Evaluating the Robustness of Interventional Normalizing Flows under Nuisance Misspecification

Ruthvik Allu

A thesis submitted to the Faculty of EEMCS, Delft University of Technology,
in partial fulfilment of the requirements for the degree of
Bachelor of Computer Science and Engineering

June 22, 2025

Name of the student: Ruthvik Allu
Final project course: CSE3000 Research Project
Supervisors: Jesse Krijthe, Rickard Karlsson
Thesis committee: Jesse Krijthe, Rickard Karlsson, Ricardo Marroquim
Faculty: EEMCS, Delft University of Technology, The Netherlands

Abstract

Interventional Normalizing Flows (INFs) are a recently proposed method for estimating interventional outcome distributions from observational data. A central component of this approach is the nuisance flow, whose function is to estimate the propensity score and the conditional outcome distribution. INFs are claimed to be doubly robust, meaning they can yield valid estimates even if only one of these components is correctly specified. This study investigates the practical limits of this robustness by asking two questions: (1) How do interventional estimates behave when nuisance flow components are entirely misspecified? and (2) How sensitive are these estimates to more realistic imperfections such as suboptimal hyperparameters or injected noise? Through experiments on four benchmark datasets with varying levels of confounding and distributional complexity, we find that INFs remain robust under low-confounding conditions even when both nuisance components are broken. However, in high-confounding settings, even partial misspecification can cause estimates to degrade substantially, undermining the doubly robust property. These results highlight the importance of carefully validating nuisance components and suggest that the theoretical guarantees of INFs may not always hold in practice.

1 Introduction

In many traditional machine learning tasks, the goal is to predict what will happen based on patterns found in past data. For example, we might try to forecast next week's energy demand based on weather trends, or estimate how likely someone is to click on an ad based on their browsing history. These kinds of problems focus on identifying associations, meaning they look at which outcomes are likely to occur given certain inputs.

However, in real-world decision-making, we often want to go a step further. Instead of just predicting what is likely to happen, we want to know what would happen if we made a specific change. For example: How would increasing the price of a subscription affect customer retention? What impact would a new marketing strategy have on user engagement? These are questions of intervention, and answering them requires more than just correlation: it requires causal reasoning.

Causal machine learning (causal ML) is the field that tries to answer these kinds of "what if" questions by modeling the effects of interventions. Such interventions, like changing a price or introducing a new policy, are referred to as treatments. Of course, in real life, we often only see what actually happened, but not what would have happened under a different treatment. So to estimate these unobserved outcomes, causal ML methods rely on structured assumptions and specialized statistical techniques (Feuerriegel et al., 2024).

However, many traditional causal ML methods estimate only the average of the potential outcome, such as the Average Treatment Effect (Hatt & Feuerriegel, 2021; Shi et al., 2019) or Conditional Average Treatment Effect (Shalit et al., 2016; Zhang et al., 2020). Prior work has shown that relying only on averages can obscure such risks and, in some cases, lead to harmful or suboptimal decisions (Spiegelhalter, 2017; Van Der Bles et al., 2019). For example, a medical treatment might reduce the average risk for a population, but still carry a high chance of severe side effects for certain individuals.

One recent method that moves beyond average effects is **Interventional Normalizing Flows (INFs)**. In their paper *"Normalizing Flows for Interventional Density Estimation"*, Melnychuk et al. (2022) introduce INFs as a deep learning-based approach designed to estimate *the entire distribution* of outcomes following an intervention, not just the mean. Mathematically, this involves estimating the **interventional density** $\mathrm{P}(\mathrm{Y}[\mathrm{a}] = \mathrm{y})$, which is a function that represents the probability of observing outcome "*y*" if treatment "*a*" were applied. By modeling this full distribution, INFs provide a richer, more nuanced view of potential consequences under different actions.

To produce these estimates, INFs are built around two key components, called the nuisance flow and the target flow. The nuisance flow learns to capture the important background information that helps explain how outcomes behave across different individuals or situations. This information is then passed to the target flow, which uses it to generate the estimated interventional density under a given treatment.

While Melnychuk et al. (2022) demonstrate the effectiveness of INFs across various experimental settings, real-world applications often involve uncertainty and imperfect model assumptions. In practice, components of the model may be misspecified or only approximately correct, raising important questions about the robustness of the estimated interventional distributions. **This paper focuses on the nuisance flow component.**

It remains unclear how much inaccuracies in the nuisance flow affect the quality of the final distribution estimates produced by the target flow. Specifically, how reliable are the interventional density estimates $\mathrm{P}(\mathrm{Y}[\mathrm{a}] = \mathrm{y})$ when the nuisance flow component of an INF is not modeled ideally, an issue that is common in real-world scenarios? The contributions of this paper are centered around answering the following two research questions:

1. **How sensitive are the estimated interventional distributions $\mathbf{P}(\mathbf{Y}[\mathbf{a}] = \mathbf{y})$, produced by the target flow, when the nuisance flow is entirely uninformative?**
2. **How sensitive are the estimated interventional distributions $\mathbf{P}(\mathbf{Y}[\mathbf{a}] = \mathbf{y})$ to minor inaccuracies in the nuisance flow, such as imperfect hyperparameter settings or noise in the estimations?**

While these questions may initially appear to differ only in degree, the distinction is practically significant. The first question addresses extreme scenarios in which the information passed from the nuisance flow to the target flow is entirely incorrect - rendering the nuisance flow effectively non-functional. In contrast, the second question focuses on common imperfections that can occur in applied settings, such as inadequately suited model architectures or noisy estimation outputs caused by approximation error. This split enables a more nuanced and structured investigation:

the first establishes boundaries for robustness under failure, while the second assesses the method1s resilience to more subtle, real-world modeling challenges.

Through this research, we seek to better understand the practical reliability and limitations of INFs in causal inference scenarios. To maintain clarity and focus, this study considers binary treatments and one-dimensional continuous outcomes. A binary treatment refers to cases where $a \in \{0, 1\}$, such as $a = 0$ for “no drug” and $a = 1$ for “drug administered.”

This paper is organized as follows: Section 2 provides background on INFs for causal ML, including their underlying principles and key components. Section 3 outlines the experimental methodology and - together with the section on Responsible Research (Section 7) - highlights the measures taken to ensure responsible and reproducible research practices. Section 4 presents the results of the experiments. Section 5 offers a discussion that interprets these findings and reflects on the experimental setup. Finally, Section 6 concludes the paper.

2 Background

This section introduces the foundational concepts behind Interventional Normalizing Flows that are essential for understanding the research presented in this paper. It begins by explaining how causal effects can be learned from observational data.

2.1 Learning Causal Effects from Observational Data

Training and evaluation of INFs are typically done using **observational datasets** (Melnychuk et al., 2022). This type of data reflects real-world scenarios, not controlled experiments. These datasets consist of tuples (X, A, Y), where X represents covariates (e.g., age, income, medical history), $A \in \{0, 1\}$ is a binary treatment indicator (e.g., drug or no drug), and Y is the observed outcome under the given treatment.

In causal inference with INFs, our goal is not merely to model the **observed conditional distributions** $\mathrm{P}(Y = y \mid A = a)$, which represent the distribution of outcomes among individuals who actually received treatment a in the observational data. Instead, we aim to estimate **interventional distributions**, denoted $\mathrm{P}(Y[a] = y)$, *which represent the distribution of outcomes if we were to intervene and set the treatment $A = a$ for everyone in the population.*

This distinction is critical. The observed quantity $\mathrm{P}(Y = y \mid A = a)$ is subject to **confounding**, because the treatment A was not randomly assigned. For example, people with certain characteristics (like being more ill) might be more likely to receive the treatment. So, differences in outcomes might reflect those characteristics, not the treatment itself. In contrast, $\mathrm{P}(Y[a] = y)$ answers a **causal** question: what would happen to the outcome if we were to *force* treatment $A = a$, regardless of any individual's covariates.

To make causal claims from observational data with INFs, we must rely on a set of crucial assumptions (Melnychuk et al., 2022; Feuerriegel et al., 2024): (1) **Consistency:** The observed outcome Y equals the potential outcome $Y[a]$ for individuals who received treatment $A = a$. (2) **Exchangeability / Unconfoundedness:** Given covariates X, the treatment assignment A is independent of the potential outcomes. This allows us to adjust for confounding using X. (3) **Positivity:** For all relevant covariate values x, there is a non-zero probability of receiving either treatment, i.e., $0 < \mathrm{P}(A = a \mid X = x) < 1$.

By operating within this setting and relying on these assumptions, INFs can turn observational data into meaningful estimates of interventional distributions.

2.2 The Architecture of INFs

In their paper, Melnychuk et al. (2022) introduce INFs as a fully parametric, deep learning-based method that stands out from prior methods for interventional density estimation. What sets INFs apart is that earlier methods, such as kernel density estimation (Kim et al., 2018) and distributional kernel mean embeddings (Muandet et al., 2018), often produced unnormalized or negative densities, and struggled to scale to large datasets. Moreover, before INFs, most alternatives were semi-parametric or non-parametric, which made them less efficient and harder to apply in complex real-world settings (Melnychuk et al., 2022).

At the core of the INF method are **normalizing flows**, a powerful and flexible technique for density estimation and generative modeling (Tabak & Vanden-Eijnden, 2010; Rezende & Mohamed, 2015). They work by learning a sequence of invertible and differentiable transformations that gradually reshape a simple, known distribution (like a standard Gaussian) into a more complex one that matches the distribution of real data, even when that data distribution is too complicated to model directly.

Building on this idea, the INF method uses a **two-step architecture**, where two separate normalizing flows are trained to estimate interventional distributions from observational data. The first component of the INF architecture is the **nuisance flow**, which is responsible for estimating two critical quantities of the observational data that are necessary for adjusting for confounding:

- **The propensity score,** $P(A = a \mid X)$, which captures the probability that an individual with covariates X receives treatment a. This reflects how treatment assignment depends on individual characteristics. For example, how older or sicker patients may be more likely to receive a certain intervention.

- **The conditional outcome distribution,** $P(Y \mid X, A)$, which models how outcomes Y vary as a function of both covariates X and the treatment actually received A. This tells us how the observed outcomes are shaped by both the characteristics of the individuals and the treatment decisions made in the data.

Importantly, rather than learning these two quantities independently, the nuisance flow models them jointly through a *shared representation.* Specifically, it consists of a feedforward neural network that maps covariates X into a hidden representation $h = f_{\text{repr}}(X)$. This shared representation captures relevant features of the covariates that are useful for both treatment and outcome modeling. From this shared representation, the network branches into two outputs. One branch produces a single scalar logit, which is passed through a sigmoid function to estimate the propensity score, defined as $\pi(X) = P(A = 1 \mid X)$. The other branch uses the hidden representation h, combined with the actual treatment value A, as conditioning input to a Conditional Normalizing Flow (CNF). This CNF forms the normalizing flow component of the nuisance flow, and is responsible for modeling the conditional outcome distribution $P(Y \mid X, A)$. This shared setup allows the network to *jointly model* treatment assignment and outcomes, while still providing task-specific outputs. This joint structure also lays the groundwork for the doubly robust properties of the model (see Section 2.3).

To train the nuisance flow, a combined objective that includes both components is optimized: (1) A **binary cross-entropy loss** for the propensity score ($\mathcal{L}_{\text{propensity}}$). (2) A **maximum likelihood loss** for the conditional normalizing flow over outcomes ($\mathcal{L}_{\text{outcome}}$). These two losses are combined into a single training objective, weighted by a hyperparameter that is refered to as **Propensity Alpha**: $\mathcal{L}_{\text{nuisance}} = \mathcal{L}_{\text{outcome}} + PropensityAlpha \cdot \mathcal{L}_{\text{propensity}}$.

Together, the two outputs from the Nuisance Flow - the estimated propensity score and conditional outcome distribution - enable the INF model to capture the data-generating processes underlying both treatment assignment and outcomes. These estimates are then passed to the second component of the two-step architecture, the **target flow**. It uses this information to adjust for confounding and estimate the interventional distribution $P(Y[a] = y)$, effectively correcting for biases that arise due to non-random treatment assignment in observational data. Further implementation details are beyond the scope of this paper. Altogether, INFs provide a practical and theoretically grounded framework for estimating causal outcome distributions.

2.3 The Doubly Robust Property and the Need for Empirical Validation

A substantial theoretical claim made by Melnychuk et al. (2022) is that the INF method possesses the **property of double robustness**. This means that the estimate of the interventional density $P(Y[a] = y)$ remains consistent as long as **either** component estimated by the nuisance flow, namely the propensity score model $P(A|X)$ or the conditional outcome model $P(Y|X, A)$, adequately captures the corresponding relationship in the data. In other words, even if one of these estimated components is misspecified or imperfect, the system can still recover accurate estimates of the outcome distribution under intervention, provided the other component is well-specified.

To support this claim, the authors state in their paper that their approach extends the work of Kennedy et al. (2023), from which they derive a tractable optimization objective for efficient and doubly robust estimation.

However, while the original INF paper provides this theoretical foundation and demonstrates superior performance against other methods, it does not include a direct empirical stress-test of the doubly robust property itself. The experiments compare the final model to baselines but do not systematically analyze how the model behaves when one nuisance component is correctly specified while the other is deliberately misspecified (or when both are deliberately misspecified).

This is a critical gap, because the referenced work by Kennedy (2022; 2023) and Kennedy et al. (2023) is primarily theoretical and general in nature. It does not directly address normalizing flows or any specific deep learning methods like INFs, while relying on specific technical assumptions. Since this paper focuses on the role and reliability of the nuisance flow, the two research questions outlined earlier effectively put the doubly robust property to the test.

2.4 Key Hyperparameters in the Nuisance Flow

Two specific hyperparameters of the nuisance flow are examined in this study, which play an important role in controlling the model's ability to balance conditional outcome expressiveness and propensity estimation accuracy:

Nuisance Count Bins: This parameter controls the number of bins in the spline-based flow used to model the conditional outcome distribution within the CNF (see Section 2.2). The number of bins dictates the model's ability to capture the shape of the conditional outcome distribution. A lower number of bins restricts the model to smoother, simpler density shapes, which can lead to underfitting if the true distribution is complex. Conversely, a higher number of bins increases flexibility but also raises the risk of overfitting, where the model captures random noise in the training data, harming its generalizability.

Propensity Alpha: As indicated in Section 2.2, this parameter is a weighting coefficient within the nuisance flow's loss function, that controls the relative importance of the propensity score loss during training. A low value directs the model to prioritize fitting the conditional outcome distribution, potentially at the expense of the propensity score's accuracy. Thus, this hyperparameter provides a direct lever to create a trade-off between the accuracy of the two nuisance components. If the propensity score is poorly estimated (possibly due to a low value of Propensity Alpha) and the outcome model is also inaccurate (e.g., due to a misspecified number of nuisance bins), the bias-correction mechanism in the subsequent target flow will be based on two flawed inputs. This violates the conditions required for the doubly robust property.

2.5 Limitations of Prior Work and Motivation for This Study

Now that the general architecture of INFs has been outlined, it is clear that they represent a promising approach for estimating full interventional outcome distributions in causal machine learning. The significance of this method is further supported by the influence of the original paper by Melnychuk et al. (2022), which has been cited in several subsequent works that advance the field.

For example, INFs have inspired research into related tasks such as individualized outcome prediction, shifting the focus from population-level to person-specific distributions. The PO-Flow framework (Wu et al., 2025) builds directly on the foundations of INFs, extending flow-based generative models to estimate individualized potential outcomes. In that work, INFs are also used as a baseline for comparison. Additionally, the work by Vanderschueren (2024) in operational decision-making discusses how causal inference tools like INFs can support individualized decision-making in high-stakes environments.

However, despite the method's theoretical appeal and influence, several practical aspects of INFs remain underexplored - for example, the lack of empirical validation of the doubly robust property discussed in Section 2.3. In particular, it is not well understood how sensitive the method is to errors in its internal components. A key open question is how inaccuracies in the nuisance flow affect the final interventional distribution estimates produced by the target flow. A review of the existing literature found no studies directly examining this question. This is likely due to the novelty of the INF method, which leaves significant practical concerns about its robustness yet to be addressed.

This question is especially important because estimating the true propensity score and conditional outcome distribution from observational data is inherently difficult. As discussed earlier, these components are critical for adjusting for confounding. Yet in real-world settings, they are often biased or imprecise due to model misspecification, limited data, or noise. Taken together, these aspects raise valid concerns about the robustness of INFs - particularly their ability to accurately estimate the interventional distribution $(Y[a] = y)$ when the nuisance flow is imperfect. This paper investigates exactly that: **How well do INFs estimate interventional distributions in the presence of nuisance flow misspecification?**

3 Methodology

This section describes the methodology used to conduct the experiments in this study. The implementation used throughout all experiments is the official INF codebase provided by the authors (Valentyn, n.d.). Specifically, the configuration "main: +model=infs_aiptw" is used, which corresponds to the main version of the INF model explored in the original paper by Melnychuk et al. (2022). The first part of this section introduces the datasets used in the experiments, which is followed by a detailed explanation of the experimental procedure. Special care has been taken to ensure the reproducibility, transparency, and integrity of all research steps. For a detailed overview of these measures, including links to the code repositories associated with this study, please refer to the Responsible Research Section 7.

3.1 Datasets

Four datasets are used in this study: IHDP_NPCI_1, ACIC_2018_14, ACIC_2018_7, and Polynomial_Normal_CovShift3. To improve clarity and readability throughout the paper, these datasets will be referred to by the following descriptive nicknames, respectively: **GaussianClean**, **BimodalClean**, **SplitPeaks**, and **SyntheticComplex**. These datasets were selected for their diverse underlying structures in both the observational distributions $P(Y = y \mid A = a)$ and the known true interventional distributions $P(Y[a] = y)$, which the INF model aims to estimate accurately for $a \in \{0, 1\}$. This diversity enables a comprehensive evaluation of INF performance under varying degrees of confounding, modality, and distributional mismatch.

3.1.1 The GaussianClean Dataset

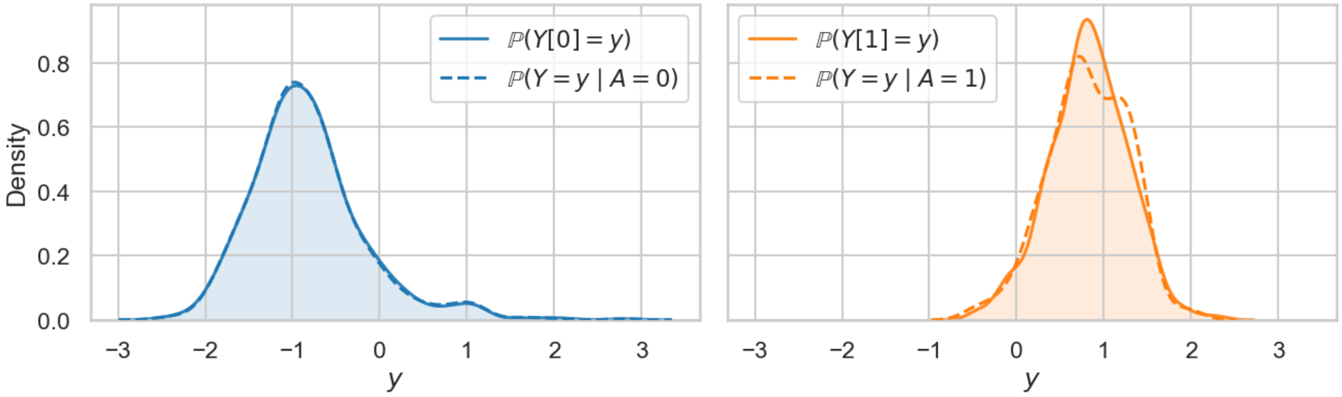

Figure 1: Visualization of the GaussianClean Dataset, with the ground truth $P(Y[a] = y)$ and $P(Y = y \mid A = a)$, for $a \in \{0, 1\}$. Simple shapes of $P(Y[a] = y)$. Their similarities to $P(Y = y \mid A = a)$ indicate low confounding.

The GaussianClean Dataset (IHDP_NPCI_1) dataset is derived from the Infant Health and Development Program (IHDP) and is widely used in causal inference benchmarks (AMLab-Amsterdam, n.d.; Hill, 2010). It is a semi-synthetic dataset, constructed by simulating treatment outcomes based on real-world covariates.

This dataset serves as a simple test case. The true interventional outcome distributions $P(Y[a] = y)$ have relatively smooth, unimodal shapes that are approximately Gaussian. Moreover, the observational outcome distributions $P(Y = y \mid A = a)$ are visually similar to their corresponding interventional counterparts, indicating limited confounding. These traits make the dataset useful for evaluating INF behavior under relatively simple conditions.

3.1.2 The BimodalClean Dataset

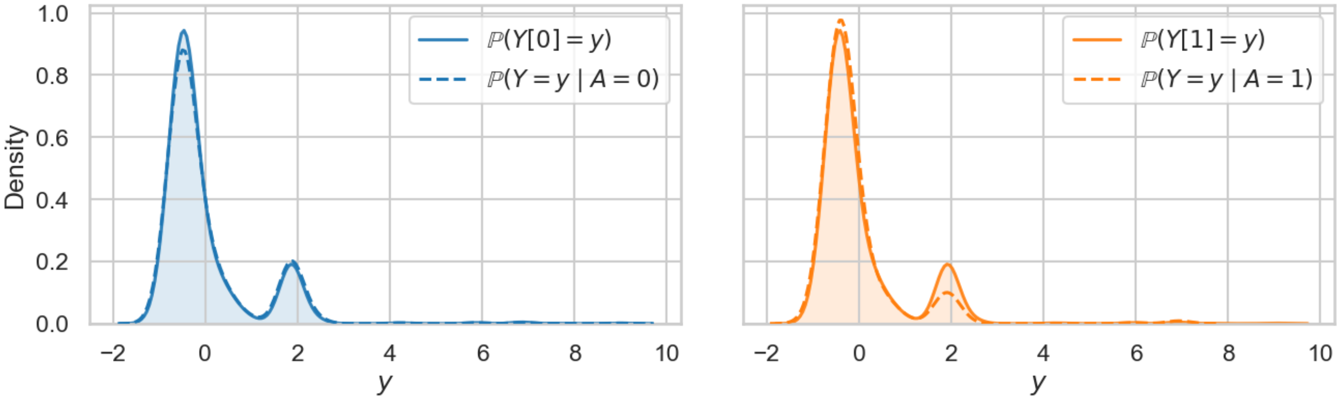

Figure 2: Visualization of the BimodalClean Dataset, with the ground truth $P(Y[a] = y)$ and $P(Y = y \mid A = a)$, for $a \in \{0, 1\}$. More complex shapes of $P(Y[a] = y)$. Their similarities to $P(Y = y \mid A = a)$ indicate low confounding.

The BimodalClean Dataset (ACIC_2018_14) is part of the 2018 Atlantic Causal Inference Conference (ACIC) data challenge (Sage Bionetworks, info@sagebase.org, n.d.). It is a semi-synthetic dataset that simulates outcomes using real covariates and designed structural causal models.

This dataset introduces more complexity. The interventional distributions $P(Y[a] = y)$ are bimodal, reflecting multiple modes in the outcome space. Similar to IHDP, the observational distributions $P(Y = y \mid A = a)$ still visually resemble the interventional ones. However, due to the added multimodality, this dataset goes beyond the simplicity of Gaussian-like shapes and presents a more challenging learning scenario.

3.1.3 The SplitPeaks Dataset

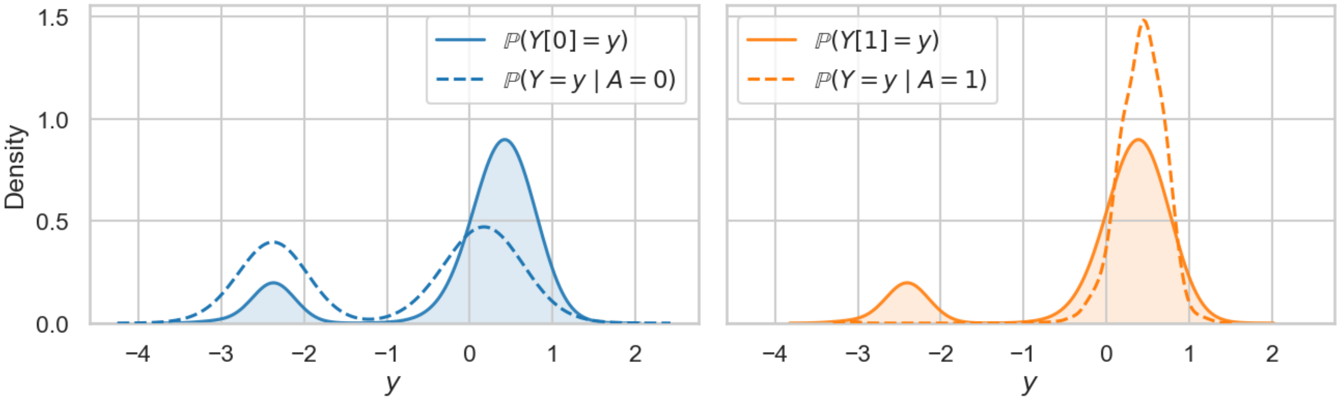

Figure 3: Visualization of the SplitPeaks Dataset, with the ground truth $P(Y[a] = y)$ and $P(Y = y \mid A = a)$, for $a \in \{0, 1\}$. More complex shapes of $P(Y[a] = y)$. Their differences to $P(Y = y \mid A = a)$ indicate higher confounding.

Also from the ACIC 2018 challenge, the SplitPeaks Dataset (ACIC_2018_7) introduces yet another layer of complexity. Like the BimodalClean, the interventional distributions of SplitPeaks are bimodal. However, here the observational distributions differ substantially from the interventional ones.

While some visual resemblance remains between the two types of distributions - such as peaks appearing in roughly the same regions of the outcome space - there are very clear differences present. Namely, the peaks between the $P(Y[a] = y)$ and $P(Y = y \mid A = a)$ distributions differ noticeably in height. Also, in the treatment group $a = 1$, one of the peaks seen in the interventional distribution is missing entirely from the observational distribution. These traits reflect a case of stronger confounding, making the dataset more challenging for causal models and a good test case for evaluating robustness under more biased data conditions.

3.1.4 The SyntheticComplex Dataset

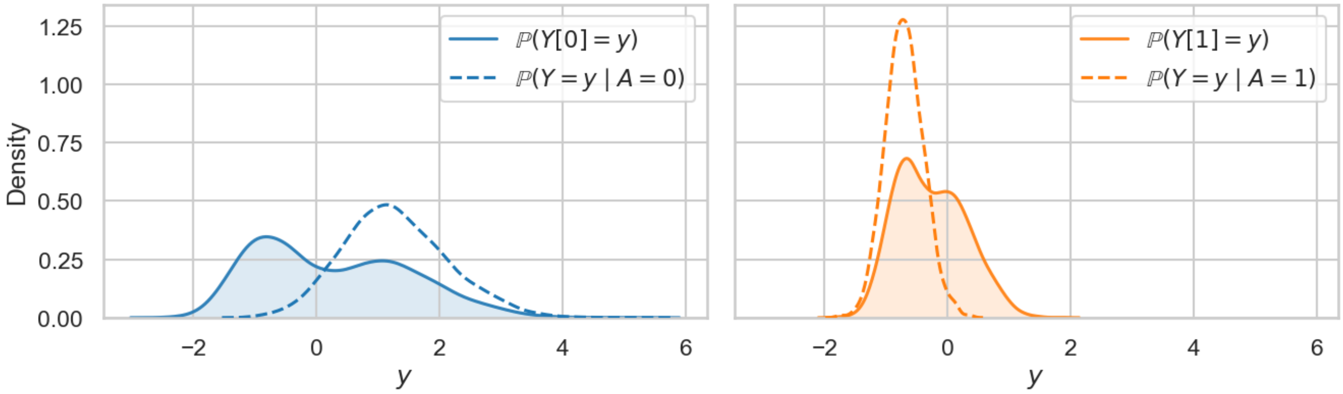

Figure 4: Visualization of the SyntheticComplex Dataset, with the ground truth $P(Y[a]=y)$ and $P(Y=y \mid A=a)$, for $a \in \{0,1\}$. Complex shapes of $P(Y[a]=y)$. Their stark differences to $P(Y=y \mid A=a)$ indicate high confounding.

The SyntheticComplex Dataset is synthetically generated using a structural causal model (SCM) implemented in the public INF codebase (Valentyn, n.d.). This dataset represents a different type of scenario. The true interventional distributions are bimodal, but with "peaks connected by high density," creating a continuous yet nontrivial landscape.

What makes this dataset particularly interesting is the significant difference between the observational outcome distributions $P(Y=y \mid A=a)$ and the true interventional counterparts $P(Y[a]=y)$. This setup stresses the INF model's ability to handle complex causal structures in the presence of dense regions.

3.2 Experimental Baselines

To ensure a fair and consistent evaluation of INF performance, this study leverages INF hyperparameter configurations identified as optimal by the authors of the method, for each of the four benchmark datasets. These configurations are documented in their official codebase (Valentyn, n.d.), and were selected based on extensive experimentation and training performance. The complete configurations are provided in Appendix A. Table 1 presents the values of the two key hyperparameters of importance to this study: nuisance count bins and propensity alpha (detailed in Section 2.4). These values are relevant for interpreting the subsequent methodology and results.

Table 1: Values of the *nuisance count bins* and *propensity alpha* hyperparameters from the optimal INF hyperparameters configuration for each dataset.

Hyperparameter	**GaussianClean**	**BimodalClean**	**SplitPeaks**	**SyntheticComplex**
Nuisance Count Bins	20	20	10	10
Propensity Alpha	1.0	1.0	1.0	1.0

In this study, the estimates produced by the optimal configurations are treated as the *"reference standard"* against which all experimental modifications are evaluated. [1] Therefore, the first step of the experimental procedure is to run the INF model on each dataset using the corresponding optimal configuration. This produces two baseline interventional density estimates, $P(Y[0]=y)$ and $P(Y[1]=y)$, which serve as reference distributions. In the following experiments, we compare the resulting interventional distributions from altered conditions to these baselines in order to quantify any deviation or degradation in model performance.

One might ask why these baseline estimates are used for comparison instead of evaluating directly against the known ground-truth distributions. The reason is twofold. First, as observed in extensive experimentation with the INF method, and as explicitly discussed in Appendix K: Qualitative Insights of Melnychuk et al. (2022), the model often struggles to accurately capture certain regions of the true outcome distributions, even when trained under optimal conditions. Second, this limitation means that direct comparison to the ground truth could overstate the negative effects of any perturbation, even when the model is performing within its typical behavior range.

Using the baseline curves as reference points allows us to more accurately isolate the impact of deliberate modifications in the experimental setup. In other words, by comparing experimental outputs to the best possible INF estimate under ideal conditions, rather than to the true distribution, we ensure a fairer evaluation of the model's robustness in practice.

[1] To ensure academic rigor, additional hyperparameter configurations were also explored during the course of this research. None of these alternatives yielded results that clearly surpassed the author-provided configurations, reinforcing their use as strong empirical baselines in our analysis.

3.3 Research Question 1: Sensitivity to Broken Nuisance Flow Components

The experiments for the first research question investigate how sensitive the INF's estimates of $P(Y[a] = y)$ are when the nuisance flow is deliberately made entirely incorrect / "broken". As discussed in Section 2, the nuisance flow is responsible for estimating two components: the propensity score and the conditional outcome model. Therefore, when we refer to breaking the nuisance flow, we consider three distinct scenarios: (1) only the propensity score estimate is broken (i.e., completely misspecified), (2) only the conditional outcome model is broken, and (3) both components are broken simultaneously. The specific methods used to break each component in isolation are as follows:

To break the **conditional outcome model**, the learned, covariate-dependent output is replaced with a fixed Normal distribution with mean 50.0 and standard deviation 2.0, completely ignoring any relationship between covariates and outcomes.[2]

To break the **propensity score model**, the learned probabilities are replaced with fixed values: 0.1 for both treatment groups, independent of the covariates. This removes any covariate-dependent treatment assignment.[2]

The INF model is trained under all three scenarios for each dataset, producing interventional density estimates in each case. To evaluate the impact of these changes, we compare the resulting estimates to the baseline by plotting the baseline estimates alongside those from the three broken scenarios.

In all cases, the INF model is trained using the same optimal hyperparameter configuration specific to the dataset. The only difference is that the output(s) of the nuisance flow are deliberately altered to be entirely uninformative before being passed to the target flow.

3.4 Research Question 2: Sensitivity to Minor Nuisance Flow Misspecifications

This research question shifts focus from extreme misspecification to more subtle modeling inaccuracies. Specifically, it investigates how minor imperfections in the nuisance flow affect the quality of the estimated interventional densities $P(Y[a] = y)$. Two distinct strategies are used to introduce such inaccuracies.

3.4.1 Hyperparameter Perturbations

The first strategy investigates the sensitivity of the INF model to changes in two key hyperparameters of the nuisance flow: **nuisance count bins** and **propensity alpha** (see Section 2.4). For brevity, we refer to these as **NBins** and **PAlpha**, respectively, in the remainder of the paper. These two parameters affect the accuracy of estimating the conditional outcome model (through NBins) and the strength of the propensity score component in the bias correction mechanism (through PAlpha). To assess their effect on model performance, each parameter is varied individually and in combination, relative to the values used in the optimal configuration (listed in Table 1). To ensure a thorough research, six perturbation scenarios are conducted: (1) **Decrease NBins only**: to examine the model's performance when relying primarily on propensity score estimation, with a less expressive conditional outcome model; (2) **Decrease PAlpha only**: to evaluate what happens when the influence of the propensity score in the bias correction is minimized; (3) **Decrease NBins with PAlpha set to zero**: to test model behavior when the conditional outcome model is underparameterized and the propensity-based bias correction is entirely suppressed; (4) **Increase PAlpha only**: to study the effects of over-reliance on the propensity score in the bias correction term; (5) **Increase NBins with PAlpha set to zero**: to isolate the effect of a highly expressive outcome model without any influence from propensity-based bias correction; and (6) **Increase NBins with PAlpha set to a large value**: to evaluate the combined effect of a more flexible outcome model and strong propensity-based correction.

The INF model is trained under each of these six scenarios for every dataset. For each case, a range of values for NBins and PAlpha are used to evaluate their individual and combined effects. All other hyperparameters remain fixed at their optimal values. The resulting interventional density estimates $P(Y[a] = y)$ are compared to the baseline to evaluate any degradation in performance.

3.4.2 Controlled Noise Injection

The second strategy introduces noise into the outputs of the nuisance flow, specifically to the estimated propensity score $P(A \mid X)$ and conditional outcome distributions $P(Y \mid X, A)$. In this setup, both estimates are first computed as usual by the nuisance flow, but noise is then introduced after estimation and before being passed to the target flow, ensuring that the target flow operates on the noise-perturbed versions.

For the **propensity score**, noise is injected into the raw logits used to calculate the original estimate. This is zero-mean Gaussian noise with a specified standard deviation is added directly to these logits. This perturbation is applied independently for each individual sample. After adding the noise, the modified logits are passed through the sigmoid function - just as in the "regular" estimation process - to map them back to a valid probability in the [0, 1] range.

For the **conditional outcome model**, in accordance with how this is implemented in the codebase (Valentyn, n.d.), noise is added to the nuisance flow's predictions for how likely different outcomes are under $P(Y \mid X, A)$. Specifically, the noise is applied to the values that represent the log-probability of each possible outcome. As before, it is zero-mean Gaussian with a specified standard deviation. This noise is applied independently at each outcome point,

[2] All 4 datasets used in these experiments are confirmed to have conditional outcome and propensity score distributions that differ significantly from the broken replacements.

introducing localized distortions to the predicted distribution. After the perturbation, the distribution is normalized to ensure it remains a valid probability distribution over a continuous outcome.

For each dataset, the INF model is trained under three scenarios: (1) noise added only to the propensity score, (2) noise added only to the conditional outcome model, and (3) noise added to both simultaneously. To study sensitivity across different perturbation magnitudes, three standard deviation levels are tested: "low" (0.2), "medium" (0.5), and "high" (1.0). These values are carefully selected based on the implementation and preliminary experiments. As in previous setups, the resulting interventional density estimates $P(Y[a] = y)$ are compared to the baseline to assess potential degradation in model performance.

3.4.3 Evaluation Metric: L1 Error for Interventional Density Comparison

Given the comprehensive investigation in Research Question 2, visualizing all comparisons as in Research Question 1 would result in excessive plotting. To ensure clarity, a quantitative metric is used instead: the **L1 error**. It is defined as the integral of the absolute difference between the baseline and experimentally altered density estimates: L1 Error $= \int |\hat{p}_{\text{exp}}(Y[a] = y) - \hat{p}_{\text{baseline}}(Y[a] = y)|\, dy$.

A smaller L1 error indicates that the estimated distribution under the experimental setting remains closer to the baseline, and thus, less affected by the modification. Since INF produces two interventional estimates (one for $a = 0$ and one for $a = 1$) the final reported value is the average of the L1 errors computed for both treatment groups.

Individual plots are still produced for each case in Research Question 2 (as was done for RQ1), and are included in the Appendix and referenced when relevant.

4 Results

This section presents the experimental results, organised by research question. Section 4.1 addresses sensitivity to complete nuisance flow misspecification, while Section 4.2 examines the impact of minor inaccuracies.

4.1 Results of Sensitivity to Completely Broken Nuisance Flow Elements (RQ.1)

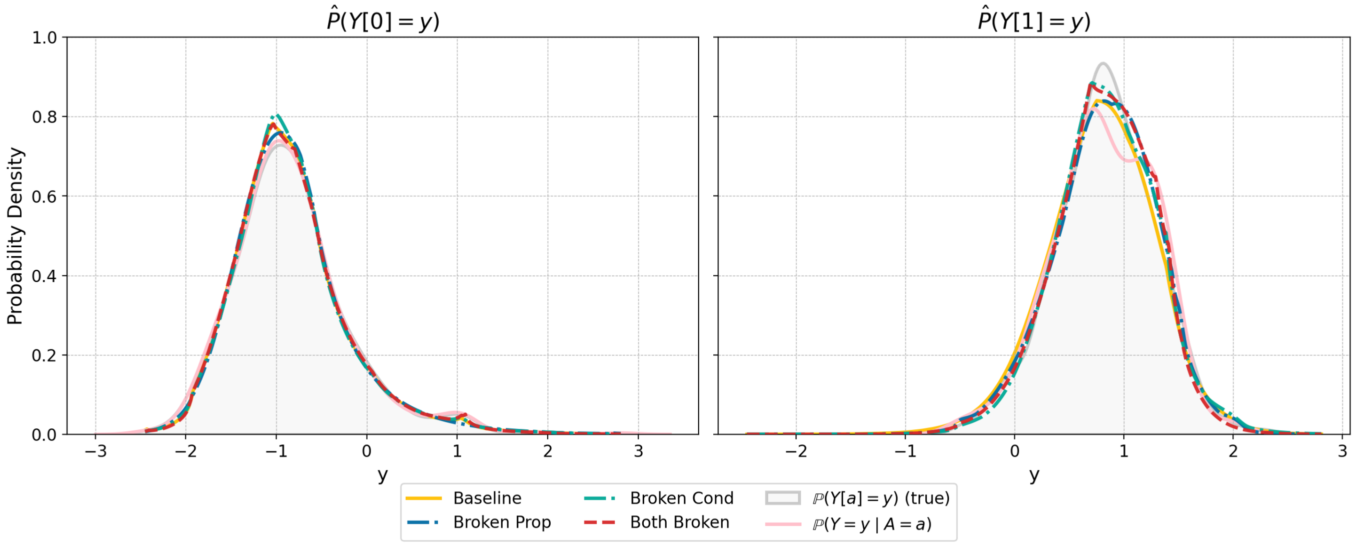

Figure 5: Estimated interventional densities $\hat{P}(Y[a] = y)$ for $a \in \{0, 1\}$ on the **GaussianClean** dataset. Shown are estimates under four configurations: baseline (no misspecification), broken propensity score only, broken conditional outcome model only, and both nuisance components broken.

First, the **GaussianClean** dataset. As shown in Figure 5, the baseline INF estimates do a good job at recovering the true interventional distributions for both treatment arms. The curves are well-aligned with the ground truth histograms and show minimal deviation. A minor exception occurs for $a = 1$, where the baseline and all other estimated curves slightly underrepresent the sharp peak present in the ground truth distribution.

Interestingly, even under extreme misspecification scenarios, where the propensity score model, the conditional outcome model, or even both components of the nuisance flow are deliberately broken, the resulting interventional estimates remain close to the baselines. For both $a = 0$ and $a = 1$, all variants track the corresponding true distribution (and therefore also $P(Y = y \mid A = a)$) closely, with only minor visual deviations.

Next, for the **BimodalClean** dataset, the results mirror those from the GaussianClean dataset (for additional context, see Appendix B.1 for the plot of the estimates). The baseline INF estimates accurately recover the underlying interventional distributions for both treatment arms. Furthermore, despite the different shape of the interventional distributions compared to the GaussianClean dataset, the estimates remain robust; even under extreme misspecification scenarios, they stay close to the baselines.

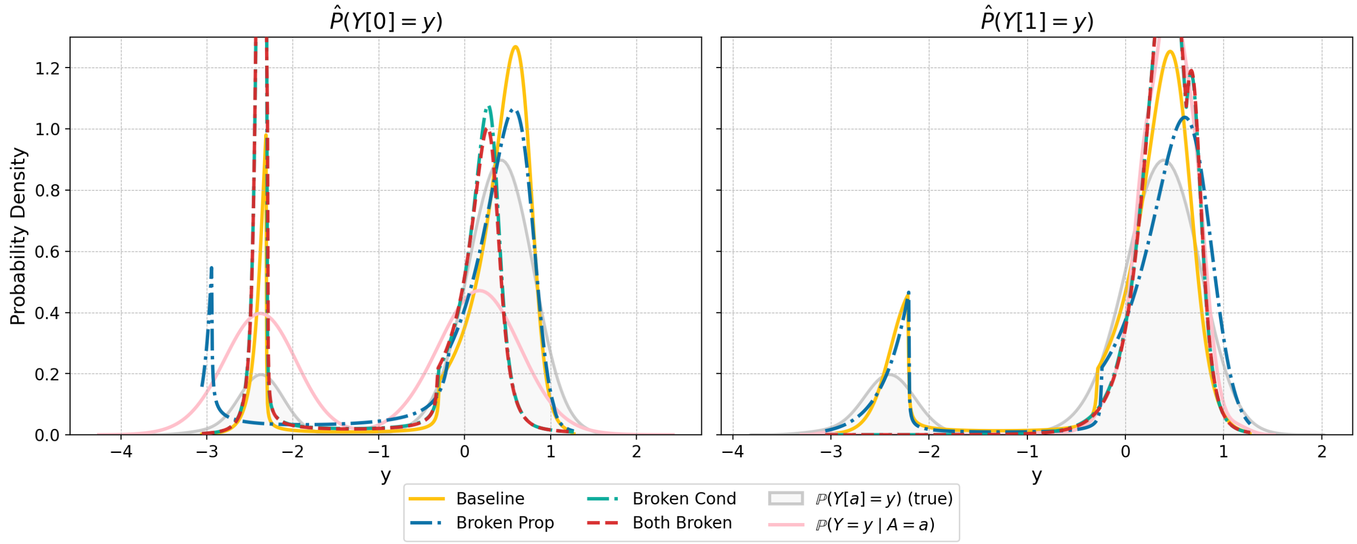

Figure 6: Estimated interventional densities $\hat{P}(Y[a]=y)$ for $a \in \{0,1\}$ on the SplitPeaks dataset. Shown are estimates under the four configurations.

Moving on, Figure 6 depicts the results for the **SplitPeaks** dataset, where we observe much greater variation among the estimated curves. Starting with the baseline INF estimates, the curve for $a=0$ captures the overall shape of the true $P(Y[0]=y)$ distribution but exhibits two notable issues: the left sharp peak overshoots the ground truth density a bit, while the right peak is shifted slightly to the right. For $a=1$, the right peak aligns reasonably well with the true $P(Y[1]=y)$ density, but the left peak undershoots the ground truth in that region, and leans more towards the right than it should.

When **only the propensity score is broken**, clear deviations already emerge. For $a=0$, the entire left side of the curve shifts further left, causing a high density estimate in a region where the true density is very low. For $a=1$, the right peak becomes short and leans noticeably more to the right, failing to capture the underlying “hill” present in the ground truth distribution. This behavior contrasts with the baseline curve for $a=1$, which managed to model this area more accurately. The degradation becomes much more severe when **only the conditional outcome model is broken**. For $a=0$, the left peak overshoots dramatically, far more than in the baseline case. The right peak becomes unnaturally narrow and shifts leftward, misrepresenting an area that should be broader and more dispersed. For $a=1$, the region where the left peak should appear becomes almost completely flat, indicating an incorrectly low density in that area. Meanwhile, the right peak significantly overshoots the true density.

When **both nuisance components are broken**, the results closely resemble those from the broken conditional model. Focussing on just the “both broken” scenario, notably, the estimates appear to drift toward the **observational** distributions $P(Y=y \mid A=a)$, rather than remaining close to the baseline / ground truth histogram. This is especially visible for $a=1$, where the red curve (both broken) closely overlaps the pink observational curve. A similar, though subtler, effect is seen for $a=0$: the peaks of the “both broken” curve shift in the direction of the corresponding peaks of the observational distribution, rather than staying aligned with the baseline estimate or the true interventional distribution.

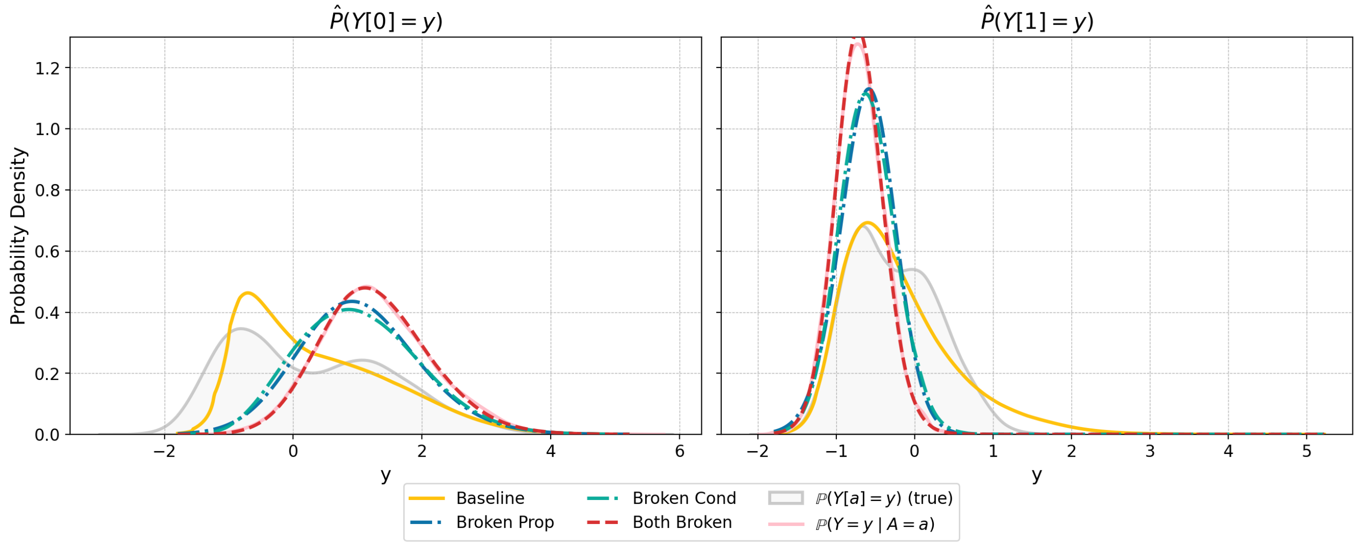

Figure 7: Estimated interventional densities $\hat{P}(Y[a]=y)$ for $a \in \{0,1\}$ on the **SyntheticComplex** dataset. Shown are estimates under the four configurations.

Figure 7 presents the results for the **SyntheticComplex** dataset. Overall, the baseline INF estimates capture the general shape of the true interventional distributions $P(Y[a] = y)$, though not perfectly. For $a = 0$, the baseline underestimates the density on the left side of the distribution. For $a = 1$, a similar underestimation occurs on the right side. Nonetheless, the overall structure of the curves remains reasonably aligned with the ground truth.

What stands out in this dataset is the consistent behavior across both treatment arms when **only the propensity score** or **only the conditional outcome model** is broken. In both cases, the resulting interventional estimates visibly shift away from the ground truth distributions and instead drift toward the observational distributions $P(Y = y \mid A = a)$. So breaking either one of the two components already results in estimates that deviate substantially from both the true interventional distributions and the baseline estimates.

This effect becomes even more pronounced when both nuisance components are broken. In this scenario, the resulting estimated densities almost perfectly overlap with the observational $P(Y = y \mid A = a)$ distributions. The alignment is so close that the red and pink curves are nearly indistinguishable.

4.2 Results of Sensitivity to Minor Nuisance Flow Misspecifications (RQ.2)

As previously noted, all results in this section are reported in terms of L1 error. The values are summarized in tables below, where higher errors are visually emphasized using darker shading.

Table 2: Average L1 error on the GaussianClean dataset under hyperparameter perturbations. The baseline configuration uses NBins=20 and PAlpha=1.0. Refer to Appendix C.1 for visualizations of the resulting estimates.

Perturbation	**L1 Error**
NBins=2 (, PAlpha = 1.0)	0.0338
NBins=5 (, PAlpha = 1.0)	0.0414
NBins=10 (, PAlpha = 1.0)	0.0384
NBins=15 (, PAlpha = 1.0)	0.0444
NBins=30, PAlpha=50.0	0.0326
NBins=50, PAlpha=50.0	0.0423
NBins=100, PAlpha=50.0	0.0473

Perturbation	**L1 Error**
PAlpha=0.00	0.0422
PAlpha=0.25	0.0347
PAlpha=0.50	0.0380
PAlpha=0.75	0.0385
PAlpha=1.50	0.0370
PAlpha=3.00	0.0397
PAlpha=50.0	0.0417

Perturbation	**L1 Error**
NBins=2, PAlpha=0.0	0.1389
NBins=5, PAlpha=0.0	0.0574
NBins=10, PAlpha=0.0	0.0521
NBins=15, PAlpha=0.0	0.0373
NBins=30, PAlpha=0.0	0.0624
NBins=50, PAlpha=0.0	0.1290
NBins=100, PAlpha=0.0	0.1565

Starting with the **GaussianClean** dataset, Table 2 reveals several noteworthy trends. First, varying the PAlpha parameter across a wide range, from 0 to 50, does not significantly degrade the quality of the resulting estimates. This is evidenced by consistently low L1 errors. Similarly, decreasing only NBins also yields comparably low errors.

However, a different pattern emerges when NBins is decreased and PAlpha is set to 0. Under these conditions, the L1 error increases substantially, and the error grows as NBins decreases. In particular, when NBins is set to 2, the error is highest. For further insight, the corresponding figure in Appendix C.2 shows that the shape of the estimate differs very noticeably from the others, especially on the left end when $a = 1$. This reflects a clear degradation in estimation quality.

Moreover, increasing NBins with PAlpha still fixed at 0 also leads to a sharp rise in error. As shown in Appendix C.5, higher bin counts result in estimates that appear clearly more erratic.

Interestingly, when PAlpha is set to a high value, such as 50, and NBins is also increased, the L1 error remains relatively low. It is only when NBins is set to an extreme value like 100 that a noticeable increase in error occurs. This rise seems to stem from a very mild presence of erratic behavior, as visualized in Appendix C.6. Still, the estimates remain considerably better than when PAlpha is set to 0.

The **BimodalClean** dataset exhibits similar trends when the hyperparameters are perturbed. However, one notable additional observation is that when NBins is reduced and PAlpha is set to 0, the smaller peak in the bimodal distribution fails to be captured accurately (for both $a = 0$ and $a = 1$), beginning as early as NBins = 10. When NBins is reduced further to 2, the resulting estimate is significantly more degraded, with higher error compared to the NBins = 2, PAlpha = 0 setting in the GaussianClean dataset. For additional context, refer to the corresponding plots and L1 error table in Appendix D.1.

Table 3: Average L1 error on the GaussianClean dataset under noise injection. Refer to Appendix C.2 for visualizations of the resulting estimates.

Noise Type (StdDev)	**L1 Error**
Prop. Only (0.2)	0.0339
Prop. Only (0.5)	0.0386
Prop. Only (1.0)	0.0350

Noise Type (StdDev)	**L1 Error**
Cond. Only (0.2)	0.0343
Cond. Only (0.5)	0.0305
Cond. Only (1.0)	0.0321

Noise Type (StdDev)	**L1 Error**
Both (0.2)	0.0366
Both (0.5)	0.0461
Both (1.0)	0.0449

Examining the estimates resulting from injecting noise into the **GaussianClean** dataset, we observe that applying either Prop. Only or Cond. Only noise does not lead to significant errors, even at the highest standard deviation. When both types of noise are applied, the error is slightly higher than with either type alone, but still not excessively large. These results suggest that introducing noise with the specified standard deviations given this dataset does not significantly degrade the quality of the estimates, even when both noise components are set to high values.

Injecting noise into the **BimodalClean** dataset results in similar observations, where increasing the noise does not lead to significant increases in error. For further context, refer to Appendix D.2, where the estimate plots and error values demonstrate that the overall structure of the estimates remains largely unchanged, even under higher noise levels.

Table 4: Average L1 error on the SplitPeaks dataset under hyperparameter perturbations. The baseline configuration uses NBins=10 and PAlpha=1.0. Refer to Appendix E.1 for visualizations of the resulting estimates.

Perturbation	**L1 Error**
NBins=2 (, PAlpha = 1.0)	0.9134
NBins=5 (, PAlpha = 1.0)	0.2846
NBins=8 (, PAlpha = 1.0)	0.1913
NBins=15, PAlpha=50.0	0.2273
NBins=25, PAlpha=50.0	0.2476
NBins=100, PAlpha=50.0	0.2866

Perturbation	**L1 Error**
PAlpha=0.00	0.2349
PAlpha=0.25	0.1461
PAlpha=0.50	0.0850
PAlpha=1.50	0.0960
PAlpha=3.00	0.1097
PAlpha=50.0	0.1816

Perturbation	**L1 Error**
NBins=2, PAlpha=0.0	0.9152
NBins=5, PAlpha=0.0	0.8433
NBins=8, PAlpha=0.0	0.2390
NBins=15, PAlpha=0.0	0.2135
NBins=25, PAlpha=0.0	0.2586
NBins=100, PAlpha=0.0	0.3168

Moving on to hyperparameter perturbation for the SplitPeaks dataset, Table 4 reveals some similarities to previous datasets, but also several distinct differences. Adjusting only the propensity score generally results in relatively low errors. However, there is a noticeable increase in error when PAlpha is set to the extreme values of 0 and 50.

Unlike in previous datasets, decreasing NBins alone leads to a significant rise in error, even when PAlpha is set to the "ideal" value of 1.0. The error becomes remarkably high when NBins = 2. When PAlpha is set to 0, the error for NBins = 2 remains similarly high, but it increases substantially for NBins = 5 and NBins = 8. For additional context, Appendix Figures E.2 and E.3 illustrate how misspecified the estimate curves become when NBins is low - especially highlighting the drastic deviation when NBins = 2.

Furthermore, we observe that increasing NBins also leads to high error. Interestingly, this occurs not only when PAlpha = 0, but also when PAlpha = 50. This rise in error correlates with the increasing erratic behavior of the estimates as NBins grows (see Appendix Figures E.5 and E.6 for visual examples).

Table 5: Average L1 error on the SplitPeaks dataset under noise injection. Refer to Appendix E.2 for visualizations of the resulting estimates.

Noise Type (StdDev)	**L1 Error**
Prop. Only (0.2)	0.0731
Prop. Only (0.5)	0.0853
Prop. Only (1.0)	0.4056

Noise Type (StdDev)	**L1 Error**
Cond. Only (0.2)	0.0525
Cond. Only (0.5)	0.3328
Cond. Only (1.0)	0.6515

Noise Type (StdDev)	**L1 Error**
Both (0.2)	0.1674
Both (0.5)	0.3785
Both (1.0)	0.6425

Table 5 shows how the **SplitPeaks** dataset responds to different types and magnitudes of noise injection. For Prop. Only noise, both low and medium levels result in relatively low error, indicating that the estimates remain similar to the baseline. However, injecting high For Prop. Only noise leads to a substantial increase in error.

In contrast, for Cond. Only noise, even a medium noise level causes a sharp increase in error - much higher than the corresponding medium Prop. Only noise. The trend continues at the high noise level, where Cond. Only noise also results in a higher error. These results show that, for this dataset, the INF model is more sensitive to Conditional noise than to Propensity noise, particularly at medium and high magnitudes.

When both noise types are injected, even the low noise configuration results in a noticeably higher error than either type alone. Medium and high levels of combined noise cause significant further degradation. Interestingly, though, the errors at these two levels are not much higher than those caused by Cond. Only noise at the same levels. At these levels, the addition of Prop. Only noise on top of Cond. Only Noise does not seem to result in more error.

Interestingly, however, the errors at these levels are not much different than those caused by Cond. Only noise at the same magnitudes. Therefore, for this dataset, it is observed that at medium and high noise levels, adding Propensity noise on top of Conditional noise does not substantially increase the overall error.

Table 6: Average L1 error on the SyntheticComplex dataset under hyperparameter perturbations. The baseline configuration uses NBins=10 and PAlpha=1.0. Refer to Appendix F.1 for visualizations of the resulting estimates.

Perturbation	**L1 Error**
NBins=2 (, PAlpha = 1.0)	0.0509
NBins=5 (, PAlpha = 1.0)	0.0545
NBins=8 (, PAlpha = 1.0)	0.0648
NBins=15, PAlpha=50.0	0.0407
NBins=25, PAlpha=50.0	0.0452
NBins=100, PAlpha=50.0	0.0753

Perturbation	**L1 Error**
PAlpha=0.00	0.0468
PAlpha=0.25	0.0230
PAlpha=0.50	0.0197
PAlpha=1.50	0.0236
PAlpha=3.00	0.0258
PAlpha=50.00	0.0289

Perturbation	**L1 Error**
NBins=2, PAlpha=0.0	0.3012
NBins=5, PAlpha=0.0	0.0590
NBins=8, PAlpha=0.0	0.0687
NBins=15, PAlpha=0.0	0.0528
NBins=25, PAlpha=0.0	0.2338
NBins=100, PAlpha=0.0	0.3266

Based on the results presented in Table 6, hyperparameter perturbations on the **SyntheticComplex** dataset exhibit overarching similarities with the GaussianClean and BimodalClean datasets. Increasing or decreasing PAlpha across a wide range does not significantly degrade the quality of the resulting estimates. Similarly, decreasing NBins alone results in comparably low errors.

As with the previous datasets, setting NBins = 2 while fixing PAlpha = 0 leads to a substantial increase in error, and the resulting estimate deviates considerably in shape from the baseline. However, the error does not consistently increase as NBins decreases. For example, when NBins = 5 or 8, the error remains relatively low, and the estimates closely resemble the baseline (see Appendix F.3 for visualizations).

Continuing, increasing NBins while keeping PAlpha = 0 results in low error at NBins = 15, but leads to a significant rise when increased to 25, and even more so at 100. These higher bin counts produce estimates that become increasingly erratic (refer to Appendix F.5 for additional context).

Finally, as seen in the GaussianClean and BimodalClean datasets, when PAlpha is set to 50 and NBins is increased, the L1 error remains relatively low. It is only at NBins = 100 that a slight but noticeable increase in error occurs, which appears to result from mild inaccuracies in the estimates, as visualized in Appendix F.6.

Table 7: Average L1 error on the SyntheticComplex dataset under noise injection. Refer to Appendix F.2 for visualizations of the resulting estimates.

Noise Type (StdDev)	L1 Error	Noise Type (StdDev)	L1 Error	Noise Type (StdDev)	L1 Error
Prop. Only (0.2)	0.0474	Cond. Only (0.2)	0.0395	Both (0.2)	0.0527
Prop. Only (0.5)	0.3022	Cond. Only (0.5)	0.3169	Both (0.5)	0.4181
Prop. Only (1.0)	0.6009	Cond. Only (1.0)	0.6313	Both (1.0)	0.8235

Looking at the results in Table 7, we observe that for the SyntheticComplex dataset, the general trend is that increasing the amount of injected noise leads to higher errors. Starting with Prop. Only noise, there is a clear and substantial increase in error as the standard deviation increases. The same pattern holds for Cond. Only noise.

For Both noise, the error is consistently higher than for either individual noise type across all three levels. This indicates that the combined noise affects the estimates more severely than each noise component alone.

5 Discussion

This section interprets the key findings from the experiments. First, the results are analyzed to understand their implications for the robustness of INFs. The discussion begins with the first research question, followed by both parts of the second research question. Throughout, the role of the doubly robust property is also examined, including cases where it appears to break down. Next, the section reflects on how the experimental setup may have influenced the outcomes and discusses the main limitations of the approach.

5.1 Impact of Severe Misspecification

The **GaussianClean** and **BimodalClean** datasets are characterized by low confounding, as indicated by the strong similarity between their observational and interventional distributions. For these datasets, even when one or both components of the nuisance flow were deliberately broken, the resulting interventional estimates remained close to the baseline. This outcome demonstrates strong robustness of INFs under low-confounding conditions - regardless of whether the true interventional distributions are simple in shape, as in GaussianClean, or more complex, as in BimodalClean.

In contrast, the **SplitPeaks** dataset reveals a different picture. Due to higher confounding, the INF model seems to have become much more sensitive to broken nuisance components. Breaking either component already leads to noticeable deviations from the baseline. However, breaking the conditional outcome model has a visibly greater negative impact than breaking the propensity score - the latter still produces estimates more similar to the baseline. Interestingly, the effect of breaking only the conditional outcome model is comparable to breaking both components simultaneously. This suggests that, in certain settings, the conditional outcome model plays a more dominant role in estimating the interventional distributions.

Next, the **SyntheticComplex** dataset, which exhibits strong confounding and a complex shape of interventional distributions, presents a different case: breaking even a single nuisance flow component already causes the model‘s estimates to drift significantly away from the baseline.

Based on the results for SplitPeaks and SyntheticComplex, severe misspecification of even a single nuisance component can substantially degrade the quality of interventional density estimates. This suggests that **the doubly robust property of INFs may not fully hold under extreme misspecification in highly confounded settings**.

A common trend across all four datasets is that, when both components of the nuisance flow are broken, the resulting estimates begin to resemble the confounded observational distributions. This effect is particularly clear in GaussianClean, BimodalClean, and SyntheticComplex, and it is also observable in SplitPeaks. These results suggest that, in the complete absence of valid nuisance information - under the specific breaking methods used in this study - the confounding patterns and biases present in the observational data seem to emerge in the estimated interventional distributions. The INF model loses its ability to recover causal quantities - as expected under the doubly robust property - and instead begins to reflect patterns that are more associative than causal.

Lastly, even when both nuisance components are broken, the resulting estimates do not behave erratically, in the sense that they are not drastically different from the baselines or the observational distributions. This pattern holds across all four datasets, regardless of the underlying interventional distribution complexity or confounding level. In other works, the INF model does not produce highly unreasonable outputs when the nuisance flow components are broken, at least under the specific breaking methods used in this study.

5.2 Impact of Hyperparameter Perturbations

A consistent trend across all four datasets is that **setting *only* PAlpha to extreme values** - either very low (0.0) or very high (50.0) - does not significantly increase error. When PAlpha is low, the loss term for the propensity score is effectively removed, but the conditional outcome model remains intact and can still be estimated accurately since NBins is unchanged. When PAlpha is high, both nuisance terms are included. Due to the doubly robust property, having at least one well-specified nuisance component is sufficient for accurately estimating the interventional density, which explains why the results remain close to the baseline.

Next, when **decreasing *only* NBins** while keeping PAlpha at its baseline value of 1.0, the error remains low for the GaussianClean, BimodalClean, and SyntheticComplex datasets. This can be attributed to the doubly robust property: although a low NBins impairs the conditional outcome model, the propensity score is still accurately estimated. However, this trend does not hold for the SplitPeaks dataset. Reducing only NBins leads to a significant increase in error, supporting the earlier observation that, for this dataset, the conditional outcome model plays a more dominant role than the propensity score in producing accurate estimates.

Now, when **NBins is set to very low values** (e.g., 2 or 5) **and PAlpha is set to 0**, the error increases significantly across all datasets. This is expected: with no propensity score term, the model relies entirely on the conditional outcome model, which is poorly estimated under such low bin settings. As a result, neither nuisance component is correctly specified, leading to inaccurate final estimates. Notably, the estimated interventional densities for NBins=2 and PAlpha=0 exhibit shapes that differ drastically from both the baseline and the observational distributions (see Appendix Figures C.3, D.3, E.3, and F.3 for details). These are worse outcomes than observed in the “both broken” scenarios (Research Question 1), even for GaussianClean and BimodalClean, where completely broken components still yielded estimates similar to the baseline.

The exact reason behind this behavior is not entirely straightforward and falls outside the core scope of this paper. However, a plausible hypothesis is as follows: in the “both broken” experiments, the conditional outcome model is explicitly replaced with a fixed Gaussian distribution - simple and smooth by design. In contrast, when using very low NBins, the CNF is still active but severely limited in its ability to model the conditional outcome distribution’s shape. This could lead to distorted, highly unrealistic estimates, essentially breaking the conditional outcome model in a different way. When combined with the absence of a propensity score (PAlpha = 0), this very poorly specified outcome model could significantly destabilize the target flow, resulting in extreme estimation errors. Thus, the nature of how the conditional model is broken could significantly affect results. This point is briefly revisited in Section 5.4.

Moving on, **increasing NBins to high values** (e.g., 50 or 100) **while setting PAlpha to 0** also results in high interventional density estimation error. This is due to the erratic shape of the estimates, characterized by sharp oscillations (see Appendix Figures C.5, D.5, E.5, and F.5 for details). Despite this, the overall shape still bears some resemblance to the baseline distribution. This behavior can be explained by the fact that with PAlpha = 0, the model depends entirely on the conditional outcome component. With a high number of bins, the conditional model becomes overfitted to the training data, capturing fine-grained variations that may not generalize well. While the precise workings of the target flow are outside the scope of this paper, it is plausible that relying solely on an overfitted conditional model leads to unstable and erratic interventional estimates.

When **PAlpha is set to a high value** (e.g., 50) **and NBins is increased**, the erratic behavior of the estimates is substantially reduced, and the error decreases. This is likely because a high PAlpha allows the model to rely on the propensity score in addition to the potentially overfitted conditional outcome model. The only exception is when NBins reaches 100, where the error slightly increases, possibly due to persistent overfitting effects. However, this improvement does not apply to the SplitPeaks dataset, where the error remains high even with large PAlpha. This further supports the finding that, for this dataset, the conditional outcome model plays a more crucial role.

Finally, across all datasets, **small deviations from the ideal baseline values** - such as using 15 or 25 NBins instead of 20 - **do not lead to significantly worse estimates**. This robustness is encouraging for real-world scenarios, where exact tuning of hyperparameters may not always be feasible.

5.3 Impact of Noise Injection

The results show that INFs exhibit varying sensitivity to noise across datasets. **In general, datasets with higher confounding appear more sensitive to noise**. For **GaussianClean** and **BimodalClean**, which have low confounding, adding noise with all three standard deviations did not lead to significantly degraded estimates. Even with medium or high noise, the increase in error is modest, and the estimated densities remain close to the baseline.

For **SplitPeaks**, the impact of noise is notably different. Adding low or medium noise to the propensity score does not lead to much error. However, introducing medium noise to the conditional outcome model already causes a significant deviation, reinforcing that this component is more critical for this dataset. Interestingly, high propensity noise does lead to a substantially larger error compared to medium noise, indicating that the propensity score is not entirely negligible and still contributes to the final estimate.

The **SyntheticComplex** dataset appears even more sensitive to noise. Any form of medium noise leads to a noticeable increase in error, with high noise amplifying it further. Moreover, applying noise to both nuisance components consistently results in higher error than applying noise to either component individually. This suggests that, for this dataset, accurate estimation of both the propensity score and the conditional outcome model is essential.

The results for SplitPeaks and SyntheticComplex show that adding noise to even a single nuisance component can substantially degrade the final estimates. Notably, both datasets exhibit high confounding. This suggests another potential limitation of the doubly robust property: **in highly confounded settings, even moderate levels of noise may be enough to undermine its expected robustness**.

Furthermore, the shape of the estimates under high noise resemble those from the "broken" scenarios in Research Question 1, in the sense that they are not drastically different from the true interventional or observational distributions (see Appendix Sections C.2, D.2, E.2, F.2 for details.) This suggests that the noise levels used are not extreme enough to fully distort the estimates, unlike the case with NBins = 2 and PAlpha = 0.

Finally, across all datasets, **injecting low levels of noise**, even into both nuisance components, **does not result in high error**. This robustness is encouraging for real-world applications, where models may inevitably not fully capture either the propensity score or the conditional outcome model with perfect precision.

5.4 Experimental Design Limitations and Directions for Future Work

Now that the results have been discussed, it is important to reflect on how the experimental setup may have influenced the findings and where its limitations lie.

One key limitation concerns **how the conditional outcome model was "broken."** In this paper, it was replaced with a fixed Normal distribution (mean 50.0, standard deviation 2.0). Combined with a broken propensity score, this led to estimates resembling the observational distribution. However, alternative ways of breaking the conditional model - e.g., using more complex distributions - were not explored. Such changes could have produced very different estimates, as seen in the NBins = 2, PAlpha = 0 case in the previous sections. This limits the generality of the findings, as different forms of model misspecification could yield substantially different outcomes.

Another limitation concerns **the noise injection experiments**. With the standard deviations tested, added noise did not significantly alter the estimated distributions compared to the baseline or $P(Y = y \mid A = a)$. It is possible that higher noise levels, beyond those used in this paper, could degrade the nuisance components more severely. Additionally, only Gaussian noise was considered, whereas real-world estimation noise may be more structured or non-Gaussian, potentially leading to different effects.

A further limitation is that the experimental setup is **restricted to one-dimensional continuous outcomes and binary treatments**. While this simplified the research, it limits generalization to higher-dimensional or multi-treatment scenarios common in practice.

Finally, this study focuses exclusively on the nuisance flow component of INF, with **the target flow architecture fixed** to the baseline configuration for each dataset across all experiments. It is possible that different target flow designs vary in their sensitivity to nuisance misspecification. As such, the robustness conclusions drawn here may not extend to alternative architectures. Future work could explore how architectural choices and target flow hyperparameters influence performance under misspecification.

These limitations indicate that while the findings offer empirical insights into INF robustness, they are constrained by the experimental design choices. roader generalization will require validating these observations across a wider range of architectures and misspecification scenarios. This presents a promising direction for future work building on this study.

6 Conclusions

This paper investigated the empirical robustness of INFs, by focussing on two research questions: (1) How do estimated interventional densities behave when the nuisance flow is completely misspecified? (2) How sensitive are these estimates to smaller, more realistic inaccuracies in the nuisance flow?

The nuisance flow estimates two key components: the propensity score model and the conditional outcome model. INFs are claimed to be doubly robust, meaning they can produce accurate interventional densities as long as either component is correctly specified. However, results show that in datasets where treatment assignment is strongly dependent on individual characteristics (i.e., high bias), misspecifying even one component leads to substantial estimation errors. In such cases, the estimated interventional distributions drift toward the observational (and biased) distributions, undermining causal validity. By contrast, in low-bias datasets, INF estimates remained accurate even when both components were deliberately broken.

For the second question, robustness was evaluated under small hyperparameter perturbations and injected noise. Estimates remained stable under minor imperfections, which is encouraging for practical use where exact tuning is rarely feasible. However, in high-bias settings, moderate perturbations could severely degrade estimate quality. In low-bias scenarios, only extreme parameter settings caused significant deviation.

These findings shine light on the INF's robustness under nuisance flow misspecification. While the method demonstrates resilience in low-bias settings, its doubly robust property weakens under strong bias. This highlights the importance of carefully validating nuisance components when using INFs. Future work should explore broader model architectures, higher-dimensional outcomes, and structured noise to better understand the boundaries of INF reliability.

References

1. Alaa, A. M., & Van Der Schaar, M. (2017). Bayesian Inference of Individualized Treatment Effects using Multi-task Gaussian Processes. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1704.02801

2. AMLab-Amsterdam. (n.d.). CEVAE/datasets/IHDP/csv/ihdp_npci_1.csv at master · *AMLab-Amsterdam/CEVAE. GitHub*. https://github.com/AMLab-Amsterdam/CEVAE/blob/master/datasets/IHDP/csv/ihdp_npci_1.csv

3. Wu, D., Inouye, D. I., & Xie, Y. (2025, May 21). PO-Flow: flow-based generative models for sampling potential outcomes and counterfactuals. *arXiv.org*. https://arxiv.org/abs/2505.16051

4. Feuerriegel, S., Frauen, D., Melnychuk, V., Schweisthal, J., Hess, K., Curth, A., Bauer, S., Kilbertus, N., Kohane, I. S., & Van Der Schaar, M. (2024). Causal machine learning for predicting treatment outcomes. *Nature Medicine*, *30*(4), 958–968. https://doi.org/10.1038/s41591-024-02902-1

5. Hatt, T., & Feuerriegel, S. (2021). Estimating average treatment effects via orthogonal regularization. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2101.08490

6. Hill, J. L. (2010). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, *20*(1), 217–240. https://doi.org/10.1198/jcgs.2010.08162

7. Jesson, A., Mindermann, S., Gal, Y., & Shalit, U. (2021). Quantifying Ignorance in Individual-Level Causal-Effect Estimates under Hidden Confounding. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2103.04850

8. Jesson, A., Douglas, A., Manshausen, P., Meinshausen, N., Stier, P., Gal, Y., & Shalit, U. (2022). Scalable sensitivity and uncertainty analysis for Causal-Effect estimates of Continuous-Valued interventions. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2204.10022

9. Kennedy, E. H., Balakrishnan, S., & Wasserman, L. A. (2023). Semiparametric counterfactual density estimation. *Biometrika*, *110*(4), 875–896. https://doi.org/10.1093/biomet/asad017

10. Kennedy, E. H. (2023). Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal Of Statistics*, *17*(2). https://doi.org/10.1214/23-ejs2157

11. Kennedy, E. H. (2022). Semiparametric doubly robust targeted double machine learning: a review. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2203.06469

12. Kim, K., Kim, J., & Kennedy, E. H. (2018). Causal effects based on distributional distances. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1806.02935

13. Melnychuk, V., Frauen, D., & Feuerriegel, S. (2022). Normalizing flows for interventional density estimation. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2209.06203

14. Muandet, K., Kanagawa, M., Saengkyongam, S., & Marukatat, S. (2018). Counterfactual mean embeddings. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1805.08845

15. Rezende, D., & Mohamed, S. (2015). Variational Inference with Normalizing Flows. *International Conference on Machine Learning*, 1530–1538. http://proceedings.mlr.press/v37/rezende15.pdf

16. Sage Bionetworks, info@sagebase.org. (n.d.). ACIC 2018 Causal Inference Challenge. https://www.synapse.org/Synapse:syn11294478/wiki/486304

17. Shalit, U., Johansson, F. D., & Sontag, D. (2016). Estimating individual treatment effect: generalization bounds and algorithms. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1606.03976

18. Shi, C., Blei, D. M., & Veitch, V. (2019). Adapting neural networks for the estimation of treatment effects. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1906.02120

19. Spiegelhalter, D. (2017). Risk and uncertainty communication. *Annual Review of Statistics and Its Application*, *4*(1), 31–60. https://doi.org/10.1146/annurev-statistics-010814-020148

20. Tabak, E. G., & Vanden-Eijnden, E. (2010). Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, *8*(1), 217–233. https://doi.org/10.4310/cms.2010.v8.n1.a11

21. Valentyn. (n.d.). INFs/README.md at main · *Valentyn1997/INFs. GitHub*. https://github.com/Valentyn1997/INFs/blob/main/README.md

22. Van Der Bles, A. M., Van Der Linden, S., Freeman, A. L. J., Mitchell, J., Galvao, A. B., Zaval, L., & Spiegelhalter, D. J. (2019). Communicating uncertainty about facts, numbers and science. *Royal Society Open Science*, *6*(5), 181870. https://doi.org/10.1098/rsos.181870

23. Vanderschueren, T. (2024). Operational decision-making with machine learning and causal inference. https://doi.org/10.63028/10067/2092470151162165141

24. Zhang, Y., Bellot, A., & Van Der Schaar, M. (2020). Learning overlapping representations for the estimation of individualized treatment effects. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2001.04754

7 Responsible Research

This section reflects on the ethical and methodological integrity of the research. It addresses three key dimensions: the use of literature and writing tools, the reproducibility of the experimental methodology, and the transparency and completeness of the reported results. Together, these considerations demonstrate a commitment to responsible and rigorous research practices throughout the study.

7.1 Writing and Literature

The literature review for this paper was conducted thoroughly and with care to ensure that relevant prior work was appropriately considered. Key theoretical concepts, such as the doubly robust property, are accurately referenced and discussed in context. All technical or factual claims made throughout the paper are supported by credible sources.

In terms of writing assistance, I used OpenAI's ChatGPT solely for grammar and language refinement during the writing process. It was not used for tasks such as forming arguments, or analyzing any part of the research. All critical thinking, research design, and analysis were independently developed and executed by me.

Typical prompts included clarification questions or minor editing requests, such as: "Is this sentence grammatically correct?" and "Can you help rephrase this paragraph more clearly?"

7.2 Methodology

To ensure the research presented in this paper is transparent, reproducible, and ethically sound, all experiments were designed and documented with care.

This research builds on the official INF codebase developed by Valentyn (n.d.-b), which is publicly available here: `https://github.com/Valentyn1997/INFs/blob/main/README.md` All four datasets used in the experiments - GaussianClean, BimodalClean, SplitPeaks, and SyntheticComplex - are also publicly accessible and are either included directly or referenced within the INF repository.

To conduct the experiments presented in this paper, I extended the original INF codebase by adding custom modules for breaking components of the nuisance flow and injecting controlled noise. The extended version of the code is also publicly available: `https://github.com/RuthvikAllu/RP_Codebase_INF`

To ensure consistency and isolate the effects of each manipulation, all experiments followed a controlled design: only the variable being tested was changed, while all other hyperparameters were kept at their default, baseline values. These baseline hyperparameters - optimized by the authors of the INF model for each dataset - are explicitly listed in the paper and are also included in the repository configuration files.

All experimental runs were conducted using fixed random seeds, specifically the default seeds provided in the original INF codebase. This supports reproducibility, by allowing others to exactly replicate the results reported in this paper, and replicability, by ensuring that the same methodology can be applied to new data.

Furthermore, detailed procedural descriptions are provided in the Methodology section of this paper, explaining exactly how components were broken, how noise was injected, and how evaluation metrics were computed. The necessary implementation additions are found in the extended codebase.

Moreover, to ensure thoroughness, the research systematically explored a wide range of experimental configurations. This included numerous hyperparameter perturbation scenarios and multiple levels of controlled noise injection. By varying these settings across both components of the INF model, the study was able to conduct a robust and comprehensive evaluation of the model's behavior under different forms and degrees of misspecification.

7.3 Results

The results presented in this paper aim to reflect a complete and honest picture of the INF model's performance under varying experimental conditions. Care was taken to include nuanced outcomes, not just those that support a particular narrative. No results were cherry-picked, and none were excluded simply because they appeared unexpected or contradicted theoretical assumptions.

For example, in some cases, the INF model's baseline estimates did not perfectly match the known ground truth interventional distributions $P(Y[a] = y)$. Rather than only selecting datasets where the baseline aligned well with the ground truth, I deliberately included all four datasets, even when their baseline behavior revealed modeling limitations. This choice reflects a commitment to transparency and allows for a more realistic evaluation of the model's practical reliability.

Furthermore, consider the SplitPeaks dataset as a specific case. Its behavior diverged from the other datasets in several ways. Notably, breaking the propensity score component led to more severe degradation in estimates compared to the GaussianClean or BimodalClean datasets. Rather than omitting these results or treating them as anomalies, I fully included and discussed them in the paper. Doing so highlighted important dataset-specific sensitivities and revealed potential weaknesses in the INF model's doubly robust property under stronger confounding.

This approach applies across the entire analysis: wherever unusual or inconsistent behavior was observed, such as non-alignment with ground truth, model drift toward observational distributions, or high L1 error values under specific perturbations, these cases were analyzed and contextualized rather than excluded. By recognizing and presenting the model's limitations alongside its strengths, the paper aims to contribute a balanced and rigorous assessment of INF robustness under real-world imperfections.

A BimodalClean's Sensitivity to Broken Nuisance Flow Components

A Hyperparameter Configurations

This appendix provides the full set of hyperparameter values used in the experiments for each dataset. These configurations correspond to the optimal settings recommended in the official INF codebase and were used as the baseline reference points throughout the study.

GaussianClean

```
nuisance_count_bins: 20
nuisance_hid_dim_multiplier: 10
noise_std_X: 0.1
noise_std_Y: 0.0
nuisance_lr: 0.005
batch_size: 32
num_epochs: 5000
noise_ce: 0.0
target_count_bins: 10
target_quadrature: rect
target_lr: 0.005
target_mode: batch
target_nce_bins: 100
target_ema: 0.995
prop_alpha: 1.0
clip_prop: 0.05
```

BimodalClean

```
nuisance_count_bins: 20
nuisance_hid_dim_multiplier: 10
noise_std_X: 0.05
noise_std_Y: 0.01
nuisance_lr: 0.001
batch_size: 64
num_epochs: 5000
noise_ce: 0.0
target_count_bins: null
target_quadrature: rect
target_lr: 0.005
target_mode: batch
target_nce_bins: 100
target_ema: 0.995
prop_alpha: 1.0
clip_prop: 0.05
```

SplitPeaks

```
nuisance_count_bins: 10
nuisance_hid_dim_multiplier: 10
noise_std_X: 0.1
noise_std_Y: 0.05
nuisance_lr: 0.001
batch_size: 64
num_epochs: 5000
noise_ce: 0.0
target_count_bins: null
target_quadrature: rect
target_lr: 0.005
target_mode: batch
target_nce_bins: 100
target_ema: 0.995
prop_alpha: 1.0
clip_prop: 0.05
```

SyntheticComplex

```
nuisance_count_bins: 10
nuisance_hid_dim_multiplier: 10
noise_std_X: 0.0
noise_std_Y: 0.05
nuisance_lr: 0.001
batch_size: 32
num_epochs: 5000
noise_ce: 0.0
target_count_bins: 5
target_quadrature: rect
target_lr: 0.005
target_mode: batch
target_nce_bins: 100
target_ema: 0.995
prop_alpha: 1.0
clip_prop: 0.05
```

B BimodalClean's Sensitivity to Broken Nuisance Flow Components

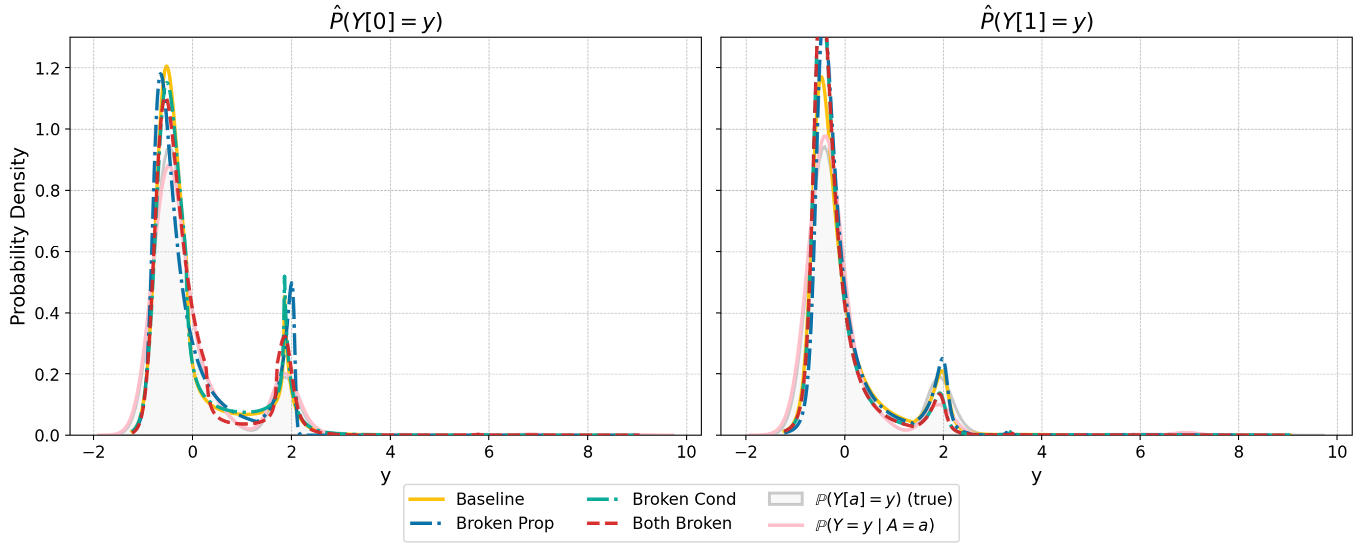

Figure B.1: Estimated interventional densities $\hat{P}(Y[a]=y)$ for $a \in \{0,1\}$ on the **BimodalClean** dataset. Shown are estimates under the four configurations.

C GaussianClean's Sensitivity to Minor Nuisance Flow Misspecifications

C.1 Hyperparameter Perturbations Results

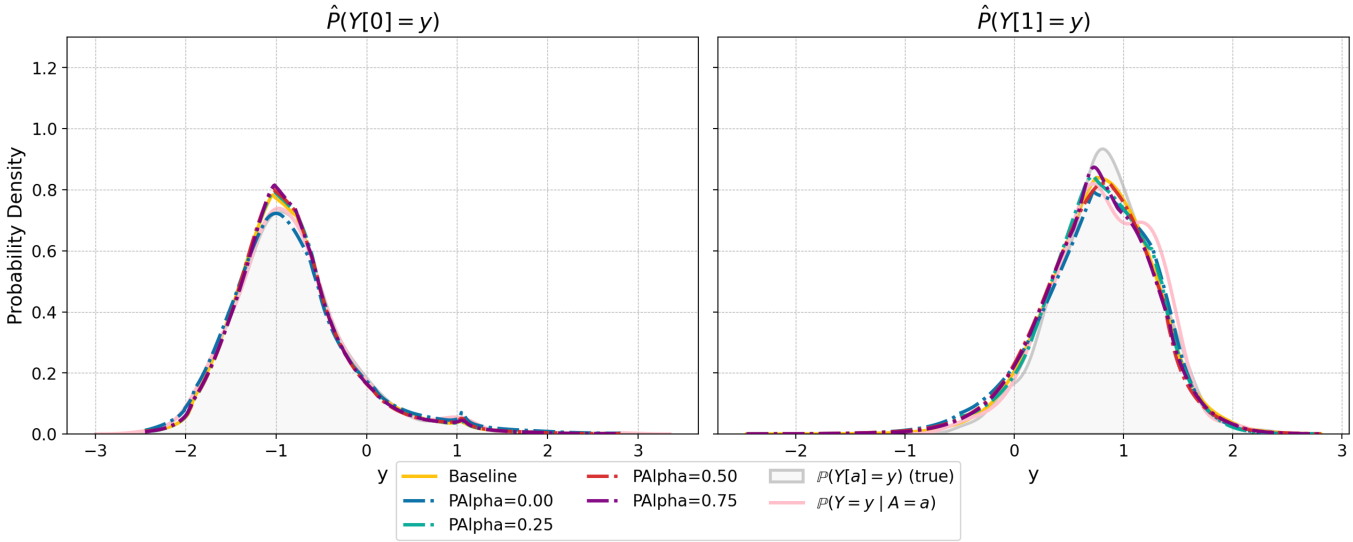

Figure C.1: Estimated interventional distributions for the GaussianClean dataset under decreasing values of PAlpha.

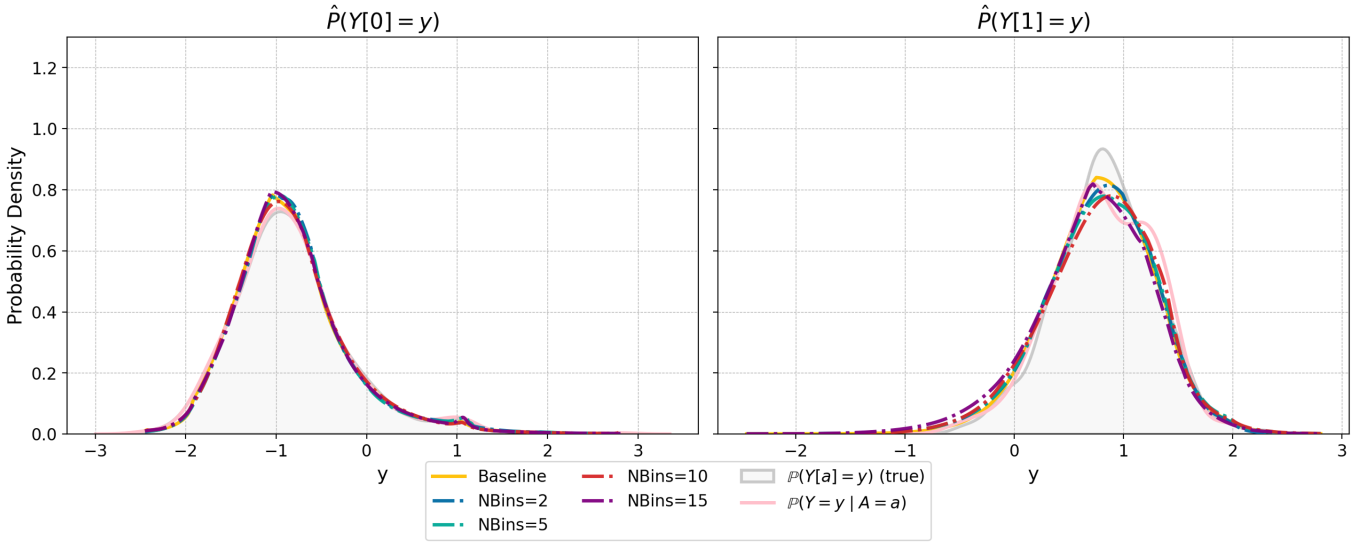

Figure C.2: Estimated interventional distributions for the GaussianClean dataset under decreasing values of NBins

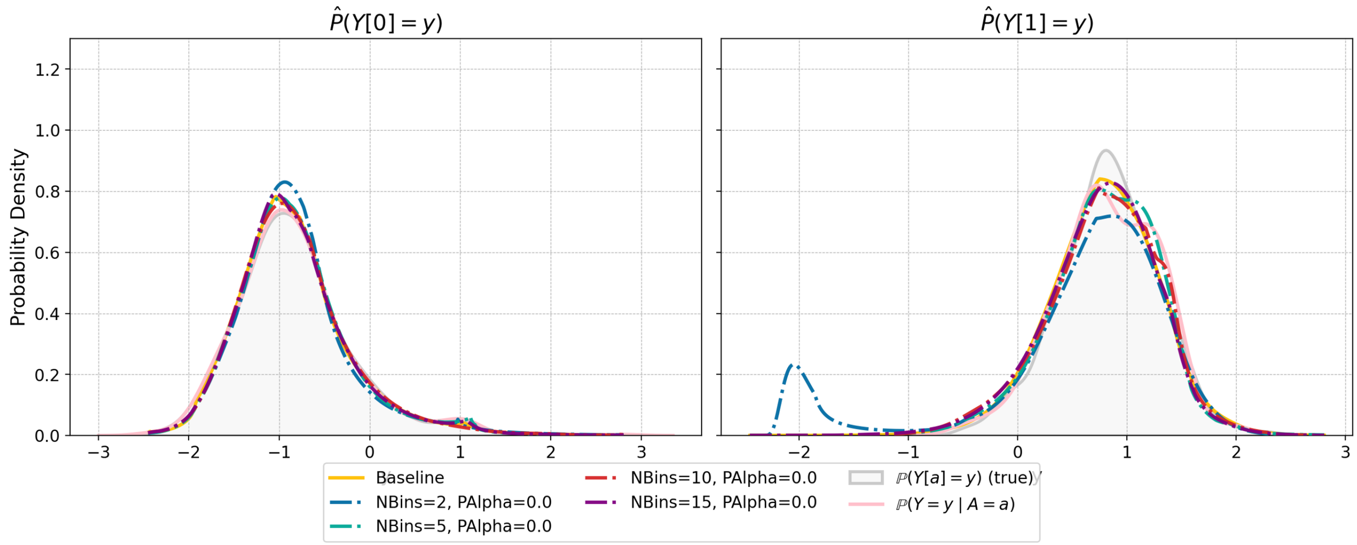

Figure C.3: Estimated interventional distributions for the GaussianClean dataset under decreasing values of NBins, with PAlpha set to 0.0.

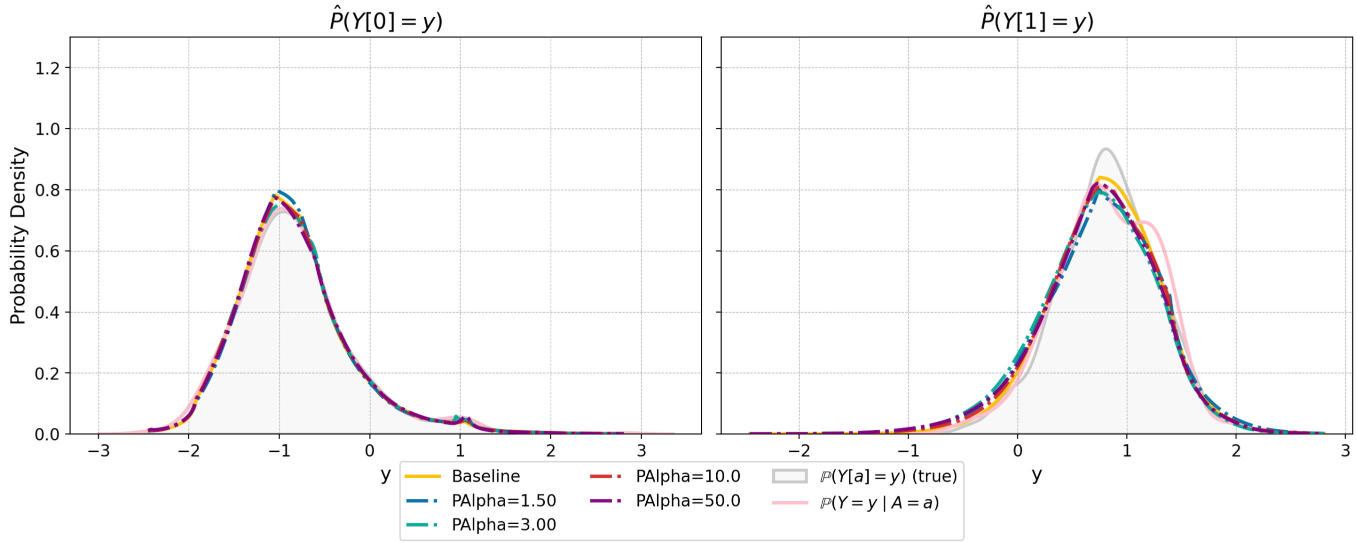

Figure C.4: Estimated interventional distributions for the GaussianClean dataset under increasing values of PAlpha.

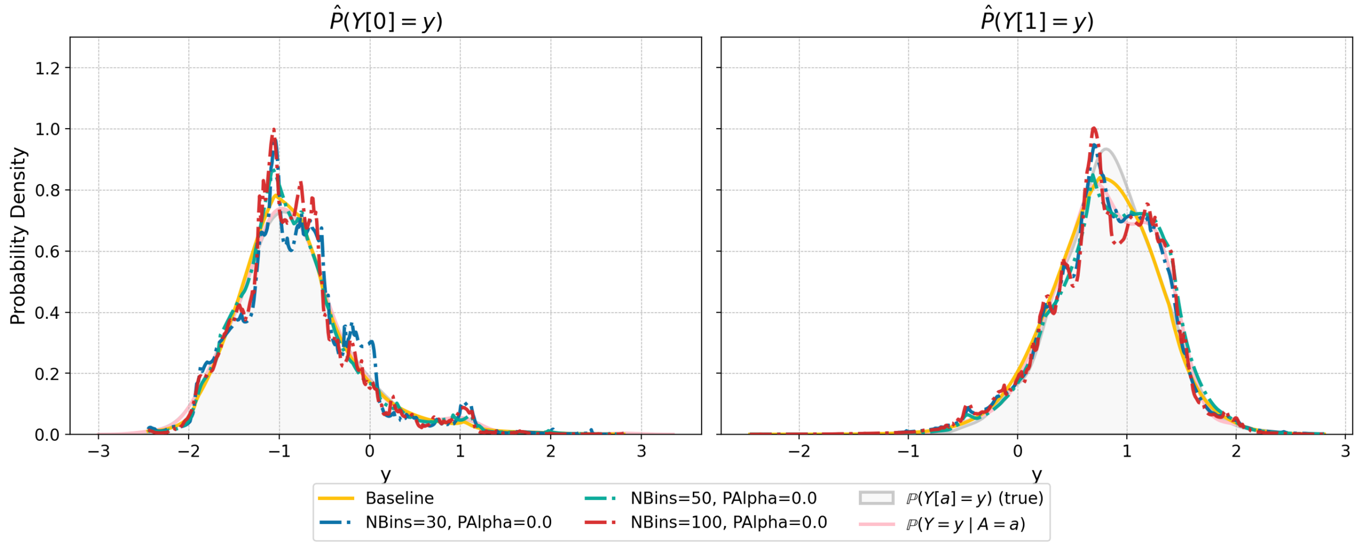

Figure C.5: Estimated interventional distributions for the GaussianClean dataset under increasing values of NBins, with PAlpha set to 0.0.

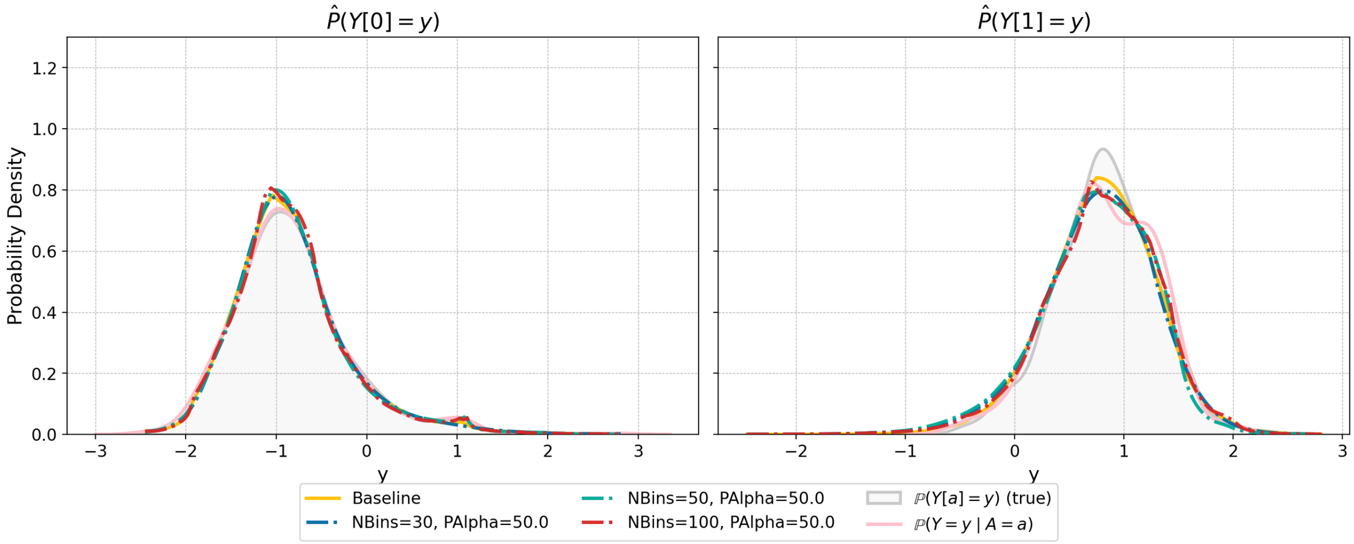

Figure C.6: Estimated interventional distributions for the GaussianClean dataset under increasing values of NBins, with PAlpha set to 50.0.

C.2 Controlled Noise Injection Results

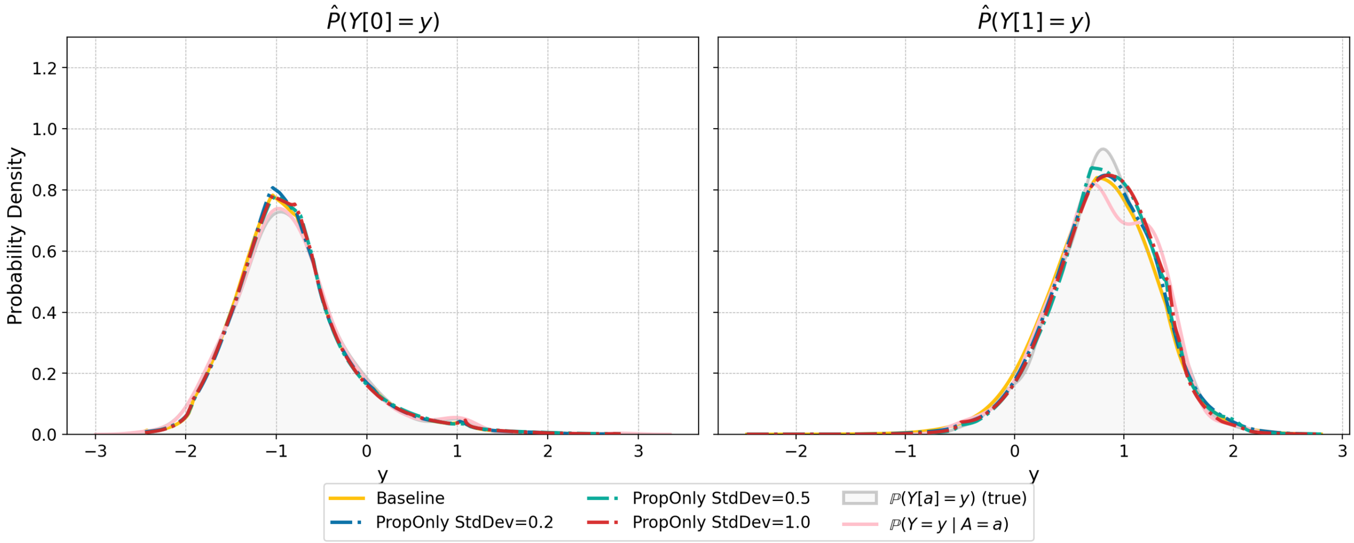

Figure C.7: Estimated interventional distributions for the GaussianClean dataset under increasing levels of noise injected into the propensity score estimate.

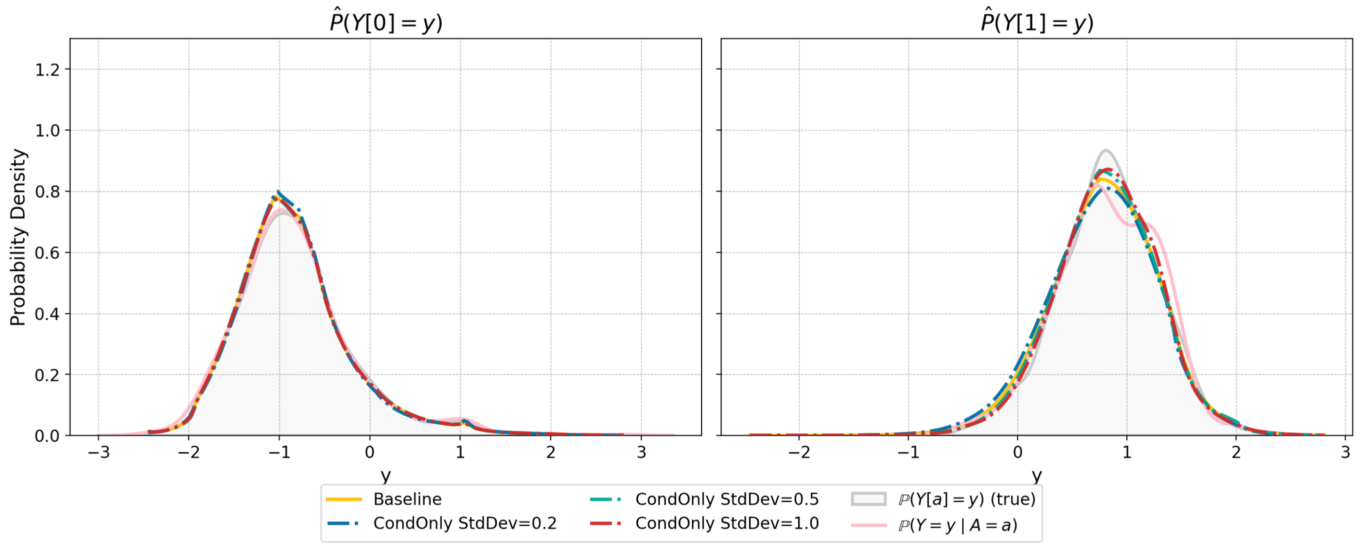

Figure C.8: Estimated interventional distributions for the GaussianClean dataset under increasing levels of noise injected into the conditional outcome model estimate.

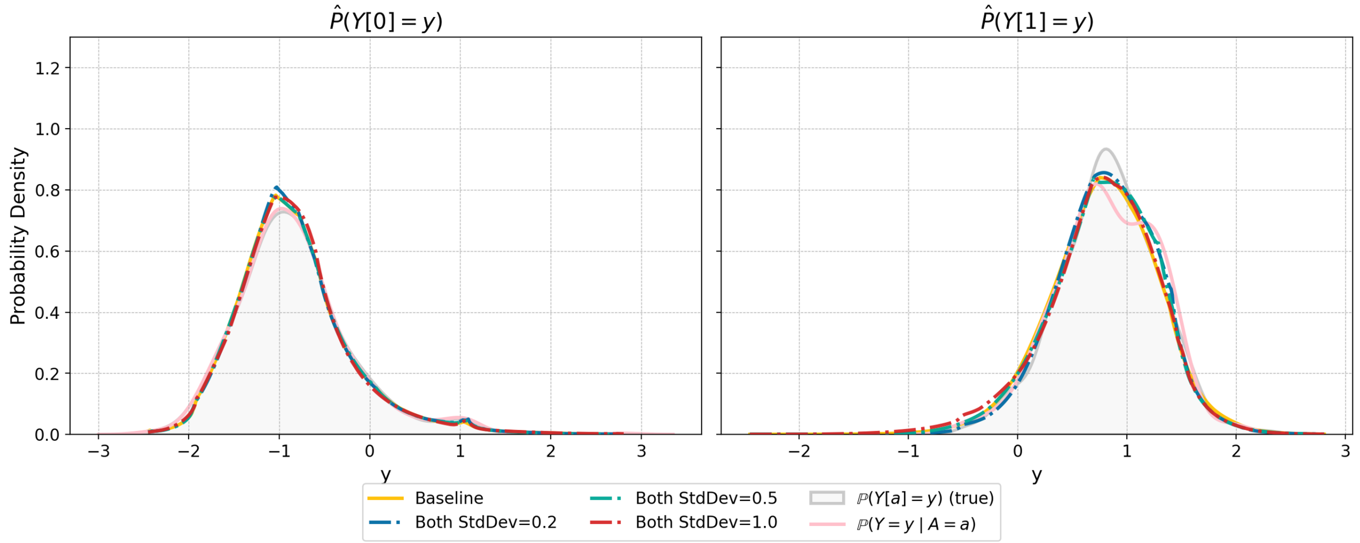

Figure C.9: Estimated interventional distributions for the GaussianClean dataset under increasing levels of noise injected into both the propensity score and conditional outcome model estimates.

D BimodalClean's Sensitivity to Minor Nuisance Flow Misspecifications

D.1 Hyperparameter Perturbations Results

Table 8: Average L1 error on the BimodalClean dataset under hyperparameter perturbations. The baseline configuration uses NBins=20 and PAlpha=1.0.

Perturbation	L1 Error
NBins=2	0.0754
NBins=5	0.0814
NBins=10	0.0887
NBins=15	0.0852
NBins=30, PAlpha=50.0	0.0680
NBins=50, PAlpha=50.0	0.0845
NBins=100, PAlpha=50.0	0.1447

Perturbation	L1 Error
PAlpha=0.00	0.0792
PAlpha=0.25	0.0867
PAlpha=0.50	0.0680
PAlpha=0.75	0.0845
PAlpha=1.50	0.1177
PAlpha=3.00	0.0610
PAlpha=50.0	0.1248

Perturbation	L1 Error
NBins=2, PAlpha=0.0	0.7618
NBins=5, PAlpha=0.0	0.2972
NBins=10, PAlpha=0.0	0.3113
NBins=15, PAlpha=0.0	0.1688
NBins=30, PAlpha=0.0	0.1239
NBins=50, PAlpha=0.0	0.2474
NBins=100, PAlpha=0.0	0.2175

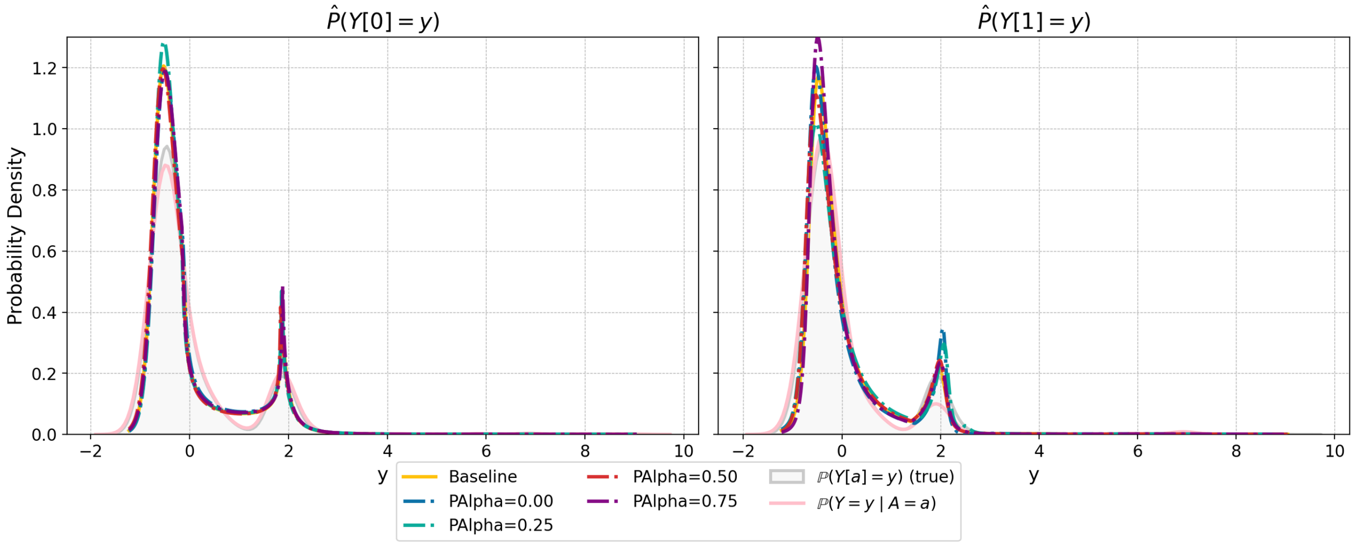

Figure D.1: Estimated interventional distributions for the BimodalClean dataset under decreasing values of PAlpha.

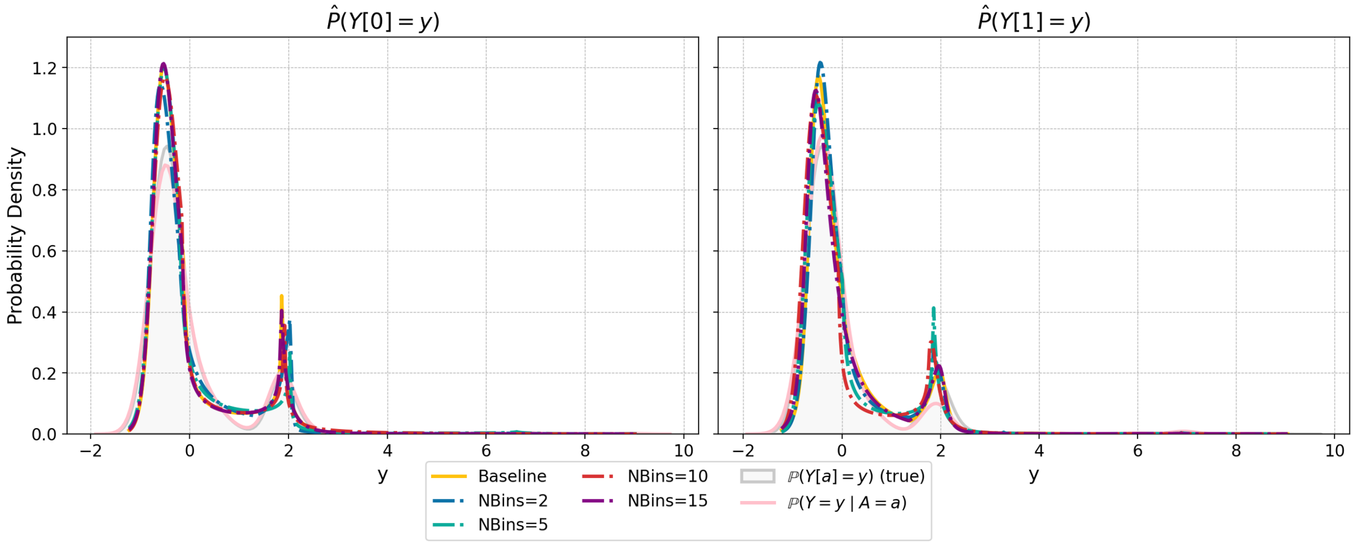

Figure D.2: Estimated interventional distributions for the BimodalClean dataset under decreasing values of NBins.

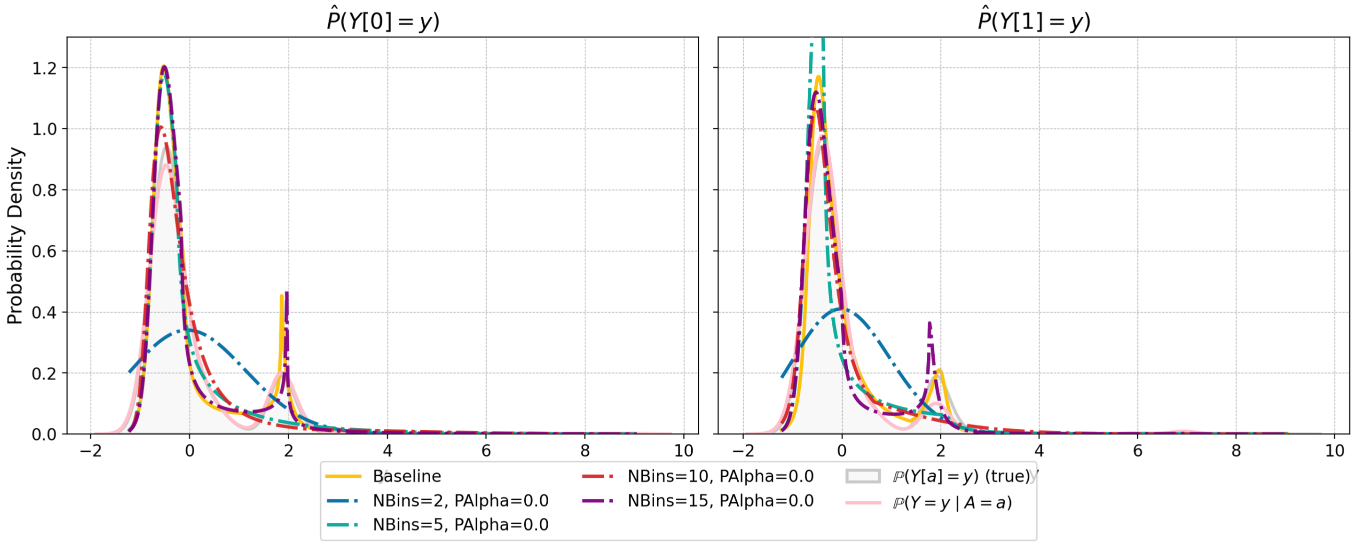

Figure D.3: Estimated interventional distributions for the BimodalClean dataset under decreasing values of NBins, with PAlpha set to 0.0.

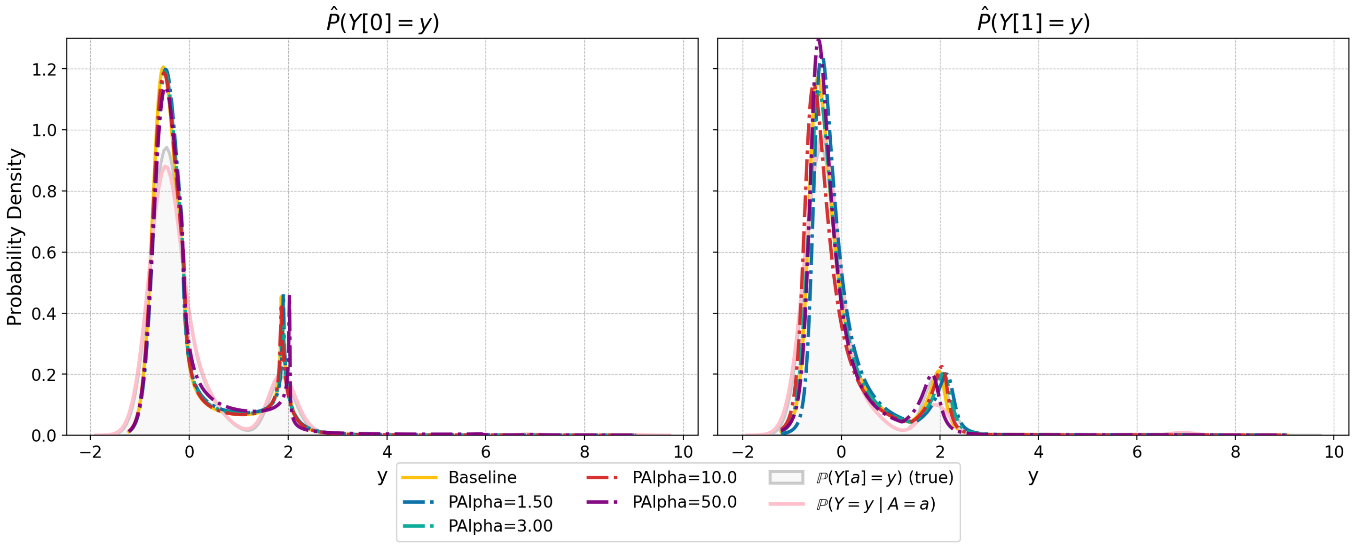

Figure D.4: Estimated interventional distributions for the BimodalClean dataset under increasing values of PAlpha.

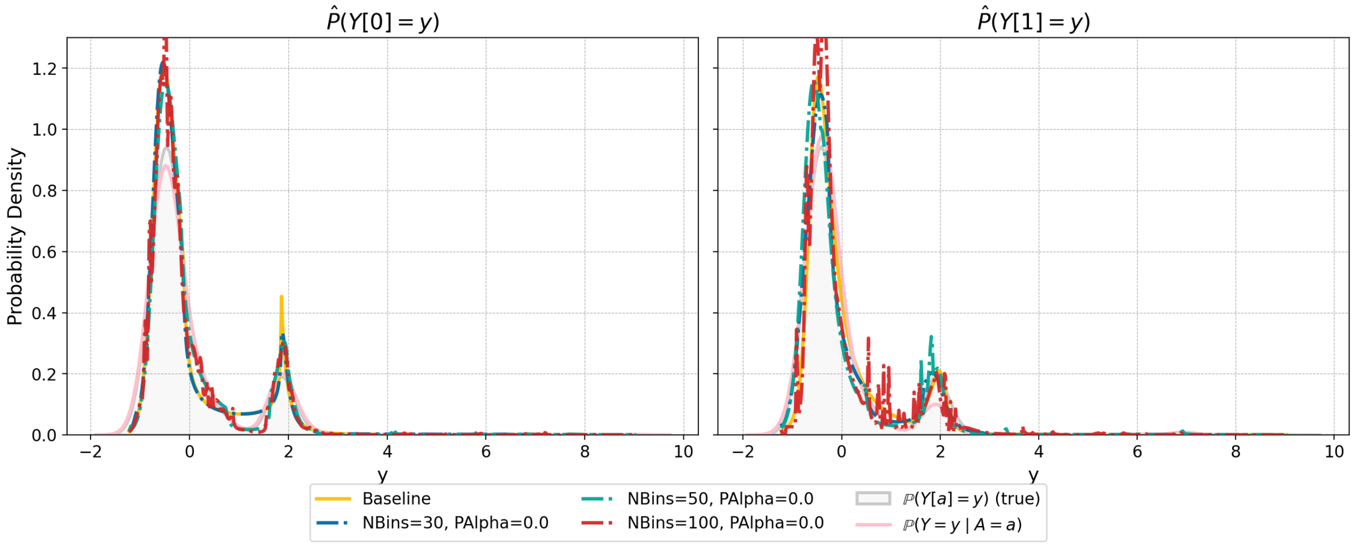

Figure D.5: Estimated interventional distributions for the BimodalClean dataset under increasing values of NBins, with PAlpha set to 0.0.

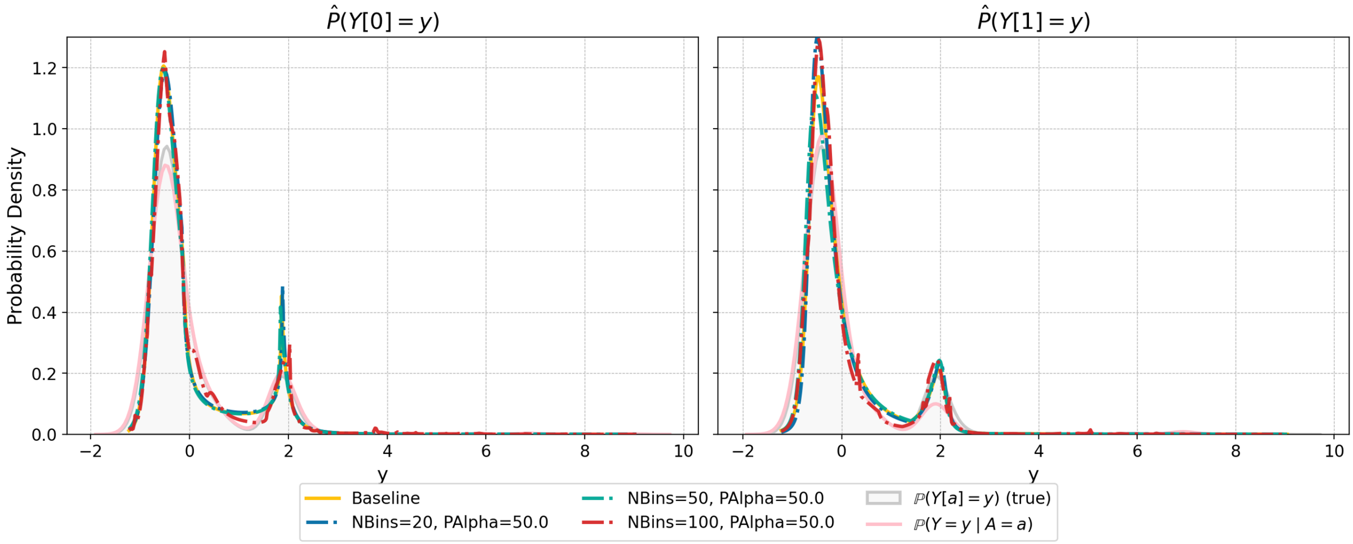

Figure D.6: Estimated interventional distributions for the BimodalClean dataset under increasing values of NBins, with PAlpha set to 50.0.

D.2 Controlled Noise Injection Results

Table 9: Average L1 error on the BimodalClean dataset under noise injection.

Noise Type (StdDev)	L1 Error
Prop. Only (0.2)	0.0735
Prop. Only (0.5)	0.0668
Prop. Only (1.0)	0.0814

Noise Type (StdDev)	L1 Error
Cond. Only (0.2)	0.0648
Cond. Only (0.5)	0.0742
Cond. Only (1.0)	0.0872

Noise Type (StdDev)	L1 Error
Both (0.2)	0.0848
Both (0.5)	0.0880
Both (1.0)	0.0959

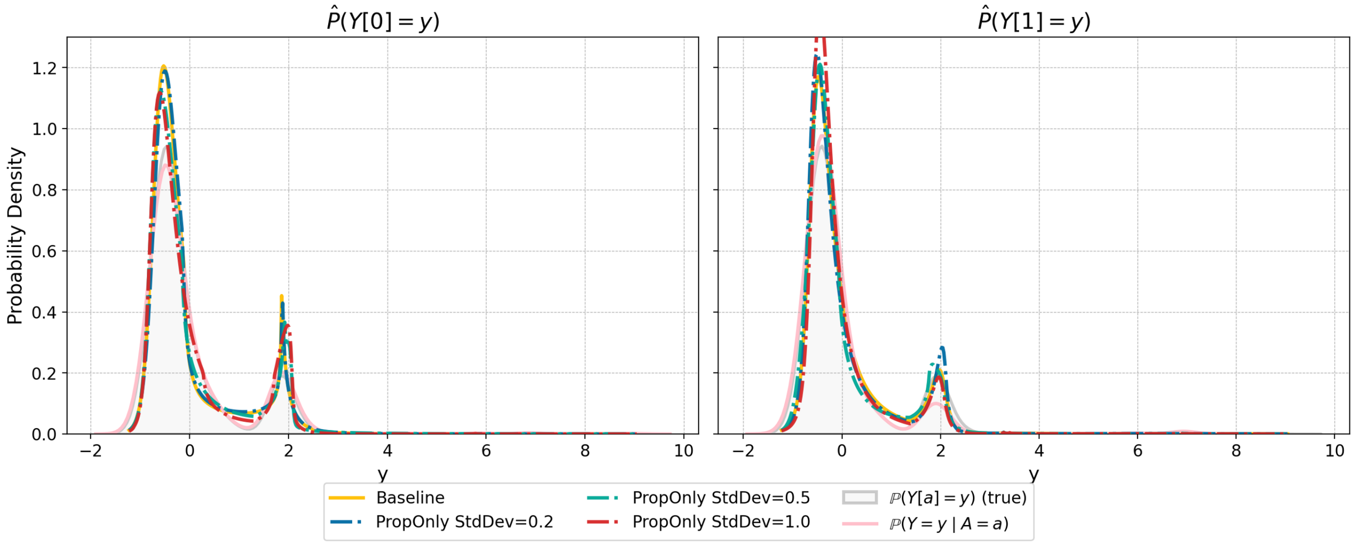

Figure D.7: Estimated interventional distributions for the BimodalClean dataset under increasing levels of noise injected into the propensity score estimate.

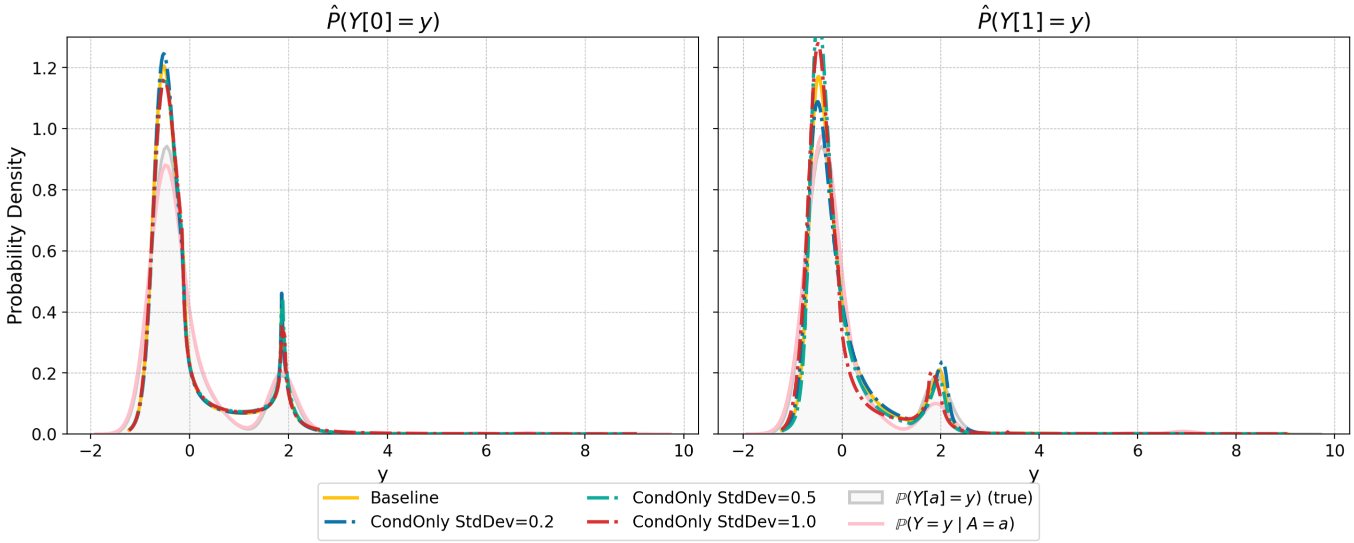

Figure D.8: Estimated interventional distributions for the BimodalClean dataset under increasing levels of noise injected into the conditional outcome model estimate.

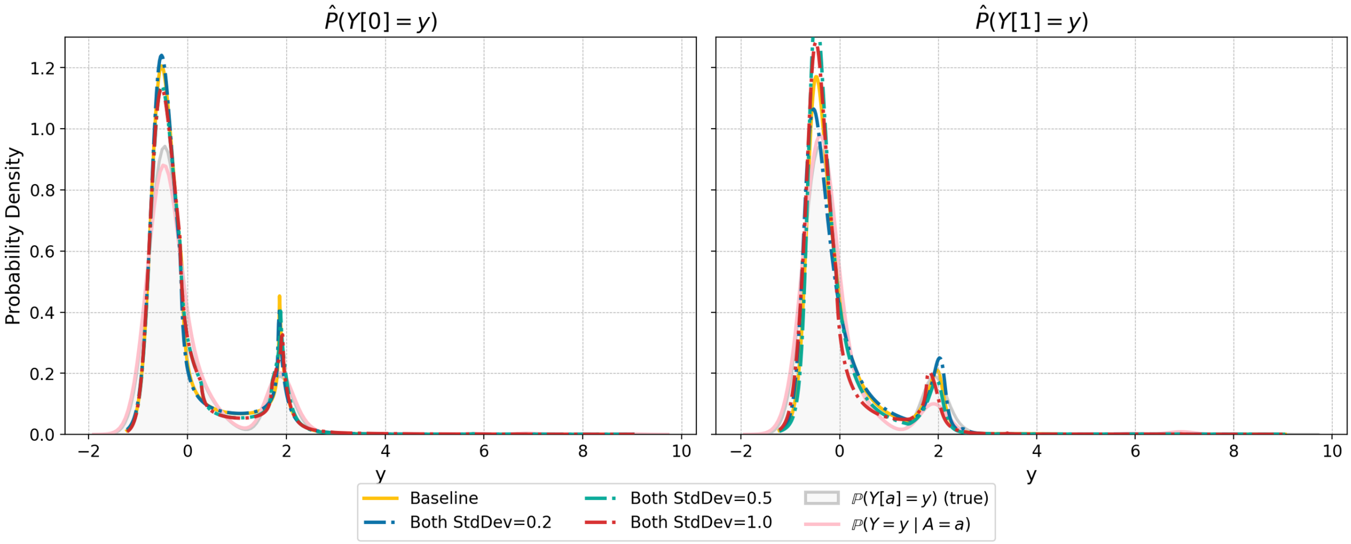

Figure D.9: Estimated interventional distributions for the BimodalClean dataset under increasing levels of noise injected into both the propensity score and conditional outcome model estimates.

E SplitPeaks's Sensitivity to Minor Nuisance Flow Misspecifications

E.1 Hyperparameter Perturbations Results

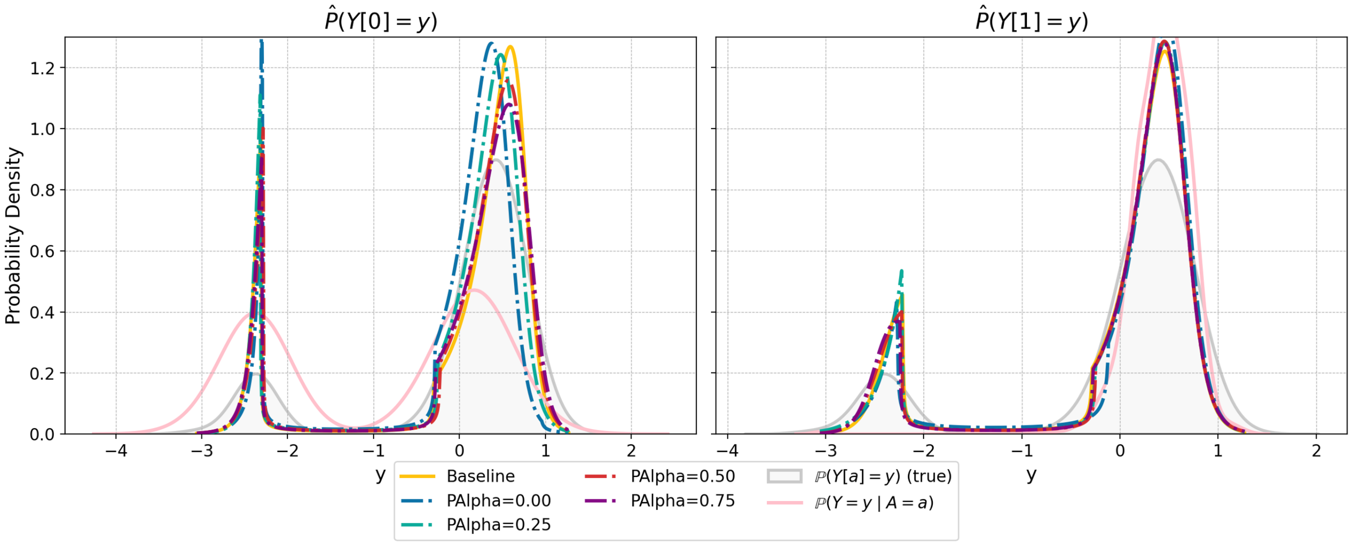

Figure E.1: Estimated interventional distributions for the SplitPeaks dataset under decreasing values of PAlpha.

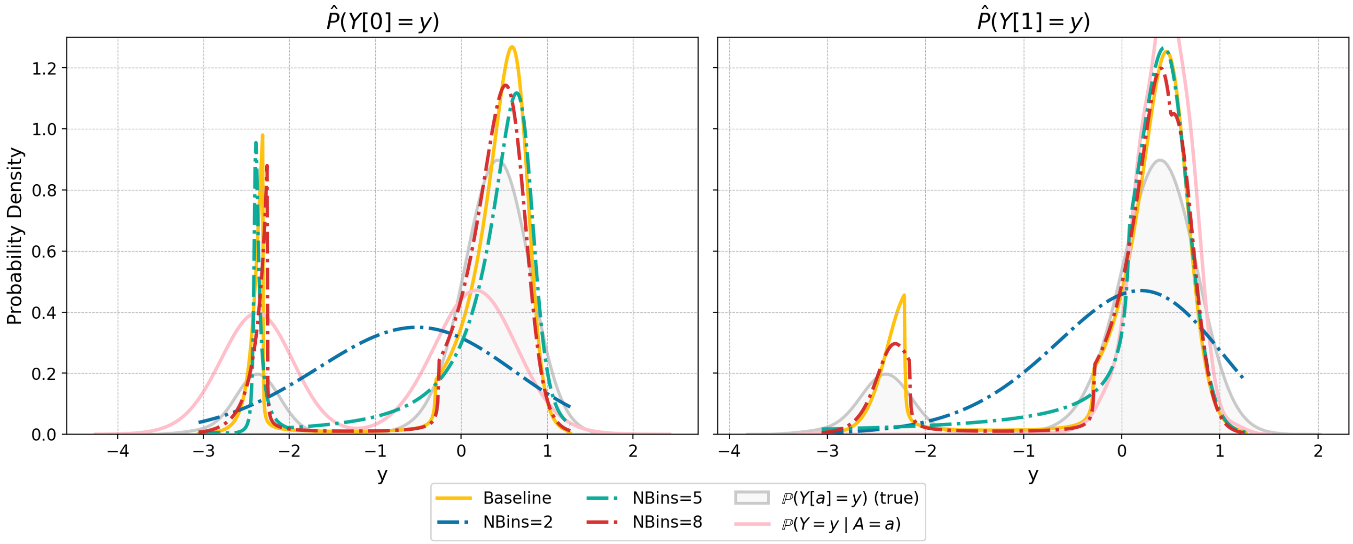

Figure E.2: Estimated interventional distributions for the SplitPeaks dataset under decreasing values of NBins

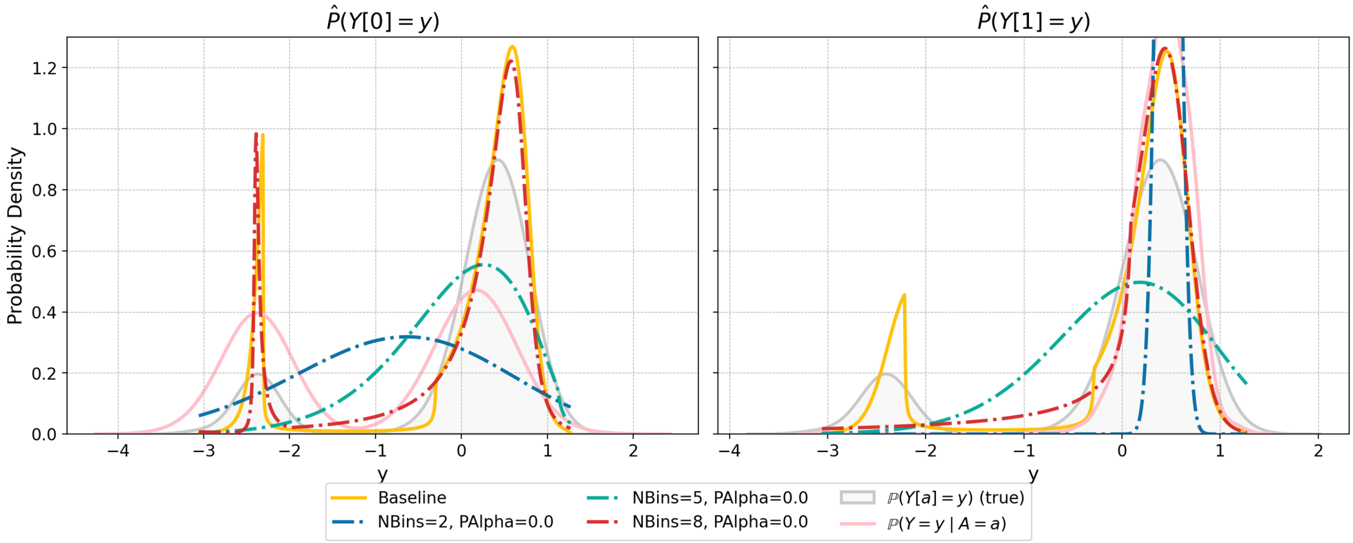

Figure E.3: Estimated interventional distributions for the SplitPeaks dataset under decreasing values of NBins, with PAlpha set to 0.0.

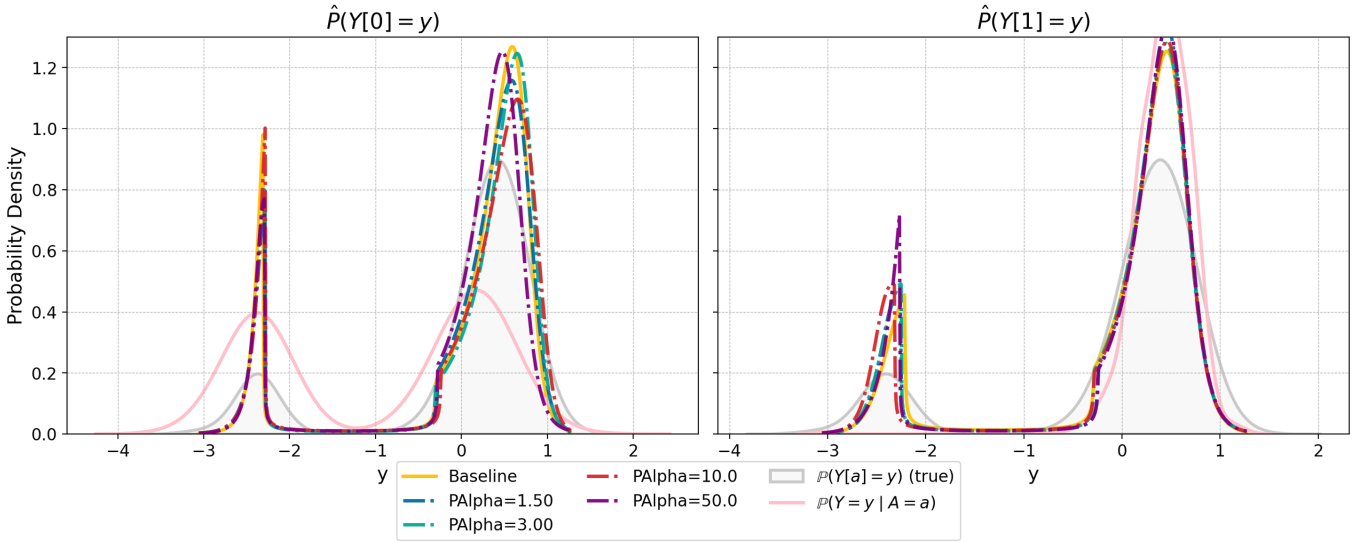

Figure E.4: Estimated interventional distributions for the SplitPeaks dataset under increasing values of PAlpha.

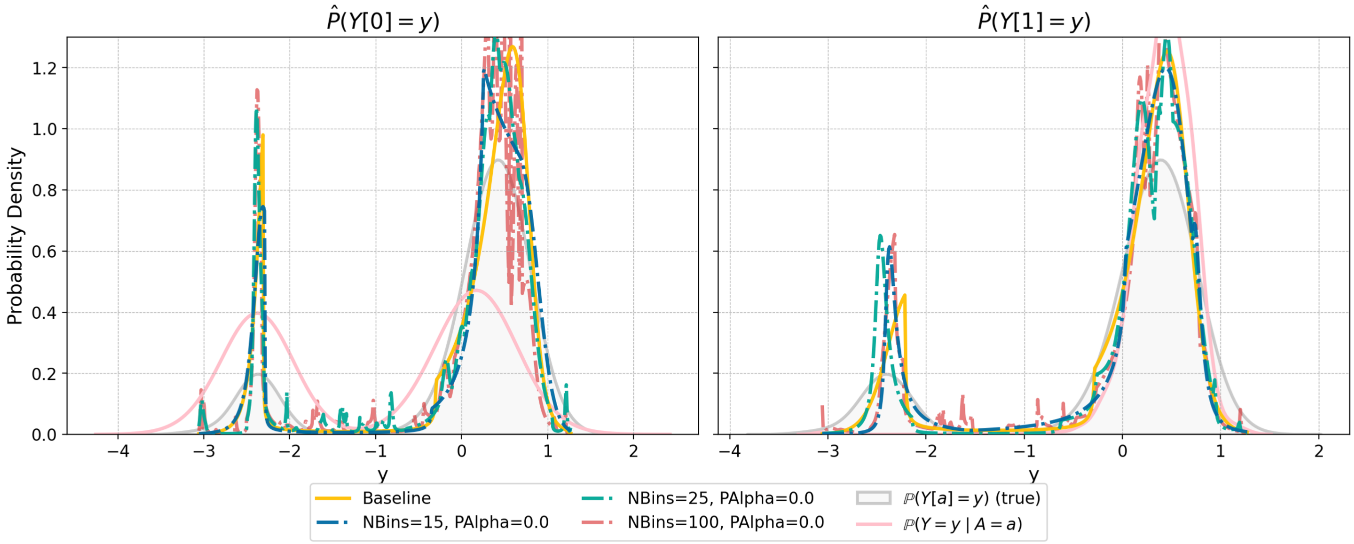

Figure E.5: Estimated interventional distributions for the SplitPeaks dataset under increasing values of NBins, with PAlpha set to 0.0.

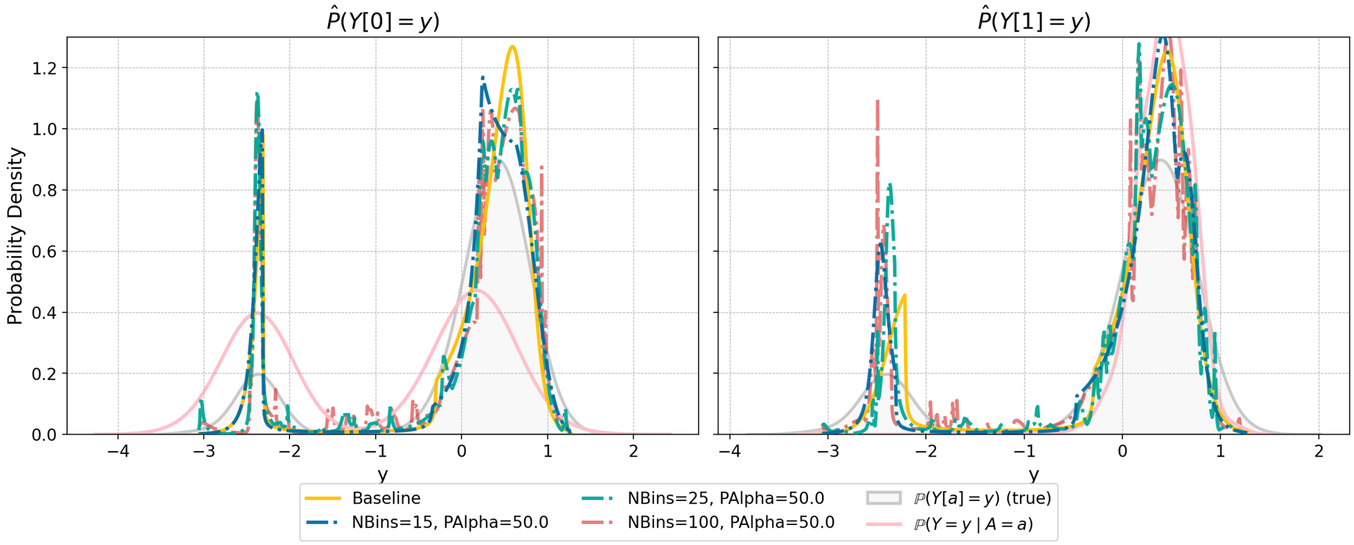

Figure E.6: Estimated interventional distributions for the SplitPeaks dataset under increasing values of NBins, with PAlpha set to 50.0.

E.2 Controlled Noise Injection Results

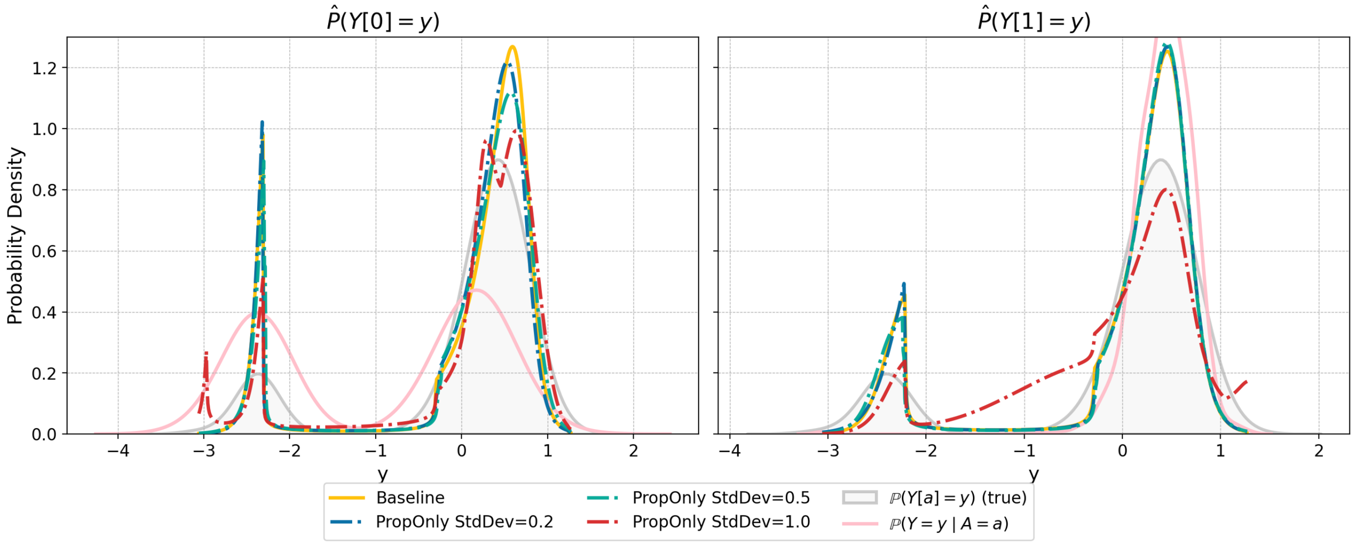

Figure E.7: Estimated interventional distributions for the SplitPeaks dataset under increasing levels of noise injected into the propensity score estimate.

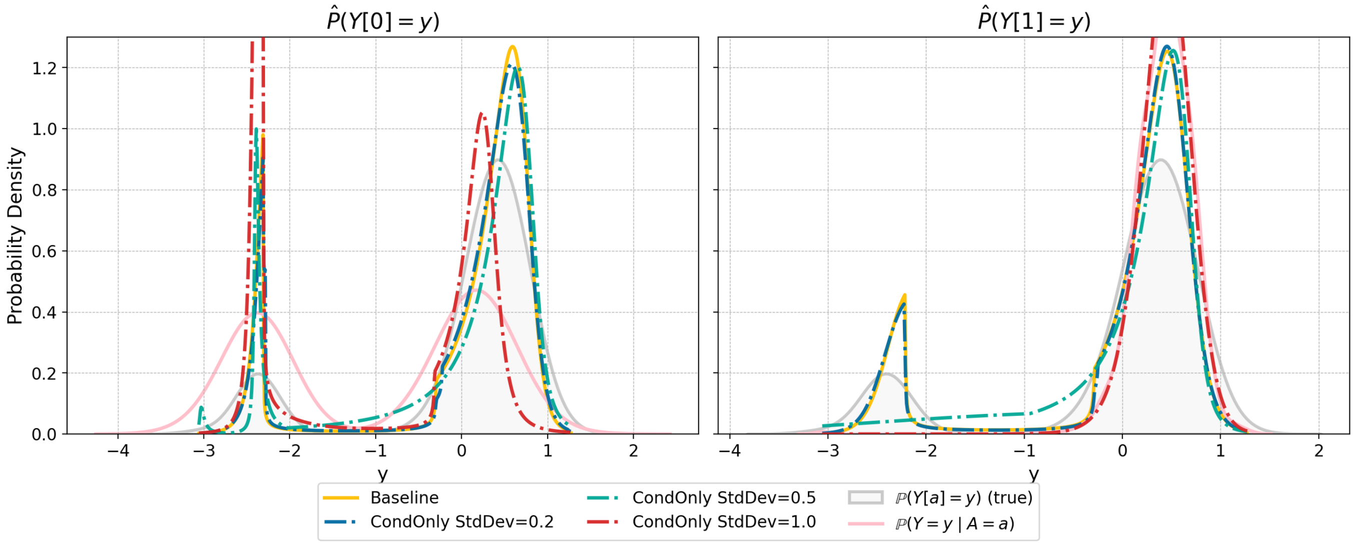

Figure E.8: Estimated interventional distributions for the SplitPeaks dataset under increasing levels of noise injected into the conditional outcome model estimate.

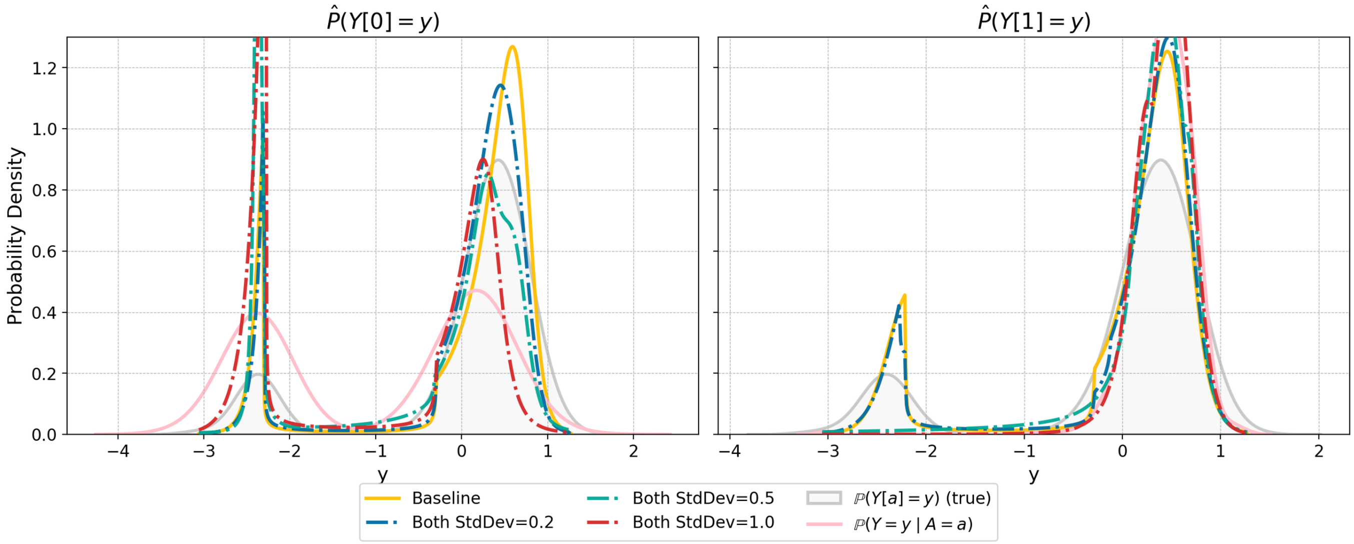

Figure E.9: Estimated interventional distributions for the SplitPeaks dataset under increasing levels of noise injected into both the propensity score and conditional outcome model estimates.

F SyntheticComplex's Sensitivity to Minor Nuisance Flow Misspecifications

F.1 Hyperparameter Perturbations Results

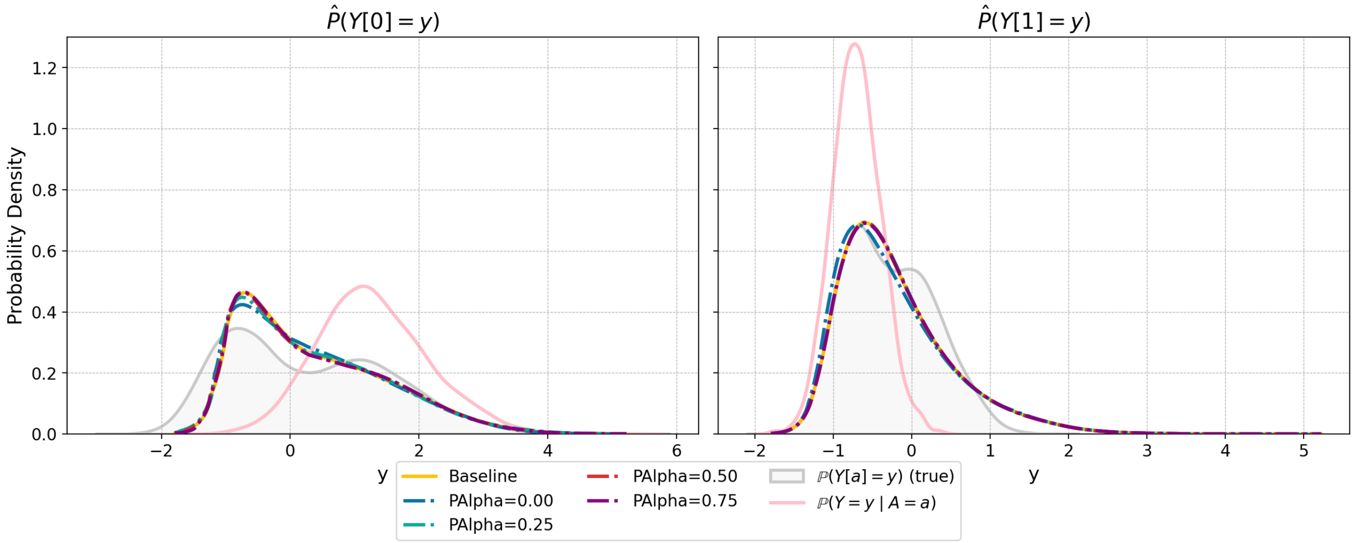

Figure F.1: Estimated interventional distributions for the SyntheticComplex dataset under decreasing values of PAlpha.

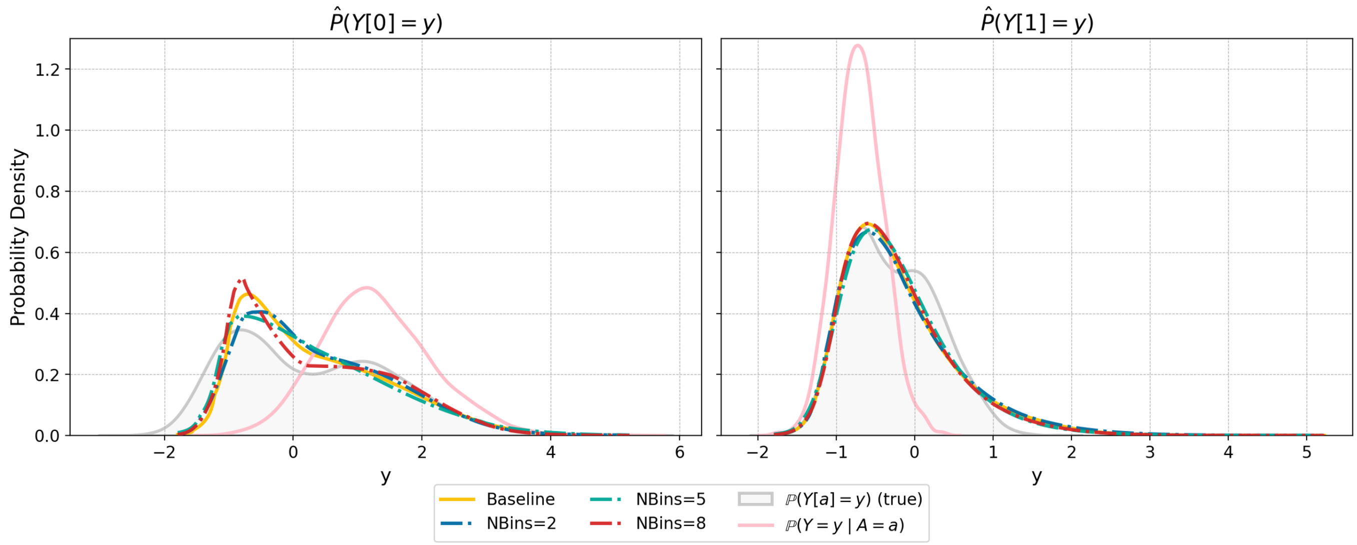

Figure F.2: Estimated interventional distributions for the SyntheticComplex dataset under decreasing values of NBins

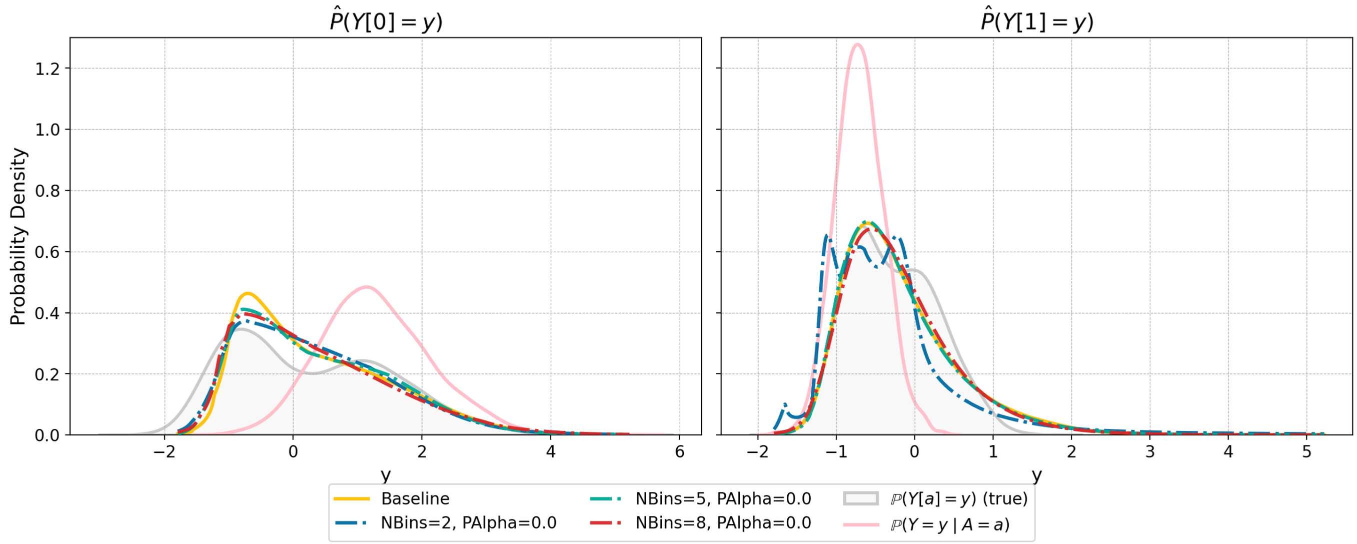

Figure F.3: Estimated interventional distributions for the SyntheticComplex dataset under decreasing values of NBins, with PAlpha set to 0.0.

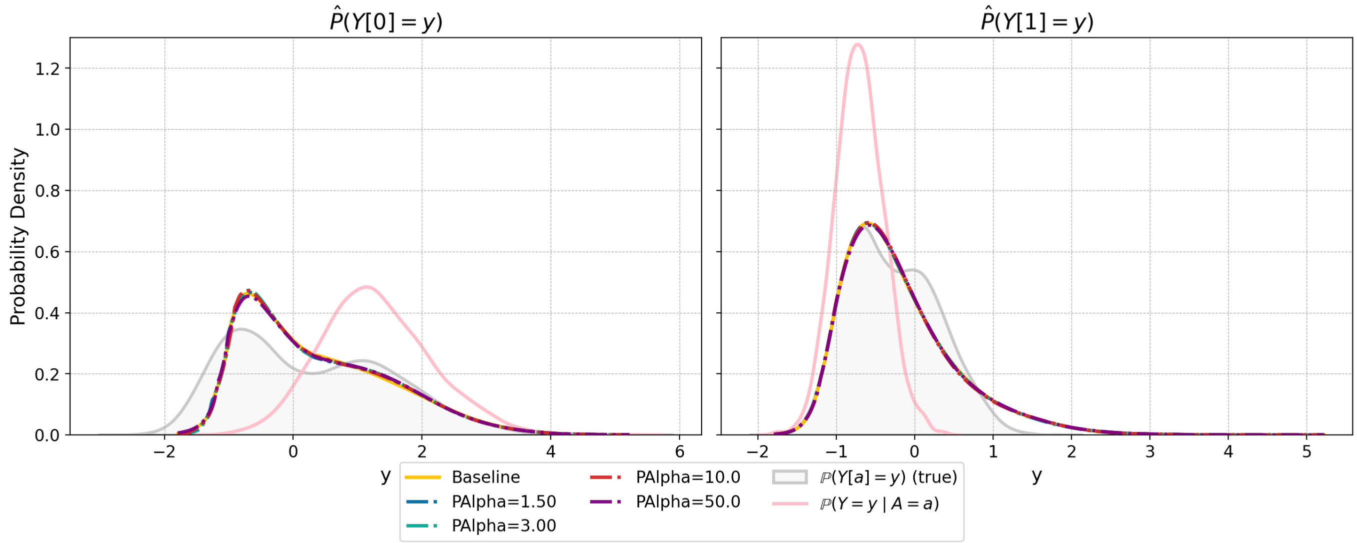

Figure F.4: Estimated interventional distributions for the SyntheticComplex dataset under increasing values of PAlpha.

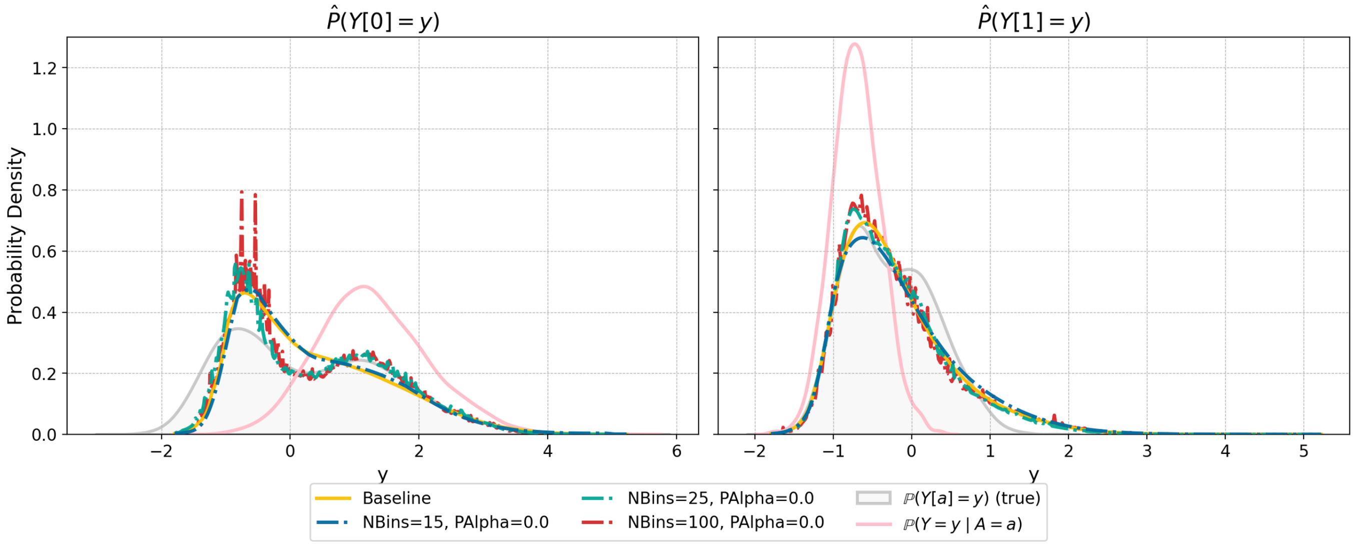

Figure F.5: Estimated interventional distributions for the SyntheticComplex dataset under increasing values of NBins, with PAlpha set to 0.0.

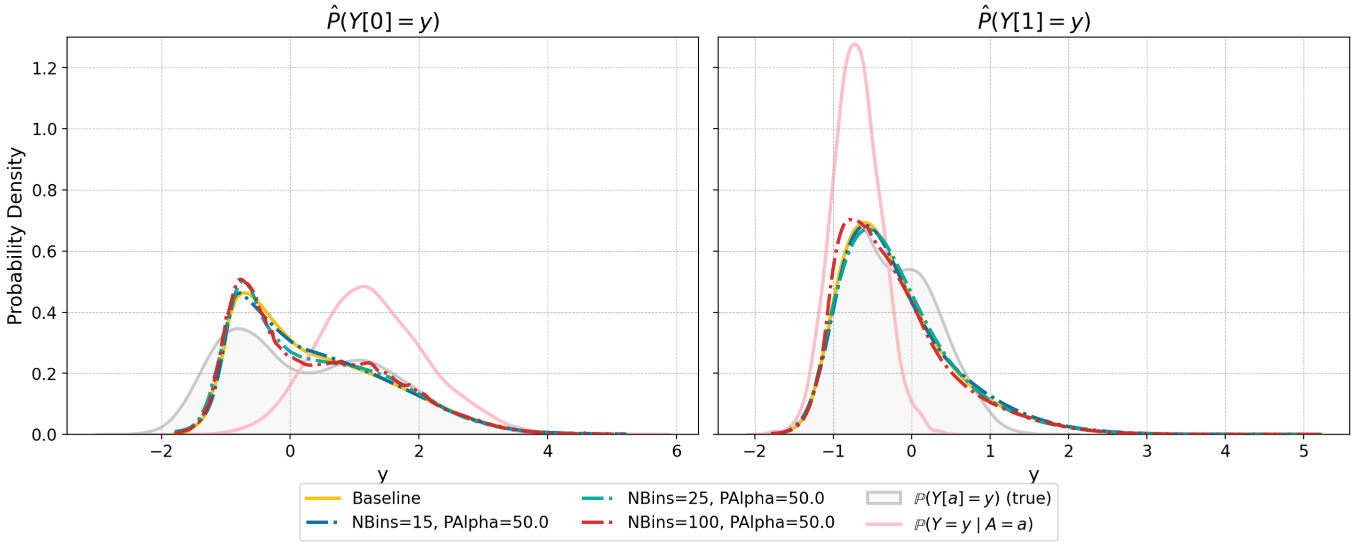

Figure F.6: Estimated interventional distributions for the SyntheticComplex dataset under increasing values of NBins, with PAlpha set to 50.0.

F.2 Controlled Noise Injection Results

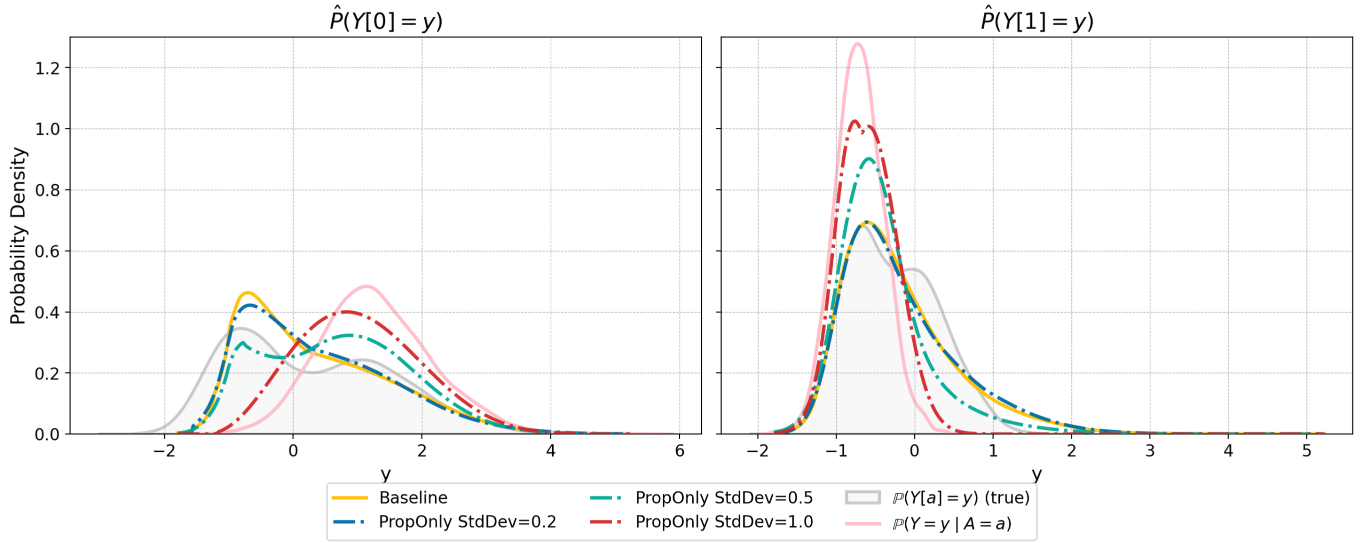

Figure F.7: Estimated interventional distributions for the SyntheticComplex dataset under increasing levels of noise injected into the propensity score estimate.

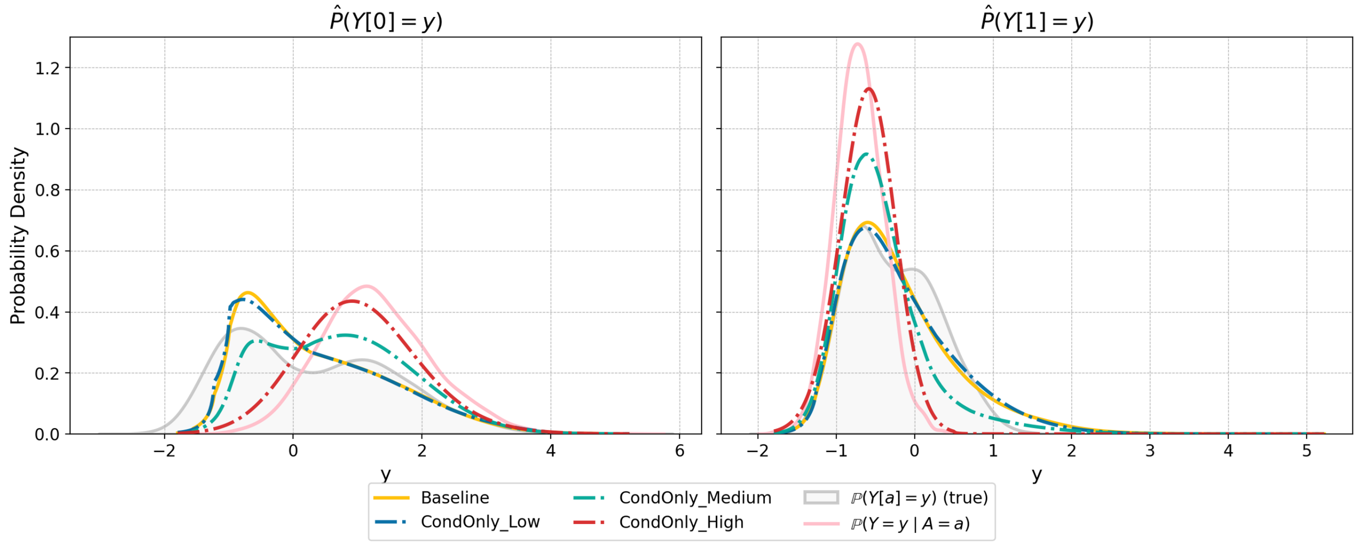

Figure F.8: Estimated interventional distributions for the SyntheticComplex dataset under increasing levels of noise injected into the conditional outcome model estimate.

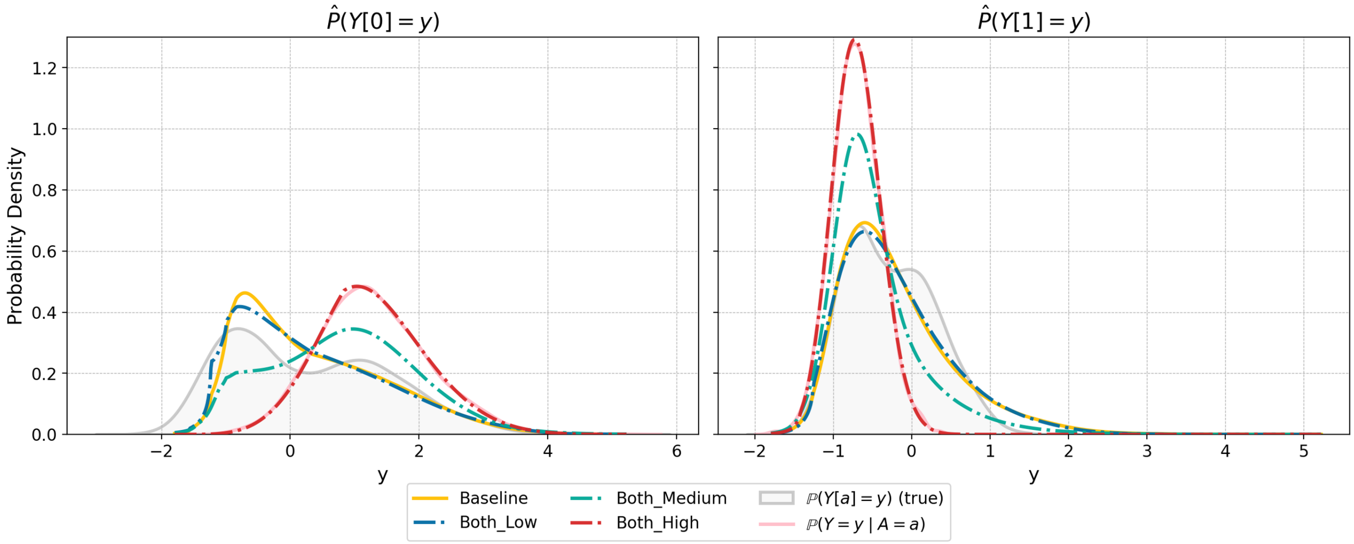

Figure F.9: Estimated interventional distributions for the SyntheticComplex dataset under increasing levels of noise injected into both the propensity score and conditional outcome model estimates.