AMORTIZING CAUSAL SENSITIVITY ANALYSIS VIA PRIOR DATA-FITTED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal sensitivity analysis aims to provide bounds for causal effect estimates in the presence of unobserved confounding. However, existing methods for causal sensitivity analysis are per-instance procedures, meaning that changes to the dataset, causal query, sensitivity level, or treatment require new computation. Here, we instead present an in-context learning approach. Specifically, we propose an amortized approach to causal sensitivity analysis based on prior-data fitted networks. A key challenge is that the sensitivity bounds are not directly available when sampling training data. To address this, we develop a general prior-data construction that is applicable across the class of generalized treatment sensitivity models. Our construction involves a Lagrangian scalarization of the objective to generate training labels for the bounds through a tradeoff between causal effect min/maximization and sensitivity model violation, which avoids model-specific analytical derivations. We further show that, under standard convexity and linearity conditions, our objective recovers the full Pareto frontier of solutions. Empirically, we demonstrate our amortized approach across various datasets, causal queries, and sensitivity levels, where our approach achieves a test-time computation that is orders of magnitude faster than per-instance methods. To the best of our knowledge, ours is the first foundation model for in-context learning for causal sensitivity analysis.

1 INTRODUCTION

Causal inference from observational data is widely used in fields such as medicine (Feuerriegel et al., 2024; Skapetze et al., 2025), public policy (Kuzmanovic et al., 2024), and the social sciences (Bär et al., 2025), but relies on untestable assumptions about the presence and strength of unobserved confounding (Pearl, 2013). ***Causal sensitivity analysis*** addresses this limitation by replacing point estimates with bounds on causal effects in the presence of unobserved confounding (Manski, 1989). In practice, such bounds can be sufficient for decision-making—for example, to assess when conclusions remain unchanged across a range of plausible confounding strengths.

Formally, causal sensitivity analysis computes lower and upper bounds (i.e., θ^- and θ^+) on a causal estimand as functions of a sensitivity level Γ. A wide range of methods exist for estimating these bounds (Dorn & Guo, 2023; Frauen et al., 2023; 2024; Jesson et al., 2021; 2022; Jin et al., 2023; 2026; Kallus et al., 2019; Manski, 1989; Marmarelis et al., 2023; 2024; Oprescu et al., 2023; Rosenbaum, 1987; Tan, 2006; Yin et al., 2024; Zhao et al., 2019), but these methods are inherently *per-instance* computations. This means that each change to the dataset, causal query, treatment arm, or sensitivity level Γ requires a new computation. The need to recompute the bounds *per-instance* often limits practical use. As a result, sensitivity analysis is frequently applied as a final check, rather than as a systematic tool to comprehensively understand how the conclusions vary across assumptions.

In this paper, we instead propose an *in-context learning* approach to causal sensitivity analysis. Our central idea is to follow an amortized approach and replace the per-instance computation with a learned predictor that, given a dataset, causal query, and sensitivity level, directly returns the corresponding sensitivity bounds. In principle, a natural framework for such amortization is through prior-data fitted networks (Müller et al., 2022); however, this is *non-trivial*. In point-identified settings, prior-data construction is straightforward: one samples from a data-generating process, which directly provides the causal effect that can be used as a label. In contrast, causal sensitivity analysis requires lower and upper bounds under a sensitivity model. Importantly, these bounds are

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.

not directly available from the data-generating process and must themselves be computed. As a result, the key challenge is therefore *how to construct such prior-data labels in a tractable manner for a broad range of sensitivity models.*

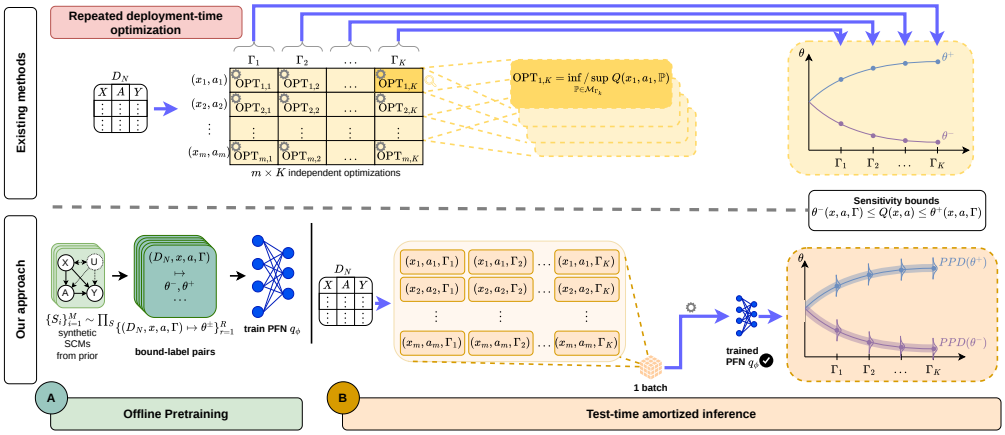


Figure 1: **Per-instance optimization vs. amortization.** Existing methods (top row) perform *per-instance* optimization: for each input query (x_i, a_i) and each level of sensitivity Γ_k, a new optimization must be instantiated. The sensitivity bound curves (right) are constructed across $m \times K$ optimizations. Our approach (bottom row) amortizes: Expensive pretraining is done once *offline* (step A). Once trained, the PFN processes all input queries in a single *batched* forward pass at trivial cost (step B).

In this paper, we introduce a general prior-data construction for obtaining labels wrt. sensitivity bounds. Our construction is broadly applicable and can be used together with the class of generalized treatment sensitivity models (GTSMs) (Frauen et al., 2024), which includes the marginal sensitivity model (MSM) (Tan, 2006), f-sensitivity models (Jin et al., 2026), and Rosenbaum's sensitivity model (Rosenbaum, 1987). To derive our construction, we formulate the problem of generating sensitivity bounds as a multi-criteria optimization problem over latent distribution shifts to tradeoff between causal effect min/max-imization and sensitivity model violation. By varying the tradeoff, we trace the Pareto frontier of bound values along different sensitivity levels and thereby yield the lower and upper bounds needed for prior-data training. Our approach does not require closed-form analytical solutions and thus applies to all sensitivity models from the GTSM class. We use the construction to train a prior-data fitted foundation model that allows us to predict sensitivity bounds based on new datasets, causal queries, and sensitivity levels.

Finally, we use the resulting labels to train a prior-data fitted *foundation model*. After training, inference reduces to a single forward pass: given a dataset, causal query, and sensitivity level, the foundation model simply predicts the lower and upper sensitivity bounds directly from data without recomputing them from scratch. This enables fast evaluation across many causal queries and assumptions. Because our prior-data construction is not tied to a specific sensitivity model, the approach applies broadly to the class of GTSMs. Finally, the foundation model outputs full posterior predictive distributions for the lower and upper bounds, thereby providing uncertainty-aware estimates rather than only point predictions. To the best of our knowledge, this is the first foundation model for causal sensitivity analysis.

Our **contributions** are three-fold:[1]

(1) **Prior-data label construction.** We introduce a general construction scheme for generating training labels with sensitivity bounds. The construction involves a Lagrangian scalarization of the objective, which we show theoretically to recover the full Pareto frontier of solutions.

(2) **A foundation model for causal sensitivity analysis.** We train a prior-data fitted foundation model to offer a new in-context learning approach for causal sensitivity analysis. Our foundation model directly predicts the sensitivity bounds given a dataset, causal query, and sensitivity level.

(3) **Empirical demonstration.** We empirically demonstrate that our foundation model can approximate sensitivity bounds across a broad range of settings within a single forward pass *and* at state-of-the-art performance.

[1]Code available at: `https://anonymous.4open.science/r/CSA-PFN-Anon-FA52/`. Upon acceptance, we move the code to a public GitHub.

2 Related Work

Causal sensitivity analysis: Causal sensitivity analysis aims at partial identification of causal queries under a sensitivity model. Several sensitivity models have been developed for this purpose, such as the marginal sensitivity model (MSM) (Tan, 2006), the f-sensitivity models (Jin et al., 2026), and Rosenbaum's sensitivity model (Rosenbaum, 1987). Recently, Frauen et al. (Frauen et al., 2023; 2024) proposed a unified approach in the form of the generalized treatment sensitivity model (GTSM), which subsumes all three aforementioned sensitivity models.

Existing approaches to estimating sensitivity bounds fall into two broad categories. On the one hand, closed-form sharp bounds are available, but only for a small number of highly structured settings (e.g., the MSM) (Dorn & Guo, 2023; Frauen et al., 2023). However, this applies only to the MSM and not, for example, the broader set of sensitivity models in the GTSM. On the other hand, GTSMs typically require a two-stage neural density estimation procedure (Frauen et al., 2024), and are thus computationally expensive. Importantly, all machine learning approaches to estimating sensitivity bounds operate under a *per-instance* approach, meaning that changes to the dataset, causal query, sensitivity level, or treatment require new computations. *As a result, causal sensitivity analysis is computationally complex at deployment, and an approach for in-context learning is still missing.*

Prior data-fitted networks: Prior-data fitted networks (PFNs) (Müller et al., 2022) are foundation models that amortize Bayesian posterior predictive inference via in-context learning. Hence, PFNs are trained on synthetic data sampled from a prespecified prior. TabPFN (Grinsztajn et al., 2026; Hollmann et al., 2023; 2025) demonstrated this paradigm at scale by combining transformers with a Bayesian neural network prior over structural causal models, thereby achieving state-of-the-art performance on tabular prediction. Subsequent work has extended PFNs to other modalities (e.g., such as time-series forecasting (Hoo et al., 2025)) and provided theoretical insights into the in-context learning behavior theoretically (Melnychuk et al., 2026; Nagler, 2023). More broadly, PFNs offer a general approach for amortized inference across tasks that would traditionally be solved on a *per-instance* basis.

Foundation models for causal inference: Only recently, first works have applied PFN-style amortization to causal inference (Balazadeh et al., 2025; Bynum et al., 2025; Ma et al., 2025; Robertson et al., 2025). The central methodological challenge in this line of work is the design of priors over data-generating processes that ensure that the learned model performs valid *causal* inference at deployment, rather than merely interpolating observational patterns. For example, Ma et al. (2025) propose a framework for constructing priors based on causal structural models. However, all of the existing approaches focus on causal inference using *point identification*, where all confounders are *observed*.

In contrast, causal sensitivity analysis focuses on settings where some confounders are *unobserved* and thus aims at *partial identification* by bounding the causal estimand. This makes the problem fundamentally harder: unlike point-identified causal effects, sensitivity bounds are not available as direct functionals of a sampled structural causal model and must themselves be computed via non-trivial optimization procedures. This also makes the development of foundation models for this task substantially more challenging. *To the best of our knowledge, no prior work has developed a foundation model for causal sensitivity analysis; we provide the first principled approach for this.*

3 Mathematical Background

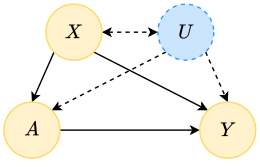


Figure 2: Causal graph. Observed variables are orange and unobserved confounding is blue.

Notation: We write random variables in uppercase (e.g., X) and their realizations in lowercase (e.g., x). We write $\mathbb{P}$ for a probability distribution, with $\mathbb{P}(x)$ denoting the probability mass/density function if X is discrete/continuous. Conditional probability mass/density functions are written as $\mathbb{P}(Y = y \mid X = x)$, or shorthand $\mathbb{P}(y \mid x)$, while a conditional distribution $Y|X = x$ is written as $\mathbb{P}(Y \mid x)$. Expectations are written as $\mathbb{E}_{\mathbb{P}}[\cdot]$, with the subscript omitted when the underlying distribution is clear from the context. We denote by $f_\sharp\mathbb{P}$ the push-forward of $\mathbb{P}$ through $f : \mathcal{A} \mapsto \mathcal{B}$, defined by $(f_\sharp\mathbb{P})(B) = \mathbb{P}(a : f(a) \in B)$ for measurable sets $B \in \mathcal{B}$.

3.1 PROBLEM SETUP

Setting: We consider a standard causal inference setting with backdoor adjustment (Dorn & Guo, 2023), with observed covariates $X \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, an unobserved confounder $U \in \mathcal{U} \subseteq \mathbb{R}$, a binary treatment $A \in \mathcal{A} = \{0, 1\}$, and an outcome $Y \in \mathcal{Y} \subseteq \mathbb{R}$ (see Fig. 2). The data-generating process is formalized as a structural causal model (SCM) $\mathcal{S} = (X, U, A, Y, f, \mathbb{P}_U, \mathbb{P}_\varepsilon)$, where $f = (f_X, f_A, f_Y)$ denotes the structural assignments and $\mathbb{P}_U, \mathbb{P}_\varepsilon$ the distributions over unobserved confounding U and exogenous noise variables $\varepsilon = (\varepsilon_X, \varepsilon_A, \varepsilon_Y)$ (Pearl, 2013). The SCM induces an observational distribution $\mathbb{P}^{\mathcal{S}}_{\text{obs}}$ over (X, A, Y), from which we observe an i.i.d. dataset $D_n = (x_i, a_i, y_i)_{i=1}^n \sim \mathbb{P}^{\mathcal{S}}_{\text{obs}}$. The interventional distributions $\mathbb{P}^{\mathcal{S}}_{\text{int}}$ are obtained by replacing the treatment assignment mechanism with the intervention $\text{do}(A = a)$ (Pearl, 2013). We denote the potential outcome under this intervention by $Y(a)$ (Rubin, 1974). Throughout, the true SCM $\mathcal{S}^\star$ is *unknown*; only samples from $\mathbb{P}^{\mathcal{S}^\star}_{\text{obs}}$ are observed.

Causal query: For a data-generating process $\mathcal{S}$, we are interested in estimating a causal query Q of the form $\mathcal{Q}\left(\mathbb{P}^{\mathcal{S}}_{\text{int}}(Y(a) \mid x)\right)$, where $\mathcal{Q}$ maps the conditional potential outcomes distribution to $\mathcal{Y}$. In this work, we focus on the conditional average potential outcome (CAPO), i.e., $Q(x, a, \mathbb{P}) = \mathbb{E}_{\mathbb{P}}[Y(a) \mid x]$, as a primitive causal estimand. Other common estimands, such as the average treatment effect (ATE) and conditional average treatment effects (CATE), can be obtained by simple (post-inference) transformations.[2].

Assumption 1. *We work under the standard causal assumption (Dorn & Guo, 2023): (i) consistency: $A = a \Rightarrow Y(a) = Y$, $\forall a$, (ii) positivity: $\mathbb{P}(a|x) > 0$, $\forall (x, a)$, and (iii) latent unconfoundedness: $Y(a) \perp\!\!\!\perp A \mid X, U$, $\forall a$.*

Partial identification: Because U is unobserved, the causal query $Q(x, a, \mathbb{P})$ is generally **not** point-identified from the observed data distribution $\mathbb{P}^{\mathcal{S}^\star}_{\text{obs}}$. In particular, there may exist different data-generating processes $\mathcal{S} \neq \mathcal{S}^\star$ that induce the *same* observational distribution $\mathbb{P}^{\mathcal{S}}_{\text{obs}} = \mathbb{P}^{\mathcal{S}^\star}_{\text{obs}}$ over (X, A, Y) but imply different causal effects $Q(\cdot, \mathbb{P}) \neq Q(\cdot, \mathbb{P}^\star)$. As a result, the causal query is only partially identified in the sense that we can only infer a set of causal effects compatible with the observed data distribution and the underlying assumptions (Manski, 1989).

3.2 CAUSAL SENSITIVITY ANALYSIS

Causal sensitivity analysis introduces a structured way to characterize the set of data-generating processes compatible with the observed data by parameterizing the strength of unobserved confounding through a sensitivity model.

Definition 1. *A sensitivity model $\mathcal{M}_\Gamma$ is a family of distributions $\mathbb{P}$ over (X, U, A, Y) parameterized by a sensitivity level $\Gamma \in \mathbb{R}_{\geq \Gamma_{\min} \geq 0}$, such that $\int_{\mathcal{U}} \mathbb{P}(x, u, a, y) \mathrm{d}u \equiv \mathbb{P}_{\text{obs}} = \mathbb{P}^\star_{\text{obs}}$ for all $\mathbb{P} \in \mathcal{M}_\Gamma$.*

In this work, we focus on the so-called *generalized treatment sensitivity model (GTSM)*, which is a broad class of sensitivity models that constrain the latent distribution shift induced by treatment intervention (Frauen et al., 2024). Under the GTSM, the admissible full distributions satisfy constraints of the form

$$\Delta_{x,a}\left(\mathbb{P}(U \mid x), \mathbb{P}(U \mid x, a)\right) \leq \Gamma, \ \forall (x, a) \tag{1}$$

where $\Delta_{x,a}$ is a divergence functional measuring the discrepancy between the latent distribution under intervention (i.e., $\mathbb{P}(U \mid x)$)[3] and the latent distribution observed under treatment assignment (i.e., $\mathbb{P}(U \mid x, a)$). The GTSM subsumes several widely used sensitivity models, including the marginal sensitivity models (Tan, 2006), the f-sensitivity models (Jin et al., 2026), and Rosenbaum's sensitivity model (Rosenbaum, 1987), depending on the choice of divergence $\Delta_{x,a}$ (see Appendix B.6 for details).

[2]For the ATE, one marginalizes over covariates the difference $\mathbb{E}_X[Q(x, 1, \mathbb{P}) - Q(x, 0, \mathbb{P})]$. Similarly, the CATE is given the difference $Q(x, 1, \mathbb{P}) - Q(x, 0, \mathbb{P})$.

[3]We recognize that $\mathbb{P}(U \mid x, \text{do}(A = a)) = \mathbb{P}(U \mid x)$ since $A \to U$ is the anti-causal direction. Unobserved confounding U *causes* treatment A (rather than the other way) and so cannot be influenced by an intervention on the treatment. Rather, the intervention removes the dependence between A and U.

Task: For a given sensitivity model $\mathcal{M}_\Gamma$, the aim of causal sensitivity analysis is to obtain the sensitivity bounds for the causal query:

$$\theta^-(x,a,\Gamma) = \inf_{\mathbb{P}\in\mathcal{M}_\Gamma} Q(x,a,\mathbb{P}), \qquad \theta^+(x,a,\Gamma) = \sup_{\mathbb{P}\in\mathcal{M}_\Gamma} Q(x,a,\mathbb{P}). \tag{2}$$

By definition, the interval $[\theta^-(x,a,\Gamma), \theta^+(x,a,\Gamma)]$ is the tightest interval of causal query values compatible with the observed distribution $\mathbb{P}^\star_{\text{obs}}$ and the sensitivity level Γ. In other words,

$$Q^\star \in [\theta^-(x,a,\Gamma), \theta^+(x,a,\Gamma)] \tag{3}$$

is the maximum information about $Q^\star$ that we can extract from D_n as $n \to \infty$, given $\mathbb{P}^\star \in \mathcal{M}_\Gamma$. The parameter Γ controls the width of the interval; when no unobserved confounding is permitted, the causal query is point identified.

For ease of reading, we omit the sensitivity model $\mathcal{M}_\Gamma$ from notation during the rest of the paper, referring to an arbitrary sensitivity model from GTSM, unless stated otherwise.

3.3 Background on PFNs

Prior-data fitted networks (PFNs) are neural foundation models trained to amortize posterior predictive inference across data-generating processes (Müller et al., 2022). Given a dataset $D_N \sim \mathbb{P}$, a trained PFN predicts the target distribution $\mathbb{P}(Y \mid X)$ without task-specific gradient updates or retraining (i.e., in-context). Training proceeds by sampling data-generating processes over a prior Π and minimizing the negative log-likelihood

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbb{P}\sim\Pi_\mathcal{S}} \mathbb{E}_{N\sim\Pi_N} \mathbb{E}_{D_N\sim\mathbb{P}^N} \mathbb{E}_{(X,Y)\sim\mathbb{P}} \left[-\log q_\theta(Y \mid X, D_N)\right]. \tag{4}$$

This yields the predictor $q_\theta(Y \mid X, D_N)$ approximating the posterior predictive distribution (PPD)

$$\Pi(Y \mid X, D_N) = \int \mathbb{P}(Y \mid X)\Pi(\mathbb{P} \mid D_N)\, \mathrm{d}\mathbb{P}. \tag{5}$$

Because the prior-data distribution can be entirely synthetic, PFNs enable foundation-model-style pretraining for broad classes of tabular prediction problems.

4 Amortized Causal Sensitivity Analysis with PFNs

Overview: We develop a PFN-based approach for amortized causal sensitivity analysis, with the aim of replacing per-instance optimization with a foundation model that predicts sensitivity bounds in a single forward pass for new datasets, causal queries, and sensitivity levels. We first formalize this amortized prediction task and describe the **high-level learning setup** in Section 4.1, and the **synthetic data generation via SCM priors** in Section 4.2. The central difficulty, however, is label construction: sensitivity bounds lack closed-form expressions in general, and existing solutions are not built for PFN-scale training label generation. To overcome this bottleneck, Section 5 introduces our core technical contribution: a general **prior-label construction procedure** based on Lagrangian scalarization, which allows us to efficiently generate sensitivity bound labels across sampled SCMs for training PFNs.

4.1 Task formulation

PFN: To adapt PFNs to causal sensitivity analysis under a fixed sensitivity model $\mathcal{M}$, we train a PFN to learn the mapping

$$D_N, x, a, \Gamma \longrightarrow \mathrm{PPD}(\theta^-), \mathrm{PPD}(\theta^+), \tag{6}$$

where $\mathrm{PPD}(\cdot)$ denotes the posterior predictive distribution induced by the prior over data-generating processes In other words, for a dataset D_N, at a specific query (x, a), and under unobserved confounding constrained by $\Delta_{x,a} \leq \Gamma$ according to $\mathcal{M}$, we learn the PPD over the lower and upper bounds θ^- and θ^+ of the causal query $Q(x,a) = \mathbb{E}_\mathbb{P}[Y(a) \mid x]$.

Training data: During training, we sample a large, diverse collection of data-generating processes $\{\mathcal{S}_i\}_{i=1}^{n_i} \sim \Pi_\mathcal{S}$, each of which generates a set of input tuple (D_N, x, a, Γ) and output label (θ^-, θ^+)

supervised training pairs $\{(D_N, x, a, \Gamma)_j$, i.e., $(\theta^-, \theta^+)_j\}_{j=1}^{n_j}$. By constructing a sufficiently rich prior-data distribution, the PFN trained on $n_i \times n_j$ such pairs is able to amortize sensitivity analysis across datasets, causal queries, and sensitivity levels.[4]

An alternative to our formulation is to consider for output a PPD on the causal query Q *directly* and to obtain bounds by taking a credible interval. While technically feasible, it is fundamentally misaligned with the nature of the setting: since the true causal query $Q^\star$ is only partially identified under the sensitivity model, the maximum information about $Q^\star$ that we can extract from the data D_n as $n \to \infty$ are the bounds θ^-, θ^+, i.e. $Q^\star \in [\theta^-, \theta^+]$. The distribution of $Q^\star$ *within* this interval *cannot* be learned from the data and is *purely* driven by the prior construction. As such, we focus directly on the identifiable and thus learnable quantities, namely, the lower and upper bounds.

4.2 Synthetic data generation via SCM priors

SCM sampling: We construct the prior-data distribution by sampling SCMs $\mathcal{S}_i \sim \Pi_{\mathcal{S}}$. Each sampled SCM $\mathcal{S}_i = (X, U, A, Y, f, \mathbb{P}_U, \mathbb{P}_\varepsilon)$ induces an observational distribution $\mathbb{P}^{\mathcal{S}}_{\text{obs}}$ over (X, A, Y), from which we sample observational datasets $D_N \sim (\mathbb{P}^{\mathcal{S}}_{\text{obs}})^N$. Following prior-data constructions for other foundation models (Hollmann et al., 2023; Ma et al., 2025), we parameterize the structural assignments $f = (f_X, f_A, f_Y)$ using randomly sampled Bayesian neural networks. For example, the outcome is generated by $Y \sim f_{Y,\text{BNN}}(X, U, A, \varepsilon_Y)$, with $f_{Y,\text{BNN}} \sim \Pi_{f_Y}$. This yields a flexible prior over nonlinear covariate distributions, treatment assignment mechanisms, and outcome functions. The distributions for noise and unobserved confounding are sampled from simple parametric families with randomly drawn scales. We provide implementation details in Appendix C.

For effective amortization, we aim to construct a prior $\Pi_{\mathcal{S}}$ that is as "broad" as possible for the setting. In contrast to prior constructions for point-identified causal inference, where the sampled SCMs must be restricted to ensure (point) identifiability of the causal query (Ma et al., 2025), our setting is only partially identified due to unobserved confounding. As a result, our prior explicitly allows for SCMs with varying degrees of unobserved confounding, up to the level Γ specified by the sensitivity model $\mathcal{M}_\Gamma$.

Challenge ($\rightarrow$ label construction). For each sampled SCM $\mathcal{S}_i$, the prior $\Pi_{\mathcal{S}}$ provides an observational dataset D_N and query points $(x, a)_j$, with sensitivity levels $\Gamma_j \sim \Pi_\Gamma$. Crucially, however, the corresponding sensitivity bound labels $\theta^\pm(\mathcal{S}_i; x_j, a_j, \Gamma_j)$ are **not** directly available from the data-generating process $\mathcal{S}_i$; instead, they must be computed. Generating these labels efficiently across many sampled SCMs, queries, and sensitivity levels is addressed in the next section.

5 Sensitivity Bound Label Construction via Lagrangian Scalarization

To construct training labels, we must compute the sensitivity bounds θ^- and θ^+ for each sampled SCM and input tuple. We first formulate this as an optimization problem with forward, backward, and bi-objective forms (Section 5.1). We then introduce a Lagrangian scalarization whose sweep traces the resulting Pareto frontier of bound values (Section 5.2). Finally, we reduce the optimization over full distributions to a tractable computation over a conditional latent distribution, which yields a scalable label-generation procedure that we use for prior-data training (Section 5.3).

5.1 Optimization problem for sensitivity bounds

Forward formulation: For a sampled SCM $\mathcal{S}_i$ and input (x_j, a_j, Γ_j), the desired label is given by the solution of a population-level partial identification problem:

$$\theta_j^-(\mathcal{S}_i; x_j, a_j, \Gamma_j) = \inf_{\mathbb{P} \in \mathcal{P}(\mathcal{S}_i)} \mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) \qquad \text{s.t. } \Delta_{x_j, a_j}(\mathbb{P}) \le \Gamma_j, \tag{7}$$

$$\theta_j^+(\mathcal{S}_i; x_j, a_j, \Gamma_j) = \sup_{\mathbb{P} \in \mathcal{P}(\mathcal{S}_i)} \mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) \qquad \text{s.t. } \Delta_{x_j, a_j}(\mathbb{P}) \le \Gamma_j, \tag{8}$$

where $\mathcal{P}(\mathcal{S}_i) = \{\mathbb{P} : \mathbb{P}_{\text{obs}} = \mathbb{P}^{\mathcal{S}_i}_{\text{obs}}\}$ denotes the set of full distributions over (X, U, A, Y) that induce the same observational distribution as $\mathcal{S}_i$. This corresponds to the standard "forward" formulation

[4] Throughout, we omit the SCM index $\mathcal{S}$ on all training pairs $\{(D_N, x, a, \Gamma), (\theta^-, \theta^+)\}$ for ease of exposition.

of sensitivity analysis: for a fixed sensitivity level Γ, one optimizes the causal query subject to observational compatibility and the sensitivity constraint to find the bound.

Backward formulation: The same problem can be approached via a "backward" formulation: given a fixed causal query Q, minimize Γ such that the causal query *becomes the bound* $Q = \theta^{\pm}$ at the Γ-level constraint, subject to observational compatibility:

$$\Gamma_j^{\downarrow}(\mathcal{S}_i; x_j, a_j, \theta_j^-) = \inf_{\mathbb{P}\in\mathcal{P}} \Delta_{x_j,a_j}(\mathbb{P}) \qquad \text{s.t. } \mathcal{Q}(\mathbb{P}) \le \theta_j^- \text{ and } \mathbb{P}_{\text{obs}} = \mathbb{P}_{\text{obs}}^{\mathcal{S}_i} \tag{9}$$

$$\Gamma_j^{\uparrow}(\mathcal{S}_i; x_j, a_j, \theta_j^+) = \inf_{\mathbb{P}\in\mathcal{P}} \Delta_{x_j,a_j}(\mathbb{P}) \qquad \text{s.t. } \mathcal{Q}(\mathbb{P}) \ge \theta_j^+ \text{ and } \mathbb{P}_{\text{obs}} = \mathbb{P}_{\text{obs}}^{\mathcal{S}_i} \tag{10}$$

Bi-objective program: The forward problem and the backward problem above are two epsilon-constraint reductions of the same underlying bi-objective program

$$\max_{\mathbb{P}\in\mathcal{P}(\mathcal{S}_i)} \left(\mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)), -\Delta_{x_j,a_j}(\mathbb{P})\right). \tag{11}$$

Maximizing the causal query and minimizing the sensitivity divergence are competing objectives: tighter compatibility with the no-confounding reference $\Delta_{x_j,a_j} \downarrow \Gamma_{\min}$ pins $\mathbb{P}$ near the point-identified value of $\mathcal{Q}$, while permitting larger Δ_{x_j,a_j} enlarges the feasible set and extends the attainable range of $\mathcal{Q}$. The set of non-dominated solutions of Eq. 11 forms a Pareto frontier in the (Γ, θ)-plane, characterized by the maps

$$\Gamma \mapsto \theta_j^-(\mathcal{S}_i; x_j, a_j, \Gamma) \quad \text{(convex, non-increasing)}, \tag{12}$$

$$\Gamma \mapsto \theta_j^+(\mathcal{S}_i; x_j, a_j, \Gamma) \quad \text{(concave, non-decreasing)}. \tag{13}$$

The forward and backward problems trace these same frontiers from opposite parametrizations and are inverse functions of one another wherever the maps are strictly monotone.

5.2 Constructing the Lagrangian

Lagrangian formulation: The two epsilon-constraint reductions both have a common Lagrangian. Weighting the two objectives by a Lagrange multiplier $\lambda > 0$ gives, for the lower and upper bounds respectively,

$$\mathcal{L}_\lambda^{\downarrow}(\mathbb{P}; \mathcal{S}_i, x_j, a_j) = -\mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) - \lambda\Delta_{x_j,a_j}(\mathbb{P}), \tag{14}$$

$$\mathcal{L}_\lambda^{\uparrow}(\mathbb{P}; \mathcal{S}_i, x_j, a_j) = \mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) - \lambda\Delta_{x_j,a_j}(\mathbb{P}). \tag{15}$$

For each fixed λ, the optima $\mathbb{P}_\lambda^{\star\downarrow}$ and $\mathbb{P}_\lambda^{\star\uparrow}$ given by

$$\mathbb{P}_\lambda^{\star\downarrow} = \arg\max_{\mathbb{P}\in\mathcal{P}(\mathcal{S}_i)} \mathcal{L}_\lambda^{\downarrow}(\mathbb{P}; \mathcal{S}_i, x_j, a_j), \tag{16}$$

$$\mathbb{P}_\lambda^{\star\uparrow} = \arg\max_{\mathbb{P}\in\mathcal{P}(\mathcal{S}_i)} \mathcal{L}_\lambda^{\uparrow}(\mathbb{P}; \mathcal{S}_i, x_j, a_j) \tag{17}$$

each yield a single point $(\Gamma^\star, \theta^\star)$ on their respective Pareto frontier:

$$\Gamma^{\star\downarrow}(\lambda) = \Delta_{x_j,a_j}(\mathbb{P}_\lambda^{\star\downarrow}), \qquad \theta^{\star\downarrow}(\lambda) = \mathcal{Q}(\mathbb{P}_\lambda^{\star\downarrow}(Y(a_j) \mid x_j)), \tag{18}$$

$$\Gamma^{\star\uparrow}(\lambda) = \Delta_{x_j,a_j}(\mathbb{P}_\lambda^{\star\uparrow}), \qquad \theta^{\star\uparrow}(\lambda) = \mathcal{Q}(\mathbb{P}_\lambda^{\star\uparrow}(Y(a_j) \mid x_j)). \tag{19}$$

For both bounds, sweeping λ from large to small traces the Pareto frontier from tight (i.e., $\Gamma^\star$ small, bounds tight) to wide (i.e., $\Gamma^\star$ large, bounds wide).

Sweeping the frontier: To construct the label, we leverage the Lagrangian formulation together with a warm-start optimization along a sweep over values of λ. Solving Eq. 14 or Eq. 15 at a fixed λ yields a point $(\theta^\star(\lambda), \Gamma^\star(\lambda))$ on the Pareto frontier. Because adjacent λ values produce nearby optima, each successive optimization can be initialized with the

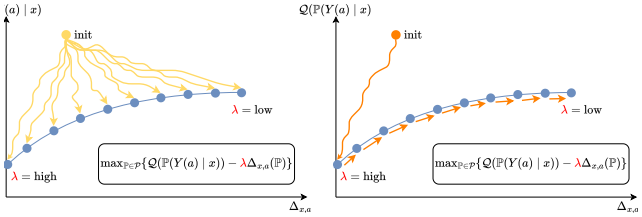


Figure 3: **Cold vs warm start.** Cold-started optimization (left) is re-initialized for each optimization as λ varies. Warm-starting (right) finds the Pareto frontier once and then walks across, starting the next optimization where the previous ended.

previous solution and converges in only a small number of additional steps, thereby shrinking computation cost across the sweep. This is a significant advantage over the traditional forward formulation: a grid of forward solves at fixed Γ targets has no analogous transferable state and each optimization is forced to cold-start. The following theorem formalizes the connection between the Lagrangian sweep and the Pareto frontier.

Theorem 1 (Lagrangian sweep recovers the Pareto frontier)**.** *Assume (A1) the divergence $\Delta_{x_j,a_j}(\mathbb{P})$ is convex in $\mathbb{P}$ on $\mathcal{P}(\mathcal{S}_i)$, and (A2) the causal query $\mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j))$ is linear in $\mathbb{P}$ on $\mathcal{P}(\mathcal{S}_i)$. Then, the following is true:*

(i) ***Frontier coverage.*** *For every $\Gamma \geq \Gamma_{\min}$, there exists $\lambda(\Gamma) \geq 0$ such that any maximizer $\mathbb{P}^{\star\uparrow}_{\lambda(\Gamma)}$ satisfies $\Delta_{x_j,a_j}(\mathbb{P}^{\star\uparrow}_{\lambda(\Gamma)}) = \Gamma$ and $\mathcal{Q}(\mathbb{P}^{\star\uparrow}_{\lambda(\Gamma)}(Y(a_j) \mid x_j)) = \theta^+_j(\mathcal{S}_i; x_j, a_j, \Gamma)$.*

(ii) ***Smooth traversal.*** *The map $\lambda \mapsto \Gamma^{\star\uparrow}(\lambda) := \Delta_{x_j,a_j}(\mathbb{P}^{\star\uparrow}_{\lambda})$ is monotone non-increasing, and $\lambda \mapsto (\Gamma^{\star\uparrow}(\lambda), \theta^{\star\uparrow}(\lambda))$ traces the upper frontier continuously as λ varies.*

The analogous statements hold for the lower bound with $\mathcal{L}^{\downarrow}_{\lambda}(\mathbb{P}) = -\mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) - \lambda\Delta_{x_j,a_j}(\mathbb{P})$.

Proof. See Appendix A.1 □

Remark 1 (Assumptions A1 and A2 hold in our setting)**.** *Assumption (A1) is satisfied for the marginal sensitivity model, f-sensitivity models with convex f (i.e., KL, χ^2, total variation, Hellinger), and Rosenbaum's sensitivity model*. Assumption (A2) holds for CAPO, CATE, and ATE, as each can be expressed as an expectation and, hence, is a linear functional of $\mathbb{P}$.*

Proof. See Appendix A.2 □

The above theorem guarantees smooth and complete traversal of the Pareto frontier. This structural property makes warm-starting effective in practice: adjacent values of λ produce optima that lie close together on the frontier, so a previously computed solution $\mathbb{P}^{\star}_{\lambda}$ provides a high-quality initialization for the subsequent optimization.

5.3 Instantiated label computation

The remaining task is to instantiate the optimization problems in Eq. 14 and Eq.15 in a form that allows for large-scale computation across the prior. To this end, we reduce the abstract objective over $\mathbb{P} \in \mathcal{P}(\mathcal{S}_i)$ to a tractable optimization over a single conditional latent distribution.

First, existing GTSM theory (Frauen et al., 2024) shows that the optimization over the full distributions $\mathbb{P} \in \mathcal{P}(\mathcal{S}_i)$ can be reduced to an optimization over a latent distribution $\mathbb{P}(U \mid x_j, A \neq a_j)$. For our objective, this translates to

$$\max_{\mathbb{P}(U|x_j)} \mathcal{Q}\left(f^{\mathcal{S}_i}_Y(x_j, \cdot, a_j)_\sharp \mathbb{P}(U \mid x_j)\right) - \lambda\Delta_{x_j,a_j}\left(\mathbb{P}(U \mid x_j), \mathbb{P}(U \mid x_j, a_j)\right), \tag{20}$$

$$\text{where } \mathbb{P}(U \mid x_j) = \pi^{\mathcal{S}_i}(a_j \mid x_j)\mathbb{P}(U \mid x_j, a_j) + \left(1 - \pi^{\mathcal{S}_i}(a_j \mid x_j)\right) \mathbb{P}(U \mid x_j, A \neq a_j). \tag{21}$$

We can fit $\mathbb{P}(U \mid x_j, A \neq a_j)$ fit with an unconstrained conditional normalizing flow (CNF) (Winkler et al., 2019) or any other conditional density estimator of choice. The reparameterization construction in Eq. 21 maps the latent distribution $\mathbb{P}(U \mid x_j, A \neq a_j)$ to $\mathbb{P}(U \mid x_j)$ and guarantees the observational compatibility, i.e., $\mathbb{P}_{\text{obs}} = \mathbb{P}^{\mathcal{S}_i}_{\text{obs}}$. We provide detailed derivation and intuition of these results in Appendix B.

Since we generate data synthetically, we obtain one additional advantage: f_Y is directly available to us and does not need to be estimated (This is unlike in Frauen et al. (2024) where a second CNF fitting stage is needed).

6 Experiments

We now evaluate our amortized foundation model (FM) for causal sensitivity analysis. Since ground-truth bounds are unavailable for real-world data, we use held-out DGPs sampled from our prior, for which reference bounds can be computed: in closed form for the MSM and via a high-budget numerical solve for KL (Appendix D.2).

Implementation details. We train two FMs with identical architecture: one for the MSM on closed-form labels and one for the KL f-sensitivity model on labels from our Lagrangian sweep, each on

10,000 synthetic DGPs. At test time, outcomes are standardized using only observed outcomes, and the predicted mean curves are projected onto monotone, non-crossing curves. Details on architecture, training, and label generation are in Appendix C.

Evaluation metrics. We report the empirical coverage of the central 90% posterior predictive interval and the RMSE of the posterior predictive mean (in units of the observed-outcome standard deviation), each with 95% whole-DGP bootstrap intervals.

Baselines. Existing methods for causal sensitivity analysis are per-instance procedures. We compare against NeuralCSA (Frauen et al., 2024), which covers the GTSM class, including sensitivity models without closed-form bounds (see Appendix D.4 for our reimplementation).

Label construction. We first evaluate the warm-started Lagrangian sweep used to generate the KL labels (128 DGPs, 50 values of λ). Warm-starting achieves **lower regret** than cold-starting each λ while being $1.90\times$ **faster** (Fig. 4; Appendix D.1).

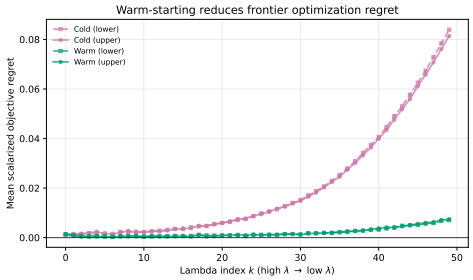


Figure 4: **Warm-started label construction.** Mean scalarized objective regret along the λ-sweep against a high-budget reference.

Calibration and accuracy. We evaluate the MSM FM on 512 held-out DGPs over 21 sensitivity levels $\Gamma \in [1, 5]$ and the KL FM on 64 held-out DGPs. Coverage is close to the nominal level for both sensitivity models (Table 1) and stable across sensitivity levels. The MSM error is nearly flat in Γ and already present at $\Gamma = 1$, where both bounds equal the point-identified CAPO. Hence, most of the error stems from estimating the CAPO from finite data rather than from the sensitivity component (see Appendix D.5 for decomposition).

Comparison with per-instance methods. On 32 held-out DGPs with matched queries and sensitivity levels, our FM is more accurate than NeuralCSA at every bound head and for both sensitivity models, with all paired differences excluding zero (Table 2). The per-instance error is dominated by NeuralCSA's first stage, which must estimate the outcome distribution from the observed data. At the same time, our FM answers a 128-instance workload in a single forward pass of 0.02 s, which is $758\times$ **faster** than NeuralCSA batched across instances and $45{,}920\times$ faster than its per-instance deployment (Fig. 5).

Table 1: **Calibration and accuracy** on held-out DGPs (MSM: 512; KL: 64), pooled over queries, arms, and sensitivity levels.

Setting	Head	90% coverage [95% CI]	RMSE [95% CI]
MSM	lower	0.896 [0.889, 0.903]	0.306 [0.295, 0.316]
	upper	0.887 [0.880, 0.894]	0.306 [0.296, 0.316]
KL	lower	0.884 [0.841, 0.922]	0.284 [0.245, 0.319]
	upper	0.878 [0.840, 0.916]	0.283 [0.249, 0.318]

Robustness. Coverage remains close to nominal under four out-of-prior shifts, including sensitivity levels up to $\Gamma = 9$, beyond the trained range (see Appendix D.6). Performance degrades for much smaller contexts or higher covariate dimensions than seen during training (see Appendix D.7).

Table 2: **Comparison with NeuralCSA** (RMSE, 32 held-out DGPs). Bold: better. Δ: paired difference.

Model	Head	FM (ours)	NeuralCSA	Paired Δ
MSM	lower	**0.282 [0.239, 0.324]**	0.430 [0.355, 0.503]	−0.147 [−0.212, −0.081]
	upper	**0.300 [0.241, 0.356]**	0.436 [0.365, 0.501]	−0.136 [−0.205, −0.064]
KL	lower	**0.330 [0.267, 0.393]**	0.448 [0.384, 0.505]	−0.118 [−0.189, −0.047]
	upper	**0.320 [0.260, 0.375]**	0.449 [0.375, 0.518]	−0.129 [−0.199, −0.058]

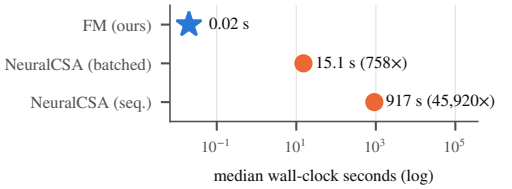


Figure 5: **Test-time cost** for a 128-instance KL workload on one GPU.

Conclusion: We introduced an amortized approach to causal sensitivity analysis based on prior-data fitted networks. Our key contribution is a general prior-data label construction for sensitivity bounds, using Lagrangian scalarization to trace Pareto frontiers across sensitivity levels without requiring model-specific analytical derivations. Building on these labels, we trained a foundation model that predicts PPDs over lower and upper sensitivity bounds directly from a dataset, causal query, and sensitivity level. Empirically, our results show that this approach can approximate sensitivity bounds accurately while replacing costly per-instance optimization with fast batched forward inference, opening a path toward practical in-context causal sensitivity analysis. We discuss limitations in Appendix E.

AI USE STATEMENT

In this work, we used generative AI tools (large language models and LLM-based coding agents) for tasks with required disclosure: to implement methods, namely the evaluation pipeline, our reimplementation of the NeuralCSA baseline, and the comparison with CausalFM; to design and provide feedback on the experimental protocol, including evaluation metrics, held-out cohorts, and robustness checks; to assist in interpreting results; and to aid in formulating mathematical claims, providing ingredients for and writing proofs, developing the theoretical framework, and proposing research hypotheses. Generating synthetic data sets with AI tools, translation, dataset cleaning, and qualitative data analysis are not applicable to this work; all synthetic data are produced by our own prior-sampling code. Additionally, we used generative AI tools for tasks with recommended disclosure: creating and editing software code, creating scientific figures and tables, drafting parts of the experimental section, editing the paper for readability. We have reviewed all AI-assisted work. All mathematical claims and proofs were checked by the authors. LLM-generated code was verified by the authors. All claims and text were checked by the authors. 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 uses only synthetic data generated from a prior over structural causal models; it involves no human subjects, personal data, or data collected from individuals, and no data set is released beyond synthetic benchmarks. The method is intended to make causal sensitivity analysis cheaper and more systematic, so that observational studies report bounds under unobserved confounding rather than overconfident point estimates. The main risks are overinterpretation and misuse: the bounds are only meaningful relative to the chosen sensitivity model, sensitivity level, and training prior, and should not be read as a certificate of causal validity. Because amortization makes large-scale sensitivity analyses easy, users could also report favorable sensitivity levels or subgroups selectively. Responsible use therefore requires reporting full sensitivity curves, documenting assumptions, choosing sensitivity ranges with domain expertise, and applying additional scrutiny in high-stakes domains such as medicine and public policy.

REPRODUCIBILITY STATEMENT

We provide anonymized source code at `https://anonymous.4open.science/r/CSA-PFN-Anon-FA52/`, including the prior sampler, label construction for both sensitivity models, PFN training, and evaluation, with configuration files for all reported runs. The prior construction and label-generation configurations are described in Appendix C.4 and Table 4, and the PFN architecture and training hyperparameters in Appendix C and Table 3. The evaluation protocol (held-out cohorts, metrics, reference bounds, and bootstrap intervals) is given in Appendix D.2, and the baseline reimplementation in Appendix D.4. All assumptions for our theoretical results are stated in the main text, with complete proofs in Appendix A and the derivation of the operational objective in Appendix B.

REFERENCES

Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Benson Li, Junwei Ma, Jesse C. Cresswell, and Rahul G. Krishnan. CausalPFN: Amortized Causal Effect Estimation via In-Context Learning. *arXiv preprint*, arXiv:2506.07918, 2025.

Dominik Bär, Nicolas Pröllochs, and Stefan Feuerriegel. The role of social media ads for election outcomes: Evidence from the 2021 German election. *PNAS Nexus*, 4(3):pgaf073, 2025.

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

Jacob Dorn and Kevin Guo. Sharp Sensitivity Analysis for Inverse Propensity Weighting via Quantile Balancing. *Journal of the American Statistical Association*, 118(544):2645–2657, 2023.

Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S. Kohane, and Mihaela van der Schaar. Causal machine learning for predicting treatment outcomes. *Nature Medicine*, 30(4):958–968, 2024.

Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. Sharp Bounds for Generalized Causal Sensitivity Analysis. *Advances in Neural Information Processing Systems*, 36:40556–40586, 2023.

Dennis Frauen, Fergus Imrie, Alicia Curth, Valentyn Melnychuk, and Stefan Feuerriegel. A Neural Framework for Generalized Causal Sensitivity Analysis. 2024.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models. *arXiv preprint*, arXiv:2511.08667, 2026.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *ICLR*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From Tables to Time: Extending TabPFN-v2 to Time Series Forecasting, 2025.

Andrew Jesson, Sören Mindermann, Yarin Gal, and Uri Shalit. Quantifying Ignorance in Individual-Level Causal-Effect Estimates under Hidden Confounding. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Andrew Jesson, Alyson Douglas, Peter Manshausen, Maëlys Solal, Nicolai Meinshausen, Philip Stier, Yarin Gal, and Uri Shalit. Scalable Sensitivity and Uncertainty Analyses for Causal-Effect Estimates of Continuous-Valued Interventions. *Advances in Neural Information Processing Systems*, 35:13892–13907, 2022.

Ying Jin, Zhimei Ren, and Emmanuel J. Candès. Sensitivity analysis of individual treatment effects: A robust conformal inference approach. *Proceedings of the National Academy of Sciences*, 120 (6):e2214889120, 2023.

Ying Jin, Zhimei Ren, and Zhengyuan Zhou. Sensitivity Analysis Under the f -Sensitivity Model: A Distributional Robustness Perspective. *Operations Research*, 74(2):860–878, 2026.

Nathan Kallus, Xiaojie Mao, and Angela Zhou. Interval Estimation of Individual-Level Causal Effects Under Unobserved Confounding. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 2019.

Milan Kuzmanovic, Dennis Frauen, Tobias Hatt, and Stefan Feuerriegel. Causal Machine Learning for Cost-Effective Allocation of Development Aid. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, 2024.

Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation Models for Causal Inference via Prior-Data Fitted Networks. *arXiv preprint*, arXiv:2506.10914, 2025.

C. Manski. Nonparametric Bounds on Treatment Effects. *The American Economic Review*, 1989.

Myrl G. Marmarelis, Elizabeth Haddad, Andrew Jesson, Neda Jahanshad, Aram Galstyan, and Greg Ver Steeg. Partial identification of dose responses with hidden confounders. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2023.

Myrl G. Marmarelis, Greg Ver Steeg, Aram Galstyan, and Fred Morstatter. Ensembled Prediction Intervals for Causal Outcomes Under Hidden Confounding. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, 2024.

Valentyn Melnychuk, Vahid Balazadeh, Stefan Feuerriegel, and Rahul G. Krishnan. Frequentist Consistency of Prior-Data Fitted Networks for Causal Inference, 2026.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers Can Do Bayesian Inference. In *ICLR*, 2022.

Thomas Nagler. Statistical Foundations of Prior-Data Fitted Networks. https://arxiv.org/abs/2305.11097v1, 2023.

Miruna Oprescu, Jacob Dorn, Marah Ghoummaid, Andrew Jesson, Nathan Kallus, and Uri Shalit. B-Learner: Quasi-Oracle Bounds on Heterogeneous Causal Effects Under Hidden Confounding. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2. ed., repr. with corr edition, 2013. ISBN 978-0-521-77362-1 978-0-521-89560-6.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-PFN: In-Context Learning for Causal Effect Estimation. *arXiv preprint*, arXiv:2506.06039, 2025.

Paul R. Rosenbaum. Sensitivity analysis for certain permutation inferences in matched observational studies. *Biometrika*, 74(1):13–26, 1987.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

Lea Skapetze, Daniela Koller, Andreas Zwergal, Stefan Feuerriegel, Anna Rubinski, and Eva Grill. Monitoring changes in vitamin D levels during the COVID-19 pandemic with routinely-collected laboratory data. *Nature Communications*, 16(1):8772, 2025.

Zhiqiang Tan. A Distributional Approach for Causal Inference Using Propensity Scores. *Journal of the American Statistical Association*, 101(476):1619–1637, 2006.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning Likelihoods with Conditional Normalizing Flows, 2019.

Mingzhang Yin, Claudia Shi, Yixin Wang, and David M. Blei. Conformal Sensitivity Analysis for Individual Treatment Effects. *Journal of the American Statistical Association*, 119(545):122–135, 2024.

Qingyuan Zhao, Dylan S. Small, and Bhaswar B. Bhattacharya. Sensitivity Analysis for Inverse Probability Weighting Estimators via the Percentile Bootstrap. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(4):735–761, 2019.

A PROOFS

A.1 THEOREM 1

Theorem 1. The argument proceeds by establishing concavity of the upper frontier and then deriving (i) and (ii) from supporting-hyperplane geometry. The lower bound case is symmetric.

Step 1: $\mathcal{P}(\mathcal{S}_i)$ is convex. The observational-compatibility constraint $\mathbb{P}_{\text{obs}} = \mathbb{P}^{\mathcal{S}_i}_{\text{obs}}$, is linear, $\int_{\mathcal{U}} \mathbb{P}\mathrm{d}u = \mathbb{P}^{\mathcal{S}_i}_{\text{obs}}$, and therefore $\mathcal{P}(\mathcal{S}_i) = \{\mathbb{P} : \mathbb{P}_{\text{obs}} = \mathbb{P}^{\mathcal{S}_i}_{\text{obs}}\}$ is convex.

Step 2: θ^+_j is concave in Γ. Let $\Gamma_1, \Gamma_2 \geq \Gamma_{\min}$, and $t \in [0,1]$. Pick $\mathbb{P}_1, \mathbb{P}_2 \in \mathcal{P}(\mathcal{S}_i)$ achieving the upper bounds at Γ_1, Γ_2, meaning that $\Delta_{x_j,a_j}(\mathbb{P}_k) \leq \Gamma_k$ and $\mathcal{Q}(\mathbb{P}_k(Y(a_j) \mid x_j)) = \theta^+_j(\Gamma_k)$ for $k = 1, 2$. Define the linear interpolation $\mathbb{P}_t = (1-t)\mathbb{P}_1 + t\mathbb{P}_2$. By Step 1, $\mathbb{P}_t \in \mathcal{P}(\mathcal{S}_i)$. By (A1), we have

$$\Delta_{x_j,a_j}(\mathbb{P}_t) \leq (1-t)\Delta_{x_j,a_j}(\mathbb{P}_1) + t\Delta_{x_j,a_j}(\mathbb{P}_2) \leq (1-t)\Gamma_1 + t\Gamma_2, \tag{22}$$

so $\mathbb{P}_t$ is feasible for the forward problem at sensitivity level $\Gamma_t = (1-t)\Gamma_1 + t\Gamma_2$. By (A2),

$$\mathcal{Q}(\mathbb{P}_t(Y(a_j) \mid x_j)) = (1-t)\theta^+_j(\Gamma_1) + t\theta^+_j(\Gamma_2). \tag{23}$$

Since this value is attained by a feasible distribution, $\theta^+_j(\Gamma_t) \geq (1-t)\theta^+_j(\Gamma_1) + t\theta^+_j(\Gamma_2)$, θ^+_j is concave in Γ.

Step 3: Achievable set has supporting hyperplanes everywhere on its upper boundary. The upper boundary of the achievable set $\mathcal{A} = \{(\Gamma, \theta) : \exists \mathbb{P} \in \mathcal{P}(\mathcal{S}_i),\ \Delta_{x_j,a_j}(\mathbb{P}) \leq \Gamma,\ \mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) = \theta\}$ is the graph of the concave map $\Gamma \mapsto \theta^+_j(\Gamma)$ from Step 2. The hypograph of a concave function is convex, so by the supporting hyperplane theorem, every point $(\Gamma_0, \theta^+_j(\Gamma_0))$ on the upper boundary admits a supporting hyperplane with normal of the form $(-\lambda_0, 1)$ for some $\lambda_0 \geq 0$.

Step 4: Frontier coverage (i). Fix Γ_0 and let λ_0 be the supporting hyperplane slope from Step 3. Membership of $(\Gamma_0, \theta^+_j(\Gamma_0))$ in the supporting hyperplane means

$$\theta - \lambda_0\Gamma \leq \theta^+_j(\Gamma_0) - \lambda_0\Gamma_0 \quad \text{for all } (\Gamma, \theta) \in \mathcal{A}. \tag{24}$$

Translating back to distributions, every $\mathbb{P} \in \mathcal{P}(\mathcal{S}_i)$ with $\Delta_{x_j,a_j}(\mathbb{P}) \leq \Gamma$ and $\mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) = \theta$ satisfies

$$\mathcal{Q}(\mathbb{P}(Y(a_j) \mid x_j)) - \lambda_0\Delta_{x_j,a_j}(\mathbb{P}) \leq \mathcal{L}^{\uparrow}_{\lambda_0}(\mathbb{P}_0) \tag{25}$$

for any $\mathbb{P}_0$ achieving $(\Gamma_0, \theta^+_j(\Gamma_0))$. Since the right-hand side is the maximum value of $\mathcal{L}^{\uparrow}_{\lambda_0}$, $\mathbb{P}_0 \in \arg\max_{\mathbb{P}} \mathcal{L}^{\uparrow}_{\lambda_0}(\mathbb{P})$. Setting $\lambda(\Gamma_0) = \lambda_0$ proves (i).

Step 5: Smooth traversal (ii). We show $\lambda \mapsto \Gamma^{\star\uparrow}(\lambda)$ is monotone non-increasing. Let $\lambda_1 > \lambda_2$ and let $\mathbb{P}_k = \mathbb{P}^{\star\uparrow}_{\lambda_k}$ with $\Gamma_k = \Delta_{x_j,a_j}(\mathbb{P}_k)$ and $\theta_k = \mathcal{Q}(\mathbb{P}_k(Y(a_j) \mid x_j))$ for $k = 1, 2$. Optimality at each λ_k gives

$$\theta_1 - \lambda_1\Gamma_1 \geq \theta_2 - \lambda_1\Gamma_2, \qquad \theta_2 - \lambda_2\Gamma_2 \geq \theta_1 - \lambda_2\Gamma_1. \tag{26}$$

Adding the two inequalities yields $(\lambda_1 - \lambda_2)(\Gamma_2 - \Gamma_1) \geq 0$, and, since $\lambda_1 > \lambda_2$, we conclude $\Gamma_2 \geq \Gamma_1$. Hence, $\Gamma^{\star\uparrow}$ is monotone non-increasing in λ.

For continuity: by Step 3, the supporting hyperplane slope $\lambda(\Gamma)$ ranges through the (negated) superdifferential of the concave function θ^+_j as Γ varies over its domain. Concavity ensures the superdifferential mapping is upper hemicontinuous and surjective onto $[0, \infty)$, so $\Gamma \mapsto \lambda(\Gamma)$ has no gaps in its range; equivalently, $\lambda \mapsto \Gamma^{\star\uparrow}(\lambda)$ has no jump discontinuities. The frontier point

$(\Gamma^{\star\uparrow}(\lambda), \theta^{\star\uparrow}(\lambda))$ thus traces the graph of θ_j^+ continuously as λ varies. (Kinks in θ_j^+, where the superdifferential is set-valued, correspond to flat regions in $\Gamma^{\star\uparrow}(\lambda)$, not to discontinuous jumps.)

□

A.2 Remark 1

Remark 1. We show that assumptions A1 and A2 hold for the settings considered.

A1: Fix (x_j, a_j) and write

$$\pi_j = \mathbb{P}_{\text{obs}}^{\mathcal{S}_i}(a_j \mid x_j), \qquad p^\star(u) = \mathbb{P}(u \mid x_j, a_j), \qquad q(u) = \mathbb{P}(u \mid x_j, A \neq a_j). \tag{27}$$

Under the reduced GTSM parametrization, the optimization variable is q, while $p^\star$ and π_j are fixed by observational compatibility. Moreover,

$$p(u \mid x_j) = \pi_j p^\star(u) + (1 - \pi_j) q(u), \tag{28}$$

and the relevant latent density ratio is

$$r_q(u) = \frac{q(u)}{p^\star(u)}. \tag{29}$$

Hence, for any two feasible latent distributions q_1, q_2 and any $t \in [0, 1]$, it holds that

$$q_t = (1 - t)q_1 + tq_2 \quad \Longrightarrow \quad r_{q_t}(u) = (1 - t)r_{q_1}(u) + tr_{q_2}(u). \tag{30}$$

Thus, $q \mapsto r_q$ is affine.

MSM: For the marginal sensitivity model, the GTSM functional can be written as

$$\Delta_{x_j, a_j}^{\text{MSM}}(q) = \max \left\{ \sup_{u \in \mathcal{U}} r_q(u), \sup_{u \in \mathcal{U}} r_q(u)^{-1} \right\}. \tag{31}$$

The map $r \mapsto r$ is linear and the map $r \mapsto r^{-1}$ is convex on $\mathbb{R}_{>0}$. Suprema of convex functions are convex, and the maximum of convex functions is convex. Therefore, $q \mapsto \Delta_{x_j, a_j}^{\text{MSM}}(q)$ is convex.

f-sensitivity: For the original binary f-sensitivity model of Jin et al. (2026), write

$$\Delta_{x_j, a_j}^{f}(q) = \max \left\{ \int_{\mathcal{U}} f(r_q(u)) p^\star(u) \, \mathrm{d}u, \int_{\mathcal{U}} f(r_q(u)^{-1}) q(u) \, \mathrm{d}u \right\}. \tag{32}$$

Using $q(u) = r_q(u) p^\star(u)$, the second term can be written as

$$\int_{\mathcal{U}} r_q(u) f(r_q(u)^{-1}) p^\star(u) \, \mathrm{d}u. \tag{33}$$

Define the conjugate generator

$$f^\diamond(t) := t f(t^{-1}). \tag{34}$$

The perspective construction implies that $f^\diamond$ is convex whenever f is convex. In particular, when f is twice differentiable,

$$(f^\diamond)''(t) = \frac{f''(t^{-1})}{t^3} \geq 0. \tag{35}$$

Because $q \mapsto r_q$ is affine, both

$$q \mapsto \int_{\mathcal{U}} f(r_q(u)) p^\star(u) \, \mathrm{d}u \tag{36}$$

and

$$q \mapsto \int_{\mathcal{U}} f^\diamond(r_q(u)) p^\star(u) \, \mathrm{d}u \tag{37}$$

are convex. Their pointwise maximum is therefore convex. Hence, $q \mapsto \Delta_{x_j, a_j}^{f}(q)$ is convex for every convex generator f, including the usual KL, χ^2, total variation, and Hellinger choices.

($\star$) **Rosenbaum:** We show Rosenbaum's model gives convex feasible sets $\{q : \Delta^{\mathrm{RB}}_{x_j,a_j}(q) \leq \Gamma\}$, which is sufficient for the constrained optimization purpose in this paper.

For Rosenbaum's sensitivity model, the scalar functional is commonly written as

$$\Delta^{\mathrm{RB}}_{x_j,a_j}(q) = \sup_{u_1,u_2 \in \mathcal{U}} \frac{r_q(u_1)}{r_q(u_2)}. \tag{38}$$

The corresponding level set at sensitivity level Γ is

$$\{q : r_q(u_1) \leq \Gamma r_q(u_2) \text{ for all } u_1, u_2 \in \mathcal{U}\}. \tag{39}$$

This level set is convex: if q_1 and q_2 both satisfy the above inequalities, then, for $q_t = (1-t)q_1 + tq_2$,

$$\begin{aligned} r_{q_t}(u_1) &= (1-t)r_{q_1}(u_1) + t r_{q_2}(u_1) \\ &\leq \Gamma\left((1-t)r_{q_1}(u_2) + t r_{q_2}(u_2)\right) \\ &= \Gamma r_{q_t}(u_2). \end{aligned} \tag{40}$$

Hence, Rosenbaum's model gives convex feasible sets $\{q : \Delta^{\mathrm{RB}}_{x_j,a_j}(q) \leq \Gamma\}$, which is a sufficient condition for the purposes of our constrained optimization. Note $\Delta^{\mathrm{RB}}_{x_j,a_j}$ itself is not generally convex as a scalar functional.

A2: Finally, Assumption (A2) holds for CAPO because

$$Q(x_j, a_j, \mathbb{P}) = \mathbb{E}_{\mathbb{P}}[Y(a_j) \mid x_j] = \int_{\mathcal{Y}} y \, d\mathbb{P}(Y(a_j) = y \mid x_j) \tag{41}$$

is linear in the potential-outcome distribution. The CATE is a difference of two CAPO functionals and is therefore linear. ATE is an average of CAPO or CATE over the covariate distribution and is therefore linear as well. □

B THEORETICAL DETAILS

This appendix derives the operational objective in Eq. 20 from the abstract Lagrangian in Eqs. 14–15, thus making explicit each reduction taken and verifying observational compatibility at each step. Throughout, we work with binary treatments $a_j \in \{0, 1\}$, scalar outcomes $Y \in \mathbb{R}$, and the conditional average potential outcome (CAPO) functional $\mathcal{Q} = \mathbb{E}[\cdot]$. Generalizations to multivariate outcomes and other monotone causal functionals follow the same steps with notational overhead. Similarly, generalization to finite categorical treatment space $\mathcal{A}$ is possible possible but omitted for notational clarity.

What we adapt and what is specific to our setting: This appendix derives the operational objective in Eq. 20 by combining (i) the latent-space reformulation theorem of NeuralCSA (Frauen et al., 2024), (ii) the transformation-invariance property required by that reformulation, and (iii) constructions specific to our prior-fitted setting. For the f-sensitivity model, we use the original binary definition of Jin et al. (2026) and derive its two directional latent-density divergences directly. In particular, we do not rely on the same-reference-measure display for f-sensitivity in NeuralCSA[5]. Theorem 1 reduction (Appendix B.2) and the standard-normal reference convention (Appendix B.3) follow the general NeuralCSA reformulation. Specific to our setting are the elimination of the first-stage outcome-density estimator (Appendix B.4), the reparameterization expressed directly through the SCM propensity (Appendix B.5), the density-ratio forms of the GTSM divergences (Appendix B.6), and their Monte Carlo implementation (Appendices B.7 and B.8). The Manski-limit characterization (Appendix B.9) provides a sanity check on the construction.

B.1 SETUP

Recall the abstract optimization problem from Eq. 15: for fixed SCM $\mathcal{S}_i$, query (x_j, a_j), and Lagrange multiplier $\lambda > 0$, we have

$$\max_{\mathbb{P} \in \mathcal{P}(\mathcal{S}_i)} \mathcal{Q}\big(\mathbb{P}(Y(a_j) \mid x_j)\big) - \lambda \Delta_{x_j, a_j}\big(\mathbb{P}(U \mid x_j), \mathbb{P}(U \mid x_j, a_j)\big), \tag{42}$$

where $\mathcal{P}(\mathcal{S}_i) = \{\mathbb{P} : \mathbb{P}_{\text{obs}} = \mathbb{P}^{\mathcal{S}_i}_{\text{obs}}\}$ is the observationally-compatible class. The optimization is over full distributions $\mathbb{P}$ on (X, U, A, Y), an infinite-dimensional object. Here, we derive the reduction of this to a tractable optimization over a single, computable quantity.

B.2 LATENT-SPACE REFORMULATION

Result. Under any transformation-invariant GTSM, the abstract optimization above is equivalent to an optimization over only the counterfactual-arm latent distribution $\mathbb{P}(U \mid x_j, A \neq a_j)$, with the queried-arm latent $\mathbb{P}^\star(U \mid x_j, a_j)$ and outcome map $f^\star_Y$ held fixed at any choice satisfying the queried-arm pushforward identity $\mathbb{P}^{\mathcal{S}_i}_{\text{obs}}(Y \mid x_j, a_j) = (f^\star_Y(x_j, \cdot, a_j))_\sharp \mathbb{P}^\star(U \mid x_j, a_j)$.

Sketch of argument. NeuralCSA (Frauen et al., 2024) Theorem 1 establishes that the supremum over $\mathbb{P} \in \mathcal{P}(\mathcal{S}_i) \cap \mathcal{M}_\Gamma$ (sensitivity-constrained candidates) is attained by a sequence of full distributions for which the queried-arm latent and outcome map are fixed. The full proof proceeds in three steps: (i) construct an alternative full distribution that induces the prescribed $\mathbb{P}^\star(U \mid x_j, a_j)$ and $f^\star_Y$; (ii) verify it remains in the GTSM via transformation invariance; and (iii) verify it attains the same query value. We refer the reader to NeuralCSA (Frauen et al., 2024, Appendix B.3) for the full construction.

Intuition. The intuition behind the theorem lies in the understanding that (a) the sensitivity model divergence constraint operates only in the latent space, and (b) the space of distributions over X, U, A, Y has unnecessarily many degrees of freedom, even when restricted to $\mathbb{P} \in \mathcal{P}(\mathcal{S}_i) \cap \mathcal{M}_\Gamma$. We can thus fix some elements in a convenient way.

[5]Private correspondence with the authors confirmed the reference-measure error in the NeuralCSA f-sensitivity display. The corrected functional remains transformation-invariant, so the latent-space reformulation used here is unaffected. Here, we work from the original definition to avoid propagating the error.

For our purposes, the consequence is: the optimization is over $\mathbb{P}(U \mid x_j, A \neq a_j)$ alone, and the objective reads

$$\sup_{\mathbb{P}(U|x_j, A\neq a_j)} \mathcal{Q}\big((f_Y^\star(x_j, \cdot, a_j))_\sharp \mathbb{P}(U \mid x_j)\big) - \lambda \Delta_{x_j,a_j}\big(\mathbb{P}(U \mid x_j), \mathbb{P}^\star(U \mid x_j, a_j)\big), \tag{43}$$

with $\mathbb{P}(U \mid x_j)$ recovered from the candidate's two arm-conditional latents by the law of total probability, as stated in Eq. 21.

B.3 Standard-normal reference convention

Result. We may fix $\mathbb{P}^\star(U \mid x_j, a_j) = \mathcal{N}(0, I)$ without loss of generality.

Justification. GTSMs are transformation-invariant (NeuralCSA; Lemma 2): for any measurable $t : \mathcal{U} \to \widetilde{\mathcal{U}}$, the divergence satisfies $D_{x,a}(\mathbb{P}(U \mid x), \mathbb{P}(U \mid x, a)) \geq D_{x,a}(\mathbb{P}(t(U) \mid x), \mathbb{P}(t(U) \mid x, a))$. The MSM, f-sensitivity, and Rosenbaum models are all transformation-invariant (NeuralCSA; Lemma 2). Applying the inverse CDF transform (or any other invertible map sending $\mathbb{P}^\star(U \mid x_j, a_j)$ to $\mathcal{N}(0, I)$) preserves both the partial-identification problem and the GTSM constraint. We adopt the density $\phi := \mathcal{N}(0, I)$ throughout the rest of the derivation.

Constructive coincidence in our setting. Our prior over SCMs imposes $\mathbb{P}^{\mathcal{S}_i}(U \mid X, A) = \mathcal{N}(0, I)$ for all (X, A) by design, so the standard-normal reference matches the true SCM's queried-arm latent exactly rather than via transformation. This is purely a parameterization convenience and does not restrict the class of observational distributions the prior can represent.

B.4 Removal of ε_Y

Result. We may take the outcome map $f_Y^\star$ to be deterministic and invertible in u for fixed (x, a). There is no exogenous noise ε_Y in the structural assignment $Y = f_Y^\star(x, U, a)$.

Justification. Theorem 1 of NeuralCSA further establishes that the bound-attaining structural model has the form $\mathbb{P}^\star(Y \mid x, a, u) = \delta(Y - f_{x,a}^\star(u))$ for an invertible $f_{x,a}^\star : \mathcal{U} \to \mathcal{Y}$. Intuitively, any distribution $\mathbb{P}(Y \mid x, a, u)$ that induces $Y \perp U \mid X, A$ would satisfy observational compatibility but imply unconfoundedness and not yield a valid bound. Maximal mutual information between U and Y (required for the bound) is achieved when Y is a deterministic invertible function of U. See Appendix B.3 in NeuralCSA for the full argument.

Computational consequence in our setting. In the two-stage approach of NeuralCSA, $f_{x,a}^\star$ is fit from observational data via a Stage-1 normalizing flow trained to satisfy the queried-arm push-forward identity. In our synthetic setting, $f_{x,a}^\star$ is the outcome BNN drawn during prior sampling, which is deterministic and invertible by construction. Stage 1 is therefore unnecessary — we read $f_Y^\star$ directly from $\mathcal{S}_i$.

B.5 The reparameterization construction

After the reductions in Appendices B.2–B.4, the optimization variable is $\nu := \mathbb{P}(U \mid x_j, A \neq a_j)$, and the candidate's covariate-conditional latent $\mathbb{P}(U \mid x_j)$ is determined from ν via the law of total probability:

$$\mathbb{P}(U \mid x_j) = \pi^{\mathcal{S}_i}(a_j \mid x_j)\mathbb{P}(U \mid x_j, a_j) + \big(1 - \pi^{\mathcal{S}_i}(a_j \mid x_j)\big)\mathbb{P}(U \mid x_j, A \neq a_j). \tag{44}$$

Substituting the queried-arm reference $\mathbb{P}(U \mid x_j, a_j) = \phi$:

$$\mathbb{P}(U \mid x_j) = \pi^{\mathcal{S}_i}(a_j \mid x_j)\phi + \big(1 - \pi^{\mathcal{S}_i}(a_j \mid x_j)\big)\nu \tag{45}$$

This is the propensity-weighted mixture used in Eq. 20. We write $\pi := \pi^{\mathcal{S}_i}(a_j \mid x_j)$ for brevity in what follows.

Observational compatibility holds by construction. The class $\mathcal{P}(\mathcal{S}_i)$ requires the candidate to reproduce $\mathbb{P}_{\text{obs}}^{\mathcal{S}_i}$ on (X, A, Y). Compatibility on X and A is automatic from sharing the SCM's marginal and propensity. Compatibility on Y decomposes by treatment arm:

$$\mathbb{P}_{\text{obs}}^{\mathcal{S}_i}(Y \mid x_j, a_j) = \big(f_Y^\star(x_j, \cdot, a_j)\big)_\sharp \mathbb{P}(U \mid x_j, a_j), \tag{46}$$

$$\mathbb{P}_{\text{obs}}^{\mathcal{S}_i}(Y \mid x_j, A \neq a_j) = \big(f_Y^\star(x_j, \cdot, A \neq a_j)\big)_\sharp \mathbb{P}(U \mid x_j, A \neq a_j). \tag{47}$$

At the queried arm, fixing $\mathbb{P}(U \mid x_j, a_j) = \phi$ together with $f_Y^\star$ from $\mathcal{S}_i$ gives compatibility automatically: $(f_Y^\star(x_j, \cdot, a_j))_\sharp \phi = \mathbb{P}^{\mathcal{S}_i}_{\text{obs}}(Y \mid x_j, a_j)$ holds because $\mathbb{P}^{\mathcal{S}_i}(U \mid x_j, a_j) = \phi$ in the SCM's own construction. At the counterfactual arm, an analogous identity must hold for $\mathbb{P}(U \mid x_j, A \neq a_j)$.

This counterfactual-arm identity is the constraint that observational compatibility imposes. It is *not* automatic for an unconstrained ν. However, Theorem 1 in NeuralCSA establishes that the reformulation is *sufficient*; that is, every $\mathbb{P} \in \mathcal{P}(\mathcal{S}_i)$ is attainable by some choice of ν in the reparameterization, and conversely, the bounds attained by sweeping ν saturate those of the original problem. The constraint at the counterfactual arm is absorbed into the reparameterization rather than appearing as a separate side condition.

Implication for parameterization. The optimization variable ν should be parameterized by an *unconstrained* density estimator (e.g., a normalizing flow) — the reparameterization handles compatibility, and no projection step or majorization constraint is needed. Naïvely parameterizing $\mathbb{P}(U \mid x_j)$ directly (rather than ν) and recovering ν implicitly fails: not every $\mathbb{P}(U \mid x_j)$ admits a valid corresponding $\nu \geq 0$ via the inverse-mixture formula, and an unconstrained flow on $\mathbb{P}(U \mid x_j)$ would explore observationally-incompatible candidates outside $\mathcal{P}(\mathcal{S}_i)$.

B.6 GTSM DIVERGENCES UNDER THE REPARAMETERIZATION

We now derive the form of the GTSM divergence $\Delta_{x_j,a_j}(\mathbb{P}(U \mid x_j), \phi)$ for each sensitivity model in scope, by expressing the result as a functional of the density ratio

$$r_\nu(u) := \frac{\nu(u)}{\phi(u)}, \tag{48}$$

which is the natural quantity to compute when ν is parameterized by a normalizing flow.

The starting point is Lemma 1 (extended in Lemma 3) in NeuralCSA, which provides a single density-ratio quantity through which all three GTSMs of interest are expressed. Define

$$\rho(x, u, a) := \frac{1}{1 - \mathbb{P}(a \mid x)} \left(\frac{\mathbb{P}(u \mid x)}{\mathbb{P}(u \mid x, a)} - \mathbb{P}(a \mid x) \right). \tag{49}$$

By Lemma 3 (Eq. 12–15) in NeuralCSA, ρ equals the full propensity odds ratio, i.e., $\text{OR}(\mathbb{P}(a \mid x), \mathbb{P}(a \mid x, u))$.

Under our reparameterization, with $\mathbb{P}(u \mid x_j) = \pi\phi(u) + (1 - \pi)\nu(u)$ and $\mathbb{P}(u \mid x_j, a_j) = \phi(u)$, and writing $\pi = \pi^{\mathcal{S}_i}(a_j \mid x_j)$, we obtain:

$$\begin{aligned} \rho(x_j, u, a_j) &= \frac{1}{1 - \pi} \left(\frac{\pi\phi(u) + (1 - \pi)\nu(u)}{\phi(u)} - \pi \right) && (50) \\ &= \frac{1}{1 - \pi} \frac{(1 - \pi)\nu(u)}{\phi(u)} && (51) \\ &= r_\nu(u). && (52) \end{aligned}$$

The propensity factor π thus cancels analytically. Henceforth, $\rho \equiv r_\nu$ in all three GTSM formulas below.

Marginal sensitivity model (MSM). By Lemma 3, Eq. (9) in NeuralCSA, the MSM divergence on the candidate is

$$\begin{aligned} D^{\text{MSM}}_{x_j,a_j} &= \max \left(\sup_u \rho(x_j, u, a_j), \sup_u \rho(x_j, u, a_j)^{-1} \right) && (53) \\ &= \max \left(\sup_u r_\nu(u), \sup_u r_\nu(u)^{-1} \right). && (54) \end{aligned}$$

The MSM constraint $D^{\text{MSM}}_{x_j,a_j} \leq \Gamma$ is equivalent to $\Gamma^{-1} \leq r_\nu(u) \leq \Gamma$ pointwise in u.

f-sensitivity models. The sensitivity divergence is

$$D^f_{x_j,a_j} = \max \left\{ \mathbb{E}_{U \sim \phi} \left[f(r_\nu(U)) \right], \mathbb{E}_{U \sim \nu} \left[f(r_\nu(U)^{-1}) \right] \right\}. \tag{55}$$

Equivalently,

$$D^f_{x_j,a_j} = \max\left\{D_f(\nu\|\phi), D_f(\phi\|\nu)\right\}. \tag{56}$$

Writing both terms under the queried-arm reference ϕ gives

$$D^f_{x_j,a_j} = \max\left\{\mathbb{E}_{U\sim\phi}\left[f(r_\nu(U))\right], \mathbb{E}_{U\sim\phi}\left[r_\nu(U) f(r_\nu(U)^{-1})\right]\right\}, \tag{57}$$

whereas writing both terms under ν gives

$$D^f_{x_j,a_j} = \max\left\{\mathbb{E}_{U\sim\nu}\left[\frac{f(r_\nu(U))}{r_\nu(U)}\right], \mathbb{E}_{U\sim\nu}\left[f(r_\nu(U)^{-1})\right]\right\}. \tag{58}$$

For the KL generator $f(t) = t\log t$, the first directional term is

$$\begin{aligned}\mathbb{E}_{U\sim\phi}\left[r_\nu(U)\log r_\nu(U)\right] &= \mathbb{E}_{U\sim\nu}\left[\log r_\nu(U)\right] &(59)\\ &= D_{\mathrm{KL}}(\nu\|\phi), &(60)\end{aligned}$$

and the second directional term is

$$\begin{aligned}\mathbb{E}_{U\sim\nu}\left[-\frac{\log r_\nu(U)}{r_\nu(U)}\right] &= \mathbb{E}_{U\sim\phi}\left[-\log r_\nu(U)\right] &(61)\\ &= D_{\mathrm{KL}}(\phi\|\nu). &(62)\end{aligned}$$

Hence,

$$D^{D_{\mathrm{KL}}}_{x_j,a_j} = \max\left\{D_{\mathrm{KL}}(\nu\|\phi), D_{\mathrm{KL}}(\phi\|\nu)\right\}. \tag{63}$$

Rosenbaum's sensitivity model. By Lemma 3, Eq. (11) in NeuralCSA, the Rosenbaum divergence is

$$D^{\mathrm{Ros}}_{x_j,a_j} = \max\left(\sup_{u_1,u_2}\rho(x_j,u_1,u_2,a_j), \sup_{u_1,u_2}\rho(x_j,u_1,u_2,a_j)^{-1}\right), \tag{64}$$

where $\rho(x,u_1,u_2,a)$ is the two-point ratio defined in NeuralCSA Eq. (17). NeuralCSA Eq. (18)–(20) establishes that $\rho(x,u_1,u_2,a) = \mathrm{OR}(\mathbb{P}(a \mid x,u_1), \mathbb{P}(a \mid x,u_2))$. Each marginal full-propensity ratio reduces to r_ν under our reparameterization (by the same calculation as the single-point case in B.6 above), so

$$\begin{aligned}\rho(x_j,u_1,u_2,a_j) &= \frac{r_\nu(u_2)}{r_\nu(u_1)}, &(65)\\ D^{\mathrm{Ros}}_{x_j,a_j} &= \frac{\sup_u r_\nu(u)}{\inf_u r_\nu(u)}. &(66)\end{aligned}$$

(The sup-over-u_1,u_2 of a ratio reduces to the ratio of sup over numerator and inf over denominator.)

B.7 Monte Carlo estimation under the reparameterization

For all three sensitivity models, the divergence reduces to a functional of $r_\nu(u) = \nu(u)/\phi(u)$. Both ν (via the normalizing flow) and ϕ (the standard normal density) are tractable, so r_ν is directly computable at any u.

For Monte Carlo estimation, the natural sampling distribution depends on the divergence. For MSM and Rosenbaum, the divergence is a sup/inf over u — sample support determines coverage, and any distribution with broad coverage of the relevant region suffices. We sample $u^{(j)} \sim \nu$ via the flow (which has sufficient support for the regions where r_ν is large). For f-sensitivity, the divergence is an expectation under ϕ, so we sample $u^{(j)} \sim \phi$ directly. For the KL reverse term $\mathbb{E}_\phi[r_\nu^{-1}\log r_\nu^{-1}]$, importance reweighting from ν-samples gives $\mathbb{E}_\nu[r_\nu^{-2}\cdot(-\log r_\nu)]/\mathbb{E}_\nu[r_\nu^{-1}]$, but since ν samples are already produced as a byproduct of the causal-query estimator (B.8), we reuse them.

B.8 THE CAUSAL QUERY UNDER THE REPARAMETERIZATION

The CAPO query at (x_j, a_j) is

$$\mathcal{Q}(x_j, a_j; \mathbb{P}) = \mathbb{E}_{U\sim\mathbb{P}(U|x_j)}\big[f_Y^\star(x_j, U, a_j)\big]. \tag{67}$$

Using $\mathbb{P}(U \mid x_j) = \pi\phi + (1-\pi)\nu$:

$$\mathcal{Q}(x_j, a_j; \mathbb{P}) = \pi \underbrace{\mathbb{E}_{U\sim\phi}[f_Y^\star(x_j, U, a_j)]}_{=:\mathcal{Q}_0(x_j, a_j; \mathcal{S}_i)} + (1-\pi)\mathbb{E}_{U\sim\nu}[f_Y^\star(x_j, U, a_j)]. \tag{68}$$

The first term $\mathcal{Q}_0$ is fixed by $\mathcal{S}_i$ (i.e., it does not depend on the optimization variable ν) and equals the queried-arm CAPO under unconfoundedness for $\mathcal{S}_i$. It contributes a constant to the Lagrangian objective and may be precomputed once per query. Only the second term is optimized through ν.

The standard ξ-mixture sampler (see Eq. 6 in NeuralCSA) implements the joint expectation: $\xi \sim$ Bernoulli(π), $\tilde{U} \sim \phi$, then $U = \xi\tilde{U} + (1-\xi)f_\eta(\tilde{U})$ where f_η is the flow-pushforward, with $f_Y^\star$ applied to U. The samples $u^{(j)} \sim \nu$ used for the divergence (see Appendix B.7) are exactly the $\xi = 0$ branch of this sampler, so the two estimators share a single MC budget.

B.9 MANSKI (NO-ASSUMPTIONS) LIMIT

As $\Gamma \to \infty$, the GTSM constraint becomes vacuous and ν ranges freely over distributions on $\mathcal{U}$. The maximum of the upper-bound CAPO is attained by a ν concentrated at the argmax of $f_Y^\star(x_j, \cdot, a_j)$:

$$\theta^+_{\text{Manski}}(x_j, a_j; \mathcal{S}_i) = \pi^{\mathcal{S}_i}(a_j \mid x_j)\mathcal{Q}_0(x_j, a_j; \mathcal{S}_i) + \big(1 - \pi^{\mathcal{S}_i}(a_j \mid x_j)\big) \sup_u f_Y^\star(x_j, u, a_j). \tag{69}$$

This is the standard Manski no-assumptions bound (Manski, 1989): a propensity-weighted average of the queried-arm CAPO and the worst-case counterfactual outcome. The reparameterization construction recovers it exactly in the $\Gamma \to \infty$ limit, providing a useful diagnostic and confirming that the bounds nest correctly.

C IMPLEMENTATION DETAILS

C.1 PFN ARCHITECTURE

Our foundation model for causal sensitivity analysis is a prior-data fitted network over mixed context/query sequences. Each input sequence contains N observational context rows $(X_i, A_i, Y_i, \text{NaN})$ followed by m query rows $(x_j, a_j, \text{NaN}, \Gamma_j)$, with Γ masked on context rows and Y masked on query rows. The split point is passed to the transformer as `single_eval_pos` $= N$. Separate input encoders for X, A, Y, and Γ handle NaNs explicitly via value-mask pairs; encoded representations are concatenated along the feature axis, processed by the per-feature transformer, and pooled by averaging over features.

The decoder has two independent Gaussian-mixture heads, where each returns posterior predictive distributions:

$$q_\phi^\downarrow(\theta^- \mid D_N, x, a, \Gamma), \qquad q_\phi^\uparrow(\theta^+ \mid D_N, x, a, \Gamma) \tag{70}$$

for the lower and upper bounds, each with K Gaussian components. Architecture and training hyperparameters are listed in Table 3. The total parameter count is `2,693,279`.

C.2 TRAINING LOSS

Each training row is a tuple $(D_N, x_j, a_j, \Gamma_j, b_j, \theta_j^\star)$ with bound type $b_j \in \{\downarrow, \uparrow\}$ and label $\theta_j^\star$ from Section 5. The lower and upper heads are trained by Gaussian-mixture NLL on their matching rows: $\mathcal{L}_{\text{NLL}}^\downarrow$ averages over the index set $\mathcal{I}_\downarrow$ of lower-bound rows, $\mathcal{L}_{\text{NLL}}^\uparrow$ over $\mathcal{I}_\uparrow$.

We additionally impose a soft monotonicity regularizer over Γ. For each query group (D_N, x_j, a_j) sampled with g sensitivity levels $\Gamma_{j,1} \le \cdots \le \Gamma_{j,g}$, let $\bar{\theta}_{j,r}^\downarrow$ and $\bar{\theta}_{j,r}^\uparrow$ denote the predictive means of

the two heads at $\Gamma_{j,r}$. The penalty enforces non-increasing lower bounds and non-decreasing upper bounds via a zero-margin hinge:

$$\mathcal{L}_{\text{mono}} = \frac{1}{|\mathcal{G}|(g-1)} \sum_{(D_N, x_j, a_j) \in \mathcal{G}} \sum_{r=1}^{g-1} \left[\text{ReLU}\left(\bar{\theta}^{\downarrow}_{j,r+1} - \bar{\theta}^{\downarrow}_{j,r} \right) + \text{ReLU}\left(\bar{\theta}^{\uparrow}_{j,r} - \bar{\theta}^{\uparrow}_{j,r+1} \right) \right]. \quad (71)$$

The total loss is $\mathcal{L} = \mathcal{L}^{\downarrow}_{\text{NLL}} + \mathcal{L}^{\uparrow}_{\text{NLL}} + \beta_{\text{mono}} \mathcal{L}_{\text{mono}}$ with $\beta_{\text{mono}} = 1$ (in practice the penalty is active only at the start of training and contributes negligibly thereafter).

The optimizer, schedule, and batch composition (i.e., DGPs per step B, query groups per DGP m', and Γ-points per group g) are listed in Table 3. Unspecified arguments use code defaults.

Table 3: PFN architecture and training hyperparameters.

Architecture	
Embedding dimension	128
Attention heads	4
Feedforward dimension	512
Transformer layers	10
Mixture components per head (K)	5
Training	
Optimizer	Adam
Learning rate	10^{-3}
Weight decay	10^{-5}
Gradient clipping (norm)	1.0
LR schedule	ReduceLROnPlateau (factor 0.5, patience 5)
Max epochs / early-stopping patience	150 / 50
Train/val split	0.8/0.2
Batch composition	
DGPs per step (B)	32
Query groups per DGP (m')	64
Γ-points per group (g)	10
Query rows per DGP ($m = m'g$)	640
Sequence length ($N + m$)	$1024 + 640 = 1664$
Loss	
Monotonicity weight (β_{mono})	1.0

C.3 Aggregation to APO, CATE, and ATE

The model is trained on CAPO bounds; APO, CATE, and ATE bounds follow as deterministic post-processing. Let $\theta^{-}_{a}(x,\Gamma) \leq \mathbb{E}[Y(a) \mid x] \leq \theta^{+}_{a}(x,\Gamma)$ be the CAPO bounds returned by the model, and let $\{x_i\}_{i=1}^{M}$ be an empirical covariate distribution (in-sample or held-out). Then

$$\theta^{\pm}_{\text{APO}}(a,\Gamma) = \tfrac{1}{M} \textstyle\sum_i \theta^{\pm}_{a}(x_i,\Gamma), \quad (72)$$

$$\theta^{-}_{\text{CATE}}(x,\Gamma) = \theta^{-}_{1}(x,\Gamma) - \theta^{+}_{0}(x,\Gamma), \quad \theta^{+}_{\text{CATE}}(x,\Gamma) = \theta^{+}_{1}(x,\Gamma) - \theta^{-}_{0}(x,\Gamma), \quad (73)$$

$$\theta^{\pm}_{\text{ATE}}(\Gamma) = \tfrac{1}{M} \textstyle\sum_i \theta^{\pm}_{\text{CATE}}(x_i,\Gamma). \quad (74)$$

These intervals are valid since each component CAPO is bounded prior to aggregation; they are not in general sharp, because sharp aggregation may exploit dependence between endpoint-achieving full distributions across covariates and arms. PPDs propagate by Monte Carlo sampling from the lower/upper Gaussian-mixture heads and applying the same transformations samplewise.

C.4 Prior-dataset generation

We generate 10,000 synthetic SCMs with $N = 1024$ observations, $d_x = 10$ covariates, scalar treatment and outcome, and a one-dimensional latent U. Outputs are normalized with $\epsilon = 10^{-6}$ and the outcome model is wrapped so that frontier construction operates in normalized coordinates. Two label-generation regimes were run, one per sensitivity model: an *analytical MSM* run (closed-form bounds) and an *optimization-based KL* run (Section 5). Configurations are summarized in Table 4. The training-facing interface consists of per-DGP

files `queries_{id}.csv` and `frontier_points_{id}.csv`, storing $(\texttt{query_id}, x, a)$ and $(\texttt{query_id}, \texttt{bound_type}, \Gamma^\star, \theta^\star)$, respectively.

Outcome standardization. Within the training prior, each synthetic DGP's outcomes are standardized with the mean and standard deviation of both potential outcomes, which are available by construction. At test time only observed outcomes exist, so all evaluations standardize with the mean and standard deviation of the observed context outcomes. For each DGP, the two standardizations differ by an affine map close to the identity (ratio of scales: median 1.002; 90% of held-out DGPs within [0.96, 1.06]). Because the sensitivity bounds are equivariant under positive affine maps of Y, the only effect is a slight mismatch in input scale, whose cost is included in all reported results.

Table 4: Label-generation runs. Frontier rows per DGP = (query covariate rows) × 2 arms × (grid size) × 2 bounds.

	Analytical MSM	**Optimization-based KL**
Sensitivity model	MSM (closed form)	KL f-sensitivity
Hardware	CPU (parallelized)	`GPU Nvidia H200`
DGP batch size B	—	128
Grid parameter	Γ	λ
Grid range	$[1.0,\ 5.0]$	$[0.08,\ 2.0]$
Grid size	50	50
Grid spacing	bounded Pareto, $\beta = 0$ (log-uniform)	log-uniform
Per-DGP grid randomization	yes	no
Query covariate rows per DGP	2048	2048
Frontier rows per DGP	409,600	409,600
MC samples k_{train}, k_{eval}	128 (single bank, reused)	128, 4096
Latent sampler	—	Sobol (sample seed 123)
Flow architecture	—	1D rational-quadratic spline
Bins / tail bound	—	16 / 6.0
Min bin width / height / derivative	—	10^{-3} each
Optimizer steps per λ (base)	—	350
Step λ-schedule	—	`inverse_sqrt_lr` (mult. cap 2.0)
LR λ-schedule	—	`sqrt`, $\lambda_{\text{ref}} = 0.25$, min mult. 0.40
LR base	—	10^{-3}
Loss reduction	—	`per_dgp_sum`

C.5 Computational complexity

For the optimization-based frontier construction, a single optimization step at fixed λ evaluates the spline flow, density ratio, divergence estimate, and outcome model on Bmk latent samples, giving per-step cost $O(Bmk)$ (with B the DGP batch size, m queries per DGP, k MC samples) plus one backward pass through the spline parameters. Total cost across the prior is

$$O(N_{\text{DGP}} \cdot |\Lambda| \cdot T \cdot m \cdot k)\,, \tag{75}$$

where T is the average number of optimizer steps per λ (bounded by the schedule at 700 for the KL run after the lambda-dependent step multiplier). This cost is paid once during prior-data generation; at test time the trained PFN replaces the per-query optimization with a single forward pass.

Warm-starting reduces effective T after the first λ: solved spline parameters initialize the next λ-solve, while Adam momentum is reset across λ changes to avoid stale state from a different scalarized objective.

C.6 Test-time inference

Test-time inference is a single transformer forward pass over a sequence of length $(N + m)$, returning the parameters of both heads' Gaussian mixtures. Posterior predictive means are the GMM means; credible intervals are GMM quantiles, estimated by Monte Carlo sampling from the predicted mixtures. APO, CATE, and ATE add only deterministic averaging and differencing on top of the CAPO PPDs (Section C.3).

C.7 Reproducibility

Run configurations appear in Tables 3 and 4. Code release: `https://anonymous.4open.science/r/CSA-PFN-Anon-FA52/`.

D Additional Experiments

D.1 Warm-start ablation.

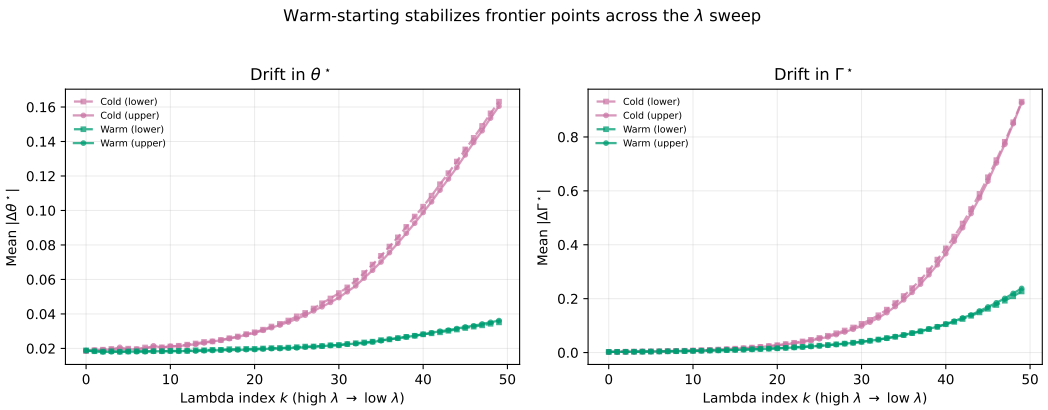


Figure 6: **Warm start ablation.** Drift in optimized causal query bound $\theta^\star$ (left) and optimized sensitivity parameter $\Gamma^\star$ (right) along the λ-sweep ($k = 0$ at $\lambda_{\max} = 2.0$, $k = 49$ at $\lambda_{\min} = 0.08$.) measured against a high-budget reference (1000 steps). **Warm starting achieves lower regret solutions while** $1.90\times$ **faster.**

We use a one-dimensional rational-quadratic spline flow with 8 bins, tail bound 6.0, $k_{\text{train}} = 128$ Monte Carlo samples, and $k_{\text{eval}} = 1024$ final-evaluation samples. Both early-stopped runs use the same optimizer settings: base maximum 350 steps, inverse-square-root λ-dependent step schedule with multiplier capped at 2.0, square-root λ-dependent learning-rate schedule with $\lambda_{\text{ref}} = 0.25$ and minimum multiplier 0.40, and early stopping after 100 minimum steps with checks every 25 steps, patience 3, absolute tolerance 2×10^{-4}, and relative tolerance 5×10^{-4}. Regret is computed relative to a warm-started fixed-budget reference with no early stopping and base maximum 1000 optimization steps. Experiment was run on an `Nvidia H200` GPU at $\sim 90\%$ utilization (no other processes running), taking in total approx. 1 hour.

D.2 Evaluation details

Cohorts. All in-prior evaluation DGPs are drawn with random seeds disjoint from those used for training and checkpoint selection. The MSM evaluation uses 512 DGPs, the KL evaluation 64 DGPs, the per-instance comparison 32 DGPs, and the point-identified comparison 512 DGPs. Each context contains $N = 1024$ observations unless stated otherwise.

Metrics. A row is one (DGP, query x, arm a, Γ, bound head). Coverage is the share of rows whose reference bound lies in the central 90% interval of the corresponding head's PPD. RMSE pools squared errors of the PPD mean over all rows before taking the square root and is reported in units of the observed-outcome standard deviation of each context. All intervals are 95% percentile intervals from 2,000 bootstrap replicates that resample whole DGPs; paired contrasts resample DGPs jointly for both methods.

References. For the MSM, reference bounds are computed in closed form from the DGP's outcome map and propensity. For KL, no closed form exists; the reference is a high-budget solve of the same Lagrangian frontier problem, and the attained divergence Γ_{KL} indexes the frontier.

D.3 EXAMPLE PREDICTIONS

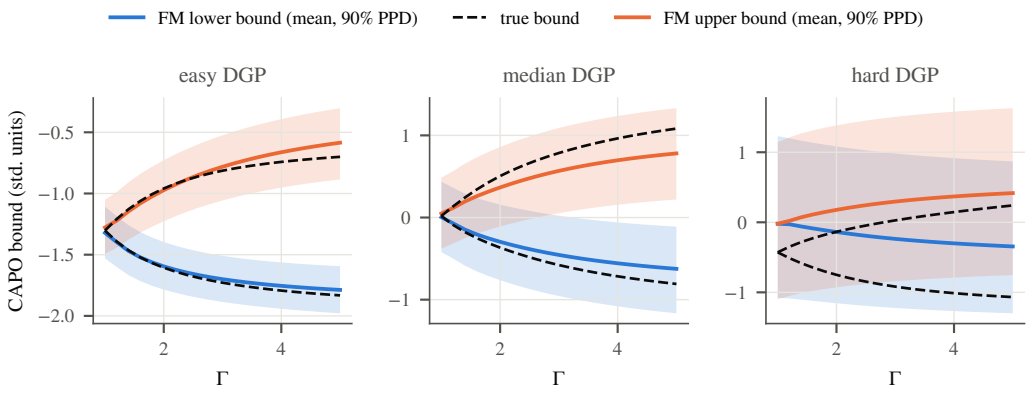


Figure 7: **Example predictions (MSM).** Posterior predictive mean and 90% interval of the lower (blue) and upper (orange) bound heads versus the closed-form bounds (dashed), for an easy, a median, and a hard held-out DGP (ranked by RMSE among 512 DGPs; control arm). Offsets at $\Gamma = 1$, where both bounds equal the point-identified CAPO, reflect CAPO estimation error.

D.4 PER-INSTANCE BASELINE DETAILS

We reimplemented the two-stage NeuralCSA procedure (Frauen et al., 2024). Stage 1 fits a conditional normalizing flow for the outcome given covariates and treatment on the observational context; Stage 2 optimizes the counterfactual latent distribution under the sensitivity constraint for each query, arm, and target sensitivity level. Its errors are dominated by Stage 1: the fitted conditional distributions are miscalibrated on held-out observations (mean Kolmogorov–Smirnov statistic of the probability integral transform 0.18). Given the true outcome map instead of the fitted flow (an infeasible oracle), the same Stage 2 attains RMSE 0.04 for both sensitivity models, against 0.43–0.45 for the full procedure; the gap to our FM is therefore attributable to Stage 1. For KL, Stage 2 also fails to reach the largest target ($\Gamma_{\mathrm{KL}} = 2$) for at least 91% of queries. We report these as diagnostics rather than exclusions. Timings in Figure 5 are medians over repeated runs on an otherwise idle GPU; the batched variant solves all instances jointly, which is the most favourable deployment of the baseline we could construct.

D.5 POINT-IDENTIFIED LIMIT

At $\Gamma = 1$, both sharp bounds equal the point-identified CAPO, so the two heads should agree. Across 512 held-out DGPs, the mean absolute gap between the heads' predictive means is 0.014 (95th percentile 0.039), and each head covers the point-identified CAPO at rates 0.889/0.888. This also gives practitioners a diagnostic for prior mismatch that requires no ground truth: the $\Gamma = 1$ prediction can be compared with any standard CAPO or CATE estimator fitted on the same data. Table 5 compares the implied CATE (midpoint at $a = 1$ minus midpoint at $a = 0$) with CausalFM (Ma et al., 2025), a foundation model for point-identified CATE estimation, on the same contexts and queries. Our FM attains a lower PEHE, and CausalFM has the lower absolute error on only 38% of queries.

Table 5: **Point-identified limit.** CATE accuracy at $\Gamma = 1$ on 512 held-out DGPs (standardized units; 95% whole-DGP bootstrap intervals). FM coverage and width refer to the CATE distributions derived from the lower and upper heads, respectively.

Method	PEHE [95% CI]	Bias	90% coverage	90% width
CausalFM (direct CATE)	0.435 [0.416, 0.453]	0.019	0.908	1.46
FM (ours, Γ=1 midpoint)	**0.350 [0.335, 0.366]**	0.003	0.926 / 0.924	1.22 / 1.22

D.6 OUT-OF-PRIOR STRESS TESTS

We construct four families of DGPs outside the training prior, each changing a single axis (64 DGPs each): oscillatory outcome mechanisms, heavy-tailed Student-t_3 conditional outcomes, overlap beyond the range the prior produces, and sensitivity levels extrapolated to $\Gamma = 9$.

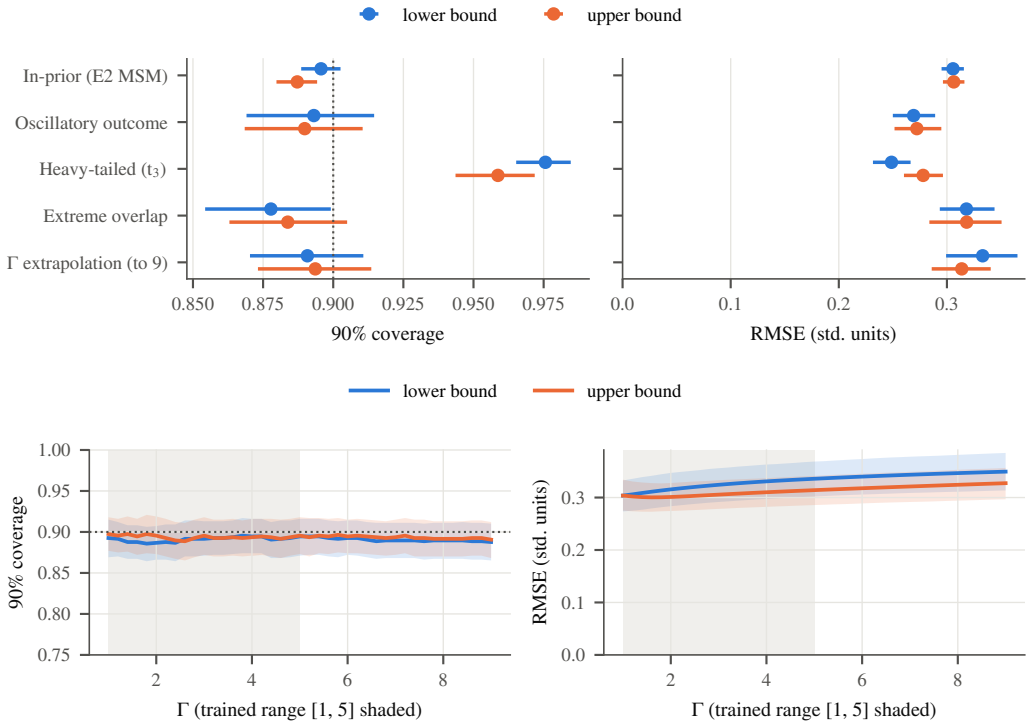


Figure 8: **Out-of-prior stress tests.** (Top) 90% coverage and RMSE (95% CIs) per bound head for four families outside the training prior versus held-out in-prior DGPs. (Bottom) Coverage and RMSE for sensitivity levels up to $\Gamma = 9$; shaded: trained range.

D.7 Context size and covariate dimension

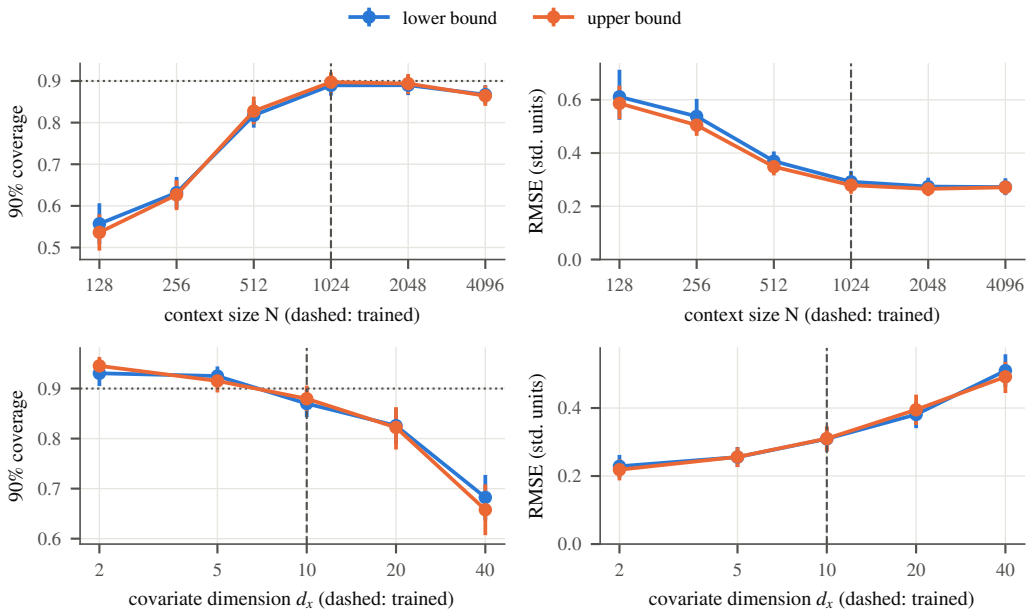


Figure 9: **Context size and covariate dimension.** Coverage and RMSE as the context size N (top) and covariate dimension d_x (bottom) move away from the training configuration (dashed). Context sizes share nested observations; dimensions are separate DGP cohorts (64 DGPs per condition).

D.8 Structural validity of predicted curves

Sharp bounds are monotone in Γ and ordered, $\theta^- \leq \theta^+$. The raw predictive means of 32% of the MSM curves cross at some Γ, but the crossings are small: median 0.007 and 99th percentile 0.035 standardized units, against a typical bound error of 0.31 (Figure 10). We therefore report jointly projected mean curves: for each (DGP, query, arm), the lower and upper mean curves are projected in Euclidean norm onto the set of monotone, non-crossing curves, which never increases squared error with respect to any valid reference. Quantiles are not modified.

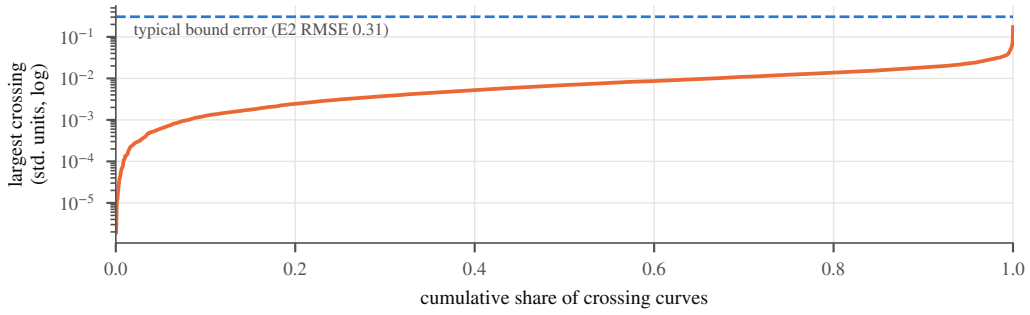


Figure 10: **Magnitude of crossings.** Largest crossing (lower minus upper) per predicted mean curve, sorted, among the curves that cross; dashed: typical bound error.

D.9 Training diagnostics (MSM)

Loss curves. We monitor the total negative log-likelihood on the training and validation splits throughout optimization. The curves provide a basic diagnostic for convergence and potential overfitting. The selected checkpoint corresponds to the epoch with the lowest validation loss.

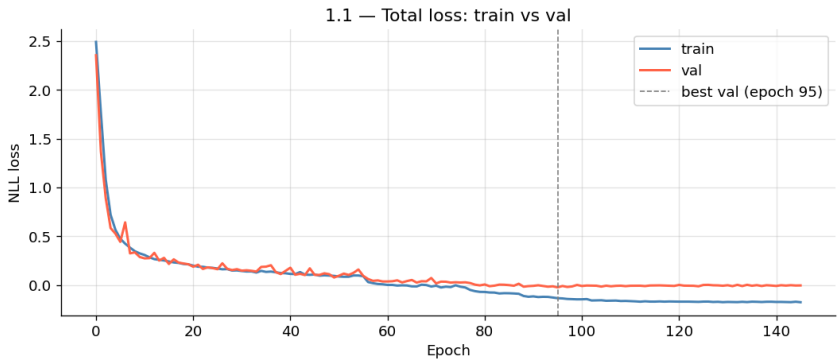


Figure 11: Training and validation negative log-likelihood decrease over epochs for the MSM foundation model.

Performance. We evaluate predictive performance using calibration and point-error diagnostics for the posterior predictive distributions of both bound heads. Coverage is reported for the central posterior predictive intervals and compared against the nominal levels. We also track the posterior predictive interval width to assess uncertainty contraction during training. Bias and RMSE summarize the accuracy of the predictive means.

Monotonicity regularization. The MSM bounds should widen monotonically as the sensitivity level increases. We therefore track both the frequency and size of monotonicity violations during training. These diagnostics show whether the soft regularization term is active and whether violations persist after convergence. The penalty is compared against the total training loss to assess its relative contribution to optimization.

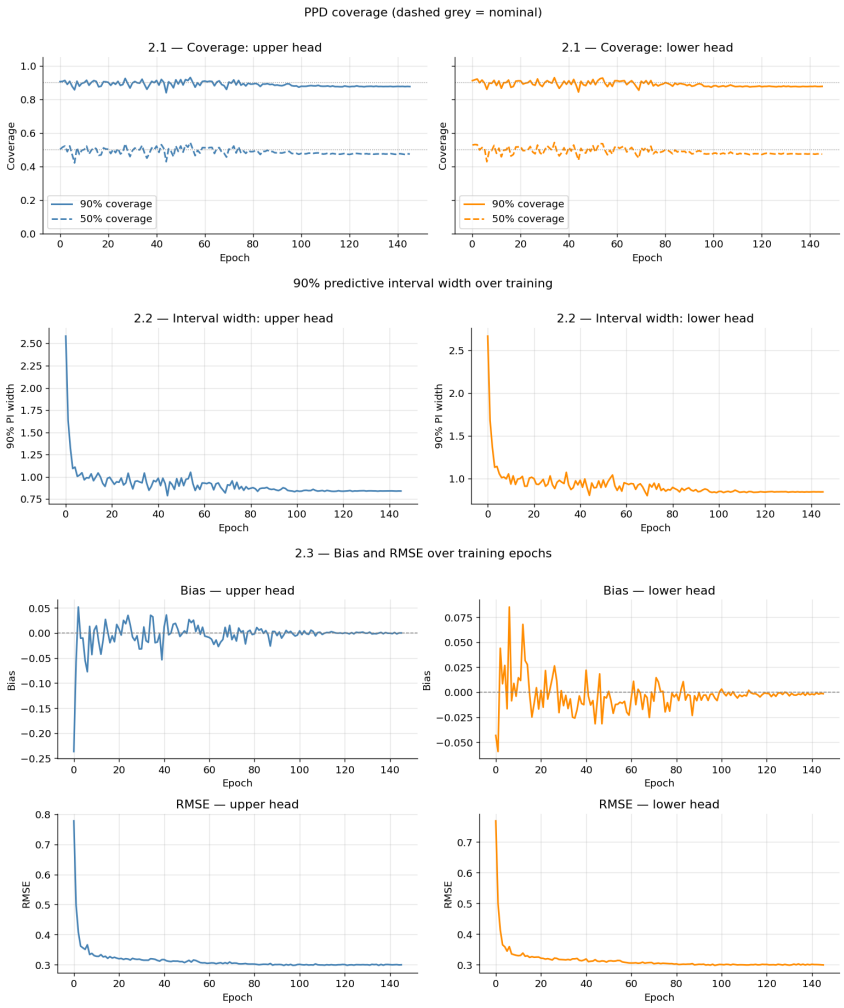


Figure 12: (Top) Posterior predictive coverage remains close to the nominal 90% and 50% levels for both bound heads. (Middle) The 90% posterior predictive interval width contracts during training for both bound heads. (Bottom) Bias and RMSE decline during training for both bound heads.

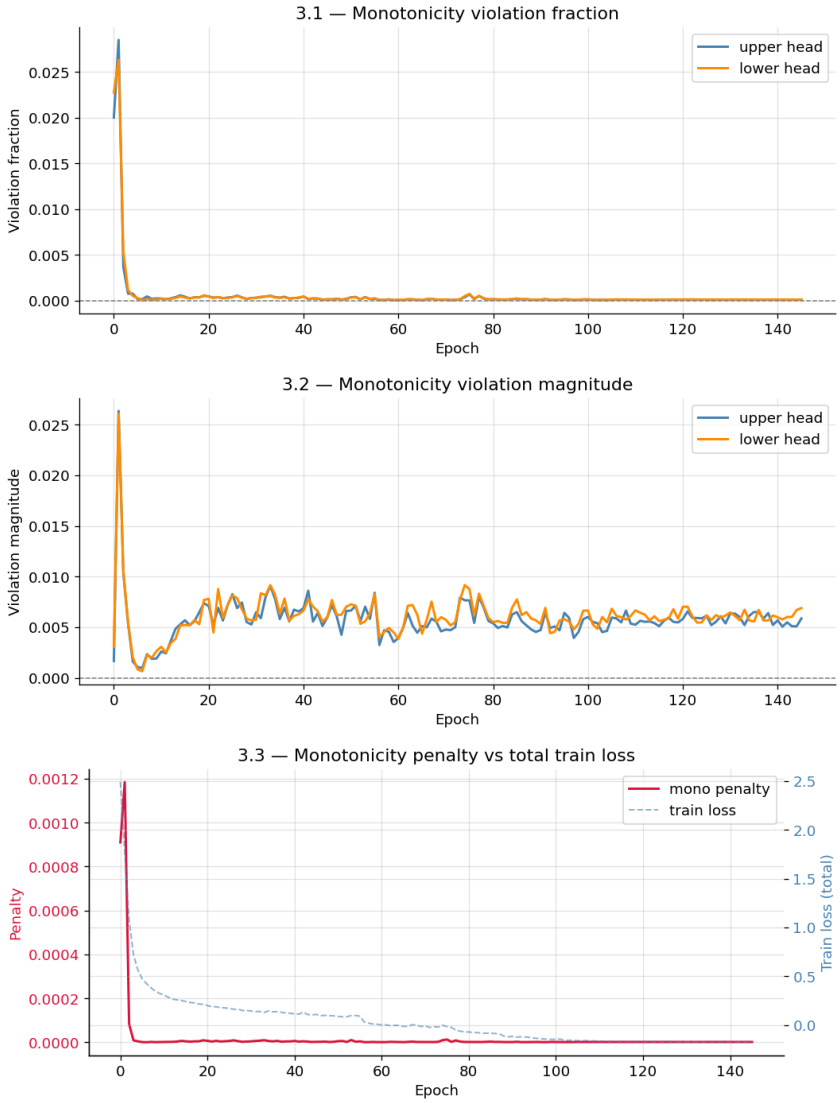


Figure 13: (Top) The fraction of monotonicity violations rapidly approaches zero for both bound heads. (Middle) The average monotonicity violation magnitude remains small after the initial training phase. (Bottom) The monotonicity penalty becomes negligible as the total training loss stabilizes.

E LIMITATIONS

As with all prior-data fitted networks, the reliability of our foundation model depends on how well the synthetic prior covers the data-generating mechanisms encountered at test time. The point-identified limit at $\Gamma = 1$ offers a practical check here, since the prediction can be compared with any standard CAPO or CATE estimator fitted on the same data (Appendix D.5). Our models are trained at a fixed context size and covariate dimension, and accuracy decreases as these move far from the training configuration (Appendix D.7); training across a range of sizes and dimensions is a natural extension. For sensitivity models without closed-form bounds, training labels are obtained by numerical optimization and are therefore approximate. Finally, our construction applies to GTSMs with convex sensitivity divergences and linear causal queries, and we focus on binary treatments and scalar outcomes; extending it to continuous treatments and other estimands is an interesting direction for future work.