LEARNING PROBABILITIES OF CAUSATION WITH MASK-AUGMENTED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Probabilities of causation play a central role in modern decision making. Tian and Pearl first introduced formal definitions and derived tight bounds for three binary probabilities of causation (PoCs), such as the probability of necessity and sufficiency (PNS). However, estimating these probabilities requires both experimental and observational distributions specific to each subpopulation, which are often unreliable or impractical to obtain from limited population-level data. To solve this problem, we propose a machine learning (ML) framework that learns from a small set of reliable subpopulations and predicts PNS bounds for all subpopulations. We further introduce a mask-based augmentation strategy that augments the data and enables prediction of masked subpopulation queries. In experiments across five ML models, four Structural Causal Models (SCMs), and four data budgets, mask-augmented neural models consistently achieve the lowest average errors. Mask-Transformer and Mask-MLP obtain average MAEs of 0.0290 and 0.0341 on masked queries, compared with 0.0985 for the plug-in baseline, and also improve exact-query MAE from 0.1866 to 0.0469 and 0.0538. These results demonstrate both the feasibility of ML models for learning PoCs and the effectiveness of mask-based augmentation.

1 INTRODUCTION

Probabilities of causation are crucial in fields such as healthcare, policy evaluation, and economics (Pearl, 2009; Imbens & Rubin, 2015; Heckman & Pinto, 2015). Unlike correlation-based methods, causal inference enables decision-makers to determine whether an action or intervention directly leads to a desired outcome. This is particularly essential in personalized medicine, where accurately assessing causal quantities ensures both efficacy and safety (Mueller & Pearl, 2023). Despite its broad significance, estimating probabilities of causation remains challenging due to data **limitations**. In this paper, we address this challenge by leveraging machine learning techniques to predict probabilities of causation for subpopulations with **insufficient** data.

The study of probabilities of causation began around 2000, when researchers introduced three probabilities of causation including PNS within the framework of SCMs (Pearl, 1999; Galles & Pearl, 1998; Halpern, 2000; Pearl, 2009). In this paper, we focus on PNS, since it is most complex and representative. Specifically, PNS represents the chance that the cause is both required and sufficient for the outcome. PNS is formally defined in Section 3; its practical value lies in helping us make informed, causality based decisions. For example, while taking Tylenol (a fever reducing drug) is correlated with having a fever, we don't give Tylenol because of the correlation. Instead, we give it because of a causal belief (PNS): without the drug, fever would likely persist; with the drug, it would likely subside. Importantly, traditional randomized controlled trials (RCTs), though powerful for estimating average treatment effects, are not always sufficient for answering such counterfactual questions as illustrated in Li & Pearl (2019).

Subsequently, Tian & Pearl (2000) derived tight bounds for these probabilities using Balke's linear programming (Balke, 1995), with both observational and experimental data[1]. Later work formally

[1]Observational data refers to data typically collected through surveys, where individuals can access and freely choose whether to receive a treatment. In contrast, experimental data is usually obtained from randomized controlled trials, where individuals are randomly assigned to receive or not receive the treatment.

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

proved these bounds, extended them to general settings, and showed that they can be further refined given specific causal structures (Li & Pearl, 2019; 2024b; Mueller et al., 2022; Dawid et al., 2017).

Moreover, population-level PNS bounds can obscure substantial heterogeneity across subpopulations. For example, Mueller et al. (2021) report a Tian–Pearl population-level bound of $[0.1, 0.5]$, while the PNS bound for individuals with a particular ancestry is $[0.55, 0.75]$. Such differences can lead totally different decision. Thereby, we need estimation of subgroup-specific PNS bounds, which requires sufficient data for each subpopulation. For the illustrative comparison below, we adopt a support criterion of at least 300 experimental and 300 observational samples per subpopulation (Li et al., 2022). Although computing PNS bounds for a small number of well-sampled subpopulations is feasible, this requirement becomes difficult to satisfy when the query space is large. To illustrate this issue, consider our setting with 10 covariates, 100k observational samples, and 100k experimental samples, where exact and mask queries are formalized in Section 4. If we directly apply plug-in Tian-Pearl bounds, approximately 17% of mask-augmented subpopulations and 46% of exact subpopulations have no *valid* estimates at all after enforcing the per-group sample requirement in *both* datasets; among the remaining exact subpopulations, the MAE ranges from 0.1499 to 0.2076, which is inadequate for practical use.

Hence, in this research, we demonstrate the potential of machine learning models to produce accurate estimates for subpopulations with insufficient data. We first use the Tian-Pearl formulas and the mask-query representation introduced to compute PNS bounds for mask-augmented subpopulations with sufficient data. Then, these bounds are used as training labels for five ML models: support vector machine (SVM) (Cortes, 1995), random forest (RF) (Ho, 1995), gradient boosting decision tree (GBDT) (Friedman, 2001), multilayer perceptron (MLP) (Rumelhart et al., 1986), and Transformer (Vaswani, 2017). These models are used to predict bounds for both unseen exact and mask queries. Besides training models on the mask-augmented query space, we also include exact-only training as an ablation. To assess prediction quality comprehensively, we evaluate our models across four distinct SCMs and report results averaged over five runs.

Contributions We formalize PNS bounds prediction as a supervised learning problem where only a small subset of subpopulations have sufficient experimental and observational data. Therefore, our models can predict the PNS bounds for all subpopulations. We also introduce a mask-augmented representation of subpopulation queries that scales supervision from 2^d to 3^d. And we train five ML models on 3^d mask-augmented subpopulations on both synthetic and real-world dataset. This model not only predicts mask queries, but also achieves better performance on the exact-query prediction.

2 Related Work

Existing research on probabilities of causation (PoC) mainly focuses on how to identify and tighten their bounds when oracle distributions are available. Tian & Pearl (2000) first derived the bounds for PoCs, which are the method used in this paper to generate the training set and also serve as the baseline for our test-set evaluation. Subsequent work has mainly focused on how to make these bounds tighter under specific conditions: Kuroki & Cai (2012) tighten the bounds of probabilities of causation under the assumption of covariate information; Dawid et al. also argued that additional information such as covariates or mediators can refine the bounds; Mueller & Pearl (2023) proved that given a causal structure, the bounds can be further tightened. These studies are complementary to ours: they improve the theoretical bounds of PoC with oracle distributions, whereas we focus on learning PNS bounds from finite data under the long-tail problem.

In addition, there is another line of work that combines PoC-related notions with ML, aiming to improve machine learning. For example, Galhotra et al. (2021) (LEWIS) and Watson et al. (2021) (LENS) use counterfactual, necessity, and sufficiency notions (not exact Pearl PoCs) to highlight influential features and explain black-box model predictions; while Yang et al. (2023) directly use PNS for invariant learning. Although these works also connect PoC-related concepts with ML, their direction is different from ours: they use causal notions to improve or interpret ML models, whereas we use ML to estimate PNS bounds from finite data.

Different from the above works that use counterfactual notions to improve or interpret ML models, there are many works that use ML for causal estimation. Chernozhukov et al. (2024) proposed double/debiased machine learning, which uses ML to estimate high-dimensional nuisance functions

and then applies orthogonalization or debiasing to estimate treatment or structural parameters. Wager & Athey (2017) proposed causal forests, which use random forests to estimate heterogeneous treatment effects. Künzel et al. (2019) proposed meta-learners for estimating CATE. These works mainly target the intervention level in Pearl's causal hierarchy (Pearl, 2009), whereas our work belongs to the counterfactual level. Unlike treatment effect targets which are usually smooth causal quantities, PNS is determined by a set of max/min counterfactual bounds. Therefore, small perturbations in the underlying distributions may cause abrupt, non-smooth changes in the bound. For this reason, treatment-effect estimators cannot serve as direct baselines for our problem

3 PRELIMINARIES

In this section, we review the fundamental concepts of causal inference necessary for understanding the rest of the paper, including the definition of PNS (Pearl, 1999) and its tight bounds (Tian & Pearl, 2000). We adopt the causal language of Structural Causal Models (SCMs) (Galles & Pearl, 1998; Halpern, 2000). In this framework, the counterfactual statement "Variable Y would have the value y had X been x" is denoted as $Y_x = y$, abbreviated as y_x. We consider two types of data: experimental data, expressed as causal effects $P(y_x)$, and observational data, represented by the joint probability function $P(x, y)$. Unless otherwise specified, we assume X and Y are binary variables in a causal model M, with x and y denoting the propositions $X =$ true and $Y =$ true, respectively, and x' and y' representing their complements. For simplicity, we focus on binary treatments, effects, and covariates. Multi-valued covariates can be encoded using multiple binary features, as illustrated in our case study in Subsection 5.4. Extensions to multi-valued treatments and effects are discussed by Pearl (2009) (p. 286, footnote 5) and Li & Pearl (2024a).

First, the definition of PNS, is defined using SCM as follow (Pearl, 1999):

Definition 3.1 (Probability of necessity and sufficiency (PNS))**.** Let X and Y be two binary variables in a causal model M, let x and y stand for the propositions $X = true$ and $Y = true$, respectively, and x' and y' for their complements. The probability of necessity and sufficiency is defined as $\text{PNS} \triangleq P(Y_{X=true} = true, Y_{X=false} = false) \triangleq P(y_x, y'_{x'})$.

Then, the oracle bounds of PNS is given by Tian & Pearl (2000).

$$\max \left\{ \begin{array}{c} 0, \\ P(y_x) - P(y_{x'}), \\ P(y) - P(y_{x'}), \\ P(y_x) - P(y) \end{array} \right\} \leq \text{PNS} \leq \min \left\{ \begin{array}{c} P(y_x), \\ P(y'_{x'}), \\ P(x, y) + P(x', y'), \\ P(y_x) - P(y_{x'}) + P(x, y') + P(x', y) \end{array} \right\}. \quad (1)$$

The advantage of PNS, as illustrated by Li & Pearl (2019), lies in capturing the portion of effects attributable specifically to the treatment, rather than the total causal effect. The primary objective of this paper is then to predict Equation (1) (i.e., the lower and upper bounds of the PNS) for any subpopulations using those with sufficient data (i.e., sufficient data to estimate the distributions $P(X, Y)$ and $P(Y_X)$.) Due to space constraints, the focus will be on the bounds of PNS (i.e., Equation (1)). Unless otherwise specified, the discussion will be limited to binary treatment and effect, meaning both X and Y are binary.

4 MAIN FRAMEWORK

4.1 THE MACHINE LEARNING PIPELINE

Our framework consists of two stages as Figure 1. Firstly, from an SCM we compute oracle PNS bounds via Equation (1) as test label, and derive sample-based bounds from finite experimental/observational data (after threshold filtering) as training labels. We then train predictors under mask-augmented supervision on $q \in \{0, 1, *\}^d$, and evaluate the predictions by using MAE against the oracle bounds, since PNS bounds are naturally defined on the $[0, 1]$ scale.

4.2 CAUSAL DATASET GENERATION

We use synthetic SCMs because exact evaluation of PNS bound requires oracle distribution, which are generally unavailable in real-world datasets. In our experiments, finite samples are used to construct training labels, while the oracle part is only used to accurate evaluation.

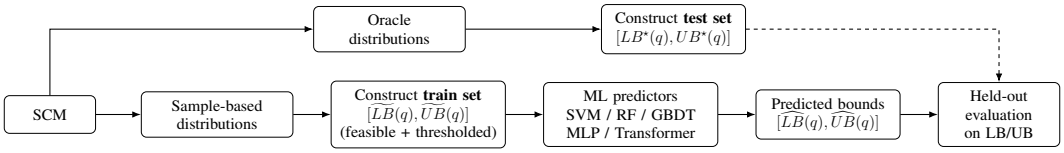
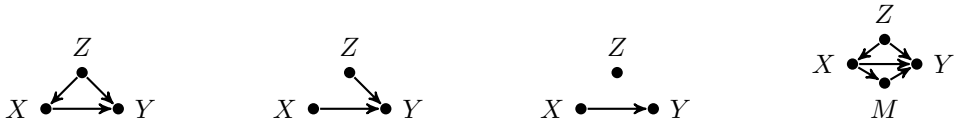


Figure 1: Pipeline for predicting and evaluating PNS bounds under exact and mask queries. Training labels $[\widehat{LB}(q), \widehat{UB}(q)]$ are sample-based bounds estimated from finite data, while oracle test labels $[LB^\star(q), UB^\star(q)]$ are computed by causal equations. Model predictions are denoted by $[\widehat{LB}(q), \widehat{UB}(q)]$.



(a) Confounder Z with direct effect (b) Covariate Z (c) Unrelated Z with Direct effect (d) Mediator M and direct effect

Figure 2: Different SCMs in this study.

SCMs and subpopulation queries. We generate synthetic data from four SCMs shown in Figure 2 (full details are provided in the appendix). Our goal is to predict PNS bounds for both exact and mask queries defined over d observed binary covariates. Exact queries $q \in \{0,1\}^d$ fully specify a covariate profile, whereas mask queries take the form $q \in \{0,1,*\}^d$, where $*$ denotes an unspecified covariate value. For example, $q = (1, *, 0)$ represents the union of the two exact profiles obtained by setting the second entry to 0 or 1. Exact queries are special cases of mask-augmented queries.

Oracle test set. To evaluate model with accurate bounds, we compute *oracle* bounds from SCM by causal equation. For any query q, we compute oracle experimental probability $P(y_x \mid q)$ and $P(y_{x'} \mid q)$ as well as the oracle observational joint distribution $P(x, y \mid q)$ by marginalizing over all hidden variables in the SCM (details in the appendix). These oracle distributions are then plugged into the Tian-Pearl bounds in Eq.(1) to obtain the oracle bounds of PNS for query q, serve as the **test labels**. And a mask-query $q \in \{0,1,*\}^d$ corresponds to a collection of fully specified covariate vectors $z \in \{0,1\}^d$ obtained by filling in the masked entries (those with $*$). Let $\mathcal{Z}(q)$ denote the set of such exact covariate vectors consistent with q, and define the normalizer $\pi(q) := \sum_{z \in \mathcal{Z}(q)} P(z)$. Then, oracle quantities conditioned on q are computed by mixture aggregation over $\mathcal{Z}(q)$:

$$P(y_x \mid q) = \frac{1}{\pi(q)} \sum_{z \in \mathcal{Z}(q)} P(y_x \mid z)\, P(z), P(x, y \mid q) = \frac{1}{\pi(q)} \sum_{z \in \mathcal{Z}(q)} P(x, y \mid z)\, P(z). \quad (2)$$

The same aggregation applies to $P(y_{x'} \mid q)$. Finally, we obtain oracle PNS bounds for q by plugging the aggregated oracle quantities into Eq. (1).

Generate samples for training labels. To emulate realistic data scarcity, we need to generate a finite number of observational and experimental samples from the SCM. For the observational setting, (X, Y) are generated according to the causal equations. And for the experimental setting, X is randomized and the remaining variables are generated by the causal equations. Only observed covariates and outcomes are retained; all hidden variables are unobserved. Note that all uncertainty are caused by those unobserved variables, which means if all variables are observed, the PNS would be an accurate value rather than bounds.

Calculate bounds for training samples Training labels are computed from finite samples by estimating. For each query q, we obtain the experimental distribution $P(y_x \mid q)$ and $P(y_{x'} \mid q)$ from the experimental samples and the observational distribution $P(x, y \mid q)$ from the observational samples. Then we plugging these empirical estimates into Eq. (1). To ensure feasibility, we add a query q into the training set only if it has sufficient support in *both* experimental and observational samples (e.g., at least 300 samples each (Li et al., 2022)) with constraints (e.g., LB $\leq$ UB). Thereby, we can generate the **mask-augmented** training set over $q \in \{0,1,*\}^d$ (used for all five ML models).

Algorithm 1 Mask-ML

Require: SCM M; budgets $(N_{\text{exp}}, N_{\text{obs}})$; threshold grid $\mathcal{T}$; models $\mathcal{A}$.
Ensure: Predictors and errors on $\mathcal{Q}_{\text{mask}}$ and $\mathcal{Q}_{\text{exact}}$.

```
Sample D_exp and D_obs from M.
Compute oracle labels B*(q) = (LB*(q), UB*(q)) for all q ∈ Q_mask using Eq. (1); for masked q, aggregate oracle quantities by Eq. (2).
for τ ∈ T do
    Estimate empirical distributions from D_exp and D_obs.
    Compute sample-based bounds B̃_τ(q) = (L̃B_τ(q), ŨB_τ(q)) for all q ∈ Q_mask using Eq. (1).
    Construct D_τ = {(φ(q), B̃_τ(q)) : q ∈ Q_mask, q passes the support threshold}, φ(q) is in Section 4.3.
    for A ∈ A do
        Split D_τ into training/validation sets.
        Train g_{τ,A} on the training split and compute E^val(τ, A) on the validation split.
    end for
end for
for A ∈ A do
    Select τ*_A = arg min_{τ∈T} E^val(τ, A).
    Predict B̂_A(q) = (L̂B_A(q), ÛB_A(q)) = g_{τ*_A,A}(φ(q)).
    If L̂B_A(q) > ÛB_A(q), replace both by their mean.
    Evaluate B̂_A(q) against B*(q) on Q_mask and Q_exact using MAE/MSE.
end for
Report selected thresholds, predictors, and oracle errors.
```

4.3 Machine Learning

We are aim to study whether ML models with mask augmentation can predict PNS bounds to subpopulation queries with data sparsity. And we formulate it as a supervised regression problem.

Input representation. Each query $q \in \{0, 1, *\}^d$ is encoded by a $3d$-dimensional one-hot vector. For each coordinate $j \in [d]$ and each symbol $c \in \{0, 1, *\}$, we define $\phi_{j,c}(q) = \mathbb{I}[q_j = c]$. Then, we form the feature vector $\phi(q) \in \{0, 1\}^{3d}$ by concatenating these indicators over all (j, c) pairs in a fixed order: $\phi(q) = \text{concat}\big(\{\phi_{j,c}(q)\}_{j\in[d],\, c\in\{0,1,*\}}\big)$.

Predictors. We evaluate five regressors for PNS bounds prediction: SVM for traditional kernel-based regression, RF for robust bagged tree ensembles, GBDT for tree models with gradient, MLP as the classic deep learning model, and Transformer for attention-based modeling of coordinate interactions. It is worth noting that we use the Mish activation function in the MLP, as it performs better than ReLU when handling many values concentrated near zero. In addition, all four models except the Transformer train two bounds separately, while Transformer train them together with joint loss (according to experiment performance). We train each ML model on the enlarged query space $q \in \{0, 1, *\}^d$ (up to 3^d queries after thresholding), and test the models on both exact and mask queries. In addition, we also train them on exact queries and predict exact queries as an ablation test.

Evaluation. We compare the predicted bounds to oracle PNS bounds computed from the SCM and report the average MAEs for both the lower and upper bounds. Since the threshold setting affects the model performance, our models will choose the best threshold from 0 to 2000 based on validation. Thereby, the model with optimal settings on validation set will be compared in test set. After selection, we apply midpoint projection: if $\widehat{LB}(q) > \widehat{UB}(q)$, both endpoints are replaced by $(\widehat{LB}(q) + \widehat{UB}(q))/2$; otherwise they remain unchanged.

4.4 Algorithm

We summarize the procedure in Algorithm 1. For each threshold τ, we compute sample-based bounds $\widetilde{B}_\tau(q)$ using Eq. (1). And a query is included in $\mathcal{D}_\tau$ iff it has enough feasible data. Then, for each ML model, $\mathcal{D}_\tau$ is split into training and validation sets, and the threshold is selected based on validation. Finally, the model with best threshold predicts $\widehat{B}_A(q)$ and is tested based on oracle bounds $B^\star(q)$.

Table 1: Mean MAE on oracle PNS bounds for exact queries $q \in \{0,1\}^d$ over five independent runs. Boldface marks the best learned model; standard deviations are reported in Appendix.

SCM	Budget	Exact queries $q \in \{0,1\}^d$											
		Baseline		MLP		Transformer		RF		GBDT		SVM	
		LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
Direct	50k	0.1719	0.2246	0.0389	0.0359	0.0157	0.0123	0.0117	0.0117	**0.0042**	**0.0042**	0.1117	0.1117
Direct	100k	0.1658	0.2200	0.0340	0.0289	0.0072	0.0059	0.0121	0.0121	**0.0041**	**0.0041**	0.0868	0.0868
Direct	200k	0.1442	0.1635	0.0276	0.0264	0.0066	0.0054	0.0152	0.0152	**0.0050**	**0.0050**	0.0128	0.0128
Direct	500k	0.1323	0.1580	0.0291	0.0304	0.0054	**0.0044**	0.0150	0.0150	**0.0052**	0.0052	0.0119	0.0119
Covariate	50k	0.2211	0.1989	**0.0711**	**0.0938**	0.0719	0.1027	0.0827	0.1174	0.0862	0.1156	0.0755	0.1313
Covariate	100k	0.1996	0.2054	**0.0571**	**0.0910**	0.0658	0.0951	0.0805	0.1120	0.0843	0.1091	0.0741	0.1235
Covariate	200k	0.1884	0.1889	**0.0368**	**0.0502**	0.0369	0.0543	0.0796	0.1148	0.0844	0.1060	0.0655	0.1007
Covariate	500k	0.1611	0.1566	**0.0325**	**0.0367**	0.0347	0.0518	0.0770	0.1100	0.0833	0.1021	0.0630	0.0965
Confounder	50k	0.2214	0.2335	**0.0681**	**0.0892**	0.0744	0.1030	0.0824	0.1171	0.0853	0.1141	0.0761	0.1306
Confounder	100k	0.1964	0.2167	**0.0512**	**0.0812**	0.0652	0.0937	0.0818	0.1074	0.0896	0.1087	0.0759	0.1318
Confounder	200k	0.1737	0.1878	0.0402	0.0627	**0.0362**	**0.0520**	0.0826	0.1106	0.0827	0.1048	0.0710	0.1016
Confounder	500k	0.1652	0.1727	0.0375	**0.0425**	**0.0358**	0.0516	0.0729	0.1062	0.0842	0.1014	0.0648	0.0966
Mediator	50k	0.2314	0.2181	0.0692	0.1141	**0.0540**	**0.1008**	0.0672	0.1423	0.0702	0.1421	0.0973	0.1893
Mediator	100k	0.1863	0.1933	0.0661	0.1176	**0.0321**	**0.0539**	0.0765	0.1544	0.0746	0.1385	0.0957	0.1806
Mediator	200k	0.1855	0.1673	0.0427	**0.0456**	**0.0348**	0.0697	0.0640	0.1510	0.0656	0.1307	0.0857	0.1772
Mediator	500k	0.1590	0.1623	0.0417	**0.0312**	**0.0239**	0.0423	0.0609	0.1246	0.0629	0.1145	0.0851	0.1675

Table 2: Mean MAE on oracle PNS bounds for mask queries $q \in \{0,1,*\}^d$ over five independent runs.Boldface marks the best one; standard deviations are reported in Appendix.

SCM	Budget	Mask queries $q \in \{0,1,*\}^d$											
		Baseline		MLP		Transformer		RF		GBDT		SVM	
		LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
Direct	50k	0.0984	0.1104	0.0247	0.0229	0.0039	**0.0036**	0.0101	0.0101	**0.0038**	0.0038	0.0759	0.0759
Direct	100k	0.0895	0.1135	0.0231	0.0202	0.0035	0.0034	0.0086	0.0086	**0.0027**	**0.0027**	0.0582	0.0582
Direct	200k	0.0741	0.0739	0.0173	0.0167	**0.0014**	**0.0013**	0.0117	0.0117	0.0039	0.0039	0.0106	0.0106
Direct	500k	0.0640	0.0730	0.0176	0.0182	**0.0014**	**0.0013**	0.0111	0.0111	0.0039	0.0039	0.0098	0.0098
Covariate	50k	0.1174	0.1158	**0.0475**	**0.0614**	0.0512	0.0698	0.0608	0.0820	0.0635	0.0799	0.0555	0.0963
Covariate	100k	0.1068	0.1234	**0.0379**	**0.0572**	0.0478	0.0662	0.0548	0.0735	0.0581	0.0738	0.0537	0.0907
Covariate	200k	0.0948	0.0920	0.0213	**0.0285**	**0.0203**	0.0286	0.0524	0.0749	0.0572	0.0702	0.0414	0.0684
Covariate	500k	0.0859	0.0723	0.0186	**0.0209**	**0.0185**	0.0270	0.0491	0.0643	0.0531	0.0640	0.0403	0.0663
Confounder	50k	0.1213	0.1278	**0.0478**	**0.0605**	0.0521	0.0680	0.0610	0.0801	0.0628	0.0779	0.0556	0.0932
Confounder	100k	0.1175	0.1197	**0.0320**	**0.0464**	0.0440	0.0602	0.0572	0.0725	0.0601	0.0724	0.0553	0.0935
Confounder	200k	0.0990	0.0946	0.0232	0.0370	**0.0193**	**0.0289**	0.0523	0.0700	0.0547	0.0686	0.0449	0.0666
Confounder	500k	0.0778	0.0800	0.0216	**0.0225**	**0.0186**	0.0270	0.0460	0.0607	0.0526	0.0614	0.0405	0.0637
Mediator	50k	0.1402	0.1171	0.0477	0.0844	**0.0354**	**0.0707**	0.0459	0.1105	0.0467	0.1108	0.0635	0.1430
Mediator	100k	0.1102	0.1082	0.0420	0.0770	**0.0193**	**0.0330**	0.0439	0.1071	0.0455	0.1048	0.0631	0.1386
Mediator	200k	0.0983	0.0851	0.0270	**0.0261**	**0.0221**	0.0456	0.0398	0.0989	0.0410	0.0942	0.0574	0.1370
Mediator	500k	0.0792	0.0713	0.0242	**0.0171**	**0.0133**	0.0224	0.0363	0.0784	0.0389	0.0819	0.0567	0.1317

5 Experimental Results

5.1 Experimental Setup

We evaluate on four SCMs (Confounder, Covariate, Direct, and Mediator) as in Figure 2. The number of observed covariates d is set to 10. Thereby, we have such two query spaces: (i) exact queries $q \in \{0,1\}^d$ (size $2^{10} = 1024$), with results in table 1; (ii) mask queries $q \in \{0,1,*\}^d$ (size $3^{10} = 59{,}049$), where $*$ denotes an unspecified covariate, with results in table 2.

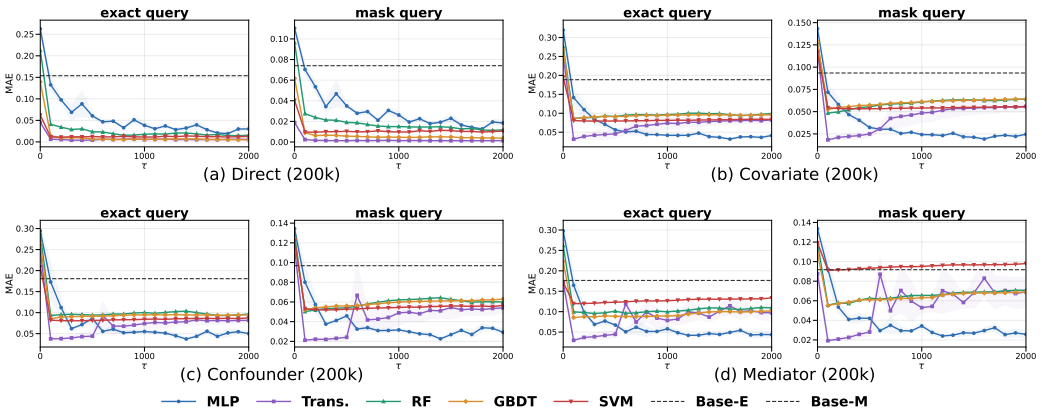


Figure 3: Threshold sweeps at the 200k budget across four SCMs. Each plot contains two panels: evaluation on exact queries (left) and on mask queries (right).

5.2 Overall Accuracy on Exact and Mask Queries

Tables 1 and 2 report the MAE for exact and mask queries, respectively. Across both tasks, ML models outperform the plug-in baseline on average. TTree-based models generally outperform SVM, while neural models perform best overall. For exact queries, MLP, Transformer, and GBDT achieve the best results in 15/32, 10/32, and 7/32 cases, with average MAEs of 0.0538, 0.0469, and 0.0743, respectively. For mask queries, Transformer, MLP, and GBDT are best in 16/32, 13/32, and 3/32 cases, with average MAEs of 0.0290 and 0.0341 for Transformer and MLP, respectively. In addition, because the Direct SCM is relatively simple, we focus on the remaining three SCMs, where either MLP or Transformer performs best. Figure 3 shows the threshold sensitivity of different models. Lower τ yields more training data but noisier labels, while higher τ improves label quality with fewer samples. Transformer tends to prefer lower τ, suggesting greater benefit from larger training sets, while MLP is more stable with higher-quality labels. Overall, both models are competitive: MLP performs better on simpler SCMs and smaller data budgets, whereas Transformer performs better on the more complex Mediator SCM and larger data budgets.

5.3 Ablation Test for Mask

Table 3 reports the mask-training ablation. Across the four SCMs, mask training reduces average exact-query MAE for MLP, Transformer, GBDT, and SVM, while RF increases by 2.63%. Transformer benefits the most from mask training, which enables it to surpass MLP in terms of average MAE. From the perspective of SCMs, since the Direct SCM is simple, mask-based data augmentation brings limited improvement. In contrast, significant improvements are observed on the other three SCMs. From the perspective of data, gains also increase with data budget. Overall, excluding the Direct SCM and RF, mask augmentation reduces the average MAE of the remaining four models by 15.7%–53.2%. These percentages are computed from the ratio of mean MAEs across the three SCMs and four data budgets. This verifies that even without considering its ability to support mask queries, mask augmentation itself is an effective training strategy.

5.4 Case Study: Treat-or-Not for a Toxic but Potentially Life-Saving Drug

Background. We consider a life-threatening disease with an effective but highly toxic drug. Each patient is described by ten observed binary covariates $Z_{1:10}$ (e.g., two bits for age Z_1Z_2, sex Z_3, and comorbidities Z_4–Z_{10}), while additional unobserved factors may also influence outcomes. We observe 200k experimental (RCT) and 200k observational records at the population level. Now consider a new patient with fully specified profile $z^\star$, the decision is whether to take the drug. However, as in 936 of the 1024 exact subpopulations, fewer than 300 patients are observed in this subgroup, making plug-in estimation of PNS bounds unreliable. Let X denote treatment (x = treated, x' = untreated) and Y denote the clinical outcome (y = survived, y' = deceased). Beyond the observed $Z_{1:10}$, a mediator M (e.g., viral load) and further unobserved factors may also affect Y.

Table 3: Mask-training ablation on exact queries $q \in \{0,1\}^d$. Entries report are over five independent runs. Within each model and row, boldface marks the lower MAE between exact and mask training.

SCM	Budget	Exact queries $q \in \{0,1\}^d$									
		MLP		Transformer		RF		GBDT		SVM	
		Exact	Mask	Exact	Mask	Exact	Mask	Exact	Mask	Exact	Mask
Direct	50k	0.0391	**0.0374**	**0.0081**	0.0140	**0.0069**	0.0117	0.0047	**0.0042**	0.2205	**0.1117**
Direct	100k	0.0404	**0.0315**	0.0066	**0.0065**	**0.0093**	0.0121	0.0043	**0.0041**	0.2117	**0.0868**
Direct	200k	0.0325	**0.0270**	0.0069	**0.0060**	**0.0057**	0.0152	**0.0014**	0.0050	0.1756	**0.0128**
Direct	500k	**0.0242**	0.0297	**0.0027**	0.0049	**0.0058**	0.0150	**0.0009**	0.0052	0.1507	**0.0119**
Covariate	50k	0.1004	**0.0825**	0.1322	**0.0873**	**0.0990**	0.1000	0.1261	**0.1009**	0.1253	**0.1034**
Covariate	100k	0.1013	**0.0741**	0.1312	**0.0805**	0.0994	**0.0962**	0.1220	**0.0967**	0.1222	**0.0988**
Covariate	200k	0.0789	**0.0435**	0.1243	**0.0456**	**0.0963**	0.0972	0.1131	**0.0952**	0.1263	**0.0831**
Covariate	500k	0.0716	**0.0346**	0.1225	**0.0432**	**0.0852**	0.0935	0.1034	**0.0927**	0.1089	**0.0797**
Confounder	50k	0.1081	**0.0786**	0.1253	**0.0887**	**0.0976**	0.0998	0.1260	**0.0997**	0.1288	**0.1033**
Confounder	100k	0.1007	**0.0662**	0.1311	**0.0794**	0.0998	**0.0946**	0.1216	**0.0992**	0.1275	**0.1038**
Confounder	200k	0.0835	**0.0514**	0.1216	**0.0441**	**0.0924**	0.0966	0.1165	**0.0938**	0.1152	**0.0863**
Confounder	500k	0.0739	**0.0400**	0.1148	**0.0437**	**0.0855**	0.0896	0.1022	**0.0928**	0.1055	**0.0807**
Mediator	50k	0.1278	**0.0916**	0.1355	**0.0774**	0.1133	**0.1048**	0.1252	**0.1061**	**0.1415**	0.1433
Mediator	100k	0.1303	**0.0918**	0.1379	**0.0430**	**0.1095**	0.1155	0.1248	**0.1065**	**0.1359**	0.1381
Mediator	200k	0.1253	**0.0442**	0.1300	**0.0522**	0.1084	**0.1075**	0.1241	**0.0982**	0.1396	**0.1314**
Mediator	500k	0.0962	**0.0365**	0.1276	**0.0331**	0.0962	**0.0927**	0.1234	**0.0887**	0.1406	**0.1263**

Table 4: Decision at $\theta = 0.5$ for the rare exact group $z =$ `1101001110` and its single-missing variants. Each masked row replaces one coordinate by $*$. N, T, and U denote no treatment, treatment, and uncertain, respectively.

Query	LB/UB				Decision			
	True	Baseline	Exact-MLP	Mask-MLP	True	Baseline	Exact-MLP	Mask-MLP
`1101001110`	0.223/0.275	1.000/1.000	0.353/0.591	0.199/0.289	**N**	T	U	**N**
`*101001110`	0.164/0.199	1.000/1.000	–	0.161/0.225	**N**	T	–	**N**
`1*01001110`	0.278/0.435	0.333/0.333	–	0.268/0.474	**N**	**N**	–	**N**
`11*1001110`	0.140/0.165	0.667/0.667	–	0.110/0.193	**N**	T	–	**N**
`110*001110`	0.308/0.491	0.361/0.444	–	0.294/0.451	**N**	**N**	–	**N**
`1101*01110`	0.324/0.528	0.324/0.532	–	0.382/0.528	**U**	**U**	–	**U**
`11010*1110`	0.170/0.205	0.000/0.000	–	0.170/0.227	**N**	**N**	–	**N**
`110100*110`	0.225/0.278	1.000/1.000	–	0.196/0.292	**N**	T	–	**N**
`1101001*10`	0.250/0.343	1.000/1.000	–	0.218/0.337	**N**	T	–	**N**
`11010011*0`	0.260/0.384	0.400/0.400	–	0.226/0.418	**N**	**N**	–	**N**
`110100111*`	0.222/0.273	1.000/1.000	–	0.196/0.286	**N**	T	–	**N**

Pipeline. In this case, the number of observed covariates is $d = 10$. Therefore, we consider exact queries $q \in \{0,1\}^{10}$ and mask-augmented subpopulations $q \in \{0,1,*\}^{10}$. Following Section 4, we train **Mask-MLP (Mish)** and obtain the predicted bounds $\widehat{\mathrm{LB}}(z^\star)$ and $\widehat{\mathrm{UB}}(z^\star)$. At the same time, we compute the plug-in baseline via Eq. (1) and train Exact-MLP for comparison. Then, we set a decision threshold $\theta = 0.5$ (Treat iff $\mathrm{PNS} \geq 0.5$): we *treat* when $\mathrm{LB} \geq \theta$, *do not treat* when $\mathrm{UB} < \theta$, and mark *uncertain* otherwise. We abbreviate decisions as **T** (treat), **N** (not treat), and **U** (unknown).

Illustrative example under $\theta = 0.5$. Consider the patient with $z =$ `1101001110` in Table 4. According to the oracle bounds ($\mathrm{LB} = 0.2229$, $\mathrm{UB} = 0.2746$), we have $\mathrm{UB} < 0.5$, so the patient should **not be treated**. In contrast, under extreme data sparsity, the plug-in baseline predicts $\mathrm{LB} = \mathrm{UB} = 1.0000$ and recommends **treatment**. As for Exact-MLP, it is uncertain because its interval crosses the threshold ($0.3529 < 0.5 < 0.5908$). Finally, Mask-MLP predicts $[0.1994, 0.2890]$ and correctly recommends **no treatment**, consistent with the oracle. In practice, patient information may be missing or unclear. We therefore construct ten *single-missing* variants by replacing one coordinate of z with $*$. Among them, nine retain the same decision as the exact query, while only one becomes **uncertain**. However, this is not a prediction error: the oracle is also uncertain for that query. This

Table 5: Hillstrom endpoint MAE against empirical reference bounds, averaged over ten training seeds. All-query evaluation includes 10 exact and 90 mask queries.

Model	Exact-only training			Mask-augmented training					
	Exact queries			Exact queries			All queries		
	LB	UB	Avg.	LB	UB	Avg.	LB	UB	Avg.
MLP	0.0218	0.0352	0.0285	0.0145	0.0082	0.0114	0.0093	0.0064	**0.0079**
Transformer	0.0205	0.0441	0.0323	0.0129	0.0098	**0.0114**	0.0090	0.0091	0.0091
RF	0.0198	0.0283	0.0241	0.0116	0.0138	0.0127	0.0074	0.0127	0.0101
GBDT	0.0205	0.0459	0.0332	0.0166	0.0211	0.0189	0.0123	0.0172	0.0148
SVM	0.0244	0.0221	**0.0233**	0.0193	0.0232	0.0213	0.0119	0.0163	0.0141

is the desired pattern: robustness to minor missingness together with uncertainty when the query is intrinsically ambiguous.

5.5 Real-World Evaluation on the Hillstrom Dataset

We evaluate our framework on the Hillstrom randomized email experiment (Hillstrom, 2008), with 64,000 customers assigned to men's email, women's email, or no email. We retain the 42,613 customers in the men's-email and no-email groups, defining X as assignment to men's email and Y as a website visit within the following two weeks.

The dataset contains only experimental data, which alone can provide valid PNS bounds (Li & Pearl, 2023). We therefore use the experimental-data PNS bounds of Tian & Pearl (2000). The empirical reference endpoints are $\widetilde{LB}(q) = \max\{0, \widehat{p}_1(q) - \widehat{p}_0(q)\}$ and $\widetilde{UB}(q) = \min\{\widehat{p}_1(q), 1 - \widehat{p}_0(q)\}$, where $\widehat{p}_x(q)$ is the empirical visit rate in treatment arm x for query q. Since oracle bounds are unavailable, we evaluate predictions against these endpoints for well-supported groups. We encode six pretreatment covariates as binary variables, yielding 621 nonempty mask-augmented queries, including 40 exact queries. We reserve 10 exact and 90 masked queries for evaluation, each containing over 1,000 customers, and split the remaining queries into training and validation sets. Support thresholds are selected by validation performance separately for each model and training regime. Test queries remain fixed across all models and ten training seeds. Additional details are in Appendix B.3.

Table 5 summarizes the results. Under this validation-based protocol, mask augmentation reduces mean exact-query MAE for all five models. Averaged over the two endpoints, the reductions are 60.3% for MLP and 64.8% for Transformer. MLP has the lowest mean MAE over all 100 test queries, while MLP and Transformer perform similarly on the ten exact queries. These results provide supplementary evidence that mask augmentation improves prediction for held-out queries in real data. These errors measure agreement with empirical rather than oracle bounds.

6 Conclusion

Our experiments show that ML models can learn PNS bounds well from insufficient data. Specifically, on exact queries, MLP and Transformer achieve average MAEs of 0.0538 and 0.0469, versus 0.1866 for the plug-in baseline; on mask queries, their MAEs are 0.0341 and 0.0290, versus 0.0985. In addition, ablation test indicates that mask augmentation further reduces MAE by 38.65% for MLP and 53.17% for Transformer (excluding the Direct SCM). After augmentation, Transformer achieves the lowest MAE, suggesting greater benefit from the expanded training data. Finally, we also demonstrate how our models work in the case study and Hillstrom dataset.

However, our work still has limitations. Our simulations use oracle bounds for every subgroup, whereas the Hillstrom evaluation relies on empirical bounds for well-supported groups. Restricting evaluation to such data rich groups may bias the results toward easier cases and may not fully reflect the difficulty of predicting long-tail subpopulations. A principled alternative is an external audit: randomly sample test groups from the target query distribution and independently collect observational and experimental data for them. Because such an audit is costly, we leave it to future work, along with evaluations on more complex SCMs, richer covariates, and smaller data budgets.

7 STATEMENT

AI USE STATEMENT

In this work, we used generative AI tools for [Grammar polishing, adaptation to real-world datasets, and table generation]. We have not used generative AI tools for [Grammar polishing, adaptation to real-world datasets, and table generation]. We have reviewed all AI-assisted work. [We checked the citations for the LLM adoption methods, as well as the associated code.]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops methods for estimating bounds on probabilities of causation for subpopulation queries. Our synthetic experiments use simulated data and involve no human participants. The real-world experiment uses the publicly available Hillstrom email-marketing dataset; we did not collect new data, conduct new interventions, or use direct personal identifiers.

REPRODUCIBILITY STATEMENT

Implementation details and source code are provided in the Appendix.

REFERENCES

Alexander Abraham Balke. *Probabilistic counterfactuals: semantics, computation, and applications*. University of California, Los Angeles, 1995.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and causal parameters, 2024. URL https://arxiv.org/abs/1608.00060.

Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

Philip Dawid, Monica Musio, and Rossella Murtas. The probability of causation. *Law, Probability and Risk*, 16:163–179, 2017.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Sainyam Galhotra, Romila Pradhan, and Babak Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. *CoRR*, abs/2103.11972, 2021.

David Galles and Judea Pearl. An axiomatic characterization of causal counterfactuals. *Foundations of Science*, 3(1):151–182, 1998.

Joseph Y Halpern. Axiomatizing causal reasoning. *Journal of Artificial Intelligence Research*, 12: 317–337, 2000.

James Heckman and Rodrigo Pinto. Causal analysis after haavelmo. *Econometric Theory*, 31(1): 115–151, 2015.

Kevin Hillstrom. The minethatdata e-mail analytics and data mining challenge. MineThatData, March 2008. URL https://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html. Accessed: 2026-09-26.

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pp. 278–282. IEEE, 1995.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

Manabu Kuroki and Zhihong Cai. Stratified analysis of 'probabilities of causation', 2012. URL https://arxiv.org/abs/1206.6861.

Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, February 2019. ISSN 1091-6490. doi: 10.1073/pnas.1804597116.

Ang Li and Judea Pearl. Unit selection based on counterfactual logic. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

Ang Li and Judea Pearl. Probabilities of causation: Role of observational data. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10012–10027. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/li23d.html.

Ang Li and Judea Pearl. Probabilities of causation with nonbinary treatment and effect. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 20465–20472, 2024a.

Ang Li and Judea Pearl. Unit selection with nonbinary treatment and effect. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(18), pp. 20473–20480, 2024b.

Ang Li, Ruirui Mao, and Judea Pearl. Probabilities of causation: Adequate size of experimental and observational samples. In *NeurIPS 2022 Workshop on Neuro Causal and Symbolic AI (nCSI)*, 2022.

Scott Mueller and Judea Pearl. Perspective on 'harm' in personalized medicine – an alternative perspective. Technical Report R-530, Department of Computer Science, University of California, Los Angeles, CA, 2023. Forthcoming, American Journal of Epidemiology.

Scott Mueller, Ang Li, and Judea Pearl. Causes of effects: Learning individual responses from population data, 2021. URL https://arxiv.org/abs/2104.13730.

Scott Mueller, Ang Li, and Judea Pearl. Causes of effects: Learning individual responses from population data. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, pp. 2712–2718, 2022.

Judea Pearl. Probabilities of causation: Three counterfactual interpretations and their identification. *Synthese*, pp. 93–149, 1999.

Judea Pearl. *Causality*. Cambridge university press, 2nd edition, 2009.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0.

Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1-4):287–313, 2000.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests, 2017. URL https://arxiv.org/abs/1510.04342.

David Watson, Limor Gultchin, Ankur Taly, and Luciano Floridi. Local explanations via necessity and sufficiency: Unifying theory and practice, 2021. URL https://arxiv.org/abs/2103.14651.

Mengyue Yang, Zhen Fang, Yonggang Zhang, Yali Du, Furui Liu, Jean-Francois Ton, Jianhong Wang, and Jun Wang. Invariant learning via probability of sufficient and necessary causes. *Advances in Neural Information Processing Systems*, 36:79832–79857, 2023.