention Effects from Observational Data. *The Annals of Statistics*, 37(6A):3133–3164, 2009. doi: 10.1214/09-AOS685.

Divyat Mahajan, Jannes Gladrow, Agrin Hilmkil, Cheng Zhang, and Meyer Scetbon. Zero-Shot Learning of Causal Models. *arXiv preprint*, 2024.

Daniel Marbach, Thomas Schaffter, Claudio Mattiussi, and Dario Floreano. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of computational biology : a journal of computational molecular cell biology*, 16(2):229–239, 2009. ISSN 1557-8666. doi: 10.1089/CMB.2008.09TT. URL https://pubmed.ncbi.nlm.nih.gov/19183003/.

Francesco Montagna, Max Cairney-Leeming, Dhanya Sridhar, and Francesco Locatello. Demystifying amortized causal discovery with transformers. *arXiv preprint*, 5 2024. URL https://arxiv.org/abs/2405.16924v1.

Hamed Nilforoshan, Michael Moor, Yusuf Roohani, Yining Chen, Anja Šurina, Michihiro Yasunaga, Sara Oblak, and Jure Leskovec. Zero-shot causal learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://github.com/snap-stanford/caml/.

Ugo Panizza and Andrea F. Presbitero. Public debt and economic growth: Is there a causal effect? *Journal of Macroeconomics*, 41:21–41, 9 2014. ISSN 0164-0704. doi: 10.1016/J.JMACRO.2014.03.009.

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

Jan Peters, Dominik Janzig, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Audrey Poinsot, Alessandro Leite, Nicolas Chesneau, Michèle Sébag, and Marc Schoenauer. Learning Structural Causal Models through Deep Generative Models: Methods, Guarantees, and Challenges. *arXiv preprint*, 2024.

Guodong Qi and Huimin Yu. CMVAE: Causal Meta VAE for Unsupervised Meta-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9480–9488, 6 2023. ISSN 2374-3468. doi: 10.1609/AAAI.V37I8.26135. URL https://ojs.aaai.org/index.php/AAAI/article/view/26135.

Md Musfiqur Rahman and Murat Kocaoglu. Modular Learning of Deep Causal Generative Models for High-dimensional Causal Inference. *arXiv preprint*, 2024.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter †4, and Bernhard Schölkopf. Do-PFN: In-Context Learning for Causal Effect Estimation. *arXiv preprint*, 2025. URL https://github.com/.

Pedro Sanchez, Sotirios A Tsaftaris, Bernhard Schölkopf, Caroline Uhler, and Kun Zhang. Diffusion Causal Models for Counterfactual Estimation, 6 2022. ISSN 2640-3498. URL https://proceedings.mlr.press/v177/sanchez22a.html.

Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. VACA: Designing Variational Graph Autoencoders for Causal Queries. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):8159–8168, 6 2022. ISSN 2374-3468. doi: 10.1609/AAAI.V36I7.20789. URL https://ojs.aaai.org/index.php/AAAI/article/view/20789.

Andreas Sauter, Erman Acar, and Aske Plaat. CausalPlayground: Addressing Data-Generation Requirements in Cutting-Edge Causality Research. *arXiv preprint*, 2024a.

Andreas W M Sauter, Nicolò Botteghi, Erman Acar, and Aske Plaat. CORE: Towards Scalable and Efficient Causal Discovery with Reinforcement Learning. *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1664–1672, 2024b.

Meyer Scetbon, Joel Jennings, Agrin Hilmkil, Cheng Zhang, and Chao Ma. A Fixed-Point Approach for Causal Generative Modeling, 2024.

Jingpu Shi and Beau Norgeot. Learning Causal Effects From Observational Data in Healthcare: A Review and Summary. *Frontiers in Medicine*, 9:864882, 7 2022. ISSN 2296858X. doi: 10.3389/FMED.2022.864882/BIBTEX. URL www.frontiersin.org.

Kihyuk Sohn, Xinchen Yan, and Honglak Lee. Learning Structured Output Representation using Deep Conditional Generative Models. *Advances in Neural Information Processing Systems*, 2015.

Gábor J. Székely and Maria L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 8 2013. ISSN 0378-3758. doi: 10.1016/J.JSPI.2013.03.018.

Toon Vanderschueren, Jeroen Berrevoets, Wouter Verbeke, and K U Leuven. NOFLITE: Learning to Predict Individual Treatment Effect Distributions. *Transactions on Machine Learning Research*, 2023.

Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 2017.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. Targeted VAE: Variational and Targeted Learning for Causal Inference. *IEEE International Conference on Smart Data Services*, 2021. doi: 10.1109/SMDS53860.2021.00027. URL https://github.com/matthewvowels1/TVAE.

Pengzhou Wu and Kenji Fukumizu. β-Intact-VAE: Identifying and Estimating Causal Effects under Limited Overlap. *International Conference on Learning Representation*, 2023.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

Matej Zečevi, Devendra Singh Dhami, Petar Veličkovi, and Kristian Kersting KERSTING. Relating Graph Neural Networks to Structural Causal Models. *arXiv preprint*, 2021. URL https://anonymous.4open.science/r/.

A Derivation of KL Decomposition

In this section, we derive the KL decomposition used in Section 4. Since the theory is stated at the level of the simulator distribution $p_{tr}(\mathcal{M})$, we work directly with the corresponding population objective. The finite-data objective used in practice can be viewed as a Monte Carlo approximation of the expectations below.

Before restricting to a fixed observational equivalence class and query, note that the KL contribution to the population objective decomposes by the law of total expectation over pairs $S = ([\mathcal{M}], do(V))$:

$$\mathbb{E}_{\mathbf{D}^{tr}\sim p_{tr}(\mathbf{D}^{tr})}\Big[\,\mathrm{KL}\left(q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})\,\|\,p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}},do(V))\right)\Big] = \mathbb{E}_{S\sim p_{tr}(S)}\Big[\mathbb{E}_{\mathbf{D}^{tr}\sim p_{tr}(\mathbf{D}^{tr}|S)}\big[\mathrm{KL}(\cdots)\big]\Big]. \tag{14}$$

Hence, it is sufficient to analyze the inner expectation for an arbitrary fixed pair S. Since the outer expectation in Equation 14 is only a weighted average over such terms, the dependence of the full objective on the prior parameters η is completely determined by the class-conditional decomposition derived below.

Fix an observational equivalence class $[\mathcal{M}^o]$ and a query intervention $do(V)$. Restricting the expected KL term in Equation 6 to tasks whose observational component lies in $[\mathcal{M}^o]$ and whose query equals $do(V)$ gives

$$\mathbb{E}_{\mathbf{D}^{tr}\sim p_{tr}(\mathbf{D}^{tr}|[\mathcal{M}^o],do(V))}\Big[\,\mathrm{KL}\left(q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})\,\|\,p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}^o},do(V))\right)\Big]. \tag{15}$$

Following the aggregated-posterior decomposition for amortized variational objectives of Hoffman & Johnson (2016), we rewrite this expected KL term by introducing the corresponding aggregated posterior over tasks in the same observational equivalence class and intervention query.

Expanding the KL divergence yields

$$\begin{aligned}&\mathbb{E}_{\mathbf{D}^{tr}\sim p_{tr}(\mathbf{D}^{tr}|[\mathcal{M}^o],do(V))}\Big[\,\mathrm{KL}\left(q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})\,\|\,p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}^o},do(V))\right)\Big]\\ &= \mathbb{E}_{\mathbf{D}^{tr}\sim p_{tr}(\mathbf{D}^{tr}|[\mathcal{M}^o],do(V))}\left[\int q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})\log\frac{q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})}{p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}^o},do(V))}\,d\boldsymbol{z}\right].\end{aligned} \tag{16}$$

Writing the outer expectation as an integral over tasks gives

$$\begin{aligned}&= \int\left[\int q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})\log\frac{q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})}{p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}^o},do(V))}\,d\boldsymbol{z}\right]p_{tr}(\mathbf{D}^{tr}\mid[\mathcal{M}^o],do(V))\,d\mathbf{D}^{tr}\\ &= \int p_{tr}(\mathbf{D}^{tr}\mid[\mathcal{M}^o],do(V))q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})\log\frac{q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})}{p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}^o},do(V))}\,d\boldsymbol{z}\,d\mathbf{D}^{tr}.\end{aligned} \tag{17}$$

Defining the joint distribution

$$q_\phi(\mathbf{D}^{tr},\boldsymbol{z}\mid[\mathcal{M}^o],do(V)) := p_{tr}(\mathbf{D}^{tr}\mid[\mathcal{M}^o],do(V))\,q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr}), \tag{18}$$

we can rewrite the expected KL term as

$$\int q_\phi(\mathbf{D}^{tr},\boldsymbol{z}\mid[\mathcal{M}^o],do(V))\log\frac{q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})}{p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}^o},do(V))}\,d\boldsymbol{z}\,d\mathbf{D}^{tr}. \tag{19}$$

We define the aggregated posterior

$$\bar{q}_\phi(\boldsymbol{z}\mid[\mathcal{M}^o],do(V)) := \int q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})\,p_{tr}(\mathbf{D}^{tr}\mid[\mathcal{M}^o],do(V))\,d\mathbf{D}^{tr}. \tag{20}$$

Adding and subtracting $\log\bar{q}_\phi(\boldsymbol{z}\mid[\mathcal{M}^o],do(V))$ inside the logarithm yields

$$\int q_\phi(\mathbf{D}^{tr},\boldsymbol{z}\mid[\mathcal{M}^o],do(V))\log\frac{q_\phi(\boldsymbol{z}\mid\mathbf{D}^{tr})}{p_\eta(\boldsymbol{z}\mid\boldsymbol{D}^{\mathcal{M}^o},do(V))}\,d\boldsymbol{z}\,d\mathbf{D}^{tr}$$

$$= \int q_\phi(\mathbf{D}^{tr}, \boldsymbol{z} \mid [\mathcal{M}^o], do(V)) \log \frac{q_\phi(\boldsymbol{z} \mid \mathbf{D}^{tr})}{\bar{q}_\phi(\boldsymbol{z} \mid [\mathcal{M}^o], do(V))} \, d\boldsymbol{z} \, d\mathbf{D}^{tr}$$
$$+ \int q_\phi(\mathbf{D}^{tr}, \boldsymbol{z} \mid [\mathcal{M}^o], do(V)) \log \frac{\bar{q}_\phi(\boldsymbol{z} \mid [\mathcal{M}^o], do(V))}{p_\eta(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V))} \, d\boldsymbol{z} \, d\mathbf{D}^{tr}. \tag{21}$$

The first term is the conditional mutual information between the task and the latent variable under q_ϕ,

$$\mathbb{I}_{q_\phi}\left(\mathbf{D}^{tr}; \boldsymbol{z} \mid [\mathcal{M}^o], do(V)\right), \tag{22}$$

while in the second term the dependence on $\mathbf{D}^{tr}$ disappears inside the logarithm. We can therefore marginalize out $\mathbf{D}^{tr}$ using the definition of $\bar{q}_\phi$ in Equation 20, which gives

$$\mathbb{E}_{\mathbf{D}^{tr} \sim p_{tr}(\mathbf{D}^{tr} \mid [\mathcal{M}^o], do(V))} \Big[\mathrm{KL}\left(q_\phi(\boldsymbol{z} \mid \mathbf{D}^{tr}) \,\|\, p_\eta(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V))\right) \Big]$$
$$= \mathrm{KL}\Big(\bar{q}_\phi(\boldsymbol{z} \mid [\mathcal{M}^o], do(V)) \,\Big\|\, p_\eta(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V))\Big) + \mathbb{I}_{q_\phi}\left(\mathbf{D}^{tr}; \boldsymbol{z} \mid [\mathcal{M}^o], do(V)\right). \tag{23}$$

In the main text, we focus on the KL term in Equation 23, since it is the only term that depends on the prior parameters η. This yields the expression used in Equation 9.

B Proposition 2 Proof

Here we give the full proof of Proposition 2 from the main text. We start by re-stating the proposition.

Proposition 2. Under Assumptions 1–3, the ACTIVA predictive distribution for an input $\left(\boldsymbol{D}^{\mathcal{M}^o}, do(V)\right)$ is given by

$$p_{\theta^*}(\boldsymbol{V} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = \int p_{\mathcal{M}}(\boldsymbol{V} \mid do(V)) \, p_{tr}(\mathcal{M} \mid [\mathcal{M}^o], do(V)) \, d\mathcal{M}. \tag{24}$$

Proof. Starting from the ACTIVA generative model in Equation 4, we marginalize out the latent variable:

$$p_{\theta^*}(\boldsymbol{V} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = \int p_{\gamma^*}(\boldsymbol{V} \mid \boldsymbol{z}) \, p_{\eta^*}(\boldsymbol{z} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) \, d\boldsymbol{z}. \tag{25}$$

Using Proposition 1, this becomes

$$p_{\theta^*}(\boldsymbol{V} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = \int p_{\gamma^*}(\boldsymbol{V} \mid \boldsymbol{z}) \, \bar{q}_{\phi^*}(\boldsymbol{z} \mid [\mathcal{M}^o], do(V)) \, d\boldsymbol{z} \tag{26}$$
$$= \int \left[\int p_{\gamma^*}(\boldsymbol{V} \mid \boldsymbol{z}) \, q_{\phi^*}(\boldsymbol{z} \mid \mathbf{D}^{tr}) \, d\boldsymbol{z} \right] p_{tr}(\mathbf{D}^{tr} \mid [\mathcal{M}^o], do(V)) \, d\mathbf{D}^{tr}. \tag{27}$$

By Assumption 2, the inner integral equals the true post-interventional distribution of the model that generated the task $\mathbf{D}^{tr}$:

$$\int p_{\gamma^*}(\boldsymbol{V} \mid \boldsymbol{z}) \, q_{\phi^*}(\boldsymbol{z} \mid \mathbf{D}^{tr}) \, d\boldsymbol{z} = p_{\mathcal{M}}(\boldsymbol{V} \mid do(V)). \tag{28}$$

Substituting this into the previous expression yields

$$p_{\theta^*}(\boldsymbol{V} \mid \boldsymbol{D}^{\mathcal{M}^o}, do(V)) = \int p_{\mathcal{M}}(\boldsymbol{V} \mid do(V)) \, p_{tr}(\mathcal{M} \mid [\mathcal{M}^o], do(V)) \, d\mathcal{M}, \tag{29}$$

which proves the claim. □

C Datasets

We evaluate the performance of the proposed method across three different types of datasets: two purely synthetic (*Gaussian* and *Beta*) and one semi-synthetic (*SERGIO*). Below, we provide an overview of each dataset category along with the relevant parameters. For every causal model in these categories, we draw N samples from both the observational and the interventional distributions to create paired datasets. Details about the exact number of models, how we split into training/test sets, and further parameter choices are given in the subsections below.

Table 2: SERGIO data generation parameters.

parameter	in-distribution	out-of-distribution
genes	8	11
b	Uniform(0, 1)	Uniform(0.5, 2.0)
k_param	Uniform(1, 5)	Uniform(3, 7)
k_sign	Beta(1, 1)	Beta(0.5, 0.5)
hill	$\in\{$ 1.9, 2.0, 2.1$\}$	$\in\{$ 1.5, 2.5$\}$
decays	$\in\{$ 0.7, 0.8, 0.9$\}$	$\in\{$ 0.5, 1.5$\}$
noise_params	$\in\{0.9, 1.0, 1.1\}$	$\in\{0.5, 1.5\}$

C.1 Synthetic Data (Gaussian and Beta Noise)

We generate data from linear additive causal models of the form:

$$V_j = \sum_{i \in \mathrm{Pa}(j)} \beta_{ij} V_i + \varepsilon_j,$$

where ε_j is drawn from either: A Gaussian distribution, $\mathcal{N}(0, \sigma^2)$, or A Beta distribution, $\mathrm{Beta}(\alpha, \beta)$. For each variable V_i, we generate interventional data by applying a single-variable intervention $do(V_i = 5)$ plus some noise $\mathcal{N}(0, 0.1)$ to increase numerical stability. In total, this yields one observational dataset and d interventional datasets for each causal model. All synthetic data is generated with the *Causal Playground* library (Sauter et al., 2024a). We split each dataset into a train (80%), test (10%) and validation (10%) set.

The causal graphs are generated according to the ER procedure (Erdós & Rényi, 1959) where first we generate an ER graph with edge-probability 0.3 and then randomly remove edges until the graph becomes acyclic. When the noise terms follow a Gaussian distribution, we sample β_{ij} uniformly from $[-2, -0.5] \cup [0.5, 2]$ and let $\varepsilon_j \sim \mathcal{N}(0, 0.5)$. For the Beta case, we again draw β_{ij} in $[-2, -0.5] \cup [0.5, 2]$, but the noise terms ε_j follow $\mathrm{Beta}(\alpha, \beta)$ with $\alpha, \beta \sim \mathrm{Uniform}(0.5, 2)$. We examine both a 2-variable and an 8-variable scenario: in the 2-variable case, we randomly generate 3000 linear models and sample 100 points each for the observational and interventional distributions, whereas in the 8-variable case, we generate 20000 linear models and, for each model, sample 80 data points for the observational and for each interventional distribution.

C.2 Semi-Synthetic Data (SERGIO)

To evaluate on data with biological realism, we employ the SERGIO simulator (Dibaeinia & Sinha, 2020), which generates single-cell gene-expression data. Notably, we use the version of SERGIO provided in (Lorch et al., 2022), which includes functionality for performing interventions. We treat single-gene knockouts as interventions, thus applying $do(V_i = 0)$ to each gene V_i. This yields one observational dataset and 8 single-gene interventional datasets per simulated gene-regulatory network.

Simulator Settings and Network Structures. Following the procedure in (Marbach et al., 2009; Lorch et al., 2021), we randomly sample subgraphs of known gene-regulatory networks from *E. coli* or *S. cerevisiae*, ensuring that each subgraph has 8 genes. For each subgraph, we draw model parameters (e.g., activation constants, decay rates, and noise magnitudes) from predefined ranges (see Table 2). We then generate observational data as well as data from each gene-knockout intervention.

Training, Testing, and Out-of-Distribution (OOD) Splits. We run 15,000 SERGIO simulations for our in-distribution dataset and sample 30 cells (data points) for each observational and interventional condition. We then split these 15,000 simulations into training (80%), validation (10%) and test (10%). We also construct an OOD set of 1800 simulations, where some SERGIO parameters (e.g., noise or decay rates) are sampled from partially disjoint ranges, creating a controlled distribution shift. For evaluation, we again sample 30 cells per observational/interventional condition in these OOD simulations.

D MACE-TNP Extension Implementation Details

This section describes our extension of MACE-TNP (Model-Averaged Causal Estimation Transformer Neural Process) (Dhir et al., 2025) in detail. MACE-TNP is a meta-learned neural process for causal inference that predicts *marginal* interventional distributions $p(x_i \mid \mathrm{do}(X_j = v), \mathcal{D}_{\text{obs}})$ for a designated outcome variable X_i, given an intervention on variable X_j. Its architecture consists of: (1) a Transformer-based encoder that alternates attention across the sample axis and the variable (node) axis to produce permutation-equivariant per-node representations, and (2) a per-variable MoG predictor that extracts the representation at the outcome node index i and decodes it into a univariate mixture of Gaussians with K components. The encoder uses three distinct embedding roles—*intervention node*, *outcome node*, and *marginalized node*—each implemented as a separate MLP that maps scalar observations to d_{model}-dimensional vectors, with separate copies for observational and interventional samples (six MLPs total). Sample-wise attention uses multi-head self-attention (MHSA) on observational tokens followed by multi-head cross-attention (MHCA) from interventional tokens to observational tokens; node-wise attention uses MHSA across all nodes. The predictive distribution factorizes over interventional samples: $p_\theta(\mathbf{x}_i \mid \mathrm{do}(\mathbf{x}_j), \mathcal{D}_{\text{obs}}) = \prod_{n=1}^{N_{\text{int}}} p_\theta(x_i^n \mid \mathrm{do}(x_j^n), \mathcal{D}_{\text{obs}})$.

Our implementation modifies MACE-TNP to estimate the *joint* interventional distribution over all variables simultaneously, rather than per-variable marginals. This requires three architectural changes:

D.1 Two-role encoder.

Since we predict the joint distribution over all variables, there is no designated outcome variable and hence no need for the outcome embedding role. We replace the original three-role encoder with a two-role encoder that uses only *marginalized* (variable) and *intervention* embeddings. For each node $d \in \{1, \ldots, D\}$, the embedding is:

$$\mathbf{h}_d = \begin{cases} f_{\text{int}}(x_d) & \text{if } d = j \text{ (intervened variable)}, \\ f_{\text{var}}(x_d) & \text{otherwise}, \end{cases} \tag{30}$$

where f_{var} and f_{int} are learned MLPs (with separate copies for observational and interventional inputs, giving four MLPs total). The Transformer encoder backbone is left unchanged.

D.2 Joint multivariate GMM predictor.

We replace the per-variable univariate GMM predictor with a multivariate predictor that outputs a K-component Gaussian mixture model via Cholesky factors. The predictor takes per-node encodings of shape, obtained by mean-pooling the encoder output over the target-sample axis. Component means are computed *per-node*: each node's d_{model}-dimensional encoding is independently mapped to K scalar means via a shared two-layer MLP, preserving the per-variable causal information encoded by the Transformer. For the Cholesky factors and mixture weights, the node encodings are mean-pooled across the variable axis and decoded through a shared backbone MLP followed by separate linear heads. The diagonal entries of each $\mathbf{L}_k$ are passed through a softplus activation with a floor of 0.1 to ensure positive-definiteness, and mixture weights are obtained via softmax.

D.3 Training and inference.

The model is trained by minimizing the negative log-likelihood under the joint GMM like in the original paper. During training, a single randomly sampled target token is used as the encoder query (matching the inference-time regime), while the loss is evaluated against all target samples in the batch. At test time, a dummy target token is constructed with only the intervention variable set to its (z-score normalized) intervention value and the remaining entries zeroed out; the encoder processes this alongside the observational context, and the predicted GMM is sampled to produce draws from the estimated joint interventional distribution.

Both observational and interventional data are z-score normalized using the observational data statistics (per-sample mean and standard deviation), following the normalization convention of the original MACE-TNP. Predictions are denormalized before metric evaluation. Training uses AdamW with cosine annealing, and gradient clipping (max norm 1.0).

D.4 Architectural symmetry properties.

The original MACE-TNP is permutation-equivariant with respect to both observational samples and variables (nodes). The latter property also allows the model to generalize to a different number of variables at test time than seen during training, since the per-variable MoG predictor extracts a single node's representation independently of D. Our joint formulation preserves the sample-axis symmetries: permutation invariance with respect to observational samples and permutation equivariance with respect to interventional samples, since the underlying module is identical to the original.

However, our predictor head does preserve permutation equivariance only for the means, not for the covariances. The component means $\boldsymbol{\mu}_k$ are computed per-node, so they transform equivariantly under variable permutations. The Cholesky factors $\mathbf{L}_k$, however, are decoded from a mean-pooled representation over the variable axis, which is permutation-*invariant*. Consequently, $\mathbf{L}_k$ is the same regardless of variable ordering. This means the model implicitly assumes a fixed variable ordering for the covariance structure and must learn the correct assignment of covariance entries to variable pairs from training data. Furthermore, because the Cholesky head outputs a fixed number of parameters, the number of variables is determined at construction time and cannot vary at inference, unlike the original per-variable formulation.

These are deliberate trade-offs: they allow us to use efficient mean-pooling while predicting a full joint distribution. In our experimental setting, these constraints mainly affect the inductive bias and sample efficiency of the baseline rather than the validity of the maximum-likelihood training objective. We therefore view this extension as a practical joint-distribution adaptation of MACE-TNP, not as a claim that the original permutation-equivariance properties are fully preserved.

E Distribution Plots

In the Figures 5- 8, each column represents samples from the observational and interventional distributions of one model, while each row corresponds to a different randomly drawn SCM from the corresponding test set. The first column shows samples from the ground-truth SCM, columns two to four show the baselines, and the last column shows our model.

Within each panel, the gray points correspond to the true observational data and are repeated across all plots to serve as a common reference. Colored points represent samples from interventional distributions, where the color encodes which variable V_i has been intervened on. All samples are projected onto the (V_0, V_1) plane, so differences between models are visible as shifts or distortions of the colored point clouds relative to the gray observational background and to the ground-truth column.

Qualitatively, a model is accurate if, for each intervention (color), the resulting cloud of points in the corresponding column closely matches the shape, location, and spread of the ground-truth interventional samples in the first column. Systematic deviations in any of these aspects indicate characteristic failure modes: for instance, collapsing the spread suggests underestimation of uncertainty, while consistent shifts or rotations of the point clouds signal misspecified causal effects. By comparing these patterns row-wise across SCMs and column-wise across methods, we can visually assess how reliably each approach captures the interventional behavior of the underlying causal model from observational data.

F LLM Usage

Throughout this work, we made use of various large language models (LLMs) as general-purpose assistive tools across different stages of the research process. Specifically, LLMs were employed to support literature exploration, clarify technical formulations, check the consistency of proofs, assist with coding and debugging, suggest concise rephrasings for improved readability, and aid with formatting tasks (e.g., LaTeX adjustments). All scientific contributions, conceptual developments, and final claims remain the responsibility of the authors.

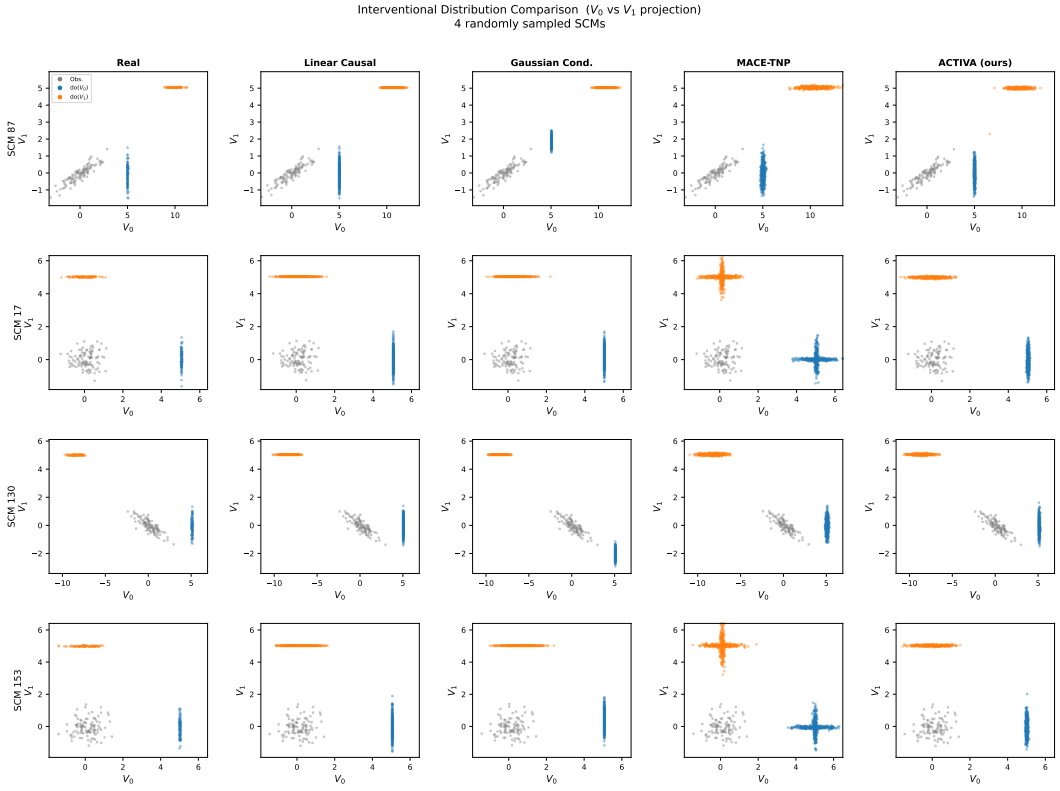

Figure 5: *Gauss 2* test data.

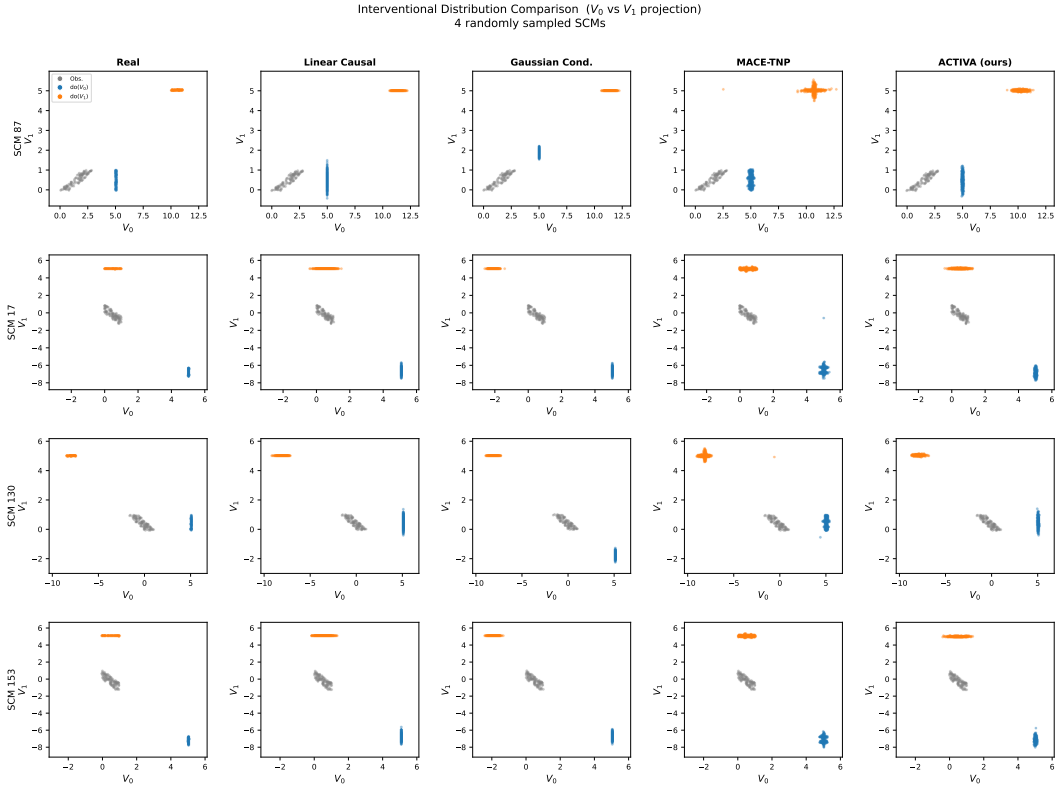

Figure 6: *Beta 2* test data.

Figure 7: *Gauss 8* test data.

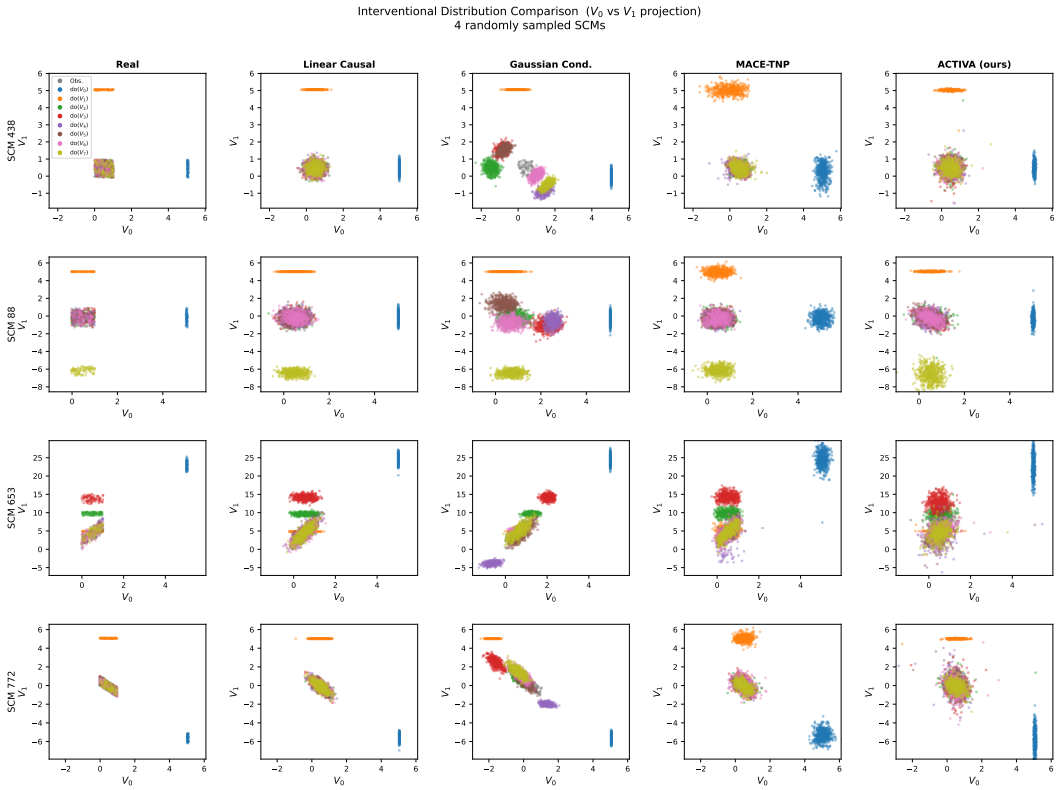

Figure 8: *Beta 8* test data.