PROXIMALFM: AMORTIZED PROXIMAL CAUSAL INFERENCE UNDER HIDDEN CONFOUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard causal identification methods often assume no unmeasured confounding and can fail when relevant confounders are unobserved. Proximal causal inference instead uses proxy variables to identify effects under hidden confounding and additional assumptions. However, nonparametric proximal estimation can be challenging in practice: recovering causal estimands such as the conditional average treatment effect (CATE) requires solving an ill-posed integral equation that is data-hungry, hyperparameter-sensitive, and optimization-unstable. Bayesian inference for such models provides a desirable alternative, mitigating these difficulties by regularizing through the prior. However, computing a posterior is itself challenging, as a typical likelihood function will include latent variables. Following the recent success of tabular foundation models in backdoor, instrumental variable, and frontdoor settings, we propose that prior-data fitted networks (PFNs) are uniquely suited to resolve this bottleneck. Indeed, by training on synthetic data sampled from compliant structural causal models with access to oracle counterfactuals, we simplify the task substantially, amortizing the implied Bayesian operator inversion into a single transformer forward pass. Compared to prior literature that focuses primarily on point estimation, our model, PROXIMALFM, explicitly targets the Bayesian posterior distribution of the CATE. One unique aspect of this problem is that we need to provide Monte Carlo estimates of the oracle CATEs, leading to a novel variation of PFNs that accounts for the added stochastic error. PROXIMALFM is competitive with existing estimators across a diverse set of proximal regimes, with its clearest advantage under substantial latent confounding and informative proxies. It attains this performance without dataset-specific tuning, in a single amortized forward pass.

1 INTRODUCTION

Causal inference from observational data is a fundamental task across disciplines, yet its validity typically hinges on the *no unmeasured confounding assumption*. This assumption requires that investigators have measured a sufficiently rich set of covariates to ensure that supervised regression and classification functions can, in theory, also provide how an outcome varies under *intervention* on a possible cause of interest (Hernán & Robins, 2020), with specialized estimation methods improving on off-the-shelf uses of supervised learning. In practice, however, there are often unmeasured variables that are plausible confounders (e.g., unrecorded disease severity).

One of the most common tasks in causal inference is estimating the *conditional average treatment effect* (CATE) function: how an outcome variable Y potentially varies in expectation, as we vary a treatment variable A by *intervening* on the latter. This is a function of an observed realization of a vector of covariates X. Using the notation $Y(a)$ to denote the *potential outcome* under a hypothetical intervention on A at level a (Rubin, 1972), the CATE function for a *binary* A is given by

$$Q(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]. \tag{1}$$

This is identifiable from observational data if unconfoundedness holds. That failing, other assumptions are needed. *Proximal causal learning* has emerged as a principled framework to identify and estimate causal effects when hidden confounders cannot be ruled out (Tchetgen Tchetgen et al., 2020). The main idea is to exploit auxiliary variables, *proxies*, which are associated with the hidden confounders without fully removing their influence. However, using proxy information while

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.

avoiding parametric models requires solving ill-posed integral equations, leading to a difficult estimation problem. As a result, current nonparametric (Ghassami et al., 2022; Sverdrup & Cui, 2023; Yang et al., 2023) and neural estimation methods (NMMR, Kompa et al., 2022) are computationally demanding, requiring intensive, dataset-specific optimization.

Simultaneously, the machine learning community is shifting toward amortized inference via foundation models. Prior-data fitted networks (PFNs, Müller et al., 2022) have demonstrated that transformers can learn to perform Bayesian inference through in-context learning. Models like TabPFN and its successors have shown state-of-the-art performance on small tabular datasets without requiring retraining (Grinsztajn et al., 2026). As we review in the next section, recent work has extended PFN-style models to causal inference settings. However, they generally do not make use of proximal assumptions, limiting their utility in the presence of hidden confounders. Our contributions are:

1. **PROXIMALFM: a foundation model for proximal causal learning.** We introduce PROXIMALFM, the first tabular foundation model designed for proximal causal inference. It leverages proxy variables without requiring dataset-specific training at deployment. Causal effects are obtained in a single forward pass through a target dataset, providing a Bayesian solution to an ill-posed inverse problem without hyperparameter tuning.
2. **A structural prior and Monte Carlo CATE targets for proximal causal learning.** We introduce a diverse prior over structural causal models that enforces the proximal assumptions for identification. We use Monte Carlo integration to approximate the CATE. These approximations are then used as targets to train PROXIMALFM. This prior determines the inductive bias regularizing the proximal estimation problem amortized by PROXIMALFM.
3. **Comprehensive evaluation, from semi-synthetic data to physical systems.** We evaluate PROXIMALFM on diverse semi-synthetic proximal benchmarks that combine real-world covariate distributions with known, controlled causal mechanisms. Beyond semi-synthetic benchmarks, we take a first step towards evaluation on real physical systems using Causal Chambers (Gamella et al., 2025), a controlled physical testbed for causal inference.

1.1 RELATED WORK

Tabular foundation models Recent advances have demonstrated the potential of transformer-based foundation models for tabular prediction. TabPFN pioneered this approach for supervised learning tasks with tabular data (Hollmann et al., 2023; 2025; Grinsztajn et al., 2026). This success prompted a rapidly expanding literature. Examples include work on more efficient tabular in-context learning, such as TabICL (Qu et al., 2025; 2026), and tuning on real-world data, as in RealTabPFN (Garg et al., 2025). This progress has also motivated standardized evaluation. TabArena provides a living benchmark for IID tabular classification and regression (Erickson et al., 2026). Extending this evaluation beyond conventional IID settings, Purucker et al. (2026) find that current tabular foundation models are particularly strong on small- to medium-scale IID datasets, but do not yet consistently outperform conventional methods on non-IID, large-scale, or high-dimensional data.

Amortized causal learning The success of tabular models has motivated their application to causal inference. Causal foundation models are pretrained on a prior over causal data-generating processes and use in-context learning to estimate causal quantities on new datasets in a single forward pass. Stith et al. (2026b) provide a recent overview of this emerging literature.

Robertson et al. (2026) introduced Do-PFN, demonstrating that transformer-based foundation models can learn causal estimation procedures directly from synthetic data and perform zero-shot inference on unseen datasets. Balazadeh Meresht et al. (2026) introduced CausalPFN, a framework for doing amortized learning of causal effects identified through unconfoundedness. This setting was also considered in Luedtke & Chung (2024), under an adversarially chosen prior. Ma et al. (2026) proposed CausalFM, extending causal foundation models beyond unconfoundedness, such as the case of identifiable additive-error instrumental-variable (IV) settings. Rather than learning a single causal estimator, CausalFM trains separate models for distinct causal identification strategies, thereby preserving the classical separation between identification and estimation. Balazadeh et al. (2026) relaxes the assumption of additive-error in the IV setting to build in-context learners for *bounding* causal effects for instrumental variables. Different from proximal learning, such par-

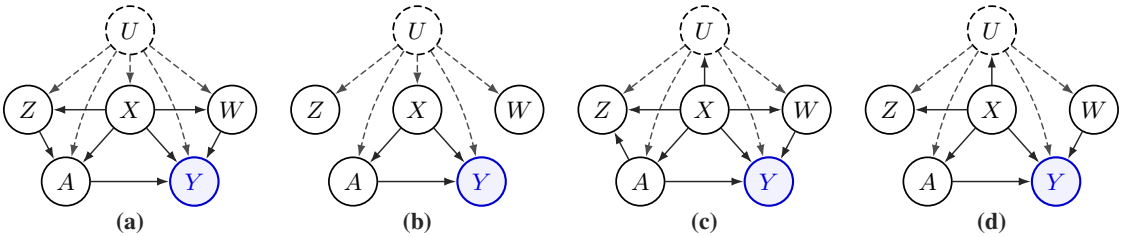


Figure 1: Examples of DAGs compatible with proximal causal identification assumptions. In every panel, Z and W are informative proxies for the unmeasured confounder U. Direct Z–A edges (in either direction) and a direct $W \to Y$ edge are optional.

tial identification IV methods do not typically exploit multiple proxies, nor exploit completeness assumptions (as defined in the next section) that could otherwise allow for identification.

Despite these advances, a foundation model designed for proximal causal inference remains absent from the literature.

2 Proximal causal inference

We consider an observational causal inference setting where the standard assumption of no unmeasured fails. Our objective is to estimate the CATE, as in Eq. (1), for binary treatments. Let the observed variables be $O = (L, A, Y)$, where $A \in \{0, 1\}$ denotes a treatment of interest, Y is the outcome, and L is a set of measured covariates. The unmeasured confounders are denoted by U.

In the presence of U, the standard exchangeability condition $Y(a) \perp\!\!\!\perp A \mid L$ is violated (Hernán & Robins, 2020). To achieve identification, we adopt the proximal causal learning framework (see, e.g., Miao et al., 2018; Tchetgen Tchetgen et al., 2020). This framework partitions L into three subsets, each with a distinct role. Let $L = (X, Z, W)$ where X contains baseline covariates, while Z and W are complementary proxies for U. Z is a *treatment-confounding* proxy: while it carries information about the latent U, it may otherwise be related directly to treatment assignment. Conversely, W is an *outcome-confounding* proxy which carries information about U and may otherwise directly shape the outcome. Figure 1 illustrates some admissible proximal directed acyclic graphs (DAGs), with formal restrictions stated in the following section.

2.1 Identifying assumptions

To identify the causal effect $Q(x)$, it suffices for the joint distribution of (X, U, Z, W, A, Y) to satisfy the following conditions:

Assumption 1 (Latent exchangeability)**.** *The potential outcomes are independent of the treatment assignment conditional on the observed covariates X and the unmeasured confounders U:*

$$Y(a) \perp\!\!\!\perp A \mid (X, U) \qquad \textit{for each } a \in \{0, 1\}.$$

Assumption 2 (Proxy exclusion restrictions)**.** *The proxies must satisfy conditional independencies relative to the treatment and outcome:*

- *Treatment-inducing proxy:* $Y \perp\!\!\!\perp Z \mid (A, X, U)$
- *Outcome-inducing proxy:* $W \perp\!\!\!\perp (A, Z) \mid (X, U)$

Assumption 3 (Latent positivity)**.** *Every observational unit has a non-zero probability of receiving any given treatment assignment, even when conditioning on the observed covariates and the unmeasured confounder:*

$$P(A = a \mid X, U) > 0 \quad \textit{for } a \in \{0, 1\}$$

Assumption 4 (Completeness)**.** *The proxies must capture enough variation in U to identify the causal effect. For any square-integrable measurable functions v of U and g of Z,*

$$\mathbb{E}[v(U) \mid Z = z, A = a, X = x] = 0 \textit{ for almost all } (z, a, x) \implies v(U) = 0 \textit{ a.s.},$$
$$\mathbb{E}[g(Z) \mid W = w, A = a, X = x] = 0 \textit{ for almost all } (w, a, x) \implies g(Z) = 0 \textit{ a.s.}$$

2.2 CLASSICAL AND MODERN ESTIMATION VIA BRIDGE FUNCTIONS

In addition to Assumptions 1–4, we assume the regularity conditions ensuring the existence of an *outcome confounding bridge function* $h_0(W, A, X)$. This function satisfies the Fredholm integral equation (Miao et al., 2018; Tchetgen Tchetgen et al., 2020)

$$\mathbb{E}[Y \mid Z, A, X] = \mathbb{E}[h_0(W, A, X) \mid Z, A, X]. \tag{2}$$

The bridge function enables identification of the conditional average treatment effect through the *proximal g-formula*

$$Q(x) = \mathbb{E}[h_0(W, 1, X) - h_0(W, 0, X) \mid X = x]. \tag{3}$$

Equation (3) identifies the effect conditional on X alone by averaging over the proxies.

Early approaches to recovering this bridge function focused on parametric proximal g-computation. Under linear specifications, proximal g-computation reduces to a two-stage estimation procedure closely related to instrumental variable methods. While computationally efficient, proximal two-stage least squares (`P2SLS`, Tchetgen Tchetgen et al., 2020) relies heavily on strict parametric assumptions. To capture more complex confounding mechanisms, recent two-stage methods have replaced linear regressions with highly flexible nonparametric approximators, such as kernel proxy variable (`KPV`, Mastouri et al., 2021) and deep feature proxy variable (Xu et al., 2021).

An alternative is to estimate the outcome bridge function by directly enforcing the conditional moment restriction equivalent to Equation (2). Proxy maximum moment restriction (`PMMR`, Mastouri et al., 2021) leverages a reproducing kernel Hilbert space, while neural maximum moment restriction (`NMMR`, Kompa et al., 2022) extends this approach by using neural networks.

A separate strand of methods are doubly-robust meta-learners for heterogeneous treatment effect estimation, e.g. `P-learner` from Sverdrup & Cui (2023) and the Forster-Warmuth learner of (Yang et al., 2023). They estimate an outcome bridge and a treatment bridge functions, then constructs a pseudo-outcome from them. These outcomes are used to fit a standard conditional mean regression mapping the domain of covariates X to treatment effect.

While these modern approaches are highly adaptive, they introduce their own practical hurdles. They demand complex, dataset-specific hyperparameter tuning and extensive model retraining from scratch for every new causal task. The kernel methods also require the computation of kernel matrices, which can impose computational and memory burden in large sample sizes. This motivates the search for an amortized alternative capable of directly estimating $Q(x)$ from observational context.

3 PROXIMALFM

We propose PROXIMALFM, a PFN for proximal causal inference. Pre-trained on synthetic proximal models, PROXIMALFM predicts a posterior distribution over $Q(x)$ from an observational context and query covariates x. Unlike existing approaches, it does this without explicitly solving an ill-posed inverse problem as an intermediate step. Instead, it directly outputs a CATE posterior from a single forward pass of a neural network.[1]

3.1 A SYNTHETIC PRIOR FOR PROXIMAL IDENTIFICATION

To train PROXIMALFM, we construct a prior Π over structural causal models (SCMs, Pearl, 2009). An SCM specifies a causal graph, structural functions mapping parent variables and exogenous noise to child variables, and distributions over the exogenous noise. Each SCM defines an episode: its structure stays identical throughout, while each of its units is generated by drawing fresh noise and propagating it through its structural equations:

$$\begin{aligned} X &= f_X(\epsilon_X), & U &= f_U(X, \epsilon_U), \\ Z &= f_Z(X, U, \epsilon_Z), & W &= f_W(X, U, \epsilon_W), \\ A &= f_A(X, U, Z, \epsilon_A), & \big(Y(0), Y(1)\big) &= f_Y(X, U, W, \epsilon_Y). \end{aligned} \tag{4}$$

[1] Our anonymous implementation of ProximalFM, including the synthetic prior and scripts to reproduce the experiments, is available at `https://anonymous.4open.science/r/ProximalFM/`. Trained model weights will be released on Hugging Face.

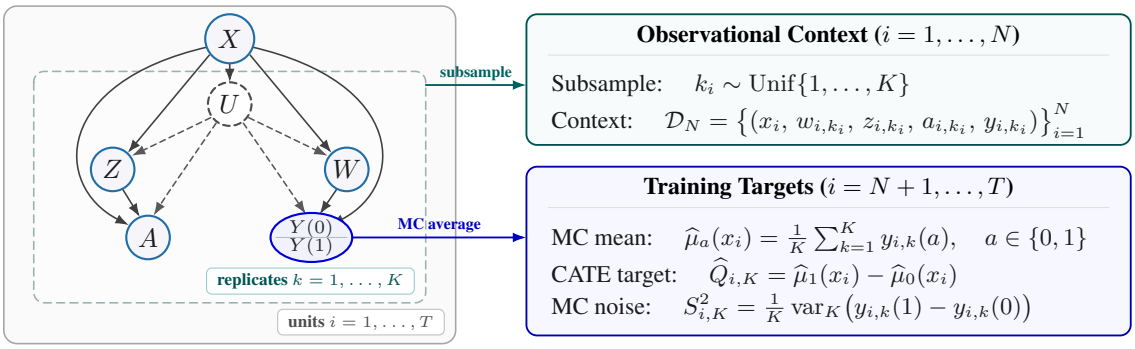


Figure 2: Synthetic-prior episode: each covariate x_i is fixed across K downstream SCM draws. One draw forms the observational context; replicate potential outcomes yield Monte Carlo CATE targets and their sampling variance. var_K is the unbiased sample variance of the K paired treatment effects.

The exogenous-noise vectors $\epsilon_{(.)}$ are drawn independently for every unit from distributions specified by the episode's SCM. X, U, Z, and W may contain continuous and categorical columns. $A \in \{0, 1\}$ is drawn from a Bernoulli propensity. $Y(0)$ and $Y(1)$ are continuous. A generated unit is the tuple $(X, U, Z, W, A, Y(0), Y(1))$: both potential outcomes are generated, but the observational dataset keeps only $Y = Y(A)$.

Π satisfies two complementary objectives. First, each SCM must be valid and satisfy the proximal assumptions. Second, the prior must be *diverse*: we build on the synthetic prior of TabICLv1 (Qu et al., 2025), in which each causal mechanism is a randomly initialized multi-layer perceptron. The prior Π induces variation across SCMs in their dimensions, nonlinear functional forms, feature types and noise distributions, as well as in confounding strength, treatment overlap, and treatment-effect heterogeneity. An overview of the synthetic prior is available in Appendix B.1.

Enforcing proximal assumptions. By construction, every SCM drawn from Π is designed to satisfy the identification conditions of Section 2.1 (see Appendix B.3 for empirical diagnostics):

- **Structural constraints (1 & 2).** By construction in (4), mutual independence of the exogenous noises $\epsilon_{(.)}$ and the specified mechanism inputs encode the required exclusions. The corresponding conditional independencies follow immediately for every sampled SCM.
- **Positivity (3).** Guaranteed by clamping treatment assignment probabilities to $[\delta, 1-\delta]$ with $\delta > 0$ before sampling A.
- **Completeness (4).** Unlike structural graph constraints, completeness is a more subtle, distribution-level condition: it is untestable in observational data and cannot be strictly guaranteed by construction for arbitrary non-linear mechanisms (Tchetgen Tchetgen et al., 2020). The prior therefore relies on a heuristic requiring the dimensions of W and Z to be greater than the one of U. Their cardinalities also dominate the one of U (see Appendix B.2.4).

Contexts & Monte Carlo targets. A naive training framework partitions each dataset of T units generated from the prior, $\{(x_i, u_i, z_i, w_i, a_i, y_i(0), y_i(1))\}_{i=1}^{T}$, into N context units and $T - N$ query units. The context becomes $\mathcal{D}_N = \{(x_i, z_i, w_i, a_i, y_i\}_{i=1}^{N}$ while the query inputs and targets are $\{(x_i, y_i(1) - y_i(0))\}_{i=N+1}^{T}$. The PFN then uses the query individual treatment effects (ITE, $y_i(1) - y_i(0)$) as targets to be predicted from $(\mathcal{D}_N, x_i)$. While simple, this framework is flawed. First, training on ITEs would target their conditional distribution, whereas our estimand is the CATE $Q(x_i)$. Second, the ITE distribution depends on the joint law of $(Y(1), Y(0))$. Our prior defines this law, but it is typically unknown and not identified from observational data, so the learned ITE uncertainty would only reflect our choice of prior.

We instead approximate the CATE of the query units. As $Q(x)$ is an average over the heterogeneity in U, W and noise, it cannot be generated in a single draw. Each query x_i is kept fixed and we run the downstream SCM K times with independent noise. We obtain the individual effects $\tau_{i,k} =$

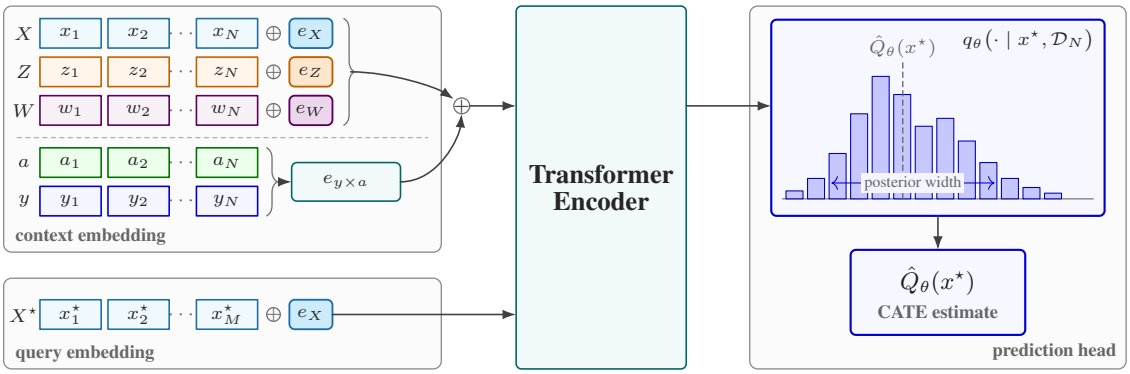


Figure 3: PROXIMALFM architecture. e_X, e_Z, and e_W are learned structural-role embeddings; $e_{y\times a}$ encodes the factual outcome according to treatment.

$y_{i,k}(1) - y_{i,k}(0)$ whose mean is $\widehat{Q}_{i,K}$, with estimated sampling variance $S^2_{i,K}$ (Figure 2). The estimator $\widehat{Q}_{i,K}$ is unbiased for $Q(x_i)$ at every K and $S^2_{i,K} \to 0$ as K grows, so K trades compute against target precision. The transformer receives exclusively the observational context $\mathcal{D}_N$ and the query covariates x_i; the Monte Carlo estimates $\widehat{Q}_{i,K}$ and $S^2_{i,K}$ are reserved for training supervision.

3.2 MODEL & OBJECTIVE

PROXIMALFM is a PFN trained to map an observational dataset directly to a Bayesian posterior distribution over the CATE estimand. Given an observational context set $\mathcal{D}_N = \{(x_i, z_i, w_i, a_i, y_i)\}_{i=1}^N$ of observed units and a query covariate vector $x^\star$, the model outputs a predictive distribution over the CATE $Q(x^\star)$ in a single forward pass. As it is trained on datasets from the proximal prior Π (Section 3.1), its output approximates the CATE posterior predictive distribution under Π (Müller et al., 2022): dataset-specific Bayesian inference is amortized once at pre-training time.

Encoding the proximal setting. The backbone of PROXIMALFM follows the TabICLv2 architecture (Qu et al., 2026); we defer architectural details to Appendix C.1. Our modifications concern the input processing which must carry the causal role of each column as the roles of (X, Z, W) are not identifiable from their position or values. PROXIMALFM differs from regular tabular models such as TabICLv2 in two respects. First, its target is not observed in the context: the ITE is never observed, only the factual outcome $Y = Y(A)$ is. These observations only provide evidence about the underlying SCM and its CATE function $Q(x)$. Second, PROXIMALFM targets the posterior distribution over the CATE, $p\big(Q(x^\star) \mid \mathcal{D}_N\big)$, rather than a distribution over an individual-level label such as the ITE.

Each query is represented by $x^\star$ only: including its proxy values $(w^\star, z^\star)$ would condition the prediction on the proxies and change the target away from the CATE $Q(x^\star)$. We add *structural role embeddings*, a learnable token per feature group (X, Z, W), added to its column embeddings. A *treatment-aware target encoder* is also implemented. It consists of two linear projections of the outcome selected by the treatment indicator A, so that outcomes observed under treatment and control are embedded separately (Figure 3).

Posterior output & noise-aware loss. PROXIMALFM uses a single distributional head to predict the posterior distribution $q_\theta(Q \mid x^\star, \mathcal{D}_N)$ directly, rather than separate heads for the two potential outcomes. We use regression-as-classification: q_θ is predicted as a histogram over 2 000 equal-width bins spanning $[-5, 5]$, in units of the standard deviation of the observed outcomes in context.

The synthetic targets $\widehat{Q}_{i,K}$ are Monte Carlo averages over K replicates. For sufficiently large K, their Monte Carlo error is approximately Gaussian by the central limit theorem, with variance estimated by $S^2_{i,K}$. Scoring them directly under a distributional loss would inflate the predicted variance

by approximately $S^2_{i,K}$. Because $S^2_{i,K}$ is estimated, we correct for it by blurring the model rather than the target: we maximize the likelihood of $\widehat{Q}_{i,K}$ under the *noise-convolved* predictive distribution

$$\tilde{q}_\theta(\cdot \mid x_i, \mathcal{D}_N) \coloneqq q_\theta(\cdot \mid x_i, \mathcal{D}_N) * \mathcal{N}\big(0,\, S_i^2\big). \tag{5}$$

giving, over the query points $i \in \{1, \dots, n_q\}$,

$$\mathcal{L}_{\text{noise_aware}} = -\frac{1}{n_q} \sum_{i=1}^{n_q} \log \tilde{q}_\theta\big(\widehat{Q}_{i,K} \mid x_i, \mathcal{D}_N\big). \tag{6}$$

The Monte Carlo inflation is removed: the minimizer of (6) approximates the posterior over Q, up to finite histogram resolution and the Gaussian approximation to the K Monte Carlo noise. The closed form of (5) for a histogram head is given in Appendix C.2. We add a squared-error regularizer on the posterior mean, $\mathcal{L}_{\text{mean}} = \frac{1}{n_q} \sum_{i=1}^{n_q} \big(\mathbb{E}_{q_\theta}[Q \mid x_i, \mathcal{D}_N] - \hat{Q}_K(x_i)\big)^2$. The total training objective is $\mathcal{L} = \mathcal{L}_{\text{noise_aware}} + \lambda_{\text{mean}} \mathcal{L}_{\text{mean}}$. At inference time, PROXIMALFM outputs the full CATE posterior distribution $q_\theta(Q \mid x^\star, \mathcal{D}_N)$ in a single forward pass, with point estimate $\hat{Q}_\theta(x^\star) = \mathbb{E}_{q_\theta}[Q \mid x^\star, \mathcal{D}_N]$. Details on the training are available in Appendix C.3.

3.3 FROM PROXIMAL IDENTIFICATION TO BAYESIAN CONSISTENCY

Let ψ parametrize an SCM drawn from Π. We adapt the Bayesian consistency argument of Balazadeh Meresht et al. (2026) to the CATE $Q(x; \psi) = \mu_1(x; \psi) - \mu_0(x; \psi)$, where $\mu_a(x; \psi) = \mathbb{E}_\psi[Y(a) \mid X = x]$, the conditional expected potential outcome (CEPO). Under the proximal identifying conditions, including completeness and existence of an outcome bridge, each CEPO is an observational functional $\mu_a(x; \psi) = \mathbb{E}_\psi[h_0(W, a, X) \mid X = x]$. Their difference therefore identifies $Q(x)$ and directly targeting CATE requires no change to this identification argument.

Suppose these conditions hold almost surely under the SCM prior Π. Under the measurability and integrability assumptions in Appendix A, for each admissible fixed query x and Π-almost every true SCM $\psi^\star$, the oracle Bayesian posterior mean satisfies

$$\overline{Q}_N(x) := \int Q(x; \psi)\, \Pi(d\psi \mid \mathcal{D}_N) \xrightarrow[N \to \infty]{\text{a.s.}} Q(x; \psi^\star). \tag{7}$$

Theorem A.4 establishes asymptotic consistency of the Bayesian CATE target underlying PROXIMALFM: as the observational context grows, the oracle posterior concentrates on the true CATE and its mean converges to that value. This provides a theoretical foundation for the inference target that PROXIMALFM learns to approximate through Monte Carlo supervision and amortized inference. Our trained model's posterior-mean predictions inherit consistency when their approximation error relative to the oracle also vanishes.

4 EVALUATION

4.1 EVALUATION SETTING

Methods evaluated. We compare PROXIMALFM against three families of methods. First, proximal estimators recover causal effects through bridge functions or conditional moment restrictions. These are the only baselines encoding the proximal identification assumptions. Second, we evaluate existing causal foundation models that assume no unmeasured confounding. Third, conventional S-, T-, and X-meta-learners provide other non-proximal backdoor baselines (Künzel et al., 2019). The backdoor methods are evaluated by either fitting X alone or fitting (X, W, Z) followed by marginalizing the proxy-conditional contrast over the proxies. Both variants return a function of x, but neither is generally identified as $Q(x)$ under the proximal assumptions. Appendix D.1 lists all methods and refers to their configuration and implementation details.

Datasets. Appendix D.2 details our semi-synthetic datasets generation. The covariates X and unobserved U are features from real-world data. The proxies W and Z are then generated from (U, X). Similarly, the treatment assignment A and outcomes $Y(1), Y(0)$ are generated from (U, X). Note that we don't use Z and W as direct inputs when generating the treatment and outcomes.

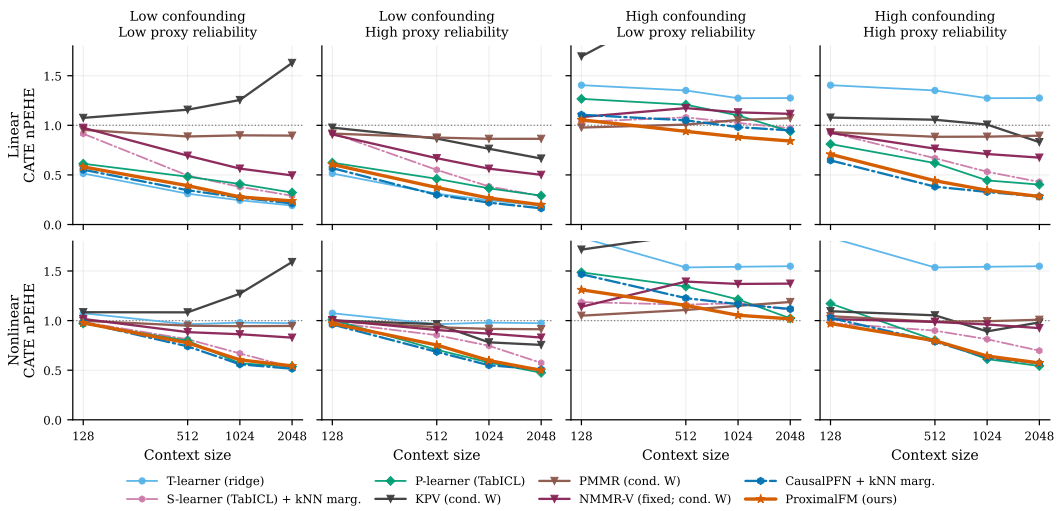


Figure 4: Median CATE nPEHE across benchmark episodes versus context size. Rows show linear and nonlinear mechanisms; columns vary latent confounding and proxy reliability. The dotted line marks the constant-ATE baseline ($\mathrm{nPEHE} = 1$); lower is better.

This semi-synthetic approach grounds the simulation on real-world features, while having control over dataset properties such as confounding strength of U and proxy informativeness of (W, Z) about U. We consider a factorial design crossing linear versus nonlinear mechanisms, low versus high latent-confounding strength, and low versus high proxy informativeness, yielding eight configurations. For each of the eight settings, we evaluate 12 datasets with three independent replicates each, giving a total of 288 evaluation instances. We set $d_X = d_U = d_W = d_Z = 5$.

4.2 Empirical results

ProximalFM is competitive across settings. Figure 4 summarizes the performance of selected methods, with detailed results available in Tables D.7–D.10. Across all eight settings, ProximalFM is competitive, with its clearest advantage at larger context sizes under high confounding and low proxy reliability. Among the proximal methods, `P-learner (TabICL)` is also consistently competitive, but typically requires larger context sizes to obtain good performance. `KPV` and `PMMR` generally underperform, possibly because of the use of kernel-based bridge functions in this moderate dimension regime ($d_X = d_U = d_W = d_Z = 5$). Flexible backdoor methods such as `S-learner (TabICL)` and `CausalPFN` perform surprisingly well when combined with simple proxy marginalization. Their performance deteriorates, however, when strong confounding is combined with low proxy information.

ProximalFM plateaus beyond its training range. We next consider the nonlinear regime with high latent confounding and low proxy reliability (Figure D.2, Table D.11). This is a particularly relevant proximal setting: the confounding cannot be ignored, the proxy signal is weak and thus conditioning on proxies is insufficient. As the context grows beyond $n_{\mathrm{ctx}} = 2\,048$, the `P-learner (TabICL)` continues to improve and becomes the strongest method, whereas ProximalFM improves little further and remains close to the constant-ATE baseline. A plausible explanation is a mismatch between evaluation and pre-training context sizes: ProximalFM was trained on contexts of at most $2\,048$ observations. To assess this, we continue training the model for limited time on episodes with contexts of up to $5\,000$ observations (Appendix C.3.1, `ProximalFM-FT`). The adapted model shows small improvements at large context sizes, consistent with limited large-context training support contributing to the observed plateau.

ProximalFM is fast and requires no task-specific tuning. ProximalFM lies on a favourable accuracy–runtime frontier (Figure 5). It achieves among the strongest CATE ranks while requiring little end-to-end runtime compared to several competing proximal methods. Moving inference from

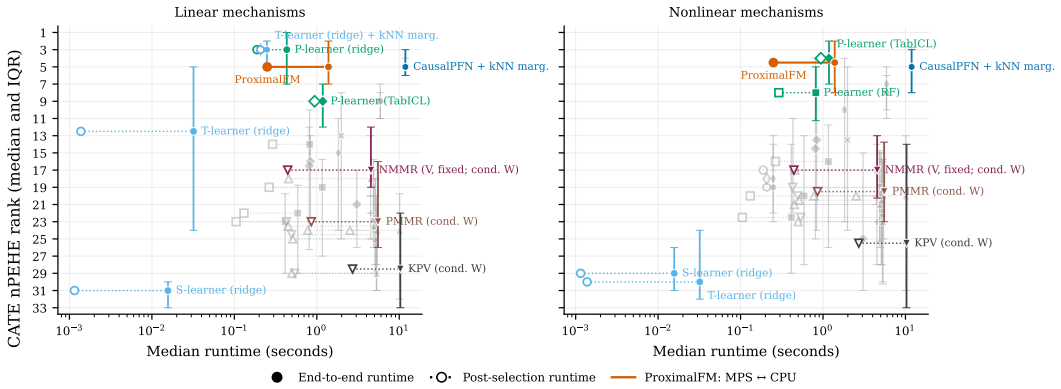


Figure 5: Accuracy and speed at context size $n = 2\,048$, shown separately for linear and nonlinear mechanisms. Points show median runtime per episode against median paired CATE nPEHE rank (IQR whiskers). Filled and hollow points respectively include and exclude hyperparameter selection. The orange segment compares PROXIMALFM on CPU and MPS. More details in Appendix D.3.1.

CPU to MPS further reduces its runtime without changing its predictions. PROXIMALFM does not require hyperparameter selection, simplifying deployment and making its reported runtime unaffected by the choice of a selection grid.

The posterior is overconfident on semi-synthetic data. Figure D.3 compares nominal and empirical credible-interval coverage. On datasets sampled from the pre-training prior, coverage is close to nominal across all context sizes, and posterior uncertainty contracts steadily as the context grows. On the other hand, on the semi-synthetic datasets, the posterior is increasingly overconfident as the context size grows. This undercoverage under distribution shift is consistent with the out-of-distribution experiments from Balazadeh Meresht et al. (2026). Notably, the posterior standard deviation continues to decrease beyond the maximum pre-training context size.

Proxy informativeness matters. The good performance of TabICL and CausalPFN in the semi-synthetic benchmark raises the question of whether flexible backdoor methods can substitute for proximal estimation. We investigate this in a setting with linear outcome and Gaussian proxy mechanisms, where we control the information that (W, Z) carries about U (Appendix D.4). At $n_{\text{ctx}} = 2\,048$, PROXIMALFM outperforms proxy-marginalized CausalPFN throughout the sweep and proxy-adjusted linear regression (OLS) in most settings, although OLS catches up when proxies nearly reveal U. Well specified proximal 2SLS (Tchetgen Tchetgen et al., 2020) provides a reference for what matching the estimator to this mechanism can achieve.

Towards a real physical evaluation. We compare ATE estimations on Causal Chambers, a physical system with a known causal graph (Gamella et al., 2025) (Appendix E). Backdoor meta-learners outperform proximal learners, including PROXIMALFM. We interpret this cautiously: PROXIMALFM is designed for CATE, this benchmark has only twelve configurations of one one-dimensional system and moderate sample size (1 000 observations each).

5 CONCLUSION

We introduced PROXIMALFM, an amortized estimator for CATE under the proximal learning assumptions. Existing causal foundation models address settings such as backdoor, instrumental-variable, and frontdoor identification, but are not designed to exploit proxies for proximal identification. Rather than solving an ill-posed bridge equation separately for each new dataset, PROXIMALFM learns from synthetic episodes to estimate CATE from observational data, with regularization implied by the prior. Its CATE posterior also provides uncertainty estimates, though credible intervals can be overconfident beyond the training prior. Together with models for other identification settings, it gives practitioners a fast way to compare estimates under different plausible identifying assumptions.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic datasets (our 'validation datasets'), provide feedback on research methodology and experiments, implement the methods (for the benchmarks, but only moderately for PROXIMALFM). We have not used generative AI tools for theoretical purpose (develop models, formulate mathematical claims, provide critical ingredients for proving claims, writing proofs, propose hypotheses), and translation assistance, cleaning datasets and qualitative data analysis are not applicable to this work.

Additionally, we used generative AI tools to create figures and tables (from results data or for visualization diagrams), create and edit code, draft parts of the appendix. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have emphasized reproducibility throughout our work. The PROXIMALFM training runs can be launched from a single script. The benchmark suites can be run with a single command. We also provide commands to reproduce most of the plots and tables. Our appendix gives exhaustive details on the implementation of our prior and of the methods (PROXIMALFM but also all methods from the benchmark).

REFERENCES

Vahid Balazadeh, Hamidreza Kamkari, Medha Barath, Ricardo Silva, and Rahul G Krishnan. Ivicl: Bounding causal effects with instrumental variables via in-context learning. *arXiv preprint arXiv:2605.12924*, 2026.

Vahid Balazadeh Meresht, Hamidreza Kamkari, Valentin Thomas, Junwei Ma, Bingru Li, Jesse Cresswell, and Rahul Krishnan. Causalpfn: Amortized causal effect estimation via in-context learning. *Advances in Neural Information Processing Systems*, 38:154945–154984, 2026.

Nishanth Dikkala, Greg Lewis, Lester Mackey, and Vasilis Syrgkanis. Minimax Estimation of Conditional Moment Models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12248–12262. Curran Associates, Inc., 2020.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2026.

Juan L. Gamella, Jonas Peters, and Peter Bühlmann. Causal chambers as a real-world physical testbed for ai methodology. *Nature Machine Intelligence*, 7(1):107–118, 2025. doi: 10.1038/s42256-024-00964-x. URL https://doi.org/10.1038/s42256-024-00964-x.

Anurag Garg, Muhammad Ali, Noah Hollmann, Lennart Purucker, Samuel Müller, and Frank Hutter. Real-tabPFN: Improving tabular foundation models via continued pre-training with real-world data. In *1st ICML Workshop on Foundation Models for Structured Data*, 2025. URL https://openreview.net/forum?id=BtEiqKsIMw.

A. Ghassami, A. Ying, I. Shpitser, and E. Tchetgen Tchetgen. Minimax kernel machine learning for a class of doubly robust functionals with application to proximal causal inference. *International Conference on Artificial Intelligence and Statistics*, pp. 7210–7239, 2022.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Miguesl Hernán and James Robins. *Causal Inference: What If*. Chapman & Hall/CRC, 2020.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second, September 2023. URL http://arxiv.org/abs/2207.01848. arXiv:2207.01848 [cs].

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, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

Benjamin Kompa, David Bellamy, Tom Kolokotrones, Andrew Beam, et al. Deep learning methods for proximal inference via maximum moment restriction. *Advances in Neural Information Processing Systems*, 35:11189–11201, 2022.

Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Meta-learners for Estimating Heterogeneous Treatment Effects using Machine Learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, March 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1804597116. URL http://arxiv.org/abs/1706.03461. arXiv:1706.03461 [math.ST].

Alex Luedtke and Incheoul Chung. Adversarial monte carlo meta-learning of conditional average treatment effects. In *Handbook of Statistical Methods for Precision Medicine*, pp. 237–248. Chapman and Hall/CRC, 2024.

Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation models for causal inference via prior-data fitted networks. In *International Conference on Learning Representations*, volume 2026, pp. 79065–79098, 2026.

Afsaneh Mastouri, Yuchen Zhu, Limor Gultchin, Anna Korba, Ricardo Silva, Matt Kusner, Arthur Gretton, and Krikamol Muandet. Proximal Causal Learning with Kernels: Two-Stage Estimation and Moment Restriction. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7512–7523. PMLR, July 2021. URL https://proceedings.mlr.press/v139/mastouri21a.html.

Wang Miao, Zhi Geng, and Eric J Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105(4):987–993, December 2018. ISSN 0006-3444. doi: 10.1093/biomet/asy038. URL https://doi.org/10.1093/biomet/asy038.

Jeffrey W. Miller. A detailed treatment of doob's theorem, 2018. URL https://arxiv.org/abs/1801.03122.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. RealCause: Realistic Causal Inference Benchmarking, March 2021. URL http://arxiv.org/abs/2011.15007. arXiv:2011.15007 [cs].

Judea Pearl. *Causality: Models, Reasoning and Inference, 2nd edition*. Cambridge University Press, 2009.

Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David Holzmüller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, Gaël Varoquaux, and Frank Hutter. Beyond iid: How general are tabular foundation models, really?, 2026. URL https://arxiv.org/abs/2606.30410.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/qu25d.html.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=SxsyLjIfWB.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-pfn: In-context learning for causal effect estimation. *Advances in Neural Information Processing Systems*, 38:174811–174848, 2026.

Donald Rubin. Estimating Causal Effects of Treatments in Experimental and Observational Studies. *ETS Research Bulletin Series*, 1972(2):i–31, 1972. ISSN 2333-8504. doi: 10.1002/j.2333-8504.1972.tb00631.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2333-8504.1972.tb00631.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2333-8504.1972.tb00631.x.

Christopher Stith, Medha Barath, Vahid Balazadeh, Jesse C. Cresswell, and Rahul G Krishnan. Causal foundation models with continuous treatments. In *2nd ICML Workshop on Foundation Models for Structured Data*, 2026a. URL https://openreview.net/forum?id=DzcWAYcR2n.

Christopher Stith, Hossein Rahmani, and Jesse C. Cresswell. Causal foundation models, 2026b. URL https://arxiv.org/abs/2609.03003.

Erik Sverdrup and Yifan Cui. Proximal causal learning of conditional average treatment effects. In *International Conference on Machine Learning*, pp. 33285–33298. PMLR, 2023.

Eric J. Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An Introduction to Proximal Causal Learning, September 2020. URL http://arxiv.org/abs/2009.10982. arXiv:2009.10982 [stat].

Liyuan Xu, Heishiro Kanagawa, and Arthur Gretton. Deep proxy causal learning and its application to confounded bandit policy evaluation. *Advances in Neural Information Processing Systems*, 34: 26264–26275, 2021.

Y Yang, AK Kuchibhotla, and E Tchetgen Tchetgen. Forster-Warmuth counterfactual regression: A unified learning approach, 2023. URL https://doi.org/10.48550/arXiv.2307.16798.

Jakob Zeitler. Physical benchmarks for testing algorithms. *Nature Machine Intelligence*, 7(2): 166–167, 2025. doi: 10.1038/s42256-025-00999-8. URL https://doi.org/10.1038/s42256-025-00999-8.

APPENDIX

A Consistency result **16**
A.1 Setup and identification . . . 16
A.2 Consistency of the oracle CATE posterior . . . 16
A.3 Proximal identification . . . 17
A.4 Application to PROXIMALFM . . . 17

B Details on the proximal prior **19**
B.1 Step-by-step procedure . . . 19
B.2 Generative model components . . . 21
B.2.1 Parameters of the prior . . . 21
B.2.2 Dimensions & dimension heuristic . . . 25
B.2.3 MLP generation, noise & edge sampling . . . 25
B.2.4 Post-process: categorisation, outlier and scaling . . . 28
B.2.5 Treatment assignment . . . 33
B.2.6 Potential outcomes & heterogeneity remix . . . 34
B.2.7 Monte Carlo CATE supervision . . . 35
B.2.8 Visualize the prior . . . 35
B.3 Proximal assumption audit . . . 39
B.3.1 Overlap . . . 39
B.3.2 Exclusion restrictions . . . 39
B.3.3 Latent exchangeability and confounding strength . . . 41
B.3.4 Proxy relevance . . . 43

C Model architecture, objective, training, and validation **45**
C.1 Architecture . . . 45
C.2 The noise-aware histogram loss . . . 47
C.3 Training the model . . . 48
C.3.1 Post-training on larger context sizes . . . 51
C.4 Validation and posterior diagnostics . . . 52
C.4.1 Validation datasets . . . 52
C.4.2 Point-estimate validation . . . 52
C.4.3 Distributional validation . . . 52
C.4.4 Posterior representation & smoothing . . . 54
C.4.5 Posterior dynamics . . . 55

D Details on results **58**
D.1 Baseline methods . . . 58
D.1.1 Backdoor meta-learners, in-context learners and proxy marginalization . . . 59

D.1.2 P-learner 61
D.1.3 Neural moment matching regression 61
D.1.4 Proxy maximum moment restriction 63
D.1.5 Kernel proxy variable 64
D.2 Semi-synthetic benchmarks from real covariates 65
D.3 Results on semi-synthetic datasets 68
D.3.1 Speed 75
D.4 Synthetic study of proxy informativeness 75

E Towards a real physical system: Causal Chambers **78**

A CONSISTENCY RESULT

We adapt the observational-equivalence argument of Balazadeh Meresht et al. (2026) to the CATE functional. The result characterizes the oracle Bayesian learner under the SCM prior Π used in the main text. Its application to PROXIMALFM is elaborated in Section A.4.

A.1 SETUP AND IDENTIFICATION

Let ψ range over a standard Borel parameter space Ψ. Each SCM induces a joint law P^ψ of the observed variables and both potential outcomes, with $A \in \{0,1\}$ and $Y = Y(A)$. Write $O = (X, Z, W, A, Y)$ for the observed variables, taking values in a standard Borel space $\mathcal{O}$, and $R(\psi) := P^\psi_{\text{obs}}$ for their law. Under the joint Bayesian model, first draw $\psi \sim \Pi$ and then draw $O_1, O_2, \ldots \mid \psi$ independently from $R(\psi)$. The observational context is $\mathcal{D}_N = (O_1, \ldots, O_N)$.

Assumption 5 (Measurability)**.** *The map $\psi \mapsto P^\psi$ is measurable, and the image $\mathcal{R} := \{R(\psi) : \psi \in \Psi\}$ is a Borel subset of $\mathcal{P}(\mathcal{O})$, the standard Borel space of probability laws on $\mathcal{O}$. We choose jointly measurable versions of*

$$\mu_a(x;\psi) := \mathbb{E}_\psi[Y(a) \mid X = x], \qquad Q(x;\psi) := \mu_1(x;\psi) - \mu_0(x;\psi). \tag{8}$$

Throughout what follows, fix an admissible query x at which these conditional means are defined and the stated assumptions apply. Our results apply pointwise in x and we treat the query as fixed (if instead a query is sampled from an SCM-dependent covariate distribution and provides additional evidence, the same argument uses the conditional prior $\Pi(d\psi \mid X_{\text{query}} = x)$, assuming the context and query are conditionally independent given ψ and the assumptions below hold for that prior.)

Assumption 6 (Integrability)**.** *At the query under consideration,*

$$\int_\Psi |Q(x;\psi)|\, \Pi(d\psi) < \infty. \tag{9}$$

Integrability of both CEPOs is sufficient, since $|Q(x;\psi)| \le |\mu_1(x;\psi)| + |\mu_0(x;\psi)|$.

Definition A.1 (Observational equivalence and CATE-identifiability)**.** *Two SCMs are observationally equivalent when $R(\psi) = R(\psi')$. The prior is CATE-identifiable at x if there is a measurable map $F_x : \mathcal{R} \to \mathbb{R}$ such that $Q(x;\psi) = F_x(R(\psi))$ for Π-almost every ψ.*

Definition A.2 (CEPO-identifiability)**.** *The prior is CEPO-identifiable at x if, for each $a \in \{0,1\}$, there is a measurable $F_{a,x}$ with $\mu_a(x;\psi) = F_{a,x}(R(\psi))$ for Π-almost every ψ.*

CEPO-identifiability implies CATE-identifiability by taking $F_x = F_{1,x} - F_{0,x}$. The converse is unnecessary and need not hold, since a common unidentified component of the two means can cancel in their difference.

The oracle CATE posterior and its mean are

$$\begin{aligned} \Pi^Q(B \mid x, \mathcal{D}_N) &:= \int_\Psi \mathbf{1}\{Q(x;\psi) \in B\}\, \Pi(d\psi \mid \mathcal{D}_N), \qquad B \in \mathcal{B}(\mathbb{R}), \\ \overline{Q}_N(x) &:= \int_\Psi Q(x;\psi)\, \Pi(d\psi \mid \mathcal{D}_N). \end{aligned} \tag{10}$$

A.2 CONSISTENCY OF THE ORACLE CATE POSTERIOR

Working directly with $\mathcal{R}$ represents observational equivalence classes by their common law. Let ν be the distribution of $R(\psi)$ under Π, and choose a measurable version of

$$g_x(r) := \mathbb{E}_\Pi[Q(x;\psi) \mid R(\psi) = r]. \tag{11}$$

Lemma A.3 (Limit within observational equivalence classes)**.** *Under Assumptions 5–6, for Π-almost every $\psi^\star$, i.i.d. observations from $R(\psi^\star)$ satisfy*

$$\overline{Q}_N(x) \xrightarrow[N\to\infty]{\text{a.s.}} g_x(R(\psi^\star)). \tag{12}$$

Proof. Conditional on R, the law of the entire observation sequence is $R^{\otimes\infty}$ and does not depend further on ψ. Thus

$$\overline{Q}_N(x) = \mathbb{E}[Q(x;\psi) \mid \mathcal{D}_N] = \mathbb{E}[g_x(R) \mid \mathcal{D}_N]. \tag{13}$$

Moreover, conditional Jensen's inequality gives $\int |g_x|\, d\nu \le \int |Q(x;\psi)|\, d\Pi < \infty$. The model indexed by $r \in \mathcal{R}$, with sampling law r, is measurable and identifiable by construction. Theorem 2.2 of Miller (2018) therefore gives $\mathbb{E}[g_x(R) \mid \mathcal{D}_N] \to g_x(R)$ almost surely under the joint Bayesian law. Intersecting the probability-one events on which (13) holds for each N and disintegrating with respect to ψ proves the stated parameterwise conclusion, as in Corollary 2.3 of Miller (2018). □

Theorem A.4 (CATE identification and oracle consistency)**.** *Under Assumptions 5–6, the following are equivalent at the fixed query x:*

1. *The prior is CATE-identifiable at x.*
2. *For Π-almost every $\psi^\star$, i.i.d. observations from $R(\psi^\star)$ satisfy*

$$\overline{Q}_N(x) \xrightarrow[N\to\infty]{\text{a.s.}} Q(x;\psi^\star). \tag{14}$$

Under these conditions the exact CATE posterior also concentrates: for Π-almost every $\psi^\star$ and every $\epsilon > 0$,

$$\Pi^Q(\{q : |q - Q(x;\psi^\star)| > \epsilon\} \mid x, \mathcal{D}_N) \xrightarrow[N\to\infty]{\text{a.s.}} 0. \tag{15}$$

Proof. If $Q(x;\psi) = F_x(R(\psi))$ almost surely, then $g_x = F_x$ ν-almost everywhere. Intersecting the pullback of this set with the full-prior-measure set in Lemma A.3 proves consistency. Conversely, consistency and Lemma A.3 imply $Q(x;\psi) = g_x(R(\psi))$ for Π-almost every ψ. The measurable function g_x therefore witnesses CATE-identifiability.

For posterior concentration, apply the same Doob argument to the bounded functions $r \mapsto \mathbf{1}\{F_x(r) \le t\}$ for all rational t. Intersecting the resulting countably many full-measure sets gives convergence of the posterior probabilities of these half-lines to their values at $Q(x;\psi^\star)$. For any $\epsilon > 0$, choose rational thresholds on either side of $Q(x;\psi^\star)$, each within ϵ of it. The posterior mass outside these thresholds tends to zero, proving (15). □

A.3 Proximal identification

Suppose the proximal assumptions hold Π-almost surely, including the completeness and bridge-existence conditions needed for the conditional proximal g-formula at the fixed query x. For an appropriate outcome bridge h_0 and the chosen identifying versions of the conditional means, the formula yields

$$\begin{aligned} \mu_a(x;\psi) &= \mathbb{E}_\psi[h_0(W,a,x) \mid X = x], \\ Q(x;\psi) &= \mathbb{E}_\psi[h_0(W,1,x) - h_0(W,0,x) \mid X = x]. \end{aligned} \tag{16}$$

The right-hand sides are identified by the observational law as their measurable representations give CEPO- and hence CATE-identifiability. Theorem A.4 consequently applies. Equivalently, when both CEPO posterior means are consistent and integrable, linearity gives

$$\overline{Q}_N(x) = \mathbb{E}[\mu_1(x;\psi) \mid \mathcal{D}_N] - \mathbb{E}[\mu_0(x;\psi) \mid \mathcal{D}_N] \xrightarrow[N\to\infty]{\text{a.s.}} Q(x;\psi^\star). \tag{17}$$

Note that this identity does not require the trained model to estimate the CEPO.

A.4 Application to ProximalFM

For independent downstream simulations from the SCM's conditional potential-outcome law given $X = x$, independent of the context conditional on ψ, the target $\widehat{Q}_K(x) = K^{-1}\sum_{k=1}^K (Y_k(1) - Y_k(0))$ satisfies

$$\mathbb{E}[\widehat{Q}_K(x) \mid \psi, \mathcal{D}_N] = Q(x;\psi), \qquad \mathbb{E}[\widehat{Q}_K(x) \mid \mathcal{D}_N] = \overline{Q}_N(x). \tag{18}$$

With finite second moments, population squared-error regression on these labels therefore targets the oracle posterior mean even at finite K. Recovering the full posterior from noisy labels additionally requires a correctly specified and identifiable measurement-error model, or a limit in which

Monte Carlo error vanishes. Our theoretical result does not establish exact recovery by the finite-K Gaussian noise-aware objective or its combination with a mean penalty but is instead concerned with an oracle learner.

To make this explicit, let $\widehat{Q}_{\theta,N}(x)$ be the mean predicted by PROXIMALFM. Then

$$\begin{aligned} |\widehat{Q}_{\theta,N}(x) - Q(x;\psi^\star)| \le |\widehat{Q}_{\theta,N}(x) - \overline{Q}_N(x)| \\ + |\overline{Q}_N(x) - Q(x;\psi^\star)|. \end{aligned} \tag{19}$$

Theorem A.4 controls the second term. Consistency of the learned estimator also requires the first term (comprising supervision, representation, and training errors) to vanish as context size grows, which extends beyond the scope of our theoretical contributions and would require substantially more developments in the theory of in-context learning.

B Details on the proximal prior

B.1 Step-by-step procedure

We generate each synthetic dataset from an independently sampled structural causal model (SCM) compatible with the proximal DAG. The full routine is sketched in Figure B.1 and specified in Algorithm 1; this section walks through it and fixes notation used across Appendix B.

Throughout, a dataset consists of T units (the sequence length) and the target estimand is the conditional average treatment effect $Q(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]$. Because $Q(x)$ marginalises over all unit-level heterogeneity beyond X, we approximate the conditional expected potential outcomes (CEPOs) $\mu_a(x) = \mathbb{E}[Y(a) \mid X = x]$ by Monte Carlo integration: every unit is replicated K times, sharing its covariate value x while drawing fresh exogenous noise—and therefore fresh latent-confounder and proxy realizations—and the potential outcomes are averaged over the K replicates.

Once per dataset: structure. We first sample the latent and observed dimensionalities, drawing the confounder dimension d_U from a law skewed toward small values and then the proxy dimensions d_W, d_Z conditionally on d_U so as to respect the completeness heuristic $d_W, d_Z \geq d_U$, with the covariate dimension d_X drawn independently and uniformly; the sampling laws and the resulting induced marginals are described below in Appendix B.2.2. We then instantiate the causal mechanisms $f_X, f_U, f_W, f_Z, f_A, f_Y$ as independently initialised random multilayer perceptrons in the style of TabICL (Qu et al., 2025), together with an auxiliary confounding-duplication network f_{aug} (described below), and sample the per-dataset hyperparameters (network depth, width, activations, noise scales; see Appendix B.2.1). Mechanisms, dimensions and hyperparameters are held fixed across all T units and their K replicates, so the entire dataset is drawn from the *same* SCM.

Per replicate: the topological pass. The baseline covariates are generated once,

$$X = \text{Corrupt}\big(f_X(\varepsilon_X)\big) \in \mathbb{R}^{T \times d_X}, \tag{20}$$

and each row is tiled K times. The remaining variables are sampled in topological order over all $T \cdot K$ rows in a single vectorised pass, with mutually independent exogenous noise $\varepsilon_{(\cdot)}$ drawn afresh for every row:

$$U = \text{Corrupt}\big(f_U(\tilde{X}_U, \varepsilon_U)\big), \quad W = \text{Corrupt}\big(f_W(\tilde{U}, \tilde{X}_W, \varepsilon_W)\big), \quad Z = \text{Corrupt}\big(f_Z(\tilde{U}, \tilde{X}_Z, \varepsilon_Z)\big), \tag{21}$$

where $\tilde{(\cdot)}$ denotes standardisation to zero mean and unit variance and $\text{Corrupt}(\cdot)$ applies the feature postprocessing of Appendix B.2.4 (optional categorisation, outlier clamping, and scaling). The exogenous-noise blocks $\varepsilon_{(\cdot)}$ are sized proportionally to each mechanism's non-noise input width via a per-dataset share parameter (see Table B.3), so the noise level does not shrink as signal width grows. Crucially, Z is excluded from the inputs of f_W and f_Y, and W is excluded from those of f_Z and f_A, so the proxy exclusion restrictions of Assumption 2 hold by construction for any realisation of the random mechanisms.

Notation: $\tilde{X}_U, \tilde{X}_W, \tilde{X}_Z, \tilde{X}_A, \tilde{X}_Y$ denote per-mechanism random subsets of $\tilde{X}$'s columns, each drawn independently with its own per-dataset keep probability (see Appendix B.2.3); every mechanism that reads X, including f_U, sees one of these subsets rather than the full block. Likewise, we'll use the notation $\tilde{W}_Y \subseteq \tilde{W}$ and $\tilde{Z}_A \subseteq \tilde{Z}$ to denote random column subsets for the direct proxy edges.

Mechanism wiring: confounding control and edge subsampling. Three per-dataset knobs then set which columns actually feed each mechanism. First, to control how strongly the confounder influences each of A and Y – independently of d_U, which in a random MLP would otherwise pin the confounding to U's accidental share of the input width – we target a share $s_A, s_Y \sim \text{Beta}(1.2, 2.8)$ of each mechanism's inputs to be U-derived and convert it to a column count $k = \max(2, \text{round}(\frac{s_c}{1-s_c} n))$ with $s_c = \min(s, 0.75)$ and n the width of the non-U inputs. The mechanism is then fed either a subsample of U's columns (when $k \leq d_U$) or all of U plus $k - d_U$ columns of a shared nonlinear re-encoding $U^{\text{aug}} = f_{\text{aug}}(\tilde{U})$ (when $k > d_U$); we write $U^{[A]}, U^{[Y]}$ for these confounding-controlled inputs. We detail this knob in Appendix B.2.3.

Algorithm 1 Generation of one proximal dataset.

Require: sequence length T, Monte Carlo count K, hyperparameters.

sample d_U from (22); sample d_W, d_Z, d_X from (23)
instantiate random MLP mechanisms $f_X, f_U, f_W, f_Z, f_A, f_Y, f_{\text{aug}}$
sample per-dataset hyperparameters: noise shares, confounding shares s_A, s_Y, edge keep-probabilities, overlap knobs $s, b, \gamma \sim \text{Unif}(0,1)$

repeat
  $X \leftarrow \text{Corrupt}(f_X(\varepsilon_X)) \in \mathbb{R}^{T \times d_X}$, tiled K times and standardised to $\tilde{X}$
  subsample covariate inputs $\tilde{X}_U, \tilde{X}_W, \tilde{X}_Z, \tilde{X}_A, \tilde{X}_Y \subseteq \tilde{X}$
  draw independent noise $\varepsilon_U, \varepsilon_W, \varepsilon_Z, \varepsilon_A, \varepsilon_Y$ over all TK rows
  $U \leftarrow \text{Corrupt}(f_U(\tilde{X}_U, \varepsilon_U))$, standardised to $\tilde{U}$
  $W \leftarrow \text{Corrupt}(f_W(\tilde{U}, \tilde{X}_W, \varepsilon_W))$, standardised to $\tilde{W}$ ▷ $Z \not\to f_W$
  $Z \leftarrow \text{Corrupt}(f_Z(\tilde{U}, \tilde{X}_Z, \varepsilon_Z))$, standardised to $\tilde{Z}$ ▷ $W \not\to f_Z$
  subsample direct-proxy inputs $\tilde{Z}_A \subseteq \tilde{Z}$ and $\tilde{W}_Y \subseteq \tilde{W}$
  form $U^{[A]}, U^{[Y]}$ from $\tilde{U}$ and $U^{\text{aug}} = f_{\text{aug}}(\tilde{U})$ via confounding shares s_A, s_Y
  $\ell \leftarrow f_A(U^{[A]}, \tilde{X}_A, \tilde{Z}_A, \varepsilon_A)$; $\tilde{\ell} \leftarrow (\ell - \bar{\ell})/(\text{sd}(\ell) + 10^{-8})$
  $\pi \leftarrow \text{clip}\left(\sigma(s\,\tilde{\ell} + b),\, \epsilon,\, 1-\epsilon\right)$
  $A \sim \text{Bernoulli}(\pi)$
  $(Y_0^{\text{raw}}, Y_1^{\text{raw}}) \leftarrow f_Y(U^{[Y]}, \tilde{X}_Y, \tilde{W}_Y, \varepsilon_Y)$
  draw $\alpha \sim \text{Unif}(0,1)^{TK}$; remix to (Y_0, Y_1) via (30)
  $\hat{\mu}_a(x_i) \leftarrow \frac{1}{K}\sum_{k=1}^{K} y_{i,k}(a)$ for $a \in \{0,1\}$
  select $k_i \sim \text{Unif}\{1, \ldots, K\}$ per unit; keep observed $\{x_i, w_i, z_i, a_i, y_i\}$
  remove constant columns from X, U, W, Z; update active dims d_X, d_U, d_W, d_Z
until $d_X, d_W, d_Z > 0$, $\min(d_W, d_Z) \geq d_U > 0$ and categorical dominance of W, Z over U.

return observed $\{x_i, w_i, z_i, a_i, y_i\}_{i=1}^{T}$ and targets $\{\hat{\mu}_0(x_i), \hat{\mu}_1(x_i)\}_{i=1}^{T}$

Second and third, the direct covariate and proxy edges are made per-column optional, mirroring each other. Every mechanism downstream of X – i.e. f_U, f_W, f_Z, f_A, f_Y – receives an independent random subset $\tilde{X}_U, \tilde{X}_W, \tilde{X}_Z, \tilde{X}_A, \tilde{X}_Y \subseteq \tilde{X}$, each column entering with its own per-dataset keep probability $\sim \text{Unif}[0,1]$. Similarly, each column of W (resp. Z) enters f_Y (resp. f_A) with its own per-dataset keep probability $\sim \text{Unif}[0,1]$, giving subsets $\tilde{W}_Y \subseteq \tilde{W}$, $\tilde{Z}_A \subseteq \tilde{Z}$. These masks alter the random-function prior without adding graph edges: they only remove direct X, W, or Z inputs, while the $U \to W$ and $U \to Z$ inputs and the completeness heuristics remain unchanged.

Potential outcomes, heterogeneity remix, and the observed dataset. The outcome mechanism returns both potential outcomes $Y_0^{\text{raw}}, Y_1^{\text{raw}}$ from a shared forward pass, which are remixed via a per-dataset heterogeneity dial $\gamma \sim \text{Unif}[0,1]$ into Y_0, Y_1, ranging continuously from the raw, fully heterogeneous effect ($\gamma = 1$) to a single constant effect ($\gamma = 0$); we detail and analyse this construction in Appendix B.2.6. Averaging the remixed potential outcomes over the K replicates of unit i gives the CEPO estimates $\hat{\mu}_a(x_i) = \frac{1}{K}\sum_{k=1}^{K} y_{i,k}(a)$, and hence the CATE target $\hat{Q}(x_i) = \hat{\mu}_1(x_i) - \hat{\mu}_0(x_i)$. To form the observed dataset, one replicate $k_i \sim \text{Unif}\{1, \ldots, K\}$ is retained per unit, yielding the tuple $\left(x_i, w_{i,k_i}, z_{i,k_i}, a_{i,k_i}, y_{i,k_i}\right)$ with $y_{i,k_i} = y_{i,k_i}(a_{i,k_i})$. The transformer sees only these observational components; the CEPO targets $\hat{\mu}_0, \hat{\mu}_1$ are used purely as training targets and are never observed.

Proximal conditions. The construction enforces latent ignorability and the proxy exclusion restrictions of Assumptions 1 and 2 through the mechanism input restrictions and independent exogenous noises. Positivity (Assumption 3) is enforced by the clipped treatment propensity (Appendix B.2.5). Completeness (Assumption 4) cannot be guaranteed for arbitrary nonlinear mechanisms; instead, the prior imposes numeric and categorical proxy-profile heuristics (Appendices B.2.2 and B.2.4). Empirical diagnostics of the resulting overlap, exclusion restrictions, confounding strength, and proxy relevance are collected in Appendix B.3.

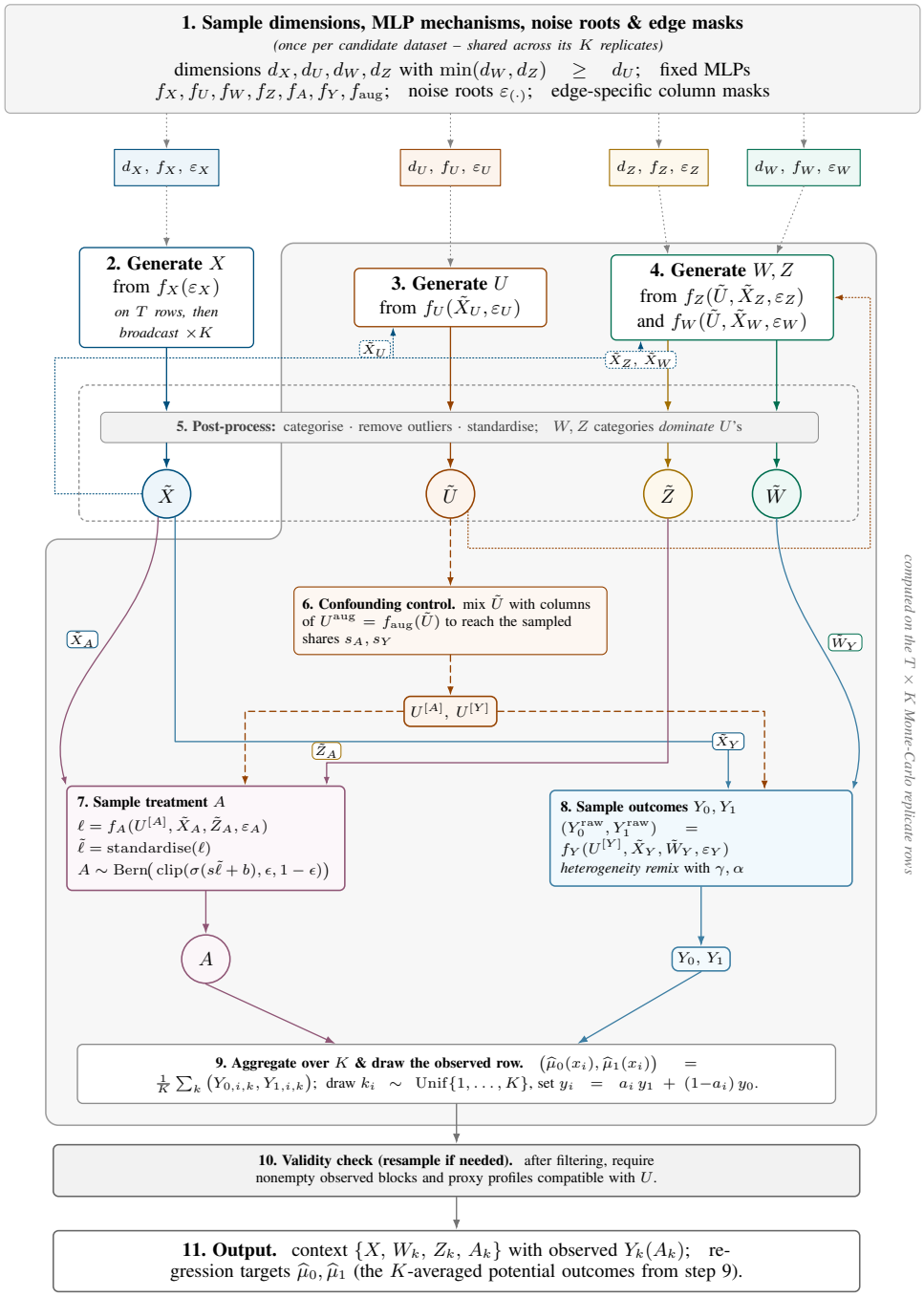


Figure B.1: Detailed diagram of the proximal prior's per-dataset generation pipeline.

B.2 Generative model components

B.2.1 Parameters of the prior

Three sets of parameters govern the prior. *Structural* parameters (Table B.1) fix the shape of a dataset – the sampling laws of the four block dimensions, the number of rows, and the number of Monte Carlo replicates behind the μ_0/μ_1 targets – and are the reference point for the dimension analysis of Appendix B.2.2. *Fixed* hyperparameters (Table B.2) are shared constants, identical across the whole prior. *Sampled* hyperparameters (Table B.3) are instead drawn once per dataset from a meta-

Table B.1: Structural parameters of the prior: the block dimensions and the dataset shape. Each dimension is drawn per dataset from the law shown, stated in terms of the ceilings listed below it. The proxy dimensions are drawn conditionally on d_U so that the completeness heuristic of Appendix B.2.2.

hyperparameter	value	description
Block dimensions		
`dim_X`	$\mathrm{Unif}\{0, \ldots, \texttt{max_dim_X}\}$	observed baseline covariate dimension
`dim_U`	triangular on $\{1, \ldots, \texttt{max_dim_U}\}$	latent confounder dimension
`dim_W`	$\mathrm{Unif}\{d_U, \ldots, \texttt{max_dim_W}\}$	outcome-inducing proxy dimension
`dim_Z`	$\mathrm{Unif}\{d_U, \ldots, \texttt{max_dim_Z}\}$	treatment-inducing proxy dimension
Dimension bounds		
`max_dim_X`	10	covariate-dimension ceiling
`max_dim_U`	10	confounder-dimension ceiling ($d^{\max}$ of Appendix B.2.2)
`max_dim_W`	10	outcome-confounding proxy dimension ceiling
`max_dim_Z`	10	treatment-confounding proxy dimension ceiling
Dataset		
`context_rows`	$\mathrm{Unif}\{64, \ldots, 2047\}$	context rows per episode
`n_query`	128	fixed query rows per episode
`mc_cate`	250	MC replicates K per row behind the μ_0/μ_1 targets
`mc_target_rows`	query rows only	rows carrying MC CEPO targets

distribution – typically a log-scaled truncated normal whose own centre is itself uniform on a log scale – so that mechanism depth, width, activation, and noise scale, as well as the causal knobs governing overlap and confounding strength, vary *across* the prior rather than being pinned to a single value.

The fixed table lists only the constants that pin down behaviour described qualitatively elsewhere in this appendix; the generator additionally exposes on/off switches for the mechanisms of the following subsections – the dimension heuristic (Appendix B.2.2), categorical completeness (Appendix B.2.4), the confounding-share subsampling of U's inputs to f_A and f_Y, and the per-column subsampling of the direct and covariate edges (Appendix B.2.3). All are enabled throughout the prior.

Table B.2: Fixed hyperparameters of the prior, held constant across every dataset. Listed are the constants that pin down behaviour the surrounding text describes only qualitatively; switches whose shipped value is that behaviour itself are stated in the text instead.

hyperparameter	value	description
Mechanism		
`noise_std`	0.001	tiny fixed per-layer dither so continuous columns realise distinct values
Feature corruption		
`max_categories`	50	cap on categories per feature
`proxy_cat_cover_prob`	0.5	chance a proxy mirrors a U category (vs. a continuous wildcard)
Positivity		
`propensity_epsilon`	0.001	positivity clamp on the propensity
Completeness		
`completeness_max_attempts`	50	resample budget before giving up

Table B.3: Per-dataset sampled hyperparameters of the prior. Each is drawn once per dataset from the listed meta-distribution; log-scaled truncated normals draw their centre log-uniformly over the stated mean range and add the lower bound as an offset.

hyperparameter	sampling law	description
Mechanism		
num_layers	log-scaled trunc. normal, mean $\sim \log \mathrm{Unif}[1, 6]$, int ≥ 2	mechanism MLP depth
hidden_dim	log-scaled trunc. normal, mean $\sim \log \mathrm{Unif}[5, 130]$, int ≥ 4	mechanism MLP width
mlp_activations	categorical over 54 random activation functions	activation, drawn from a random library
init_std	log-scaled trunc. normal, mean $\sim \log \mathrm{Unif}[0.01, 10]$, real ≥ 0	weight-initialisation scale
block_wise_dropout	categorical $\{True, False\}$ (random weights)	block-sparse vs. dense weight init
mlp_dropout_prob	scaled Beta on $[0, 0.9]$, shapes $\sim \mathrm{Unif}[0.1, 5]$	dropout rate (dense init only)
Noise		
sampling	categorical $\{\texttt{normal}, \texttt{mixed}, \texttt{uniform}\}$ (random weights)	family of the root cause variables
pre_sample_cause_stats	categorical $\{True, False\}$ (random weights)	pre-sample per-cause mean/std (normal sampling)
noise_share_of_input_in_U	$\mathrm{Beta}(1.2, 2.8)$	target noise share of f_U inputs (ε_U width)
noise_share_of_input_in_W	$\mathrm{Beta}(1.2, 2.8)$	target noise share of f_W inputs (ε_W width)
noise_share_of_input_in_Z	$\mathrm{Beta}(1.2, 2.8)$	target noise share of f_Z inputs (ε_Z width)
noise_share_of_input_in_A	$\mathrm{Beta}(1.2, 2.8)$	target noise share of f_A inputs (ε_A width)
noise_share_of_input_in_Y	$\mathrm{Beta}(1.2, 2.8)$	target noise share of f_Y inputs (ε_Y width)
Causal		
propensity_shift	$\mathrm{Unif}[-2.5, 2.5]$	treatment-logit bias (marginal treated fraction)
propensity_scale	log-scaled trunc. normal, mean $\sim \log \mathrm{Unif}[0.5, 3]$, real ≥ 0.1	treatment-logit gain (overlap vs. confounding)
confounding_share_of_U_in_Y	$\mathrm{Beta}(1.2, 2.8)$	target share of f_Y inputs drawn from U
confounding_share_of_U_in_A	$\mathrm{Beta}(1.2, 2.8)$	target share of f_A inputs drawn from U
direct_edge_share_of_W_in_Y	$\mathrm{Beta}(1, 1)$	per-column keep probability of the direct $W \to Y$ edge
direct_edge_share_of_Z_in_A	$\mathrm{Beta}(1, 1)$	per-column keep probability of the direct $Z \to A$ edge
covariate_share_of_X_in_U	$\mathrm{Beta}(1, 1)$	per-column keep probability of the direct $X \to U$ edge
covariate_share_of_X_in_W	$\mathrm{Beta}(1, 1)$	per-column keep probability of the direct $X \to W$ edge
covariate_share_of_X_in_Z	$\mathrm{Beta}(1, 1)$	per-column keep probability of the direct $X \to Z$ edge
covariate_share_of_X_in_A	$\mathrm{Beta}(1, 1)$	per-column keep probability of the direct $X \to A$ edge
covariate_share_of_X_in_Y	$\mathrm{Beta}(1, 1)$	per-column keep probability of the direct $X \to Y$ edge
Feature corruption		
cat_col_prob	$\mathrm{Beta}(1, 3)$	per-column categorisation probability

B.2.2 DIMENSIONS & DIMENSION HEURISTIC

The block dimensions are sampled according to the completeness heuristic introduced in Appendix B.1. The confounder dimension is drawn from a triangular (linearly decreasing) law on $\{1, \ldots, d_U^{\max}\}$,

$$\Pr[d_U = d] \;\propto\; (d_U^{\max} + 1 - d), \tag{22}$$

which skews d_U toward small values. Conditional on d_U, the two proxy dimensions are sampled uniformly subject to $d_W, d_Z \geq d_U$, while the covariate dimension d_X is sampled independently and uniformly:

$$d_W \sim \mathrm{Unif}\{d_U, \ldots, d_W^{\max}\}, \quad d_Z \sim \mathrm{Unif}\{d_U, \ldots, d_Z^{\max}\}, \quad d_X \sim \mathrm{Unif}\{d_X^{\min}, \ldots, d_X^{\max}\}. \tag{23}$$

This conditional-uniform sampling has a convenient consequence in the common-bound configuration used here, where d_U, d_W, and d_Z all range from 1 to a shared maximum m. When d_U follows the triangular law, the weight $(m + 1 - d)$ placed on $d_U = d$ exactly cancels the normaliser $1/(m - d + 1)$ of the conditional uniform, so the induced marginal of each proxy dimension is the linear ramp

$$\Pr[d_W = w] \;=\; \frac{w}{\sum_{j=1}^{m} j} \;=\; \frac{2w}{m(m+1)}, \qquad w \in \{1, \ldots, m\}, \tag{24}$$

and likewise for d_Z.

The triangular choice for d_U is what buys this. Had d_U been uniform, the same conditional sampling would induce a *harmonic* proxy marginal, $\Pr[d_W = w] = (H_m - H_{m-w})/m$ with H_n the n-th harmonic number, which piles proxy mass against the ceiling m: the value $w = m$ is reachable from every d_U, whereas small w requires a small d_U, so probability accumulates at the top (for $m = 10$, $\Pr[d_W = 10] = 0.29$). The triangular law flattens this to the linear ramp of (24) ($\Pr[d_W = 10] = 0.18$), and lowers the mean confounder and proxy dimensions from 5.5 and 7.75 to 4.0 and 7.0 respectively. Figure B.2 shows the delivered marginals of both schemes against these analytic laws.

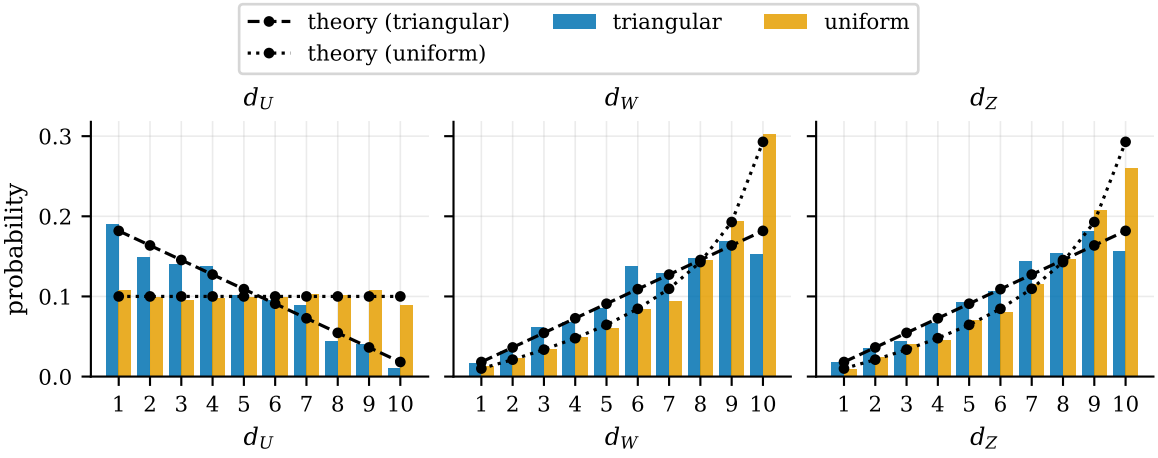


Figure B.2: Delivered marginal distributions of the block dimensions d_U, d_W, d_Z across the prior, for the shipped (triangular d_U) and the alternative (uniform d_U) sampling, overlaid with their analytic sampling laws (dashed: triangular; dotted: uniform). Bars are empirical frequencies over 1024 datasets. Uniform d_U pushes proxy mass against the ceiling $d^{\max} = 10$; the triangular law spreads it into the linear ramp of (24). Delivered bars track the theoretical curves, confirming the generator samples the intended laws.

B.2.3 MLP GENERATION, NOISE & EDGE SAMPLING

TabICL builds a single random MLP that turns a block of root "causes" into one feature table, reading its columns from the pooled activations of the MLP's intermediate layers. We reuse that construction unchanged as the mechanism for each *individual* node – the layer stack, the random activation library, the block-sparse or dropout-masked Gaussian initialisation, and the pooled-output

selection, all sampled once per dataset (Table B.3); we refer to TabICL for these details (Qu et al., 2025). In PROXIMALFM, the innovative step is that a proximal dataset is not one block but a *chain* of them, $f_X, f_U, f_W, f_Z, f_A, f_Y$ along a proximal learning DAG. Wiring one block into the next raises two questions a single-MLP generator never faces:

- How should a parent block enter the child's mechanism?
- How should we inject fresh per-node randomness while keeping it proportional to the signal?

Parents and noise. The root mechanism has no parents and maps a sampled noise input block straight to the covariates, $X = f_X(\varepsilon_X)$. Every downstream mechanism instead concatenates its standardised parent blocks with a fresh input block of its own, e.g. $U = f_U(\tilde{X}_U, \varepsilon_U)$ as in (21). These per-node blocks $\varepsilon_{(\cdot)}$ play the role of exogenous noise: we draw them from the *same* sampler TabICL uses for its root causes – each column independently normal, uniform, multinomial, or clamped Zipf under per-dataset mixing weights – and size each to a per-dataset share of its mechanism's signal width (Table B.3), so the noise-to-signal ratio stays controlled rather than shrinking as the parent blocks grow. The MLP's own within-layer Gaussian perturbation is left small ($\texttt{noise_std} = 10^{-3}$), so in practice the input blocks carry essentially all of a node's exogenous randomness. Postprocessing $\mathrm{Corrupt}(\cdot)$ (Appendix B.2.4) then reshapes each block before it is consumed.

Edge sampling. At the MLP level, the parent blocks above are passed through dataset-specific column masks. For each mechanism that depends on X, we draw a keep probability from $\mathrm{Unif}[0, 1]$ and retain the columns of $\tilde{X}$ independently according to that probability. The same procedure is applied to the $W \to Y$ and $Z \to A$ links, producing the masked proxy inputs $\tilde{W}_Y$ and $\tilde{Z}_A$. Thus edge sampling can only remove potential parents from the fixed proximal DAG: the completeness links $Z \leftarrow U \to W$ remain intact, and no new dependencies are introduced. The resulting subgraphs therefore retain the dimensionality heuristic and the structural proximal restrictions. The separate control of how strongly U enters f_A and f_Y is described next.

Confounding control. In a random MLP the confounder's influence on A and Y would be pinned by U's share of each mechanism's input width, hence by d_U; to decouple confounding strength from d_U we target a share $s_A, s_Y \sim \mathrm{Beta}(1.2, 2.8)$ of the inputs to f_A, f_Y to be U-derived and translate it into a column count k. When k does not exceed d_U the mechanism sees a k-column subsample of U; when it does, it sees all of U plus $(k - d_U)$ extra columns drawn from a shared nonlinear re-encoding $U^{\mathrm{aug}} = f_{\mathrm{aug}}(\tilde{U})$, itself a random MLP, giving the confounding-controlled inputs $U^{[A]}, U^{[Y]}$. Since f_{aug} is a deterministic function of U, these extra columns carry no information beyond U – they only let the prior push confounding above what d_U alone would permit, without altering the DAG or the completeness edges.

Figure B.3 evaluates the effect on the mechanisms' input variance of the four signal knobs, one per channel[2] governing the U-, X-, proxy- and noise-channels respectively. We observe clearly the expected behaviour, as higher values for a knob correspond to a higher proportion of input variance carried by its own channel, on average. Then, Figure B.4 considers the outputs of each mechanism, and evaluates their dependence on the input. On the delivered data we predict each mechanism's output from its parent blocks with a cross-fitted in-context regressor (TabICL, out-of-fold so every row is predicted exactly once), and read each block's *unique* contribution as the leave-one-out drop $R^2(\text{all}) - R^2(\text{all} \setminus b)$. For f_A we predict the raw treatment logit $\mathrm{logit}(\pi)$ from $\{U, X, Z\}$; for f_Y the potential outcome Y_0 from $\{U, X, W\}$; the noise channel, having no block to drop, is read as the unexplained share $1 - R^2$.

Figure B.4 shows each channel's contribution rising with its own knob, mirroring Figure B.3 at the output level: the mechanisms are predicted with median $R^2 \approx 0.5$, and the noise knob visibly inflates the unexplained share. Two caveats with this procedure. First, the blocks are supersets of the subsampled columns each mechanism actually consumed, so a drop is a *lower bound* on the true dependence. Second, because leave-one-out credits only *unique* signal, the confounder

[2]Confounder: `confounding_share_of_U_in_*`. Covariate: `covariate_share_of_X_in_*`. Proxy: `direct_edge_share_of_*_in_*`. Noise: `noise_share_of_input_in_*`.

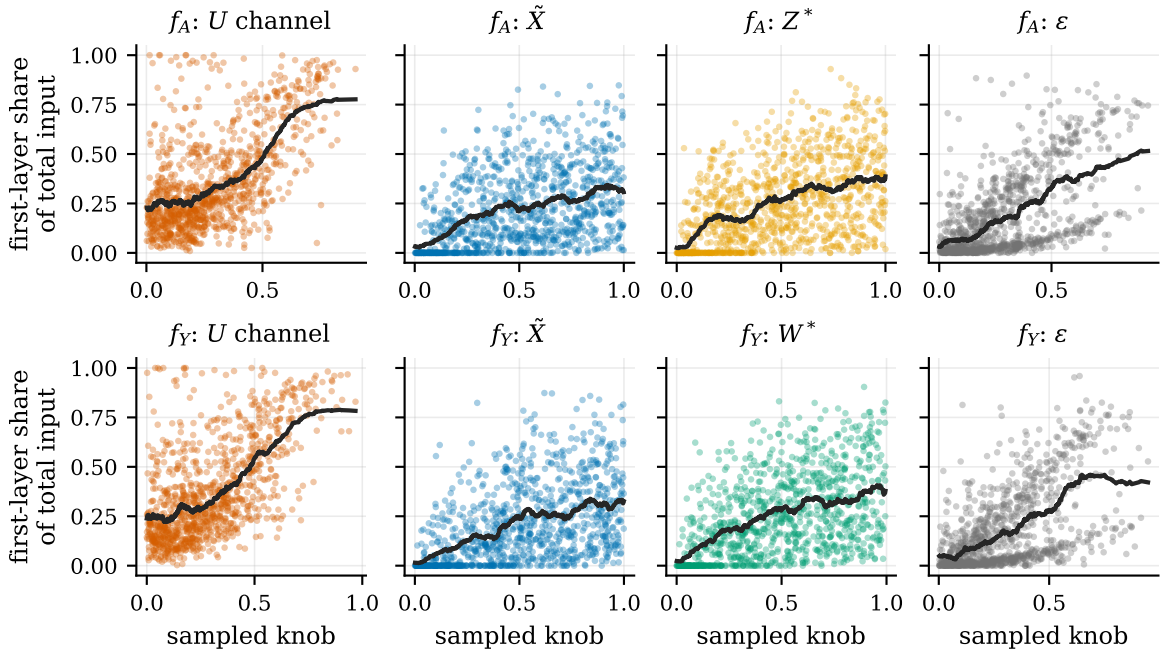


Figure B.3: Every dial: each input channel's share of f_A's (top) and f_Y's (bottom) total input *variance*, against its own sampled knob (s for the U-channel, the covariate and proxy keep-probabilities for $\tilde{X}$ and $\tilde{Z}_A/\tilde{W}_Y$, the noise share for ε). Solid line: moving average. Each knob steers its channel's delivered variance share.

channel reads low wherever U is partly recoverable from its child proxy. On the Y side, Y_0 is post heterogeneity-remix, so its $1 - R^2$ also absorbs the exogenous mixing (which vanishes as $\gamma \to 1$).

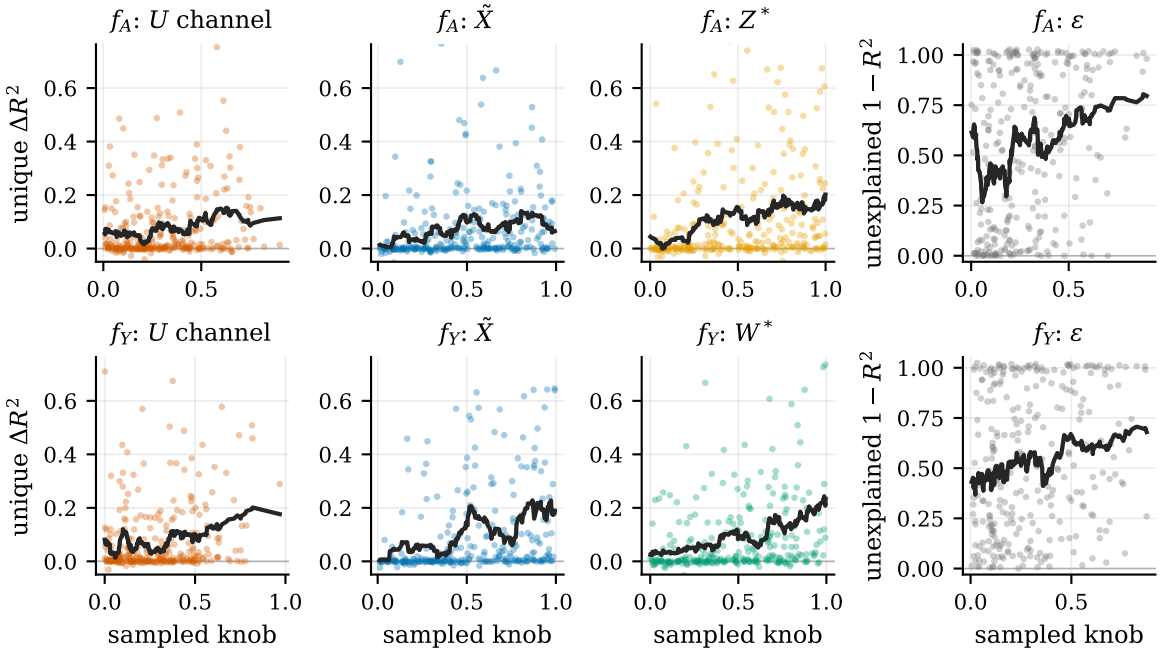


Figure B.4: Output side of the composition: what each mechanism's output *depends on*, measured by a cross-fitted TabICL regression on the delivered data. Columns 1–3 plot each parent channel's unique leave-one-out ΔR^2 (drop in out-of-fold R^2 when that block is removed from the predictors) against its own knob, for f_A (top, target $\mathrm{logit}(\pi)$) and f_Y (bottom, target Y_0); column 4 plots the unexplained share $1 - R^2$ against the noise knob. Solid line: moving average. Each channel's dependence rises with its knob, as at the input level. The U-channel reads modestly because leave-one-out credits only signal *unique* to U, not the part shared with its child proxy.

B.2.4 Post-process: categorisation, outlier and scaling

Each block passes through the $\mathrm{Corrupt}(\cdot)$ pipeline of TabICL (Qu et al., 2025) before being consumed in (21): a random subset of columns is discretised into categories, outliers are clamped at 4 standard deviations, and the result is standardised. Like TabICL, we apply this to each mechanism's output rather than *within* its MLP layers, as Stith et al. (2026a) instead suggest.

Categorisation interacts with the completeness assumptions of Appendix B, which we preserve through two heuristics on the *delivered* blocks. *Numerically*, we keep $d_W, d_Z \geq d_U$. *Categorically*, every column of U must be dominated by a distinct proxy column: a numerical column of U by a numerical proxy column, and a categorical column of U by either a numerical proxy column or a categorical one with at least as many categories; as exemplified in Table B.4.

Numeric completeness. The dimension heuristic is imposed at sampling time, but the constant-column filtering that follows (Algorithm 1, line 20) can shrink a block afterwards, so we re-check it on the delivered dimensions. Figure B.5 illustrates this heuristic enforcement.

Categorical completeness. We call P *categorically complete* w.r.t. U when every column of U matches a distinct, at-least-as-informative column of P ("dominates"). Whenever U is categorised, the prior builds each proxy to satisfy this by construction: it reserves one numerical column per numerical column of U, covers each categorical column of U with either a numerical column or – with probability $\texttt{proxy_cat_cover_prob} = 50\%$ – a categorical column drawing strictly more categories, and fills any surplus proxy columns freely. Figure B.6 contrasts the block compositions this yields with and without this construction: *default* (enforcement on) and *no_enforcement* (U categorised, proxies left unconstrained) start from the same categorised U, but under *no_enforcement* the proxies end up no more numerical than U itself, so nothing forces domination.

Table B.4: Worked examples of the categorical-completeness heuristic for the fixed confounder profile $U = [\,\text{num},\ 5\text{-cat},\ 3\text{-cat}\,]$ (one numerical column and two categorical columns, with 5 and 3 categories). A proxy block P satisfies the heuristic when its columns can be matched one-to-one to those of U so that each is at least as informative: a numerical column dominates anything, and a categorical column dominates a categorical requirement only if it has at least as many categories. A numerical proxy column may thus cover a categorical requirement, but a categorical proxy column can never cover a numerical one.

Proxy profile P	Heuristic?	Reason
$[\,\text{num}, 5\text{-cat}, 3\text{-cat}\,]$	**ok**	exact match, column for column
$[\,\text{num}, 8\text{-cat}, 4\text{-cat}\,]$	**ok**	categorical columns carry more categories than required
$[\,\text{num}, \text{num}, 3\text{-cat}\,]$	**ok**	a numerical column covers U's 5-cat column
$[\,\text{num}, \text{num}, \text{num}\,]$	**ok**	numerical columns dominate every requirement
$[\,\text{num}, \text{num}\,]$	**no**	too few columns ($d_P = 3 < d_U = 4$)
$[\,\text{num}, 4\text{-cat}, 3\text{-cat}\,]$	**no**	the 4-cat column cannot cover U's 5-cat column
$[\,5\text{-cat}, 5\text{-cat}, 3\text{-cat}\,]$	**no**	only one numerical column for U's two numerical columns
$[\,\text{num}, \text{num}, 2\text{-cat}\,]$	**no**	a numerical wildcard covers the 5-cat column, leaving the 2-cat column to cover U's 3-cat column, but $2 < 3$

Enforcement scales with U's demand. Figure B.7 reads the covering rule off the delivered data from two angles. Conditioning on a single U column's cardinality (panels a–b), the proxy block's numerical fraction is roughly flat in that cardinality and sits *above* the unconstrained arm (≈ 0.70 vs. ≈ 0.60), while the mean category count of the proxy's categorical columns grows with the U column's cardinality under *default* (≈ 12 at 2 categories to ≈ 19 in the 30+ bin), exactly as the "strictly more categories" covering rule dictates, and stays flat at ≈ 10 under *no_enforcement*. Enforcement therefore acts on the *depth* of the proxy's categorical columns, not on how many of them there are.

Panel (c) makes the same point negatively, at the dataset level. The categorical share of the proxies rises with U's categorical demand $|\text{cats}(U)|$ in *both* arms, because the per-dataset categorisation propensity is shared across blocks: a dataset that discretises many columns of U also discretises many columns of its proxies, enforcement or not. Enforcement in fact leaves the proxies *less* categorical than the free *no_enforcement* arm at every demand level ($0.08 \to 0.43$ vs. $0.14 \to 0.58$), consistent with the reserved numerical columns of panel (a). So the categorical share is not the signature of enforcement; the per-column depth of panel (b), and the survival rate below, are.

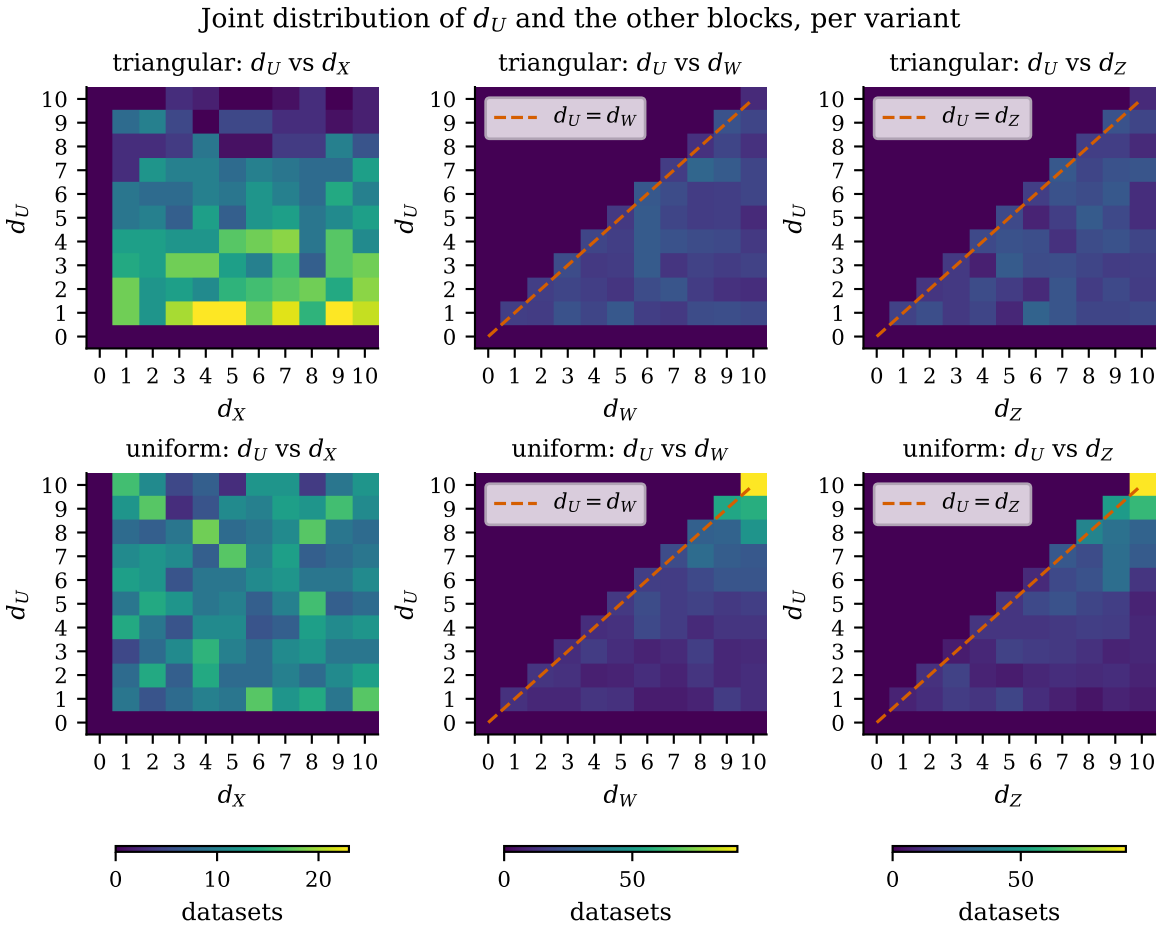


Figure B.5: Joint distribution of the latent dimension d_U against each feature block (d_X, d_W, d_Z), one row per prior variant (*triangular*, *uniform*). Colour encodes the number of delivered datasets falling in each $(d_U, \cdot)$ cell. The diagonal line on the d_W/d_Z panels marks the completeness heuristic $d_W, d_Z \geq d_U$; both variants place their mass on or below it, i.e. constant-column filtering does not measurably break the heuristic imposed at sampling time. d_X carries no such constraint and is shown for reference.

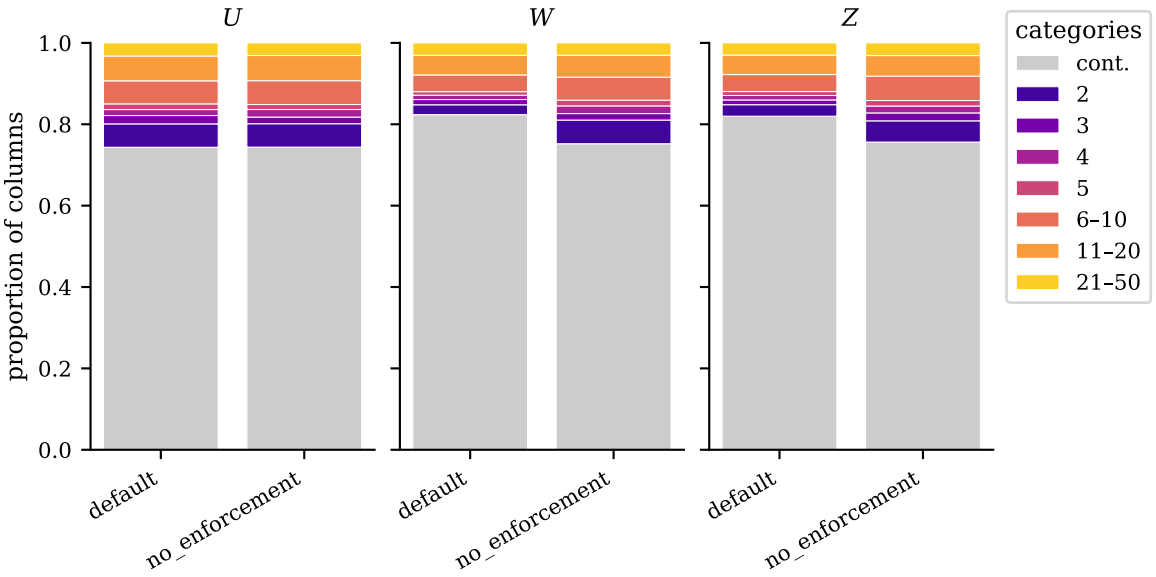


Figure B.6: Column-type composition of each block (U, W, Z), one 100%-stacked bar per variant (*default*: categorical completeness enforced; *no_enforcement*: proxies left unconstrained). Each bar partitions a block's columns into continuous (grey, "cont.") and categorical, the latter binned by category count (2–5 singletons, then coarser 6–10, 11–20, 21–$d_{\text{cat}}^{\max}$ tail bins). Continuous columns dominate every block; contrasting W/Z across the two variants shows how much enforcement pushes proxy columns categorical relative to the unconstrained *no_enforcement* arm.

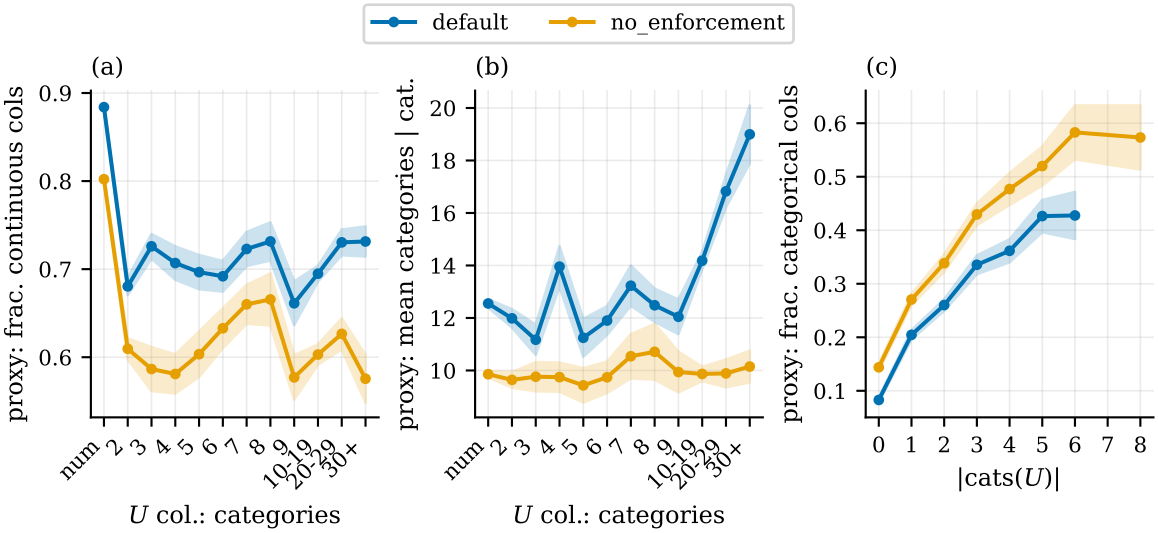


Figure B.7: How enforcement makes proxy structure track U's categorical demand, pooling W and Z across the *default* and *no_enforcement* variants; shaded bands are $\pm$ one SEM. **(a, b)** condition on a single U column's own cardinality: the x-axis bins a U column by its category count (continuous "num", then $2, \ldots, 9$, and coarser tail bins up to "30+"), and each point averages a statistic of the co-occurring proxy block over every dataset containing a U column of that kind – (a) the fraction of proxy columns that are continuous, essentially flat in U's cardinality but higher under *default*, and (b) the mean category count of the proxy's categorical columns, which rises with U's cardinality under *default* but stays low under *no_enforcement*. **(c)** moves to the dataset level: the mean fraction of categorical proxy columns against U's total categorical demand $|\mathrm{cats}(U)|$, with bins of fewer than 5 datasets dropped. Both arms rise (the per-dataset categorisation propensity is shared across blocks) and *default* sits below *no_enforcement*: enforcement deepens the proxy's categorical columns rather than multiplying them. Because proxy statistics in (a, b) are block-level rather than tied to a matched column, those trends reflect association with "the dataset contains a U column of this kind," not a one-to-one mapping.

B.2.5 TREATMENT ASSIGNMENT

The treatment mechanism f_A maps the confounding-controlled confounder input $U^{[A]}$, the covariate subset $\tilde{X}_A$, the direct proxy subset $\tilde{Z}_A$ and fresh noise ε_A to a single scalar logit per row,

$$\ell \;=\; f_A\big(U^{[A]},\, \tilde{X}_A,\, \tilde{Z}_A,\, \varepsilon_A\big) \;\in\; \mathbb{R}^{TK}, \tag{25}$$

with W excluded from the inputs so that the proxy exclusion restriction of Assumption 2 holds by construction (Equation 21). The treatment A is binary; the steps below turn ℓ into a propensity score $\pi = \Pr[A = 1 \mid \cdot]$ and draw A.

Logit normalisation. A random MLP fixes neither the location nor the spread of its output, both of which drift with depth, width and input scale; left raw, the induced treatment prevalence and the amount of overlap would be an uncontrolled by-product of the sampled architecture. We therefore standardise the logits across the $T \cdot K$ rows of the dataset to zero mean and unit variance,

$$\tilde{\ell} \;=\; \frac{\ell - \bar{\ell}}{\mathrm{sd}(\ell) + 10^{-8}}, \tag{26}$$

so that the assignment distribution is decoupled from the accidental scale of f_A and every dataset starts from a common footing before the overlap knobs act.

Shift and scale. Two per-dataset propensity knobs then set the assignment regime through an affine map of the standardised logits,

$$\ell' \;=\; s\,\tilde{\ell} + b, \qquad b \sim \mathrm{Unif}(-2.5,\, 2.5), \quad s \sim \text{log-scaled truncated normal}, \tag{27}$$

both drawn once per dataset (see Table B.3). The shift b moves the marginal treatment prevalence away from the balanced split – negative b makes treatment rare, positive b makes it common – while the scale s controls how sharply assignment depends on the covariates and confounder: small s flattens the logits toward a coin flip (strong overlap, weak confounding channelled through A), whereas large s sharpens them toward near-deterministic assignment (weaker overlap).

Propensity and sampling. A sigmoid turns the shifted logits into a propensity score, from which the treatment is drawn Bernoulli,

$$\pi \;=\; \sigma(\ell'), \qquad A \sim \mathrm{Bernoulli}(\pi). \tag{28}$$

Overlap clamp. Because a large s can push π arbitrarily close to 0 or 1, we clamp the propensity into $[\epsilon, 1-\epsilon]$ with $\epsilon = 10^{-3}$ before sampling,

$$\pi \;\leftarrow\; \mathrm{clip}(\pi,\, \epsilon,\, 1-\epsilon), \tag{29}$$

guaranteeing a floor on the probability of each treatment arm and hence enforcing positivity/overlap ($0 < \Pr[A = 1 \mid \cdot] < 1$) for every row by construction (Assumption 3). We audit the delivered overlap across the prior in Appendix B.3.1.

Figure B.8 shows this five-step pipeline on a single representative dataset, following one unit (in red) from its raw logit through to its treatment draw.

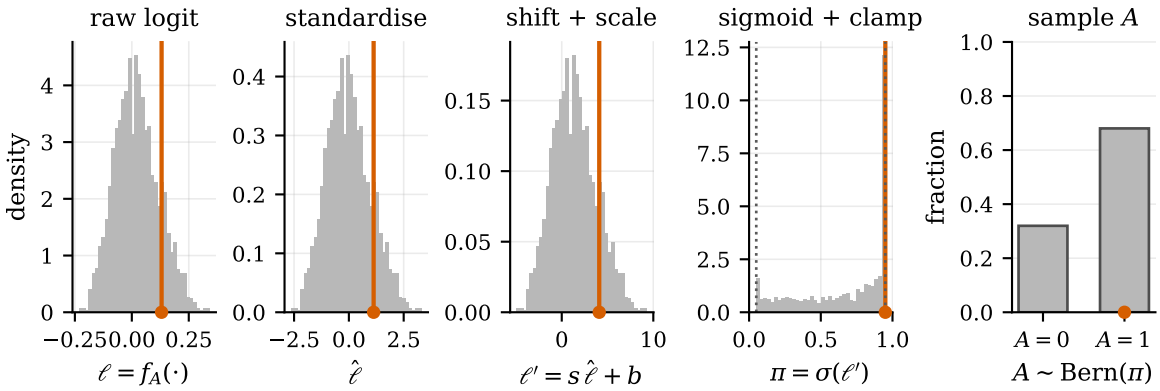


Figure B.8: The treatment-assignment pipeline of (25)–(29) on one representative dataset, tracking a single unit (red line) across all five stages. (a) The raw logit $\ell = f_A(\cdot)$, whose scale and offset are an uncontrolled by-product of the sampled architecture. (b) The standardised $\tilde{\ell}$ of (26), pinned to zero mean and unit variance. (c) The affine $\ell' = s\,\tilde{\ell} + b$ of (27), shifted and sharpened by the two sampled overlap knobs. (d) The propensity $\pi = \sigma(\ell')$ of (28), confined to $[\epsilon, 1-\epsilon]$ by the clamp of (29) (dotted bounds, drawn at an enlarged ϵ for visibility; production $\epsilon = 10^{-3}$). (e) The realised treatment $A \sim \mathrm{Bernoulli}(\pi)$ across the batch (fraction in each arm), with the tracked unit's realised arm marked in red.

B.2.6 Potential outcomes & heterogeneity remix

The outcome mechanism f_Y emits both potential outcomes from a single forward pass, returning the pair $\big(Y_0^{\mathrm{raw}}, Y_1^{\mathrm{raw}}\big)$ from a common input and exogenous draw ε_Y; the two arms differ only in their read-out, so the raw effect $\tau^{\mathrm{raw}} = Y_1^{\mathrm{raw}} - Y_0^{\mathrm{raw}}$ stays coupled through the shared latent state.

Left as is, a dataset's effect heterogeneity would be an uncontrolled by-product of the sampled MLP. To let the prior span the full range from strongly modulated effects to a single constant effect, we remix the raw outcomes before forming the targets. With a per-dataset dial $\gamma \sim \mathrm{Unif}(0,1)$, a per-row weight $\alpha \sim \mathrm{Unif}(0,1)$, and the raw average effect $\bar{\tau} = \frac{1}{T \cdot K}\sum(Y_1^{\mathrm{raw}} - Y_0^{\mathrm{raw}})$,

$$\begin{aligned} Y_1 &= \big(\alpha + (1-\alpha)\gamma\big)\, Y_1^{\mathrm{raw}} + (1-\gamma)(1-\alpha)\big(Y_0^{\mathrm{raw}} + \bar{\tau}\big), \\ Y_0 &= \big((1-\alpha) + \alpha\gamma\big)\, Y_0^{\mathrm{raw}} + (1-\gamma)\alpha\big(Y_1^{\mathrm{raw}} - \bar{\tau}\big). \end{aligned} \tag{30}$$

This construction follows the heterogeneity knob of RealCause (Neal et al., 2021), later re-used in the causal-foundation-model prior of CausalPFN (Balazadeh Meresht et al., 2026).

Substituting (30) collapses the per-row weights to $Y_1 - Y_0 = \gamma\,\tau^{\mathrm{raw}} + (1-\gamma)\bar{\tau}$, so γ linearly interpolates each unit's effect between its raw value ($\gamma = 1$, fully heterogeneous) and the dataset mean $\bar{\tau}$ ($\gamma = 0$, fully homogeneous). Anchoring at $\bar{\tau}$ rather than zero preserves the ATE at every γ; the per-row α simply shares the required shift between Y_0 and Y_1 for each unit; averaged across rows, this allocation and the compensating $\pm\bar{\tau}$ terms leave the marginal means of Y_0, Y_1 (hence the outcome levels and the ATE) unchanged in expectation, so the dial reshapes only the *spread* of the effect.

Because the collapse also gives $\mathrm{sd}(Y_1 - Y_0) = \gamma\,\mathrm{sd}(\tau^{\mathrm{raw}})$, the dial should act on the delivered data as a pure, proportional spread control. Figure B.9 checks this across the prior: γ is drawn independently per dataset and recorded, so the shipped prior is itself a γ sweep, and the (outcome-scaled) effect dispersion tracks it closely, while the outcome coupling $\mathrm{corr}(Y_0, Y_1)$ approaches its exact $\gamma \to 0$ value of 1.

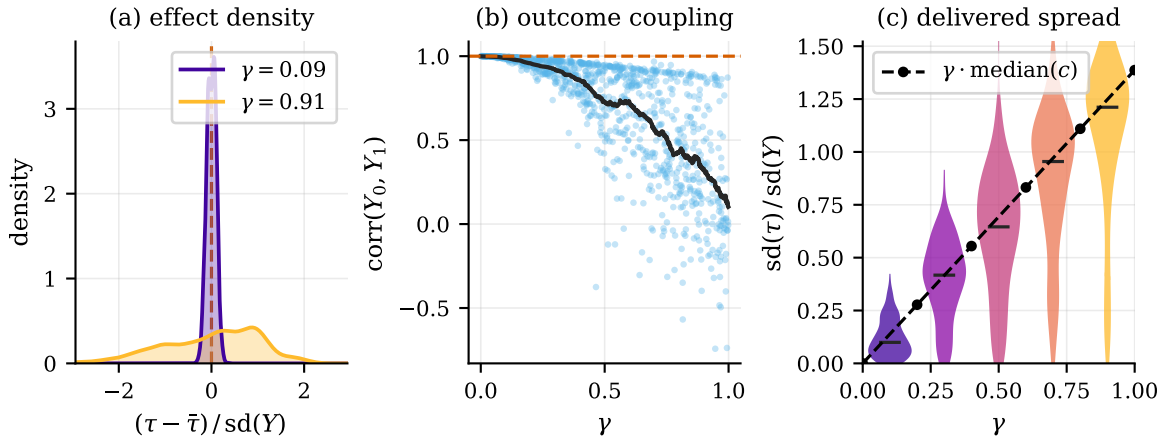


Figure B.9: The heterogeneity dial on the delivered prior, over 1024 datasets. All statistics are ratios, since each SCM's outcome carries its own arbitrary scale. (a) Two datasets' effect densities on a shared outcome-scaled axis, each centred on its own $\bar{\tau}$ (dashed line), one gathered ($\gamma = 0.09$) and one spread ($\gamma = 0.91$); the pair is chosen with *matched* raw dispersion $c = \mathrm{sd}(\tau^{\mathrm{raw}})/\mathrm{sd}(Y)$ (1.127 vs. 1.125), so the contrast is attributable to γ rather than to SCM variation. (b) The outcome coupling $\mathrm{corr}(Y_0, Y_1)$ against γ (solid line: moving average). At $\gamma = 0$ the effect is the constant $\bar{\tau}$, so $Y_1 = Y_0 + \bar{\tau}$ and the coupling is *exactly* 1 (dashed reference). (c) Distribution of the delivered dispersion $\mathrm{sd}(\tau)/\mathrm{sd}(Y)$ per γ bin (Spearman 0.78 overall), against the proportionality $\gamma \cdot \mathrm{median}(c)$ implied by the collapse; within-bin spread is dominated by variation in c. Because the reference line is anchored at zero and uses the overall median c for its slope, the panel verifies strict proportionality to γ—linear scaling starting from the origin—rather than predicting the absolute magnitude of dispersion.

B.2.7 Monte Carlo CATE supervision

For a query covariate row x_i, `mc_cate` specifies the number K of independent downstream SCM draws used to approximate its CATE. The covariates X remain fixed, while the latent confounder U, proxies, and exogenous outcome noise are redrawn. Within draw k, both potential outcomes are evaluated on the same latent state, yielding the paired effect $\tau_{i,k} = Y_{i,k}(1) - Y_{i,k}(0)$. We supervise on $\widehat{Q}_K(x_i) = K^{-1} \sum_k \tau_{i,k}$, an unbiased estimate of $Q(x_i)$, with estimated sampling variance $S_i^2 = \widehat{\mathrm{Var}}_k(\tau_{i,k})/K$. Thus K controls target precision, not the SCM or estimand: its standard error decreases as $K^{-1/2}$, at generation cost proportional to K.

B.2.8 Visualize the prior

We close Appendix B.2 with three galleries of the delivered data, each a fixed-seed cross-section of the prior's diversity: Figure B.10 shows individual *feature marginals*, one column per block and one sampled dataset per row (the propensity π for A, both potential outcomes Y_0, Y_1 for Y); Figure B.11 shows *within-block* dependence, scattering two features of the same block (π against the draw A, and Y_0 against Y_1 for the scalar blocks); and Figure B.12 shows *cross-block* dependence for a chosen few block pairs – the parent→child edges U–W, U–Z, the proxy pair W–Z (associated only through the shared latent U), and the confounder against the propensity U–π.

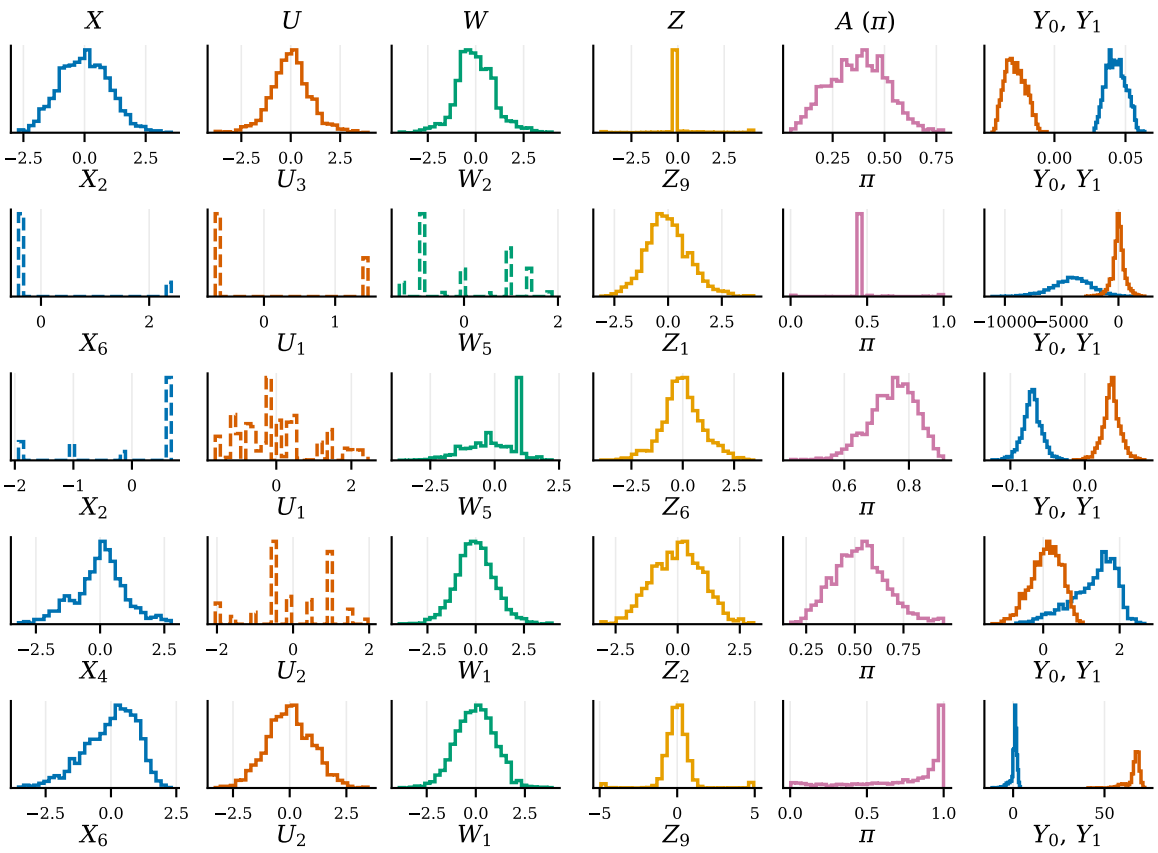


Figure B.10: Gallery of feature marginals: one column per block (X, U, W, Z, the propensity $\pi = P(A{=}1 \mid \cdot)$ for A, both potential outcomes for Y), one row per randomly sampled dataset. Each panel is one feature column's histogram, solid for continuous columns and dashed for categorical ones; the Y column overlays Y_0 (blue) and Y_1 (red). Feature blocks share the standardised axis Corrupt leaves; $\pi \in [0, 1]$ and Y keep their natural scale.

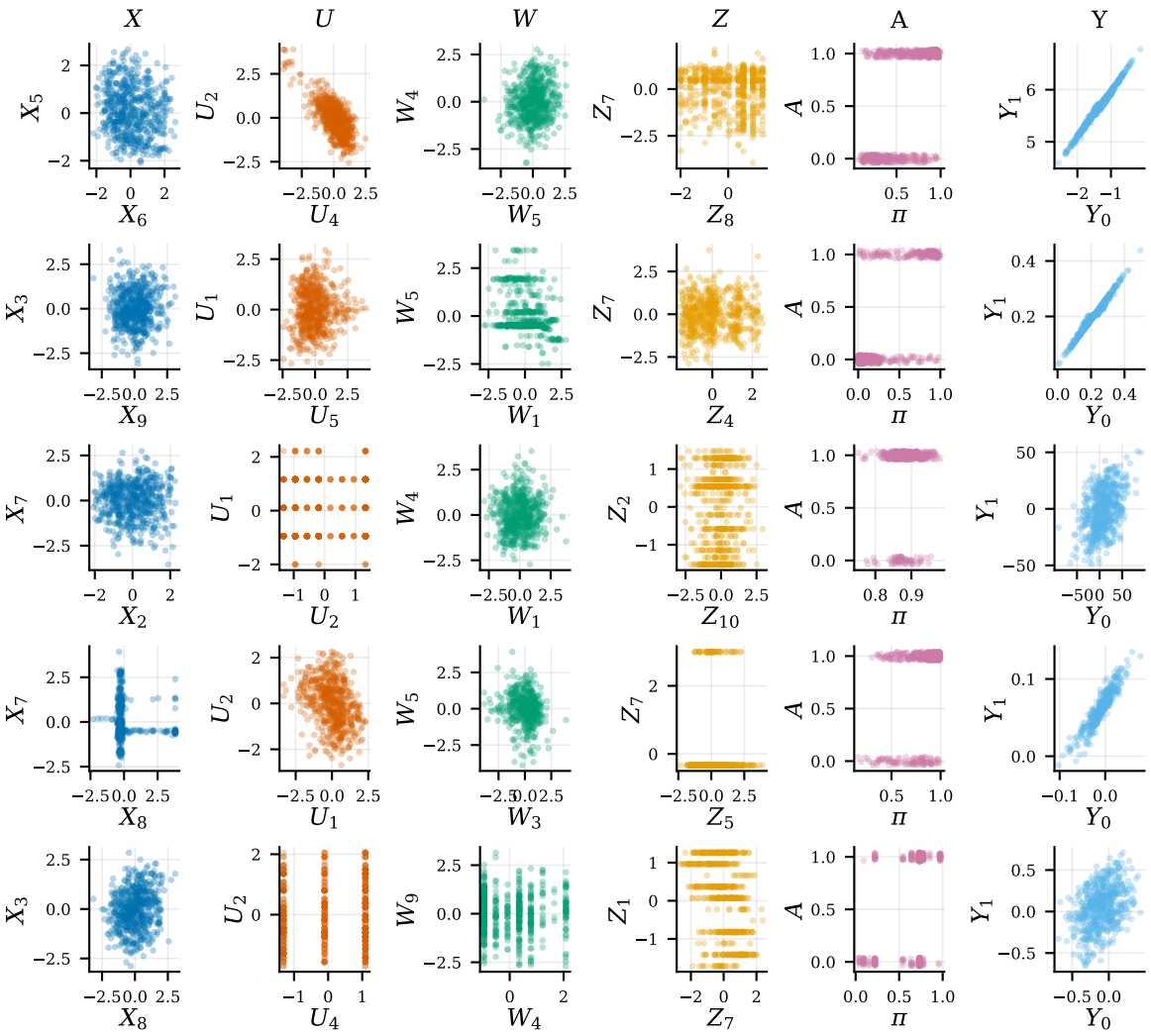


Figure B.11: Within-block dependence: one column per block, one randomly sampled dataset per row, each panel a scatter of two features from that block. For X, U, W, Z the two features are random columns of the block (sampled from datasets of width ≥ 2); for A the panel plots the propensity π against the treatment draw A; for Y the two potential outcomes Y_0 vs Y_1. Rows are subsampled for legibility; feature axes inherit the standardised scale.

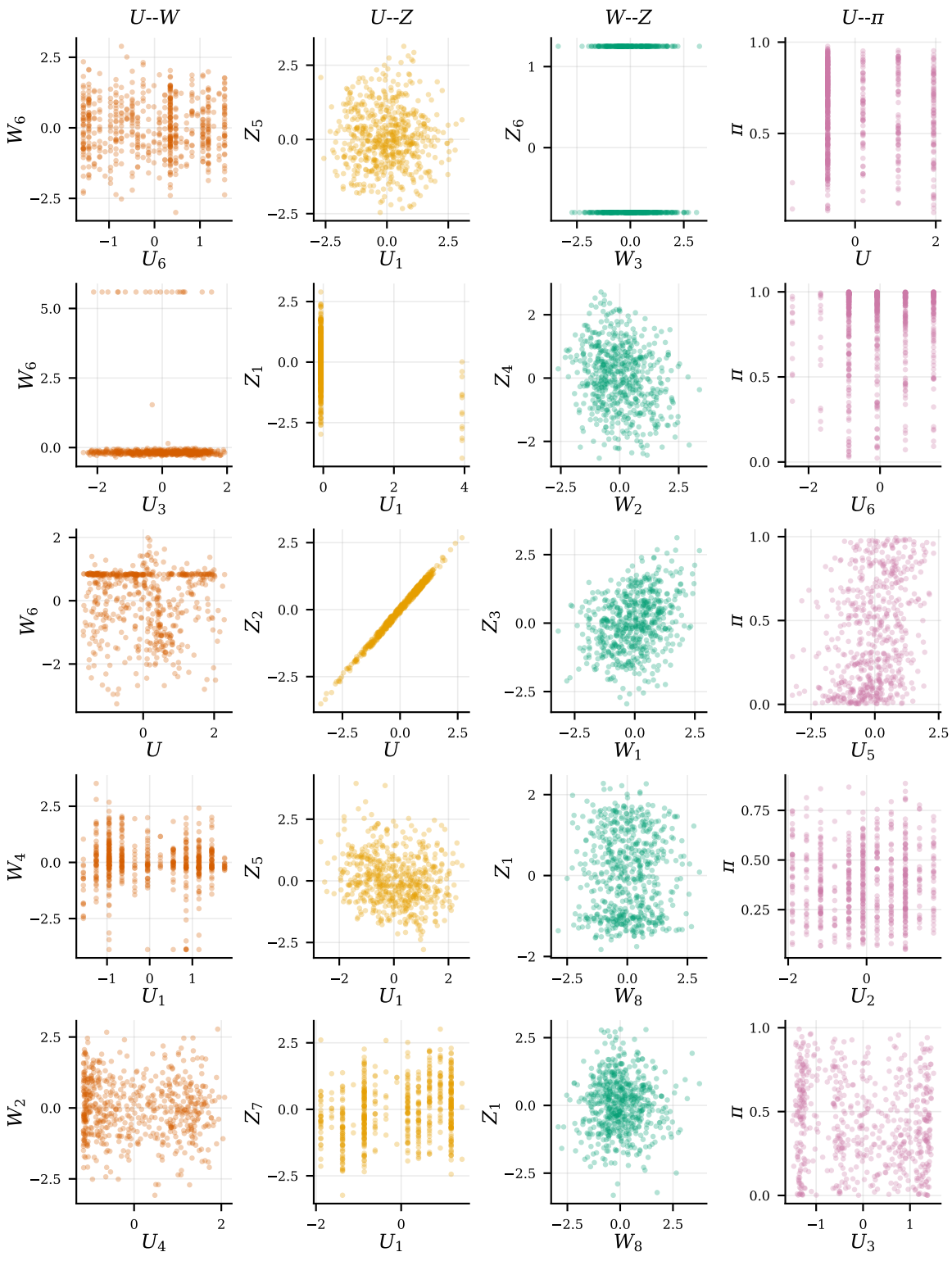


Figure B.12: Cross-block dependence: one column per chosen block pair, one sampled dataset per row, each panel a scatter of one feature from each block. U–W and U–Z are the parent→child edges; W–Z is the proxy pair, associated only through the shared latent U (no direct edge); U–π is the confounder against the propensity $\pi = P(A{=}1 \mid \cdot)$. Rows are subsampled for legibility; feature axes inherit the standardised scale.

B.3 Proximal assumption audit

B.3.1 Overlap

Positivity (Assumption 3) cannot fail in this prior: the clamp of (29) confines π to $[\varepsilon, 1-\varepsilon]$ with $\varepsilon = 10^{-3}$ chosen deliberately small (see Table B.2), so π is never exactly 0 or 1 and every row carries nonzero mass in both treatment arms by construction. Strictly, the stored π is realised conditional on the unit's full set of mechanism inputs (X, U, Z, ϵ_A) rather than the marginal $\Pr[A = 1 \mid X, U]$ that Assumption 3 is stated over, since A's logit also depends on the proxy Z and the mechanism's own exogenous noise; but the marginal is the expectation of this finer-grained π over Z, ϵ_A, so keeping every realised π away from $0, 1$ is a sufficient, if anything conservative, condition for the marginal to be as well, and auditing the stored values directly is valid. The question worth auditing is therefore not whether positivity holds but how close the sampled overlap knobs push π toward that clamp in practice, since a propensity pinned near 0 or 1 leaves little residual randomness in treatment assignment for any estimator to exploit even though positivity technically holds. The audit needs no additional estimation: Figure B.13 is a direct summary of these stored propensities.

Because ε is small, the clamp only bites at the very extremes and the vast majority of rows sit well within it: across the prior, 84.0% of all rows fall inside the conventional band $[0.05, 0.95]$, and only 42.6% of datasets contain even a single row on the clamp, with the median dataset containing none. Overlap is nonetheless deliberately stressed away from that safe interior: the median dataset still places 13.6% of its rows outside $[0.05, 0.95]$, since the sharpness s of (27) is sampled precisely to span easy and hard assignment regimes.

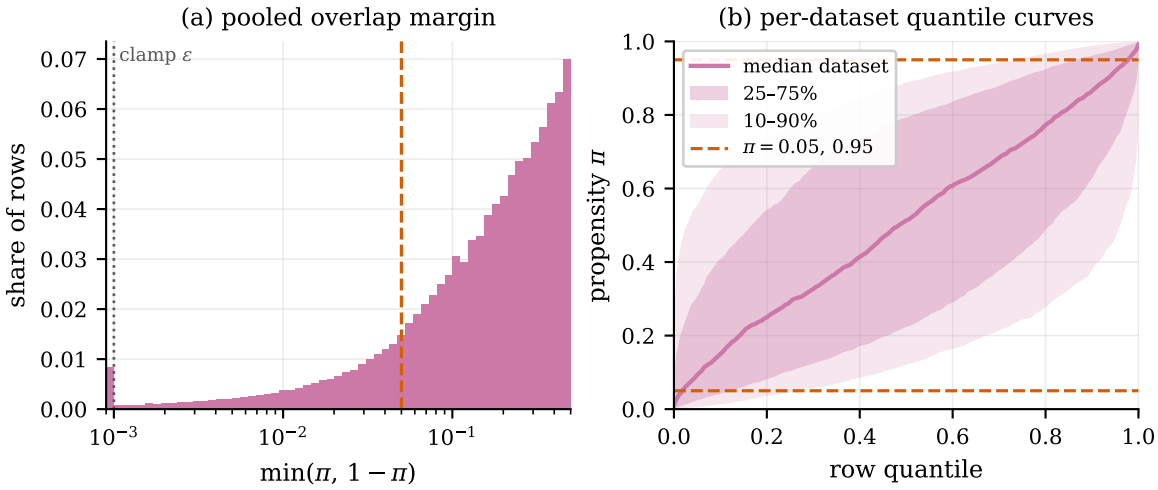


Figure B.13: Delivered overlap across the prior, from the stored true propensity $\pi = \Pr[A = 1 \mid X, U]$ (no estimation involved). (a) Pooled distribution of the overlap margin $\min(\pi, 1-\pi)$ over all rows of 1024 datasets, on a logarithmic axis: the margin measures each unit's distance from a degenerate assignment, folding both tails into one axis, and the log scale is what makes the mass piled on the clamp $\varepsilon = 10^{-3}$ (dotted) visible at all. Bars are per-bin shares of all rows on a logarithmic grid, so bin width grows to the right. (b) Per-dataset quantile function of π, summarised across datasets by the median curve with the 25–75% and 10–90% inter-dataset bands. Dashed lines mark the practical-overlap band $[0.05, 0.95]$ in both panels.

B.3.2 Exclusion restrictions

The proxy exclusion restriction of Assumption 2, $Y \perp\!\!\!\perp Z \mid (A, X, U)$ and $W \perp\!\!\!\perp (A, Z) \mid (X, U)$, holds *by construction*: (21) never wires Z into f_W or f_Y, nor W into f_Z or f_A. We audit this with a cross-fitted regression: on the Y side, the incremental $\Delta R^2_Z = R^2(Y_{\text{obs}} \mid A, X, U, Z) - R^2(Y_{\text{obs}} \mid A, X, U)$ of adding Z to a conditioning set that already contains everything the clause names; on the A side, the analogous ΔR^2_W for W against the raw treatment logit $\text{logit}(\pi)$ given (X, U, W). Both are expected to be ≈ 0.

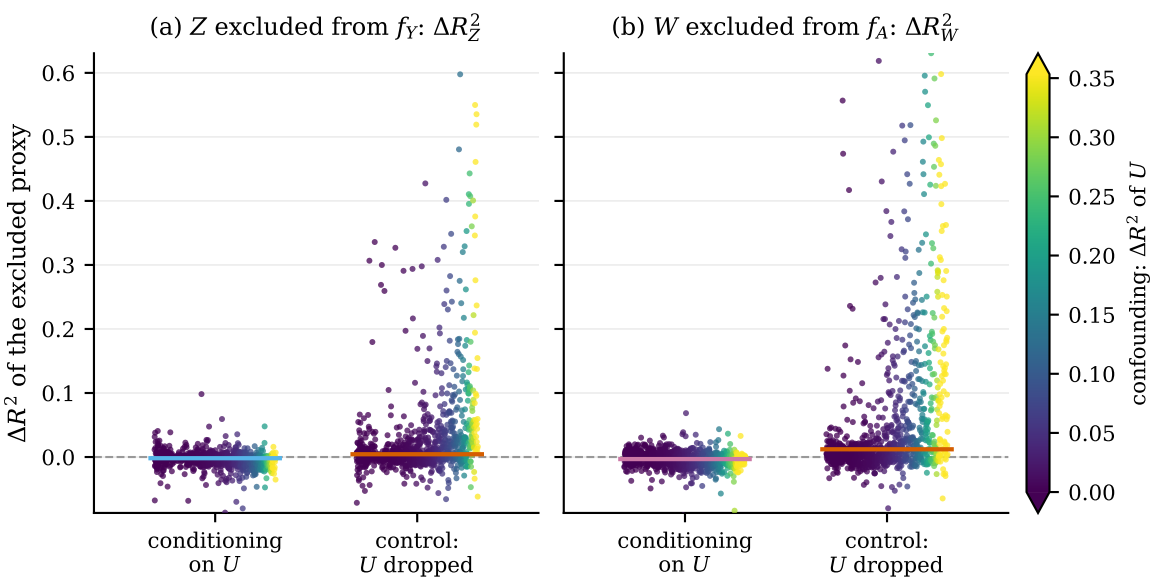


Figure B.14: Exclusion-restriction audit (Assumption 2) over 1024 datasets. Each panel is one clause – (a) Z excluded from f_Y, (b) W excluded from f_A – and each column a cross-fitted incremental ΔR^2 of the excluded block: *conditioning on* U is the check (expected ≈ 0), *control: U dropped* the positive control (expected clearly positive). Colour and horizontal position both encode $u_dependence = R^2(\text{with } U) - R^2(\text{without } U)$, i.e. how much confounding that draw carries (linear ramp, floored at 0, capped at the 95th percentile). Both checks stay on zero regardless of colour; both controls fan upward with warmer colour, most visibly in the top tercile of $u_dependence$ (Table B.5).

A near-zero ΔR^2 is only informative if the estimator could have detected a leak had one been present. We therefore pair each check with a positive control that repeats the same regression with U dropped from the conditioning set (Y-side: $\{A, X, Z\}$; A-side: $\{X, W\}$): since U is the only path linking Z to Y and W to A, removing it makes the excluded block genuinely predictive, and the control must light up clearly nonzero. Because the control's power scales with how much confounding a given draw actually carries, the figure also colours and ranks each point by *$u_dependence$* $= R^2(\text{with } U) - R^2(\text{without } U)$, the unique contribution of U to the target: the control should fan upward with warmer colour, while the check stays flat and near zero regardless of colour.

Figure B.14 confirms this reading. Both checks sit on zero (median $\Delta R^2_Z = -0.0019$, $\Delta R^2_W = -0.0032$), while both controls are slightly positive and rise further on the third of datasets with the largest U-dependence, with a positive Spearman correlation between $u_dependence$ and the control (0.375 Y-side, 0.415 A-side) visible as the colour gradient climbing alongside the control column. The zero-leak reading is therefore not an artefact of a blind test: the same regression detects a real, confounder-mediated dependence whenever U is removed, and detects none when the exclusion restriction is left intact.

Table B.5: Exclusion-restriction audit (Assumption 2): cross-fitted incremental ΔR^2 of the excluded block over the stated conditioning set, for the check (conditions on U) and the positive control (U dropped). *Top confounded* is the third of datasets with the largest $u_dependence$ (R^2 with U minus R^2 without it).

conditioning set	role	datasets	median ΔR^2	90th pct	share > 0.02
Z **excluded from** f_Y					
$\{A, X, U, Z\}$	conditions on U	all	-0.002	0.007	0.021
$\{A, X, U, Z\}$	conditions on U	top confounded	-0.005	0.004	0.012
$\{A, X, Z\}$	positive control	all	0.004	0.099	0.313
$\{A, X, Z\}$	positive control	top confounded	0.029	0.222	0.598
W **excluded from** f_A					
$\{X, U, W\}$	conditions on U	all	-0.003	0.009	0.020
$\{X, U, W\}$	conditions on U	top confounded	-0.004	0.004	0.021
$\{X, W\}$	positive control	all	0.012	0.218	0.415
$\{X, W\}$	positive control	top confounded	0.060	0.395	0.677

B.3.3 Latent exchangeability and confounding strength

Latent exchangeability (Assumption 1) holds by construction. Treatment is drawn from $f_A(X, U, Z, \varepsilon_A)$, while both potential outcomes are generated without treatment as an input and depend on X, U, W and independent outcome noise (Equation (21)). Conditional on (X, U), the exogenous draws generating Z and treatment are independent of the ones that generate W and the potential outcomes. So, $Y(a) \perp\!\!\!\perp A \mid (X, U)$ for each a.This establishes the assumption; we investigate the strength of the confounding of U below, but this does not test the conditional independence or prove finite-sample unbiasedness.

Figure B.15 compares five estimators (unadjusted; T-learners adjusting for X or (X, W, Z); and oracle T-learners adjusting for (X, U) or (X, U, W)) against the generated oracle ATE $\mathrm{mean}(Y_1 - Y_0)$, with absolute errors divided by $\mathrm{sd}(Y_{\mathrm{obs}})$ (Table B.6). Median errors are close (unadjusted 0.041; oracle-(X, U, W) 0.025). Errors above $0.1\,\mathrm{sd}(Y_{\mathrm{obs}})$ occur in 24.1% of datasets without adjustment but only 10.7–10.8% with either oracle. This reduction is consistent with residual confounding in some draws, although finite-sample estimation error also contributes to each rate. Figure B.16 groups datasets by the delivered U share of f_A and f_Y: the backdoor-X error is higher mainly when the treatment mechanism receives a larger share of U-derived inputs. The pattern is modest at this sample size; adjusting for U removes this confounding pathway, explaining why the oracle error does not rise in the same way.

Table B.6: ATE error across adjustment sets. Errors are absolute deviations from the in-sample $\mathrm{mean}(Y_1 - Y_0)$, divided by $\mathrm{sd}(Y_{\mathrm{obs}})$. All adjusted estimates use cross-fitted T-learners; the oracle rows include the latent U.

rung	adjustment set	role	median error	90th pct	share > 0.1
naive	none	unadjusted	0.041	0.195	0.241
backdoor X	X	correct if no latent confounder	0.037	0.163	0.211
backdoor XWZ	(X, W, Z)	DAG-blind: everything measured	0.030	0.121	0.148
oracle XU	(X, U)	identification: minimal sufficient set	0.028	0.104	0.108
oracle XUW	(X, U, W)	efficiency: adds W, a parent of f_Y	0.025	0.105	0.107

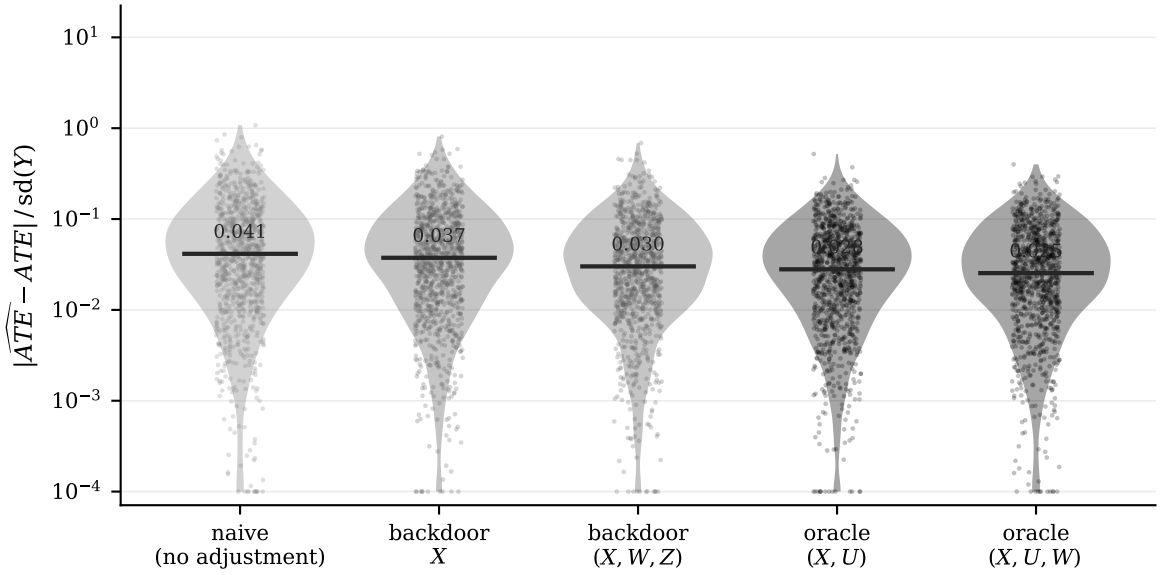


Figure B.15: Confounding-strength comparison (Table B.6) over 1024 datasets. Each violin shows normalised absolute ATE error $|\widehat{\mathrm{ATE}} - \mathrm{ATE}|/\mathrm{sd}(Y_{\mathrm{obs}})$ on a log axis, scored against $\mathrm{mean}(Y_1 - Y_0)$: *naive* (no adjustment), *backdoor* X and (X, W, Z) (T-learners on measured covariates), and *oracle* (X, U) and (X, U, W) (T-learners also given the latent confounder).

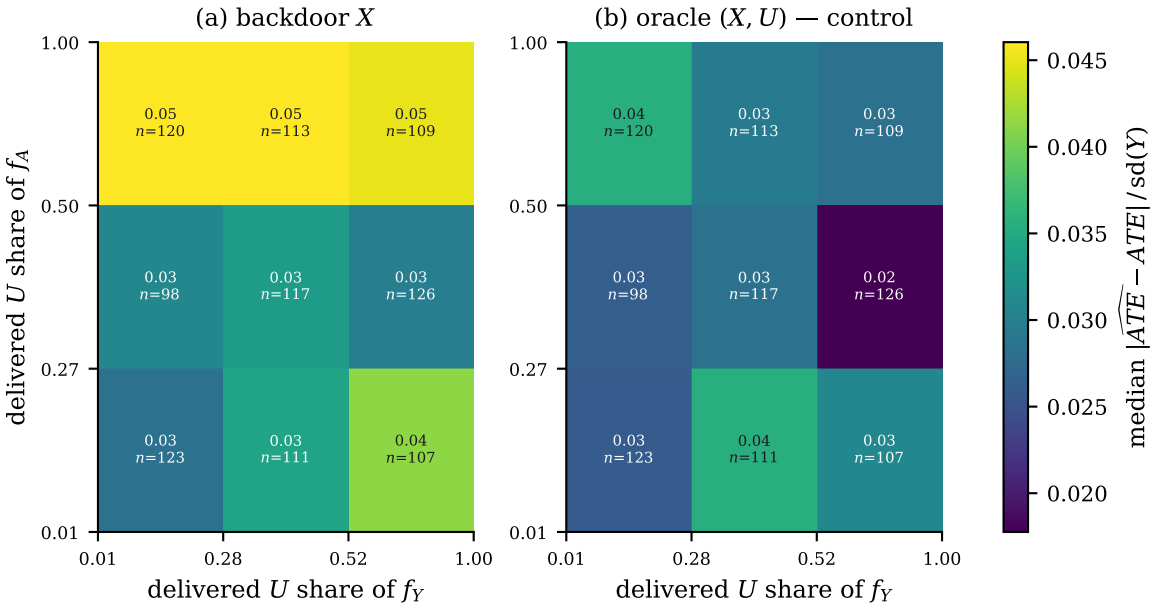


Figure B.16: Backdoor-X and oracle-(X, U) error over the delivered confounding-share grid: 1024 datasets binned 3×3 on the quantiles of the delivered U share of f_A (rows) and f_Y (columns) (Figure B.3). Cells report the median normalised error on a shared colour scale. The backdoor panel's top row (delivered U share of f_A above 0.495) is elevated at 0.045–0.046 against 0.028–0.041 in the two rows below, and is nearly flat across f_Y's share. The oracle panel shows no comparable pattern.

B.3.4 PROXY RELEVANCE

Completeness (Assumption 4) asks that W and Z carry *enough variation* in U for the bridge functions to be identified, not merely that they are wide enough. Appendices B.2.2 and B.2.4 enforce only a counting surrogate – $d_W, d_Z \geq d_U$ and its categorical analogue $\mathrm{dominates}(P, U)$ – which holds on 100% of delivered datasets by construction and is silent on whether satisfying it buys *functional* recoverability of U. We audit that directly with a cross-fitted regression: for each column u_j of U and each proxy $P \in \{W, Z\}$, the incremental $\Delta R^2_P(u_j) = R^2(u_j \mid X, P) - R^2(u_j \mid X)$, incremental over X because X is itself a parent of U (Equation 21) and a proxy that merely re-encoded it would be useless for identification however high its raw R^2. We aggregate each proxy's ΔR^2 over U's columns two ways: the mean (treating every column as equally important) and the minimum, since completeness is a claim about *all* of U and one unrecoverable direction is enough to break it.

Figure B.17 reads the same quantity three ways. Panel (a) shows both proxies are relevant on the typical draw (median $\Delta R^2 \approx 0.28$, over 93% of datasets positive) but the weakest U column is a materially harder story (median $\Delta R^2 \approx 0.05$, roughly 60% of datasets below the 0.1 weak line). Panel (b) conditions on each proxy's own dimension slack $d_P - d_U$: relevance rises with slack, validating the heuristic's spirit. Panel (c) conditions on the proxy mechanism's sampled noise share, and relevance falls as that share grows, as the dilution story predicts. Table B.7 separates the two components of slack. Mean relevance has a modest positive correlation with proxy width ($\rho = 0.166$ for both proxies). Weakest-column relevance rises with slack ($\rho \approx 0.28$) and falls with d_U ($\rho \approx -0.26$), but has almost no association with proxy width alone ($\rho = 0.034$ for W, 0.017 for Z). Because the prior samples proxy width conditional on d_U, wider proxies tend to accompany higher-dimensional U; recovering the weakest column can therefore remain difficult as proxy width increases, especially since taking a minimum over more U columns tends to lower that statistic.

Table B.7: Proxy-relevance audit (Assumption 4): cross-fitted incremental $\Delta R^2 = R^2(u_j \mid X, P) - R^2(u_j \mid X)$ of each proxy P for the latent columns, aggregated over U by the mean or by the weakest column. The counting heuristic (Appendix B.2.4) passes for every dataset, so the *share* column is the rate at which it and functional recoverability disagree. ρ is Spearman's correlation with each candidate driver: the proxy's own width d_P, d_U, the slack $d_P - d_U$ that conflates the two, and the sampled noise share of the f_P input.

aggregation over U	median ΔR^2	share < 0.1	$\rho\, d_P$	$\rho\, d_U$	$\rho\, d_P - d_U$	ρ noise
W						
mean	0.273	0.226	0.166	-0.002	0.208	-0.202
weakest	0.053	0.593	0.034	-0.249	0.277	-0.140
Z						
mean	0.271	0.235	0.166	-0.023	0.226	-0.208
weakest	0.052	0.588	0.017	-0.272	0.281	-0.095

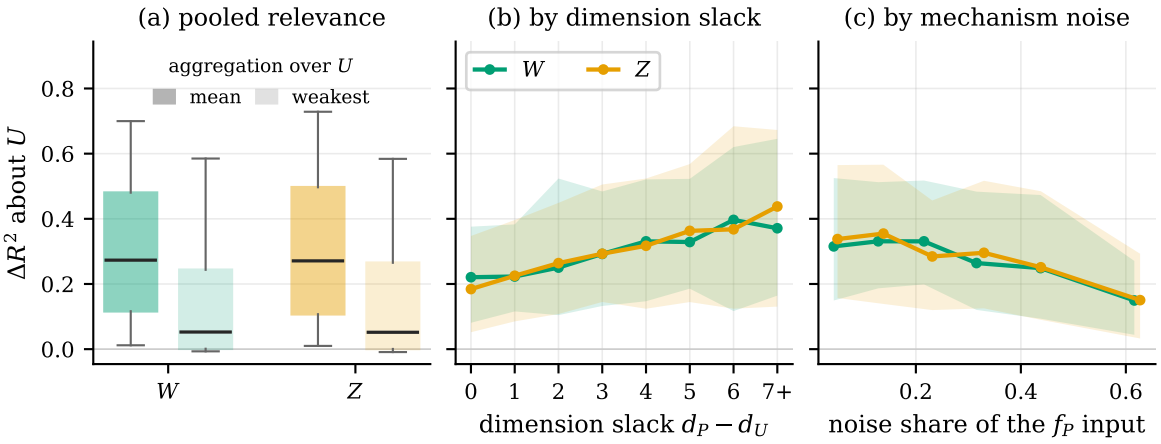


Figure B.17: Proxy relevance measured by the cross-fitted increment $\Delta R^2_P(u_j) = R^2(u_j \mid X, P) - R^2(u_j \mid X)$, shown separately for W and Z. (a) Distribution of the mean and weakest-column ΔR^2 over U; boxes show the IQR and whiskers the 10th–90th percentiles. (b) Median ΔR^2 by proxy dimension slack $d_P - d_U$, with slack ≥ 7 pooled. (c) Median ΔR^2 by sampled noise share of f_P, in six equal-frequency bins per proxy. Bands in (b) and (c) show the IQR.

C MODEL ARCHITECTURE, OBJECTIVE, TRAINING, AND VALIDATION

C.1 ARCHITECTURE

PROXIMALFM encodes a dataset in the three sequential stages of TabICLv2 (Qu et al., 2026), which we use unchanged. A *column embedder* maps each scalar cell to an embedding through a shared inducing-point set transformer applied along the unit axis, so that a cell is represented relative to the empirical distribution of its own column. A *row interactor* then treats the embedded cells of a unit as a sequence of tokens along the column axis, applies a transformer with rotary positional encoding, and concatenates the outputs of n_{cls} prepended learnable CLS tokens into a single vector per unit. Finally, an *in-context learner* runs a transformer over the sequence of unit vectors in which query units attend to context units but not to one another; no positional encoding is applied along the unit axis, so the model is equivariant to the ordering of the dataset. All blocks are pre-norm and use the query-aware scalable softmax of Qu et al. (2026). Our only removals are the classification-specific components, which a continuous causal estimand does not use; the remaining departures are described below.

A target that is never observed. Every departure below follows from one change of task. In tabular in-context learning the query is a context row with a missing label: the model predicts $y^\star$, the same variable it has just seen realized in the context, and the query differs from a context unit only in that one entry is withheld. PROXIMALFM predicts $Q(x^\star) = \mathbb{E}[Y(1) - Y(0) \mid X = x^\star]$, which is realized for no unit in the table. The observed columns are evidence about the mechanism that generates it, and supervision comes from the prior's oracle targets rather than from held-out labels.

Two context columns consequently have no query counterpart. The outcome y is context-only in TabICL as well, but there it is withheld because it is the answer; here it is withheld because a query unit is a covariate profile, not a realized unit, and the same holds for the treatment a: the estimand averages over both arms, so no value of a can sensibly be attached to the query vector $x^\star$. The proxies are a different case. A query unit can have a $w^\star$ and a $z^\star$, and the prior generates them, so withholding them is a design choice rather than a structural necessity: conditioning on them would target $\mathbb{E}[Y(1) - Y(0) \mid X, W, Z]$ instead. We follow the proximal literature in conditioning on the baseline covariates alone (Section 2.2) and marginalizing over the proxies. At inference the model therefore sees the context table in full and a query reduced to $x^\star$ alone. The modifications below adapt the backbone to exactly this interface.

Structural role embeddings. The backbone from TabICL consumes a single undifferentiated feature matrix, whereas proximal identification treats the three observed blocks asymmetrically: the roles of Z, W and X are necessary to be known in order to infer the causal estimand of interest. Nothing in the values themselves reveals that role. Column order alone cannot supply it either: the block sizes (d_X, d_Z, d_W) are resampled for every dataset, so no fixed column index marks where one block ends and the next begins, and the model would have to infer the two boundaries from the data before it could use them. We therefore tag the role explicitly. Writing $\text{lin}(\cdot)$ for the shared cell projection into $\mathbb{R}^E$, a cell v of block $g \in \{X, Z, W\}$ enters the column embedder as

$$\text{lin}(v) + r_g, \qquad r_X,\, r_Z,\, r_W \in \mathbb{R}^E \text{ learnable}, \tag{31}$$

after which the three blocks are concatenated along the column axis into a single sequence of $d_X + d_Z + d_W$ columns. Role identity is thus carried by the embedding rather than by position. The treatment A is not passed as a column at all: it enters only through the target encoder described next.

Treatment-aware target encoder. The remaining two observed variables, y and a, are not handled the way X, Z and W are. They are present for context units and absent for queries, so instead of being concatenated to the input row as columns, they are added to the tokens of context units. This also settles how a is encoded: it qualifies y rather than standing beside it, since an outcome means one thing under treatment and another under control. We therefore hold two projections and select between them with the observed treatment indicator,

$$\text{enc}(y_i, a_i) = (1 - a_i)\,\text{lin}_{y_0}(y_i) + a_i\,\text{lin}_{y_1}(y_i), \qquad \text{lin}_{y_0},\, \text{lin}_{y_1} : \mathbb{R} \to \mathbb{R}^E. \tag{32}$$

Outcome and arm are thus encoded jointly, and the treated and control response surfaces may occupy different directions in embedding space from the first layer onwards. The encoder is applied at two points, the column embedder and the in-context learner, with untied parameters.

Standardization and query-row proxy masking. Each of X, Z, W and y is standardized column-wise using the empirical mean and standard deviation of the *context* units only, so that no query information reaches the model through the scale and predictions are invariant to affine rescaling of the inputs.

A query unit is then presented without its proxies. The Z and W positions of query rows are filled with a fixed value of 0 and then masked out of the attention in the row interactor, so those tokens do not merely carry a neutral value but are invisible to the query representation altogether. The same protocol applies during training and at inference time.

Output head. For each query unit the in-context learner emits B logits, passed through a softmax to give the predictive distribution $q_\theta[\cdot \mid x^\star, \mathcal{D}_N]$ over a fixed grid of B equal-width bins spanning $[v_{\min}, v_{\max}]$, expressed in units of the context standard deviation of y. Discretizing the predictive distribution in this way is standard practice for PFNs (Hollmann et al., 2023) and is what allows a single forward pass to return a full distribution without committing to a parametric family; amortized causal models adopt it as well (Balazadeh Meresht et al., 2026). A single head is placed directly over the CATE rather than one head per arm. Although $Q(x) = \mu_1(x) - \mu_0(x)$, the two CEPOs are functionals of the same unknown SCM and are therefore generally dependent under its posterior given $\mathcal{D}_N$. Their marginal posterior distributions alone do not determine the posterior of their difference: convolving them would implicitly impose posterior independence and omit their covariance, which can misstate the CATE uncertainty. Directly learning the distribution of $Q(x)$ avoids this unsupported joint-posterior assumption.

Two properties of the discretization matter for what follows. First, we read the bin probabilities as a piecewise-uniform *density* rather than as point masses at the bin centers. This is what makes the noise convolution of (34) exact, and it puts a floor of $w/\sqrt{12}$ — the standard deviation of a single bin, of width $w = (v_{\max} - v_{\min})/B$ — on any posterior standard deviation the head can report. Second, the grid is a hard support constraint: a target outside $[v_{\min}, v_{\max}]$ cannot be represented at all, so the bounds must be wide enough to cover the CATE distribution the prior produces. Chosen $v_{\max}$, $v_{\min}$ and B are reported in Table C.1.

Hyperparameters. Table C.1 lists the configuration of the three stages. Two entries are worth reading together. The column embedder and the row interactor operate at the embedding dimension E, but the in-context learner receives the concatenation of the n_{cls} CLS tokens and therefore runs at $E \cdot n_{\text{cls}} = 512$; since it is also the deepest stage, it accounts for 28.3M of the 29.6M parameters, the column embedder and row interactor contributing 0.9M and 0.4M respectively. These parameters are kept identical to those of TabICLv2, so that its pre-trained weights can be used to initialize every component we did not modify.

Table C.1: Architectural hyperparameters of PROXIMALFM. The three stages are those of TabICLv2, kept at its values so that its pre-trained weights initialize every component we did not modify; the output head is ours. Parameter counts are per stage, the in-context learner's including the head.

Stage	Hyperparameter	Value
Shared	Embedding dimension E	128
	Feedforward expansion factor	2
	Activation	GELU
	Normalization	pre-norm
	Dropout	0.0
Column embedder	Blocks	3
	Attention heads	8
	Inducing points	128
	Scalable softmax	query-aware, elementwise
Row interactor	Blocks	3
	Attention heads	8
	CLS tokens n_{cls}	4
	RoPE base	10^5
In-context learner	Blocks	12
	Attention heads	8
	Model dimension $E \cdot n_{\mathrm{cls}}$	512
Output head	Bins B	2 000
	Grid $[v_{\min}, v_{\max}]$	$[-5, 5]$
Parameters	Column embedder	0.9 M
	Row interactor	0.4 M
	In-context learner	28.3 M
	Total	29.6 M

C.2 THE NOISE-AWARE HISTOGRAM LOSS

This section gives the closed form of the training objective of Section 3.2, together with the motivation for scoring a *noise-convolved* predictive distribution rather than fitting the Monte Carlo targets directly.

Notation and standardization. All quantities below live on the context-standardized scale. For a dataset with N context rows, let m_N and s_N denote the empirical mean and standard deviation of the observed outcomes $\{y_i\}_{i=1}^N$. Outcomes are standardized as $(y - m_N)/s_N$, so the CATE target and its Monte Carlo variance enter as

$$t_i \coloneqq \frac{\hat{Q}_K(x_i)}{s_N}, \qquad \varsigma_i^2 \coloneqq \frac{S_i^2}{s_N^2}, \tag{33}$$

the location shift m_N cancelling in the difference $\hat{\mu}_1 - \hat{\mu}_0$. Because m_N and s_N are computed on context rows only, no query information leaks into the scale.

The head emits B logits over a fixed uniform grid $e_1 < e_2 < \cdots < e_{B+1}$ spanning $[v_{\min}, v_{\max}]$ with bin width $w = (v_{\max} - v_{\min})/B$, and $q_\theta[b \mid x_i, \mathcal{D}_N]$ denotes the predicted probability of bin $b = [e_b, e_{b+1})$. We write Φ for the standard normal CDF. The quantity σ^2 denotes the variance of an auxiliary Gaussian *smoothing kernel*, expressed on this standardized scale; it is distinct from the row-specific Monte Carlo variance ς_i^2 and is annealed during training according to the schedule described in Appendix C.3.

Closed form. Whereas q_θ is the model's predictive distribution over the estimand $Q(x_i)$, the quantity actually observed during training is the noisy measurement t_i; $\tilde{q}_\theta$ is the predictive density that q_θ induces over that measurement. Write the per-row variance as $\kappa_i^2 \coloneqq \varsigma_i^2 + \sigma^2$, combining the known target noise with the annealed smoothing scale (also known). Conditioning on which bin contains

the estimand, the law of total probability gives

$$\tilde{q}_\theta\big(t \mid x_i, \mathcal{D}_N\big) = \sum_{b=1}^{B} \mathbb{P}\big(Q \in [e_b, e_{b+1}) \mid x_i, \mathcal{D}_N\big) \;\cdot\; p\big(t \mid Q \in [e_b, e_{b+1})\big)$$

$$= \sum_{b=1}^{B} q_\theta[b \mid x_i, \mathcal{D}_N]\; \frac{1}{w}\left[\Phi\left(\frac{e_{b+1}-t}{\kappa_i}\right) - \Phi\left(\frac{e_b - t}{\kappa_i}\right)\right]. \tag{34}$$

The first factor is read straight off the head. The second follows from the piecewise-uniform continuous reading of the histogram: given that Q lies in bin b, the head places it uniformly within that bin, so

$$p\big(t \mid Q \in [e_b, e_{b+1})\big) = \int_{e_b}^{e_{b+1}} \frac{1}{w}\,\mathcal{N}\big(t;\, u,\, \kappa_i^2\big)\,\mathrm{d}u = \frac{1}{w}\left[\Phi\left(\frac{e_{b+1}-t}{\kappa_i}\right) - \Phi\left(\frac{e_b - t}{\kappa_i}\right)\right], \tag{35}$$

where the integral runs over the *mean* of the Gaussian. The training objective is the negative log-likelihood of the observed target under (34), evaluated at $t = t_i$:

$$\mathcal{L}_{\text{noise_aware}} = -\frac{1}{n_q}\sum_{i=1}^{n_q} \log\left(\sum_{b=1}^{B} q_\theta[b \mid x_i, \mathcal{D}_N]\; \frac{1}{w}\left[\Phi\left(\frac{e_{b+1}-t_i}{\kappa_i}\right) - \Phi\left(\frac{e_b - t_i}{\kappa_i}\right)\right]\right). \tag{36}$$

Why convolve the model rather than fit the target. Consider the population risk of (36) at fixed context. The target is distributed as $t \sim p(\cdot \mid \mathcal{D}_N) * \mathcal{N}(0, \varsigma^2)$, the posterior over the estimand blurred by Monte Carlo error. A negative log-likelihood is minimized when the scored density matches the density of the data, i.e. when

$$q_\theta * \mathcal{N}(0, \varsigma^2 + \sigma^2) = p(\cdot \mid \mathcal{D}_N) * \mathcal{N}(0, \varsigma^2) \quad \Longleftrightarrow \quad q_\theta * \mathcal{N}(0, \sigma^2) = p(\cdot \mid \mathcal{D}_N). \tag{37}$$

The Monte Carlo variance cancels on both sides: the correction is exact for *any* value of σ, and the recovered q_θ converges to the posterior as σ is annealed. Fitting the same head directly to t_i instead—the noise-unaware objective—has population optimum $q_\theta = p(\cdot \mid \mathcal{D}_N) * \mathcal{N}(0, \varsigma^2 + \sigma^2)$, over-dispersed by exactly the target noise. The distortion is not a constant offset that could be calibrated away post hoc: $\varsigma_i^2 = \mathrm{Var}(\tau \mid X = x_i)/K$ varies across query points, and is largest precisely where the individual treatment effect is most heterogeneous, so the inflation is heaviest in the regions where posterior width is most informative.

Full objective. To align the point prediction with the estimand, a squared-error penalty on the posterior mean is added,

$$\mathcal{L}_{\text{mean}} = \frac{1}{n_q}\sum_{i=1}^{n_q}\Big(\mathbb{E}_{q_\theta}\big[Q \mid x_i, \mathcal{D}_N\big] - t_i\Big)^2, \qquad \mathbb{E}_{q_\theta}\big[Q \mid x_i, \mathcal{D}_N\big] = \sum_{b=1}^{B} q_\theta[b \mid x_i, \mathcal{D}_N]\, c_b, \tag{38}$$

with $c_b = (e_b + e_{b+1})/2$ the bin centers, giving $\mathcal{L} = \mathcal{L}_{\text{noise_aware}} + \lambda_{\text{mean}}\mathcal{L}_{\text{mean}}$. The penalty does not compete with the distributional objective: because the Monte Carlo error is zero-mean, the minimizer of (36) already carries the posterior mean and hence minimizes (38) as well, leaving the population optimum of $\mathcal{L}$ unchanged for any $\lambda_{\text{mean}} \geq 0$. At inference time the predictive distribution is reported *un-blurred*, as $q_\theta(\cdot \mid x^\star, \mathcal{D}_N)$.

C.3 Training the model

All results in this paper come from a single training run, and every number below is a property of that one run.

Configuration. Table C.2 reports the configuration in full: the prior the datasets are drawn from, the objective and its smoothing schedule, the optimizer, the pre-trained backbone the run is initialized from, and the validation sets used for checkpoint selection. Training runs for 75 000 steps at an effective batch of 64 datasets, accumulated in micro-batches of 16, for 4.8M datasets consumed in total — each drawn fresh from the prior and seen exactly once, so there is no notion of an epoch and no train/validation split on the prior itself.

Table C.2: Training configuration of PROXIMALFM: the prior the model is trained on, the objective and its smoothing schedule, the optimizer, the pre-trained backbone it is initialized from, and the validation sets used for checkpoint selection.

Stage	Hyperparameter	Value
Prior	Confounder dimension d_U	1–10
	Treatment proxy dimension d_Z	1–10
	Outcome proxy dimension d_W	1–10
	Covariate dimension d_X	0–10
	Context units N	64–2 048 (uniform)
	Query units n_q	128
	Monte Carlo worlds K	250
	MC targets on context units	no
Objective	Loss	`noise_aware`
	Mean penalty λ_{mean}	0.5
	Bins B	2 000
	Grid $[v_{\min}, v_{\max}]$	$[-5, 5]$
	Bin width w	0.005
	Initial smoothing σ_0	0.5
	σ held constant for	500 steps
	σ linear decay over	3 000 steps
	Final smoothing σ	0.01 ($2w$)
Optimization	Optimizer	AdamW
	Learning rate	1×10^{-4}
	Schedule	`cosine_warmup`
	Warmup	2 500 steps
	Weight decay	0
	Gradient clipping	1
	Training steps	75 000
	Datasets per step	64 (16×4 accumulated)
	Precision	`bfloat16` (AMP)
	Seed (numpy / torch)	1234 / 1234
Initialization	Init. checkpoint	`ckpt_tabicl/tabicl-regressor-v2-20260212.ckpt`
	Transfer rules	`ckpt_tabicl/transfer_rules.yaml`
	Weights only (fresh optimizer)	yes
Validation	Prior set	`prior_val_v1`
	Prior set: episodes / interval	500 / 200 steps
	Realistic set	`synt_val_v1`
	Realistic set: episodes / interval	480 / 100 steps
	Selection metric	`synt_dist/crps` (min)

Schedules and training curves. Figure C.1 shows the objective and the two schedules. The learning rate follows a linear warmup over 2 500 steps into a cosine decay; the smoothing scale σ of (36) is held at its initial value for 500 steps and then annealed linearly over 3 000 steps to two bin widths, its final value for the remaining of the training (panel c). The anneal is the reason the loss level is not comparable across the shaded window: changing σ changes the training objective, so movement during the anneal cannot be attributed solely to learning.

Panels (a) and (b) show that, after the initial phase, the total objective becomes dominated by the distributional term, the mean term staying a small regularizer term. Once the σ anneal is complete, both the distributional term and the total objective continue to decrease through step 75 000, but at a diminishing rate. Panel (d) zooms on the mean term of the loss. We observe that it follows the behaviour of the distributional term closely.

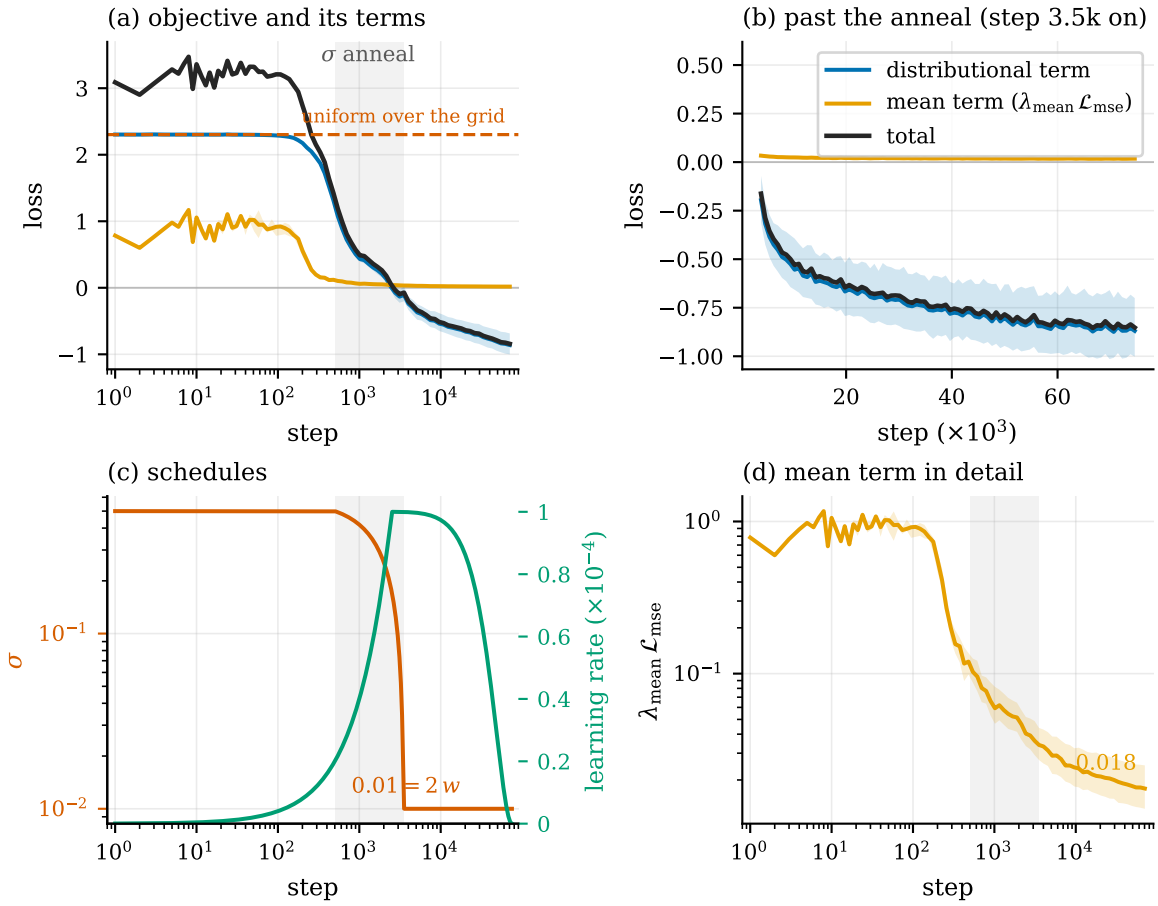


Figure C.1: Training curves for PROXIMALFM. **(a)** The objective and its two terms against log step, with the σ-annealing window shaded; the dashed line is the loss of a uniform density over the CATE grid, matching the value at initialisation. **(b)** The same curves on a linear step axis after the anneal. **(c)** The σ and learning-rate schedules. **(d)** A magnified view of the weighted mean term, $\lambda_{\text{mean}}\mathcal{L}_{\text{mse}}$, with the σ-annealing window again shaded. Stochastic curves show the median and interquartile band within log-spaced step bins in panels (a) and (d), and linearly spaced bins in panel (b); the deterministic schedules in panel (c) are shown without binning.

Table C.3: Cost of the PROXIMALFM training run, as measured GPU-hours. Validation is the two frozen evaluation datasets, scored at their logging intervals throughout training.

		Hours	%
Compute	Prior generation	43.46	65.7
	Forward / backward	18.12	27.4
	Validation, prior set (376 passes)	1.27	1.9
	Validation, realistic set (751 passes)	2.34	3.5
	Checkpointing, logging, startup	0.96	1.5
Total	GPU-hours (1× H100 NVL)	66.15	100.0
Scale	Training steps	75 000	
	Synthetic datasets consumed	4 800 000	
	Seconds per step	3.17	

Validity of the objective. Three logged diagnostics establish that the loss the curves above report was well posed throughout. First, the grid of the output head is a hard support constraint: a target outside $[v_{\min}, v_{\max}]$ cannot be represented by any forecast on the grid, so a non-negligible out-of-grid fraction would invalidate every number scored on it. Averaged over the CATE targets in the on-the-fly synthetic training prior, the share of targets falling outside the grid is 2.0×10^{-4}, i.e. 0.02%, exceeding 1% on 275 of 75 000 steps.

Second, the Monte Carlo targets carried their full replicate budget. Each query row's CATE target is an average over K replicate draws, and the realized K can be capped below the requested value by the generator's memory budget. The mean realized K was 250 — the requested value — at every logged step, so the cap never bound.

Third, sampling noise remained material at $K = 250$ Monte Carlo replicates. The ratio of the mean target-noise standard deviation ς_i to the mean predicted posterior standard deviation averaged 0.20, rising from 0.15 shortly after the anneal to 0.22 over the final third as the posterior sharpened while $\varsigma_i \propto K^{-1/2}$ remained fixed. Ignoring this noise would therefore add the heterogeneous per-row variance ς_i^2 to the learned posterior variance, motivating the noise-aware correction.

Cost. Table C.3 summarizes the computational cost of the run. Interestingly, model optimization was not the dominant expense: generating the synthetic prior consumed 43.46 GPU-hours (65.7%), approximately $2.4\times$ the cost of the forward and backward passes (27.4%). Validation accounted for a further 5.4%, while checkpointing, logging, and startup contributed only 1.5%. In total, training consumed 66.15 H100 GPU-hours over $75\,000$ steps and 4.8 million synthetic datasets, averaging 3.17 seconds per step.

C.3.1 POST-TRAINING ON LARGER CONTEXT SIZES

The main training run represents contexts of at most $2\,048$ observations. We therefore perform a post-training experiment to test whether the limited support at larger context sizes contributes to the plateau observed in Section 4.2. Starting from the completed main-training checkpoint, we optimize for a total of $30\,000$ steps on the same causal prior and noise-aware objective, but draw context sizes uniformly from 64 to $5\,000$ observations (with 128 query rows). Uniform, rather than log-uniform, sampling ensures that the adaptation gives substantial weight to the newly introduced large contexts while retaining smaller contexts for rehearsal. We retain the effective meta-batch size of 64 datasets, use a lower learning rate of 2×10^{-5} with a 500-step warmup, and enable activation recomputation to fit the longer sequences.

To make this adaptation inexpensive and limit changes to the learned representation, we load the complete main-training checkpoint and freeze all parameters except the in-context predictor. Within that predictor, its first six transformer blocks are frozen as well, so only its later layers and unfrozen predictor parameters are optimized. We evaluate the resulting checkpoints on separate frozen prior and realistic validation suites constructed at large context sizes.

C.4 Validation and posterior diagnostics

C.4.1 Validation datasets

We monitor the training of ProximalFM on a fixed suite of 480 datasets, balanced across 12 hand-written realistic proximal mechanisms. These mechanisms cover clinical severity and frailty, latent patient subtypes, threshold-based treatment assignment, utilization counts, censored survival, marketing uplift, educational selection, wage returns, hierarchical site effects, nonlinear ordinal proxies, interaction-driven confounding, and near-null effects under strong confounding. These settings span a diverse range of datasets: continuous, binary, ordinal and categorical variables, the treatment is binary, the outcomes may be continuous, skewed, count-valued, zero-inflated, or censored.

For each dataset, the dimensions of U, X, W and Z are sampled with the same procedure as in the prior (Appendix B.2.2). Each episode contains exactly 2048 context rows and 128 query points. The query target CATEs are computed by averaging $10\,000$ Monte Carlo replicates of ITE draws. At this replicate count, the Monte Carlo sampling noise is approximately 0.025 times the predictive posterior standard deviation (of the last model checkpoint). The same fixed validation set is evaluated at every training checkpoint.

C.4.2 Point-estimate validation

We measure the point-estimate performance at both ATE and CATE levels. The ATE prediction is estimated as the average of the CATE predictions from the query set,

$$\widehat{\mathrm{ATE}} = \frac{1}{n_q}\sum_{i=1}^{n_q}\hat{\tau}_i,$$

and so is the target ATE also computed from the references CATEs. To evaluate the ATE estimation, we use the root mean squared error (RMSE) and the bias of the ATE normalized by the standard deviation of the episode's CATEs.

Regarding CATE evaluation, we use normalized precision in estimation of heterogeneous effect (nPEHE),

$$\mathrm{nPEHE} = \sqrt{\frac{\frac{1}{n_q}\sum_{i=1}^{n_q}(\hat{\tau}_i - \tau_i)^2}{\mathrm{Var}_i(\tau_i)}}. \tag{39}$$

We also measure the shape-R^2, which isolates the recovery of treatment-effect heterogeneity from constant level error,

$$R^2_{\mathrm{shape}} = 1 - \frac{\mathrm{Var}_i(\hat{\tau}_i - \tau_i)}{\mathrm{Var}_i(\tau_i)}. \tag{40}$$

A constant-ATE prediction gives $\mathrm{nPEHE} = 1$ and $R^2_{\mathrm{shape}} = 0$.

Figure C.2 shows the evolution of point-estimate performance on the fixed realistic-mechanism validation suite. In the initial stage of training, the model performs worse than the constant-ATE baseline (CATE nPEHE above 1 and the shape R^2 is close to or below zero). After approximately $3\,500$ training steps, the median nPEHE falls below the constant-ATE baseline and the shape R^2 becomes positive. Performance then continues to improve throughout training: ATE RMSE decreases, normalized ATE bias moves towards zero, nPEHE falls further, and the shape R^2 increases. The narrowing interquartile bands indicate that performance also becomes more consistent across validation episodes.

C.4.3 Distributional validation

We evaluate the predictive posterior using the continuous ranked probability score (CRPS), the predictive standard deviation relative to the prediction error, and empirical coverage of central credible intervals. For query row i, let q_i denote the predicted distribution over $Q(x_i)$, with CDF F_i, mean μ_i, and variance v_i^2. Let $Q_{K,i}$ denote the Monte Carlo CATE target and let ς_i^2 denote its estimated sampling variance.

The CRPS evaluates the full predictive distribution against the target:

$$\mathrm{CRPS}(q_i, Q_{K,i}) = \int \left[F_i(t) - \mathbf{1}\{t \geq Q_{K,i}\}\right]^2 \,\mathrm{d}t. \tag{41}$$

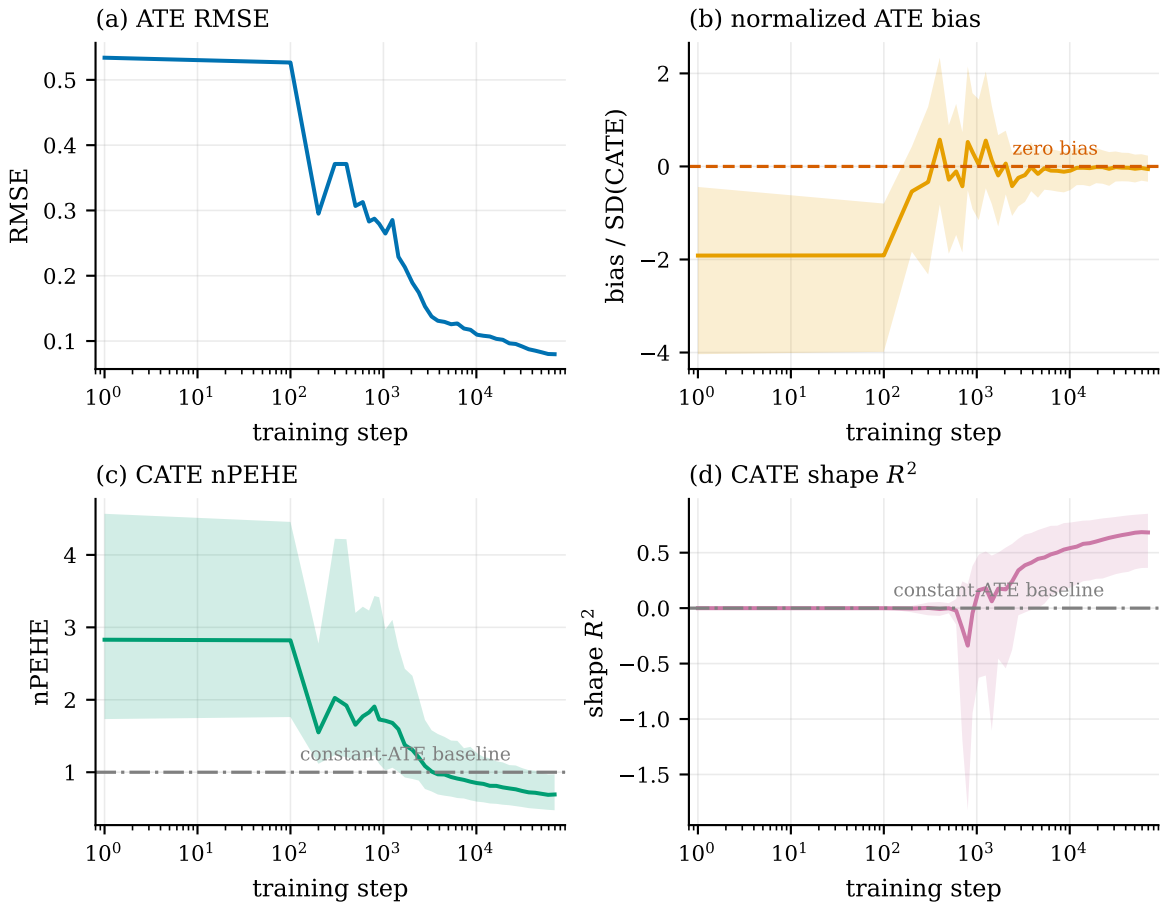


Figure C.2: Evolution of point-estimate performance during training on the fixed realistic-mechanism validation suite (Appendix C.4.1). Curves show the validation metric at each evaluation step. The shaded regions in panels (b)–(d) indicate the interquartile range across validation episodes. **(a)** ATE RMSE, a pooled metric and therefore shown without an episode-level band. **(b)** normalized ATE bias, where zero denotes no bias. **(c)** CATE nPEHE, with the dashed line marking the constant-ATE baseline ($\mathrm{nPEHE} = 1$). **(d)** CATE shape R^2, with the dashed line marking the corresponding constant-ATE baseline ($R^2_{\mathrm{shape}} = 0$). Lower values are preferred in panels (a) and (c), while panel (b) is centered at zero and higher values are preferred in panel (d). For readability, curves and shaded bands are lightly smoothed by taking medians within log-spaced training-step bins.

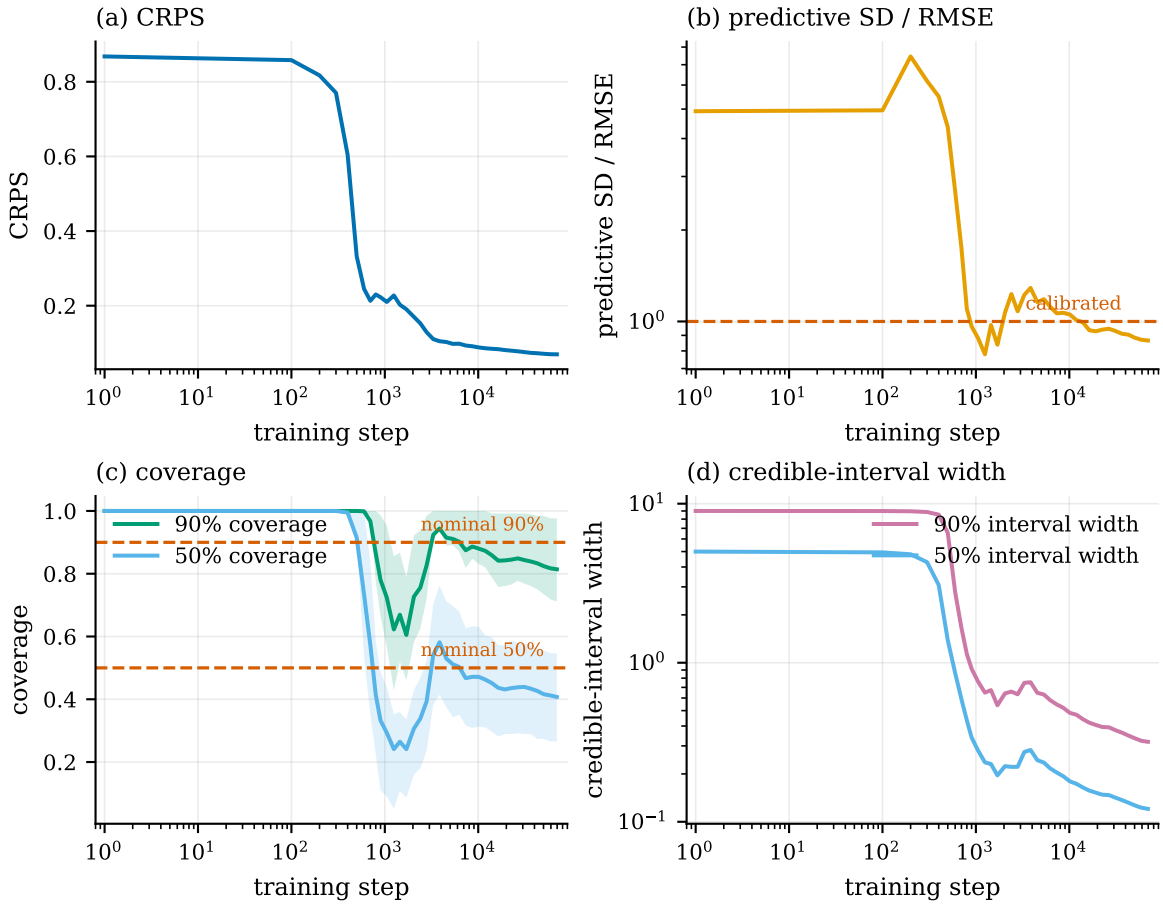


Figure C.3: Evolution of distributional predictive performance during training on the fixed realistic-mechanism validation suite (Appendix C.4.1). **(a)** CRPS, for which lower values indicate better probabilistic predictions. **(b)** predictive standard deviation divided by RMSE, with the dashed line at one denoting calibrated predictive spread. **(c)** empirical 90% and 50% coverages, with the dashed lines marking the nominal coverage levels and the shaded regions showing the interquartile ranges across validation episodes. **(d)** widths of the central 90% and 50% credible intervals, pooled over validation rows. Panels (b) and (d) use logarithmic y-axes. For readability, curves are lightly smoothed by taking medians within log-spaced training-step bins.

We also report the ratio of predictive standard deviation to prediction error,

$$\mathrm{SD/RMSE} = \sqrt{\frac{\frac{1}{M}\sum_{i=1}^{n_q} v_i^2}{\frac{1}{n_q}\sum_{i=1}^{n_q}\left(\mu_i - Q_{K,i}\right)^2 - \frac{1}{M}\sum_{i=1}^{n_q}\varsigma_i^2}}, \tag{42}$$

where n_q is the total number of validation query rows. The subtraction of ς_i^2 removes the Monte Carlo noise from the observed prediction error. This denominator equals the average predictive variance under calibration, in expectation. Values below one indicate overconfident predictions, whereas values above one indicate underconfidence.

Figure C.3 displays the evolution of these distributional metrics during training. CRPS decreases as the model learns. Both the ratio SD/RMSE and the empirical coverages converge to overconfident states.

C.4.4 Posterior representation & smoothing

The histogram representation of the predictive posterior can produce visually spiky densities. Small changes in the allocation of probability across neighbouring bins may have little effect on the training

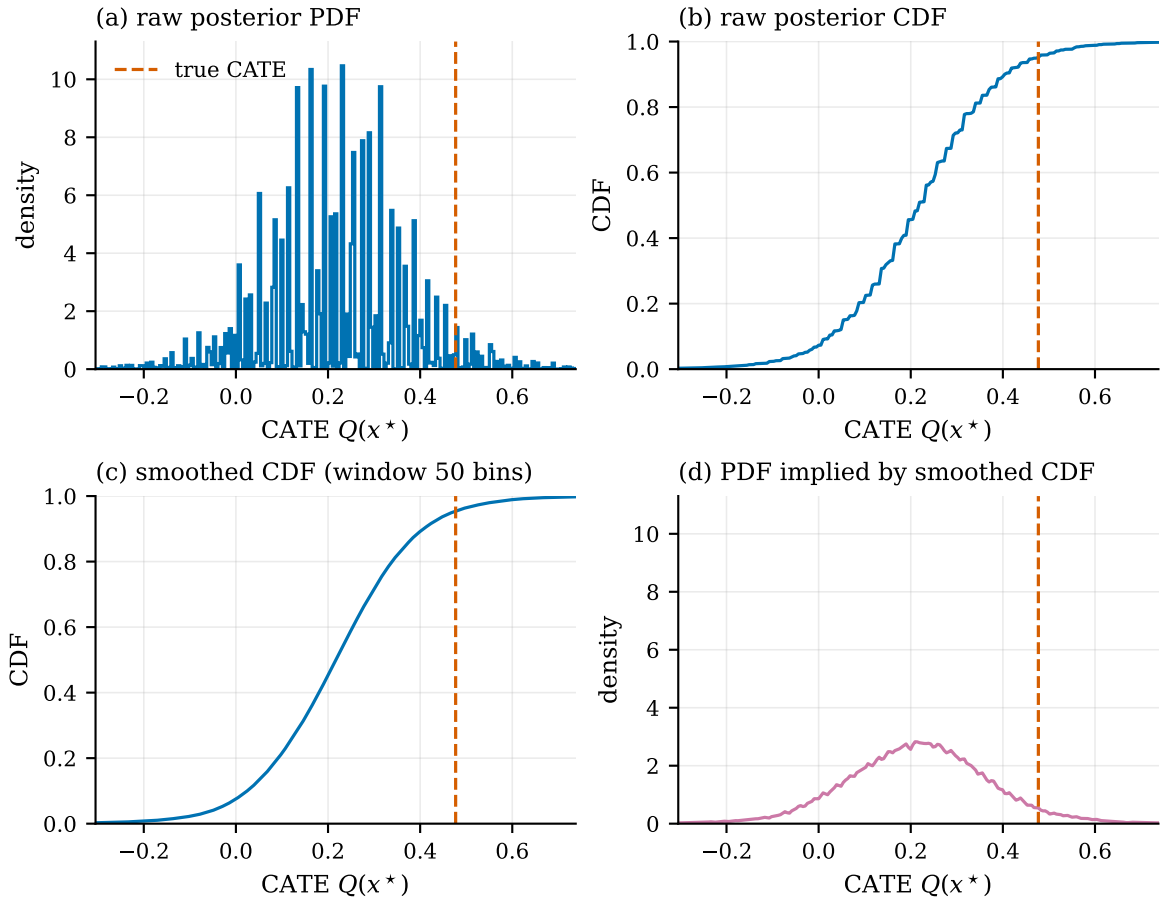


Figure C.4: Illustration of the spiky posterior-density representation for one selected query point from the validation suite. **(a)** Raw posterior density induced by the model's histogram output. **(b)** Corresponding raw CDF. **(c)** CDF after Savitzky–Golay smoothing with a 50-point window and polynomial degree three. **(d)** Density obtained by differentiating the smoothed CDF. The smoothed CDF closely follows the raw CDF while yielding a more visually interpretable density. The dashed line marks the true CATE. Smoothing is used for visualization only; all validation metrics are computed from the original posterior representation.

objective while producing noticeably different density shapes. We therefore use the CDF as an intermediate representation for visualization. A Savitzky–Golay filter is applied to the raw CDF, and the corresponding density is obtained by numerical differentiation. As illustrated in Figure C.4, the smoothed CDF stays very close to the original CDF, but its implied density is easier to interpret. This postprocessing is applied only for visual inspection and does not alter the predictive distribution used to compute validation metrics. Future work may investigate regularization techniques during training to encourage smoother posterior estimates.

C.4.5 Posterior dynamics

Figure C.5 shows that posterior width is initially independent of context size: at the first checkpoints, all context sizes produce similarly wide predictions. Over the first thousands of training steps, the width stays independent of the context size, but becomes narrower. Later in training, context-dependent contraction becomes evident, with larger context sizes producing narrower posteriors while smaller contexts remain wider. At the final checkpoint, the mean predictive standard deviation decreases from approximately 0.54 at $n_{\text{ctx}} = 16$ to 0.10 at $n_{\text{ctx}} = 2\,048$.

Figure C.6 illustrates the evolution of the predictive posterior for a fixed query point. Early in training, the posterior is diffuse and weakly dependent on context size. As training progresses, it

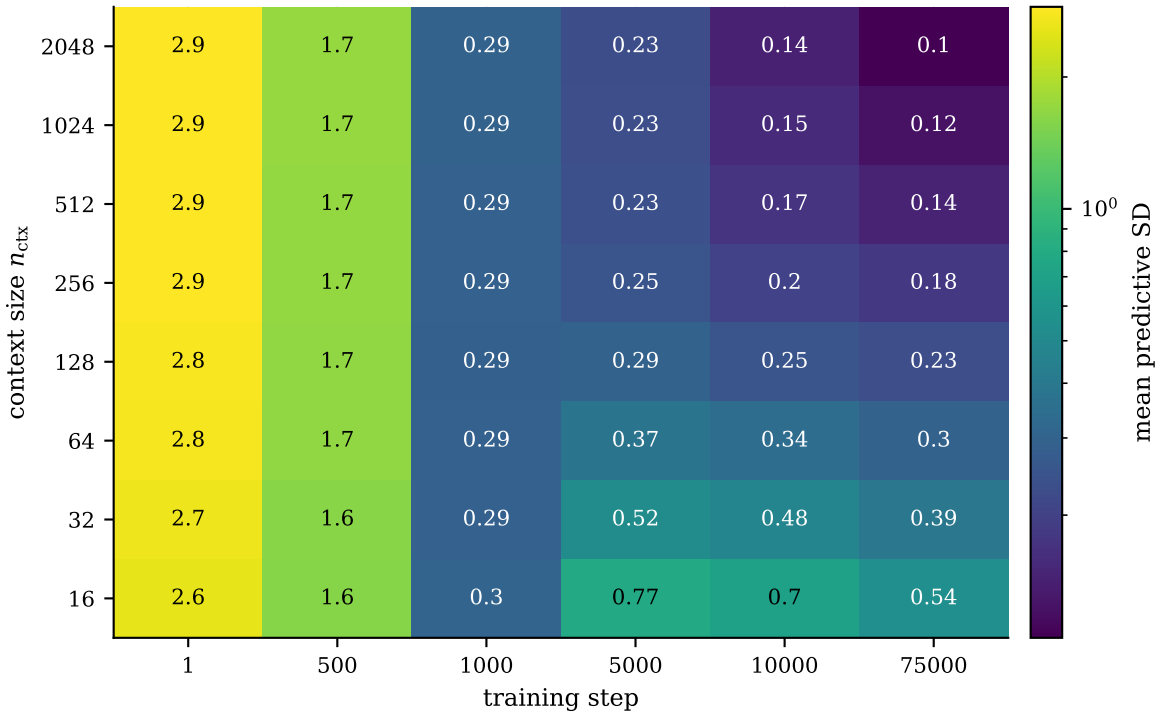


Figure C.5: Posterior-width contraction during training on the fixed realistic-mechanism validation suite (Appendix C.4.1). Each cell reports the mean predictive posterior standard deviation averaged over all validation query rows, for a given training checkpoint and context size. Widths are expressed in standardized units and use the piecewise-uniform posterior variance. The shared colour scale is logarithmic.

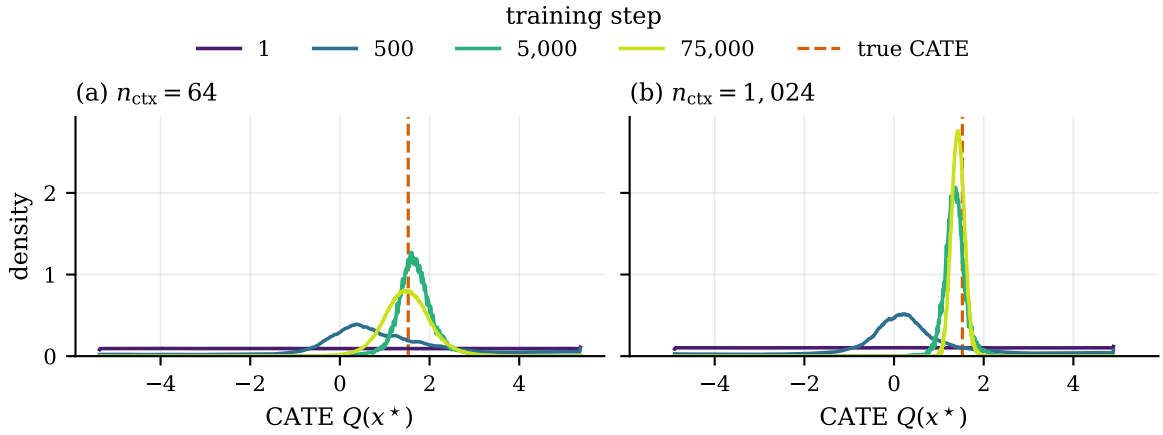


Figure C.6: Evolution of the predictive CATE posterior for one fixed query point from the realistic-mechanism validation suite (Appendix C.4.1). Curves show the density obtained from the posterior CDF at four training checkpoints, with training steps indicated by colour. The same query point is evaluated with **(a)** $n_{\text{ctx}} = 64$ and **(b)** $n_{\text{ctx}} = 1\,024$. The dashed line marks the true CATE. Densities are shown in standardized units. The CDFs are smoothed for visualization, following Appendix C.4.4.

concentrates around the true CATE, with the larger context producing a noticeably sharper final posterior.

D Details on results

D.1 Baseline methods

The baseline methods are either newly implemented or adjusted from existing codebases. All relevant code and configurations are available in our codebase.

Table D.1: Catalogue of CATE predictors selected in the main benchmark.

Predictor	Family	Fitted estimator and prediction rule	Details
S-learner (ridge)	Backdoor meta-learner	S-learner with ridge regression.	Appendix D.1.1
S-learner (RF)	Backdoor meta-learner	S-learner with random forest.	Appendix D.1.1
S-learner (TabICL)	Backdoor meta-learner	S-learner with zero-shot TabICL.	Appendix D.1.1
S-learner (TabICL) + kNN marg.	Backdoor meta-learner	S-TabICL fit on (X, W, Z), then 16-neighbour proxy marginalisation.	Appendix D.1.1
S-learner (TabICL) + kernel marg.	Backdoor meta-learner	S-TabICL fit on (X, W, Z), then 16-neighbour kernel proxy marginalisation (scale 1).	Appendix D.1.1
T-learner (ridge)	Backdoor meta-learner	T-learner with ridge regression.	Appendix D.1.1
T-learner (RF)	Backdoor meta-learner	T-learner with random forest.	Appendix D.1.1
T-learner (TabICL)	Backdoor meta-learner	T-learner with zero-shot TabICL.	Appendix D.1.1
T-learner (ridge) + kNN marg.	Backdoor meta-learner	T-OLS fit on (X, W, Z), then 16-neighbour proxy marginalisation.	Appendix D.1.1
T-learner (ridge) + kernel marg.	Backdoor meta-learner	T-OLS fit on (X, W, Z), then 16-neighbour kernel proxy marginalisation (scale 1).	Appendix D.1.1
X-learner (RF)	Backdoor meta-learner	X-learner with random forest.	Appendix D.1.1
X-learner (TabICL)	Backdoor meta-learner	X-learner with zero-shot TabICL.	Appendix D.1.1
P-learner (ridge)	Proximal	Kernel-bridge P-learner with ridge final-stage CATE regression.	Appendix D.1.2
P-learner (RF)	Proximal	Kernel-bridge P-learner with random-forest final-stage CATE regression.	Appendix D.1.2
P-learner (TabICL)	Proximal	Kernel-bridge P-learner with TabICL final-stage CATE regression.	Appendix D.1.2
KPV (ind. W)	Proximal	KPV bridge with W integrated over the context empirical marginal.	Appendix D.1.5
KPV (cond. W)	Proximal	KPV bridge with locally conditional $W \mid X = x$ integration.	Appendix D.1.5
PMMR (ind. W)	Proximal	PMMR bridge with W integrated over the context empirical marginal.	Appendix D.1.4
PMMR (cond. W)	Proximal	PMMR bridge with locally conditional $W \mid X = x$ integration.	Appendix D.1.4

Continued on next page

Table D.1: Catalogue of CATE predictors selected in the main benchmark (continued).

Predictor	Family	Fitted estimator and prediction rule	Details
NMMR-V (fixed; ind. W)	Proximal	V-statistic NMMR, fixed RBF width, with marginal W integration.	Appendix D.1.3
NMMR-V (fixed; cond. W)	Proximal	V-statistic NMMR, fixed RBF width, with conditional $W \mid X = x$ integration.	Appendix D.1.3
NMMR-U (fixed; ind. W)	Proximal	U-statistic NMMR, fixed RBF width, with marginal W integration.	Appendix D.1.3
NMMR-U (fixed; cond. W)	Proximal	U-statistic NMMR, fixed RBF width, with conditional $W \mid X = x$ integration.	Appendix D.1.3
NMMR-V (median; ind. W)	Proximal	V-statistic NMMR, median-heuristic RBF width, with marginal W integration.	Appendix D.1.3
NMMR-V (median; cond. W)	Proximal	V-statistic NMMR, median-heuristic RBF width, with conditional $W \mid X = x$ integration.	Appendix D.1.3
NMMR-U (median; ind. W)	Proximal	U-statistic NMMR, median-heuristic RBF width, with marginal W integration.	Appendix D.1.3
NMMR-U (median; cond. W)	Proximal	U-statistic NMMR, median-heuristic RBF width, with conditional $W \mid X = x$ integration.	Appendix D.1.3
CausalFM	Causal ICL	Pre-trained causal foundation model, X only.	Appendix D.1.1
Do-PFN	Causal ICL	Pre-trained causal foundation model, X only.	Appendix D.1.1
CausalPFN	Causal ICL	Pre-trained causal foundation model, X only.	Appendix D.1.1
CausalPFN + kNN marg.	Causal ICL	CausalPFN fit on (X, W, Z), then 16-neighbour proxy marginalisation.	Appendix D.1.1
CausalPFN + kernel marg.	Causal ICL	CausalPFN fit on (X, W, Z), then 16-neighbour kernel proxy marginalisation (scale 1).	Appendix D.1.1
PROXIMALFM	Proximal foundation model	Amortized proximal transformer; query proxies are masked.	Section 3

D.1.1 BACKDOOR META-LEARNERS, IN-CONTEXT LEARNERS AND PROXY MARGINALIZATION

We evaluate some methods estimating the CATE under the backdoor adjustment assumptions. First, we evaluat standard causal meta-learners from the backdoor adjustment framework (Künzel et al., 2019), specifically the S-learner, T-learner, and X-learner. We also implement some in-context learners: CausalPFN (Balazadeh Meresht et al., 2026), CausalFM (Ma et al., 2026) and Do-PFN (Robertson et al., 2026). In our proximal causal setting, the presence of the unobserved confounder U directly violates the conditional ignorability assumption these methods rely on ($Y(a) \perp\!\!\!\perp A \mid X$). Consequently, all backdoor methods are misspecified. They serve as empirical baselines.

Because the causal estimand of interest is the covariate-conditional effect $Q(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]$, yet observational data contain proxy variables (W, Z) that carry information about U, we evaluate backdoor methods under two distinct options:

1. **Discard proxies (X-only):** The learners are trained exclusively on baseline covariates X, discarding (W, Z) entirely. This reflects the standard practice of ignoring unobserved confounding and assuming $Y(a) \perp\!\!\!\perp A \mid X$.
2. **Proxy conditioning with marginalization ($(X, W, Z) \to X$):** To avoid disadvantaging backdoor methods by withholding proxy features, we also fit learners on the full feature set (X, W, Z). Because this yields a proxy-conditional effect $\hat{\tau}(x, w, z)$ rather than $Q(x)$, we post-process the predictions by integrating out (W, Z) with respect to a local empirical estimate of $P(W, Z \mid X = x)$.

Note that even with proxy features and marginalization, Option 2 does not identify $Q(x)$: naive regression is insufficient under the proximal assumptions (Section 2).

Meta-learner formulations. We evaluate three meta-learning architectures (Künzel et al., 2019):

- **S-Learner:** Fits a single regressor $\hat{\mu}$ on the pooled data predicting Y from features and treatment A. The effect is estimated as $\hat{\tau}_{\mathrm{S}}(x) = \hat{\mu}(x, 1) - \hat{\mu}(x, 0)$. When proxies are included, it estimates $\hat{\tau}_{\mathrm{S}}(x, w, z) = \hat{\mu}(x, 1, w, z) - \hat{\mu}(x, 0, w, z)$.
- **T-Learner:** Fits separate regressors $\hat{\mu}_0$ on the control cohort ($A = 0$) and $\hat{\mu}_1$ on the treated cohort ($A = 1$), predicting $\hat{\tau}_{\mathrm{T}} = \hat{\mu}_1 - \hat{\mu}_0$. This allows separate response surfaces for each arm, at the cost of not sharing information across arms.
- **X-Learner:** A four-stage crossover architecture designed for imbalanced treatments. After fitting base models $\hat{\mu}_0, \hat{\mu}_1$, it imputes counterfactual pseudo-effects $D_1 = Y_1 - \hat{\mu}_0(X_1)$ and $D_0 = \hat{\mu}_1(X_0) - Y_0$, trains second-stage models $\hat{\tau}_1$ and $\hat{\tau}_0$ to predict them, and combines them using an estimated propensity score $\hat{e}$:
$$\hat{\tau}_{\mathrm{X}} = \hat{e}\,\hat{\tau}_0 + (1 - \hat{e})\,\hat{\tau}_1. \tag{43}$$
In our implementation, $\hat{e}$ is estimated via logistic regression.

Each meta-learner accepts three distinct base model classes:

- **Ridge regression:** Standardized linear model with penalty $\alpha \in \{0.1, 1, 10, 100, 1000\}$ selected via 4-fold cross-validation.
- **Random forest:** 100 trees with maximum depth $\in \{1, 3, 5\}$ and minimum leaf size $\in \{1, 5, 10\}$ chosen via 4-fold cross-validation.
- **TabICLv2 (Qu et al., 2026):** In-context tabular foundation model evaluated zero-shot.

In-context learners. We evaluate CausalPFN, CausalFM and Do-PFN in their pretrained, zero-shot form: each model uses the episode's observed data as context, with no episode-specific parameter updates. CausalPFN and CausalFM directly return a CATE prediction for each query. Do-PFN instead predicts outcomes under each treatment arm, and we take their difference to form the CATE.

Proxy marginalization. When trained on (X, W, Z), a meta-learner predicts a proxy-conditional contrast $\hat{\tau}(x, w, z)$. Because the evaluation target is the covariate-conditional effect $Q(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]$, we recover a function of x alone by integrating out the proxies against a local empirical estimate of $P(W, Z \mid X = x)$:
$$\hat{\tau}(x) = \sum_{i \in \mathcal{N}_k(x)} q_i(x)\,\hat{\tau}(x, W_i, Z_i). \tag{44}$$

Crucially, the integration uses the aligned context pairs (W_i, Z_i) rather than a Cartesian product of marginals, preserving the empirical joint dependence between the two proxies. The neighborhood $\mathcal{N}_k(x)$ comprises the k context points closest to x under a feature-standardized Euclidean distance:
$$d(x, X_i) = \sqrt{\sum_{d=1}^{d_X} \frac{(x_d - X_{i,d})^2}{\sigma_d^2}}, \qquad \sigma_d = \sqrt{2}\,\mathrm{median}_{j,l}\big(|X_{j,d} - X_{l,d}|\big). \tag{45}$$

We evaluate two local weighting schemes for $q_i(x)$:

- k**-NN weighting:** Uniform weights over the k nearest context atoms, $q_i(x) = 1/k$.
- **Gaussian kernel weighting:** Nadaraya-Watson exponential weights with bandwidth scale $s > 0$:

$$q_i(x) = \frac{\exp\big(-\big(d(x, X_i)^2 - \min_{j\in\mathcal{N}_k(x)} d(x, X_j)^2\big)/s\big)}{\sum_{l\in\mathcal{N}_k(x)} \exp\big(-\big(d(x, X_l)^2 - \min_{j\in\mathcal{N}_k(x)} d(x, X_j)^2\big)/s\big)}. \tag{46}$$

In our benchmarks, we use $k = 16$ and fixed kernel scale $s = 1.0$.

D.1.2 P-LEARNER

We implement the P-learner of Sverdrup & Cui (2023). The cross-fitted score and final-stage regression follow their procedure. For bridge estimation, we use a Nyström-approximation of the minimax RKHS formulation, described below.

For each data point, we compute the doubly robust proximal score:

$$\Gamma_i = (-1)^{1-A_i}\, q(Z_i, A_i, X_i)\big(Y_i - h(W_i, A_i, X_i)\big) + h(W_i, 1, X_i) - h(W_i, 0, X_i), \tag{47}$$

The outcome and treatment bridges are characterized, for $a \in \{0, 1\}$, by the conditional moments

$$\begin{aligned} \mathbb{E}[\mathbf{1}\{A = a\}\{Y - h(W, a, X)\} \mid Z, X] = 0, \\ \mathbb{E}[\mathbf{1}\{A = a\}q(Z, a, X) - 1 \mid W, X] = 0. \end{aligned} \tag{48}$$

We obtain the CATE estimate by regressing the cross-fitted scores on X, $\hat{Q}(x) = \hat{\mathbb{E}}[\hat{\Gamma} \mid X = x]$. The mean of the same scores gives the corresponding proximal ATE estimate.

Kernel bridge estimation. We estimate both bridges using the regularised minimax RKHS formulation of Dikkala et al. (2020), following Sverdrup & Cui (2023). The outcome bridge h is represented by RBF features Φ_H of (W, X), and the treatment bridge q by Φ_Q of (Z, X). Because of computational and memory constraints, we use the Nyström method with r landmark points sampled from the training fold. The kernel bridge estimators have closed-form solutions with penalties λ_h and λ_q on the two bridges.

Hyperparameter tuning. Several components of the method is subject to a choice of parameter. First of all, the CATE predictor $\hat{Q}(.)$ can be made using any general predictor. We use a ridge regularized linear regression, a random forest or TabICL (Qu et al., 2026). The parameters of the random forest and ridge regressor are chosen by cross-validation, TabICL does not require parameter tuning.

Then, regarding the bridge functions h and q, the regularizer λ_h and λ_q are chosen by cross-validation to minimize their moment loss. The Nyström rank r is fixed. The kernel bandwidth is chosen by the median heuristic, and not searched. Table D.2 summarizes the hyperparameters.

Score stabilisation. The estimated treatment-bridge weight can have large magnitude: we cap it at $q_{\max}$ before forming (47).

D.1.3 NEURAL MOMENT MATCHING REGRESSION

We adapted the implementation of the neural moment matching regression (NMMR) method of Kompa et al. (2022)[3] to our setting. Their code estimates one population-level quantity per dataset, $\mathbb{E}[Y \mid \mathrm{do}(A = a)]$, evaluated on a grid of treatment values, which for a binary treatment is simply the two arms whose difference is the ATE. Our benchmark instead asks for the CATE $\hat{Q}(x)$ at each query row of every episode.

[3]Original implementation available at `https://github.com/beamlab-hsph/Neural-Moment-Matching-Regression`.

Table D.2: P-learner settings. Bridge regularization parameters are selected separately within each cross-fitting fold using held-out moment loss. Final-stage hyperparameters are selected per episode using four-fold cross-validated prediction MSE.

Component	Setting	Value
Cross-fitting	Folds (stratified on A)	4
Bridge class	Kernel	RBF, blockwise median heuristic
	Nyström rank r	256
Bridge regularization	λ_h, λ_q	*sweep*: $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
	Selection criterion	Held-out proximal moment loss
	Validation fraction	0.25
	λ_g (critic loss)	10^{-3}, fixed
Proximal score	Weight clipping $q_{\max}$	20
Final-stage CV	Folds	4
	Selection criterion	Mean squared error
Ridge final stage	Model	Standardized ridge regression
	Ridge penalty α	*searched*: $\{0.1, 1, 10, 100, 1000\}$
Random-forest final stage	Number of trees	100, fixed
	Maximum depth	*searched*: $\{1, 3, 5\}$
	Minimum leaf size	*searched*: $\{1, 5, 10\}$
	Maximum feature fraction	1, fixed
TabICL final stage	Model	Zero-shot prediction, no tuning

The method. NMMR replaces the outcome bridge h_0 of (2) by a neural network $g_\phi(x, w, a) \in \mathbb{R}$ and fits it directly, with no two-stage regression. This neural network is trained on an observational dataset, with simple gradient descent. It uses the following loss function, determined by an RBF kernel matrix on the conditioning variables A, Z and X:

$$r_i = Y_i - g_\phi(X_i, W_i, A_i), \qquad K_{ij} = \exp\Big(-\tfrac{\|\kappa_i - \kappa_j\|^2}{2\ell^2}\Big), \quad \kappa_i = \big(A_i,\ Z_i/\hat{\sigma}_Z,\ X_i/\hat{\sigma}_X\big), \tag{49}$$

which determines the quadratic forms to minimize

$$\hat{R}_V = \frac{r^\top K\, r}{n^2} \qquad \text{or} \qquad \hat{R}_U = \frac{r^\top \big(K - \mathrm{diag}(K)\big)\, r}{n(n-1)}, \tag{50}$$

the V- and U-statistics. The U-statistic drops the diagonal: it is unbiased but noisier, while the V-statistic is biased but lower variance. We report both as separate entries, as in the original paper.

Deviations at prediction time. The reference code never produces a conditional ATE, even though the bridge it fits supports one. It returns a single scalar per arm, $\mathbb{E}_{W,X}[g_\phi(a, W, X)]$, whose difference is an ATE. Recovering $Q(x)$ of (3) instead means marginalising W *at each query point*; we consider two possibilities.

1. **We assume** $W \perp\!\!\!\perp X$ ("ind. W"). The direct transposition of the reference code: average the fitted bridge over the empirical marginal of W in the context, $\hat{Q}(x) = \frac{1}{n}\sum_i g_\phi(x, W_i, 1) - g_\phi(x, W_i, 0)$. This is a valid strategy only if one assumes that $W \perp\!\!\!\perp X$.
2. **We do not assume** $W \perp\!\!\!\perp X$ ("cond. W"). We approximate $\mathbb{E}[g_\phi(x, W, a) \mid X = x]$ by averaging over the marginal of W implied by the context, reweighted by a Gaussian kernel on X. This computes the proximal g-formula, without independence assumption, but at a cost of an extra smoothing bandwidth.

Implementation corrections & computational notes. We do the following changes in the code compared to original code. First, in the MLP used to model the bridge function, the original implementation applied a ReLU transformation on the last layer. This made the prediction of negative outcomes impossible, and was removed in our implementation. Second, for computational optimization, we avoid recomputing the kernel matrix at each iteration mini-batch, but compute the complete matrix once and sub-sample from it.

Table D.3: NMMR settings. Values marked *searched* are selected per episode by the procedure described below; the remainder are fixed across all episodes. The U- and V-statistic variants are reported as separate methods, so the loss is fixed within a run rather than searched.

Component	Setting	Value
Bridge class	MLP, ReLU, linear output	
	Hidden width	*searched*: $\{16, 32, 64\}$
	Depth (hidden layers)	*searched*: $\{2, 4, 6\}$
Moment kernel	RBF over $(A, Z/\hat{\sigma}_Z, X/\hat{\sigma}_X)$	
	Length scale ℓ	fixed at 1.0 or median heuristic
Loss	U- or V-statistic (50)	one method each
Optimisation	Adam, learning rate	*searched*: $\{10^{-3}, 5 \cdot 10^{-3}\}$
	Weight decay (L_2)	10^{-4}
	Epochs	*sweep*: $\{25, 50, 75, 100\}$
	Minibatch size	256
Prediction	W strategy	ind. W / cond. W
	X smoothing length scale (`Wcond`)	median heuristic
	Neighbours k (`Wcond`)	512
Tuning	Validation fraction	0.2
	Search budget	10 of 18 configurations

Kernel bandwidth. The original implementation uses a fixed bandwidth of $l = 1.0$ in the Equation (49). This has obvious drawbacks as the average distance between data points grow with the dimensionality of these points. In non-trivial dimensions, the kernel becomes the identity matrix, and all information from the proxy Z is lost: the bridge function becomes a direct estimator of Y in the V-statistic case, and very unstable in the U-statistic one. Therefore, we implement an alternative bandwidth which is defined by the median heuristic and does not suffer from this degeneration in higher dimension. Both alternatives are evaluated.

Hyperparameter selection. We follow the tuning from Kompa et al. (2022) and choose hyperparameters of the MLP based on the V- and U-statistics on a held-out validation split. The model is then refitted on the whole context using the final parameters. The parameter space searched is described in Table D.3. Note that we do not evaluate every configuration, but sample without repetitions 10 of them.

D.1.4 Proxy maximum moment restriction

We reimplemented the proxy maximum moment restriction (PMMR) method of Mastouri et al. (2021), instead of reusing the authors' experimental code[4]. This allowed us to adjust the code for our usecases (prediction of CATE, with binary treatments). We preserved the PMMR's RKHS bridge class and objective, but adapted treatment kernel to binary setting, and changed the prediction heads to the same two cases as described for NMMR (Appendix D.1.3).

The method. PMMR fits the same outcome bridge h_0 as NMMR, but takes it in an RKHS rather than in a neural network class, which makes the minimiser available in closed form. We use column-wise Gaussian kernels for W, Z and X, and a binary indicator kernel for A. With their products kernels over (A, X, W) and (A, Z, X), respectively, write the two matrices:

$$L = K_A \odot K_W \odot K_X, \qquad \Phi = K_A \odot K_Z \odot K_X, \tag{51}$$

so that $h(\cdot) = \sum_i \alpha_i k_L(\cdot, (A_i, W_i, X_i))$ and the V-statistic (50) of the residuals $Y - L\alpha$ under Φ is minimised by

$$\alpha = \left(L\Phi L + \lambda n^2 L + \epsilon n^2 I\right)^{-1} L\Phi Y. \tag{52}$$

Here λ penalises $\|h\|_{\mathcal{H}_L}$ and is the regulariser of the original method; ϵ is added for numerical stability.

[4]Original implementation available at `https://github.com/yuchen-zhu/kernel_proxies`.

Deviations at prediction time. As for NMMR, the reference code returns only $\mathbb{E}[Y \mid \mathrm{do}(A = a)]$, which here is $\hat{y}(a) = \alpha^\top [\, k_A(A, a) \odot \overline{wx} \,]$ with $\overline{wx}_i = \frac{1}{n} \sum_j k_W(W_i, W_j)\, k_X(X_i, X_j)$. We obtain $\hat{Q}(x)$ through the same two strategies as NMMR ("ind. W", "cond. W"), with the same caveats. Unlike NMMR, `Wcond` does not need extra hyperparameters: the weights used to approximate $W \mid X = x$ are determined by the $k_X(X_j, x)$ from the model.

Hyperparameter selection. We select per episode on the V-statistic of a held-out fraction of the context, with the scoring kernel fixed at the median heuristic of the held-out rows, so that all candidates are compared on one metric rather than each on its own. The parameter space is described in Table D.4. It splits in two: the bandwidth multipliers enter the matrices L and Φ, while λ enters only the matrix assembled in (52). We therefore sample the former and enumerate the latter exhaustively for each draw, re-solving (52) at every λ off a single pair of L, Φ.

Computational notes. For binary A the kernel k_A is an indicator, so (52) is block diagonal by arm and we assemble and solve it one arm at a time; this is exact and reduces computational and memory costs.

Table D.4: PMMR settings. Bandwidth multipliers are *sampled*; the regularisers are *swept*, i.e. every combination is evaluated for each bandwidth draw. Both are selected per episode on the held-out V-statistic.

Component	Setting	Value
Kernels	Bandwidths baseline	per-column median heuristic
	Kernel bandwidth multiplier (W,X)	*searched*: $\{0.8, 1, 1.2\}$
	Kernel bandwidth multiplier (Z)	*searched*: $\{0.8, 1, 1.2\}$
Regularisation	λ (RKHS norm)	*swept*: $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
	Cholesky jitter	10^{-8}
Prediction	W strategy	"ind. W", "cond. W"
Tuning	Validation fraction	0.2
	Search budget	all 9 bandwidth multipliers $\times$ 5 regularisers

D.1.5 Kernel proxy variable

We reimplemented the kernel proxy variable (KPV) method of Mastouri et al. (2021), presented in the same paper as PMMR and sharing its released code. Again, the reimplementation allowed us to adjust the code for our usage: binary kernel and CATE predictions.

The method. KPV fits the same outcome bridge h_0 as PMMR and in the same RKHS, but reaches it through two ridge regressions rather than one moment restriction. The theoretical formulation uses two samples, D_1 of size m_1 and D_2 of size m_2. Following the released implementation, we reuse the complete context in both stages, so $D_1 = D_2$ and $m_1 = m_2 = n_{\text{context}}$. The first stage regresses the feature map of W on (A, X, Z), giving the conditional mean embedding $\hat{\mu}_{W|a,x,z}$; writing $\mathcal{K} = K_A \odot K_X \odot K_Z$ for the product kernel over the conditioning set, its solution is

$$\Gamma = \big(\mathcal{K}_{D_1 D_1} + \lambda_1 m_1 I\big)^{-1} \mathcal{K}_{D_1 D_2}. \tag{53}$$

The second stage regresses Y on that embedding, which yields the bridge coefficients in closed form,

$$\Sigma = \big(\Gamma^\top K_W \Gamma\big) \odot K_A \odot K_X, \qquad \alpha_{ij} = \Gamma_{ij} \big[(\Sigma + \lambda_2 m_2 I)^{-1} Y_{D_2}\big]_j, \tag{54}$$

with K_W taken on D_1 and K_A, K_X on D_2. As done in PMMR, we use columnwise Gaussian kernels for K_X, K_W and K_Z, and a binary indicator one for K_A. Unlike PMMR, the conditional distribution of W given the conditioning set is estimated explicitly, in the first stage, rather than left implicit in a moment condition.

Hyperparameter selection. We use the authors' closed-form leave-one-out criterion for both regularisers, run inside each fit. On the other hand, we modify the hyperparameter tuning of the kernel bandwidth. As suggested by the authors, we initialize the search by the median heuristic. We then

search around this initialization by minimizing the stage-2 leave-one-out criterion (instead of the stage-1 one, used in their implementation). We prefer this as it compares bandwidths using prediction error against the same outcome Y, whereas stage-1 LOOCV changes both the input kernel and the W-kernel defining the loss, making scores across bandwidths less directly comparable. Table D.5 summarises the tuning parameters.

Deviations at prediction time. As for NMMR and PMMR, the reference returns only $\mathbb{E}[Y \mid \mathrm{do}(A = a)]$, here $\hat{\beta}(a) = \frac{1}{n}\sum_{i,j,k} \hat{\alpha}_{ij}\, k_A(\tilde{a}_j, a)\, k_X(\tilde{x}_j, x_k)\, k_W(w_i, w_k)$. We obtain $\hat{Q}(x)$ through the same two strategies ("ind. W", "cond. W"), with the same caveats. As in PMMR, the marginalisation is over the *joint* (X, W), whereas the released code averages the two grams separately, which agrees only if $X \perp\!\!\!\perp W$ in sample. The reference estimates the population effect by averaging the fitted bridge over a separate test sample; we instead use the context rows, which also serve as the fitting sample.

Implementation corrections & computational notes. In the original implementation, the median bandwidth is recomputed at every kernel call, and prediction uses the default scale rather than using the training one. Therefore, training and prediction do not use the same fitted kernel. We instead estimate each bandwidth once from the context and reuse it consistently. The released driver also fits a separate input scaler to the test data; we reuse the transformation from the context. Finally, we leverage the binary aspect of the treatment kernel to solve the closed-form formulas arm by arm, reducing computational and memory costs without changing the estimator.

Table D.5: KPV settings. Both stages use all context rows. The regularizers λ_1 and λ_2 are selected within each fit by their closed-form leave-one-out criteria. The shared kernel-scale multiplier is selected per episode using the stage-2 LOOCV loss.

Component	Setting	Value
Data split	m_1 (stage-1 rows)	n_{context}
	m_2 (stage-2 rows)	n_{context}
Kernels	Kernel family	Column-wise Gaussian RBF for X, W, Z; binary for A
	Base bandwidths	Per-column median heuristic, estimated on the context
	Shared multiplier c on σ_d^2	Shared across X, W, Z; *searched:* $\{0.8, 1.0, 1.2\}$
Regularization	λ_1, λ_2	*searched* by LOOCV; 10-point log grid in $[10^{-5}, 10]$
Prediction	W strategy	"ind. W", "cond. W"
Tuning	Selection score	Stage-2 leave-one-out loss
	Bandwidth candidates	3 per episode; no separate validation split

D.2 Semi-synthetic benchmarks from real covariates

We construct proximal benchmarks from twelve numerical datasets in the INRIA-SODA tabular benchmark collection (Grinsztajn et al., 2022).[5] Real source rows supply the joint distribution of observed and hidden variables. We generate treatment, proxies, and outcomes with known causal mechanisms. Crossing linear/nonlinear mechanisms, weak/strong confounding, and weak/strong proxy information gives $12 \times 2 \times 2 \times 2 = 96$ configurations. Three seeded realizations per configuration give 288 realizations. Figure D.1(a) shows how the real covariate distribution enters the construction.

Sources and calibration. We select twelve datasets from the INRIA-SODA tabular benchmark collection with at least 20 features and 10 000 rows. For each source, we reserve a fixed set of rows for calibration and use the remaining rows to construct evaluation episodes. Calibration rows determine missing-value handling, clipping, and feature standardisation; the resulting transformations remain fixed across episodes. We exclude features that cannot be reliably calibrated.

We assign the retained source columns, which may include the dataset's original prediction target, to observed covariates X and hidden source variables U_{src} using a deterministic ordering. The two

[5] https://huggingface.co/datasets/inria-soda/tabular-benchmark

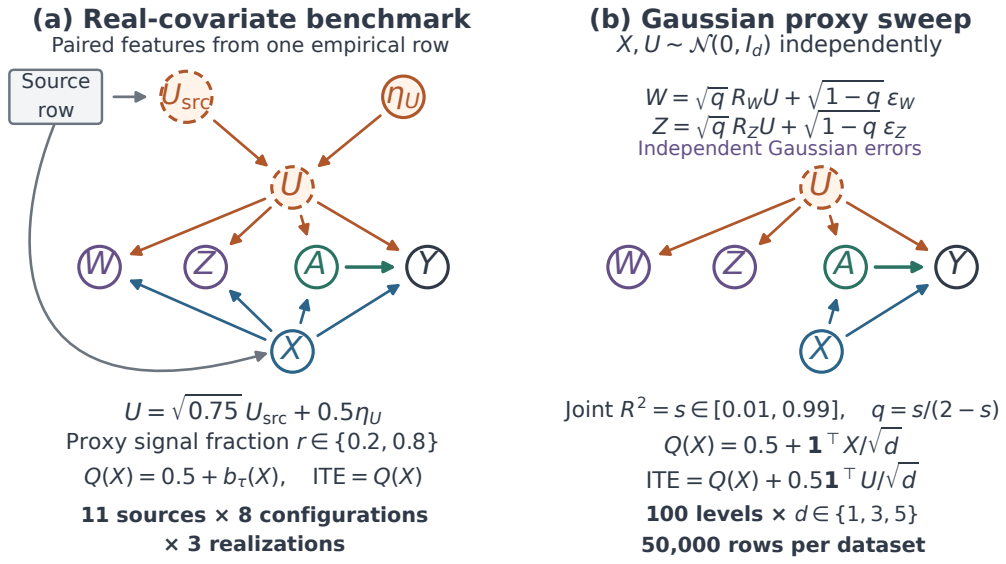


Figure D.1: Construction of the generated proximal benchmarks. (a) A shared real source row supplies dependent observed covariates and hidden source features. Independent Gaussian innovation augments the hidden features. Both X and U enter the proxy measurements and the treatment and outcome mechanisms. (b) Independent Gaussian X and U define the controlled synthetic study. The joint proxy information parameter changes the measurement channels while preserving the causal variables within each dimension. The diagrams display the generated causal structure, with independent disturbances suppressed where indicated.

Table D.6: Original sizes of the twelve selected datasets. Feature counts exclude the original prediction target, although that column remains eligible for selection as a source variable in our construction.

Source	Rows	Features
pol	10 082	26
MiniBooNE	72 998	50
default-of-credit-card-clients	13 272	20
Higgs	940 160	24
jannis	57 580	54
heloc	10 000	22
covertype	423 680	54
albert	58 252	31
road-safety	111 762	32
Ailerons	13 750	33
superconduct	21 263	79
Allstate_Claims_Severity	188 318	124

groups contain the same number of features; we use $d = 5$ for our experiments. Each realization samples context and query rows from the evaluation pool without replacement. Table D.6 reports the original sizes of the selected datasets.

Latent and observed mechanisms. For each row, we draw $\eta_U \sim \mathcal{N}(0, I_{d_U})$ independently and set

$$U = \sqrt{0.75}\, U_{\mathrm{src}} + 0.5\eta_U.$$

Because X and U_{src} come from the same source row, U retains some of their dependence. The independent noise term adds variation in U even when X is fixed. Let g be identity in the linear

family and componentwise asinh in the nonlinear family. A separate calibration innovation stream determines

$$m = \mathrm{mean}_{\mathrm{cal}}\, g(U), \qquad C = \mathrm{Cov}_{\mathrm{cal}}\, g(U), \qquad V = (C + 0.05I)^{-1/2}\{g(U) - m\}.$$

The covariance uses the population denominator and the inverse square root is symmetric. We draw $\widetilde{c} \sim \mathcal{N}(0, I_{d_U})$ and rescale it to c so that $L = c^\top V$ has unit calibration standard deviation. Treatment and baseline outcome share this latent score.

The linear observed basis contains the coordinates of X. The nonlinear basis applies one uniformly selected transformation from t, $\mathrm{asinh}(t)$, $\sin(t)$, and t^2 to each coordinate, then adds products for up to eight distinct coordinate pairs selected uniformly without replacement. After standardising these basis columns on calibration rows, let K denote their number. For each role $r \in \{a, y, \tau, w, z\}$, we draw independent coefficients from $\mathcal{N}(0, 1/K)$ and standardise the resulting linear combinations on calibration rows to obtain $b_r(X)$. The functions b_a, b_y, b_τ are scalar and b_w, b_z each have d_U coordinates. Standard deviations used to normalise the score, basis, and functions have a floor of 10^{-8}. We set

$$f_A(X) = a_0 + 0.5 b_a(X), \quad f_Y(X) = b_y(X), \quad f_W(X) = 0.5 b_w(X), \quad f_Z(X) = 0.5 b_z(X),$$

where $a_0 \sim \mathrm{Unif}[-0.25, 0.25]$, and define $Q(X) = 0.5 + b_\tau(X)$. The CATE function therefore has calibration mean 0.5 and standard deviation one.

Proxy channels. We obtain orthogonal B_W, B_Z from independent Gaussian matrices using QR decomposition. We choose signs for a nonnegative diagonal in the triangular factor. For $J \in \{W, Z\}$, define

$$\sigma_{J,j} = \max\{\mathrm{sd}_{\mathrm{cal}}[(B_J V)_j], 10^{-8}\}, \qquad D_J(r) = \sqrt{\frac{1-r}{r}}\, \mathrm{diag}(\sigma_{J,1}, \ldots, \sigma_{J,d_U}).$$

With independent standard-normal errors, the proxy measurements are

$$W = \phi\{B_W V + f_W(X) + D_W(r)\epsilon_W\}, \qquad Z = \phi\{B_Z V + f_Z(X) + D_Z(r)\epsilon_Z\}.$$

The map ϕ is identity in the linear family and componentwise asinh in the nonlinear family. Weak and strong proxy settings use $r = 0.2$ and $r = 0.8$. This parameter is the calibration latent-signal fraction in each measurement coordinate before the observed offset and observation map. Joint predictive information about U also depends on the source distribution and both proxy channels.

Treatment, outcomes, and bridge. Using independent treatment and outcome disturbances, we generate

$$e(X, U) = \mathrm{clip}\{\mathrm{sigmoid}(f_A(X) + \alpha L), 0.02, 0.98\}, \tag{55}$$

$$A \sim \mathrm{Bernoulli}\{e(X, U)\}, \tag{56}$$

$$Y(0) = f_Y(X) + \gamma L + \epsilon_Y, \qquad \epsilon_Y \sim \mathcal{N}(0, 1), \tag{57}$$

$$Y(1) = Y(0) + Q(X), \qquad Y = Y(0) + AQ(X). \tag{58}$$

Weak confounding uses $(\alpha, \gamma) = (0.35, 0.5)$ and strong confounding uses $(1.25, 1.5)$. The potential outcomes share their disturbance, giving the exact heterogeneous effect

$$Y(1) - Y(0) = Q(X) = \mathbb{E}[Y(1) - Y(0) \mid X].$$

An explicit outcome bridge is

$$h(w, a, x) = f_Y(x) + \gamma c^\top B_W^{-1}\{\phi^{-1}(w) - f_W(x)\} + aQ(x).$$

Here ϕ^{-1} is identity or componentwise $\sinh$. Independent, zero-mean measurement error gives $\mathbb{E}[h(W, a, X) \mid X, U] = \mathbb{E}[Y(a) \mid X, U]$. The graph in Figure D.1(a) also gives $W \perp (A, Z) \mid (X, U)$ and $Y \perp Z \mid (A, X, U)$, establishing

$$\mathbb{E}[Y \mid A, Z, X] = \mathbb{E}[h(W, A, X) \mid A, Z, X].$$

The nonlinear family consequently retains an explicit bridge through invertible measurement transformations.

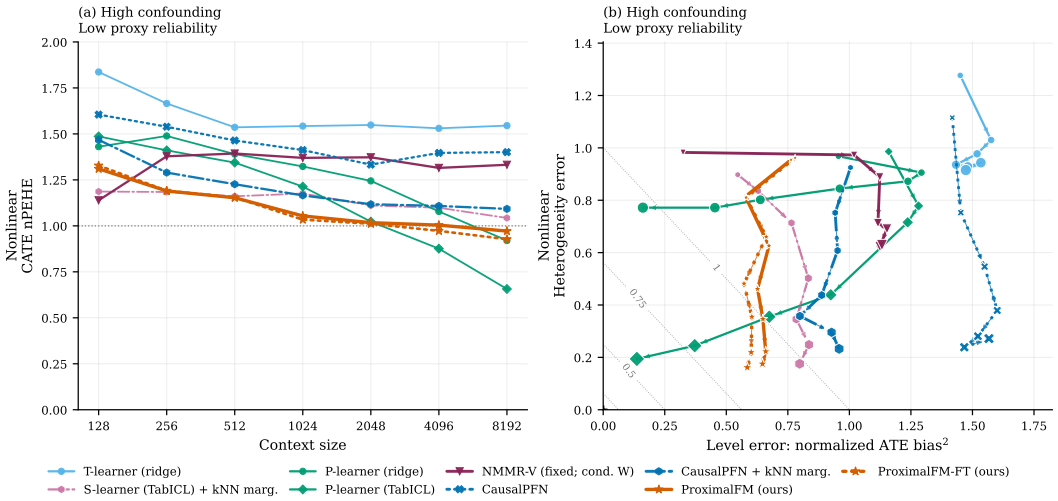


Figure D.2: Learning curves and error decomposition for the nonlinear, high-confounding, low-proxy-reliability setting (36 episodes: 12 sources with three realizations each). **(a)** Median CATE nPEHE versus context size; the dotted line marks the constant-ATE benchmark (nPEHE $= 1$). `ProximalFM-FT` continues training the main checkpoint on the same prior and objective with contexts up to $5\,000$ rows, updating only part of the in-context predictor (Appendix C.3.1). **(b)** For query-row errors $e_i = \hat{\tau}_i - \tau_i$, their mean $\bar{e}$, and true CATE variance $s_\tau^2 = \mathrm{Var}_i(\tau_i)$, the episode-level decomposition is $\mathrm{nPEHE}^2 = \bar{e}^2/s_\tau^2 + \mathrm{Var}_i(e_i)/s_\tau^2$. Panel (b) plots the first term horizontally (squared bias in the query-average treatment effect) and the second vertically (heterogeneity error after removing that bias). Dotted contours show constant $\sqrt{x+y}$; plotted coordinates are separate component medians, so they need not reproduce panel (a)'s median nPEHE. Arrows and larger markers indicate increasing context size (128–8 192). `ProximalFM-FT` moves mainly leftward: post-training improves the average effect estimate, with little improvement in CATE shape.

Paired configurations. For each source and dimension, feature assignments are fixed across all configurations and realizations. Within a realization, all configurations use the same sampled rows; configurations in the same mechanism family also share fitted functions and random disturbances. Different realizations independently sample rows from the same evaluation pool and redraw the mechanisms and disturbances. Their sampled rows may overlap.

D.3 Results on semi-synthetic datasets

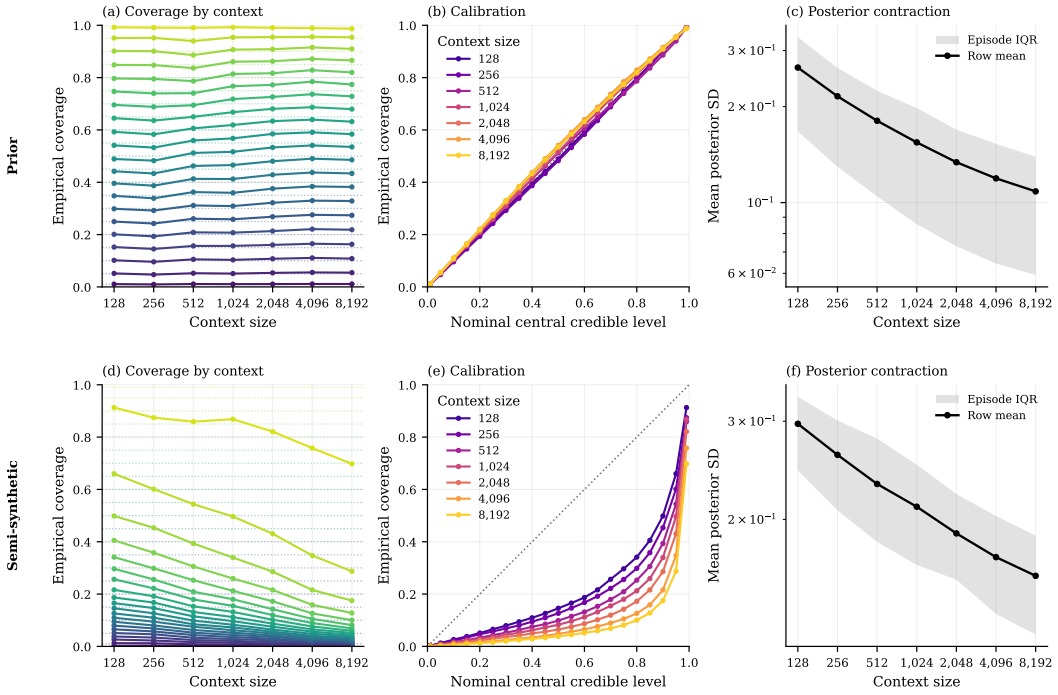


Figure D.3: Posterior calibration and contraction. The top row evaluates 400 datasets drawn from the pre-training prior; the bottom row evaluates 120 semi-synthetic datasets with nonlinear mechanisms, high confounding, and low proxy reliability. In every episode, the same 256 query points are evaluated as the context grows through nested prefixes from 128 to 8 192 observations. **(a,d) Coverage by context:** each solid curve gives empirical coverage of a central CATE credible interval at one nominal level (1%, 5%, 10%, …, 95%, 99%); the faint dotted line of the same colour marks that level's nominal coverage. **(b,e) Calibration:** each curve fixes a context size, coloured from purple (128) to yellow (8 192), and plots empirical against nominal coverage. The dotted diagonal denotes perfect calibration; curves below it indicate undercoverage. **(c,f) Posterior contraction:** the black curve is the mean predicted CATE posterior standard deviation across query rows. The grey band is the interquartile range of episode-level mean standard deviations, showing variation between datasets. These panels use logarithmic vertical axes. Coverage stays close to nominal on the prior, whereas the semi-synthetic intervals increasingly undercover as context grows, even though their predicted uncertainty contracts.

Table D.7: Exhaustive semisynthetic CATE nPEHE results at $n_{\mathrm{ctx}} = 128$. Each entry first averages the three replicates of each source dataset, then averages equally over the 12 sources; lower is better. The lowest mean in each column is bold. Other underlined methods are not significantly worse than that empirical best under one-sided exact paired sign-flip tests across source-level means, with Holm correction within the column ($\alpha = 0.05$).

	Linear				Nonlinear			
	Low confounding		High confounding		Low confounding		High confounding	
Method	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy
S-learner (ridge)	1.027	1.027	1.458	1.458	1.059	1.059	1.459	1.459
S-learner (RF)	0.868	0.868	1.229	1.229	0.986	0.986	1.542	1.542
S-learner (TabICL)	0.782	0.782	1.336	1.336	**0.937**	**0.937**	1.457	1.457
S-learner (TabICL) + kNN marg.	0.885	0.895	1.070	0.921	0.973	0.982	1.259	0.986
S-learner (TabICL) + kernel marg.	0.888	0.888	1.064	0.914	0.966	0.966	1.245	**0.966**
T-learner (ridge)	0.603	0.603	1.466	1.466	1.122	1.122	1.837	1.837
T-learner (RF)	0.846	0.846	1.580	1.580	1.157	1.157	1.751	1.751
T-learner (TabICL)	0.865	0.865	1.637	1.637	1.232	1.232	1.810	1.810
T-learner (ridge) + kNN marg.	0.639	0.603	1.255	0.694	1.063	1.103	1.631	1.398
T-learner (ridge) + kernel marg.	0.665	0.610	1.276	0.704	1.064	1.074	1.634	1.395
X-learner (RF)	0.718	0.718	1.436	1.436	1.025	1.025	1.605	1.605
X-learner (TabICL)	0.737	0.737	1.509	1.509	1.065	1.065	1.669	1.669
CausalFM	0.738	0.738	1.306	1.306	1.048	1.048	1.566	1.566
Do-PFN	0.806	0.806	1.294	1.294	1.606	1.606	1.604	1.604
CausalPFN	0.641	0.641	1.408	1.408	1.055	1.055	1.687	1.687
CausalPFN + kNN marg.	0.577	**0.565**	1.112	**0.654**	0.971	0.945	1.460	1.048
CausalPFN + kernel marg.	0.597	0.574	1.133	0.668	0.979	0.956	1.466	1.072
P-learner (RF)	0.731	0.708	1.267	0.916	0.993	0.980	1.514	1.222
P-learner (ridge)	**0.571**	0.581	1.181	0.771	1.001	0.995	1.511	1.221
P-learner (TabICL)	0.652	0.664	1.289	0.891	0.984	0.972	1.535	1.210
KPV (cond. W)	1.325	1.021	1.905	1.192	1.300	1.014	2.054	1.204
KPV (independent W)	0.955	0.953	1.395	0.970	1.040	1.003	1.551	1.028
PMMR (cond. W)	0.957	0.922	1.011	0.953	1.006	1.002	1.105	1.072
PMMR (independent W)	0.993	0.989	**0.982**	0.953	1.017	1.021	**1.039**	1.006
NMMR-V (fixed; cond. W)	0.954	0.910	1.090	0.917	1.012	1.013	1.290	1.006
NMMR-V (fixed; ind. W)	0.957	0.915	1.094	0.926	1.016	1.017	1.292	1.006
NMMR-U (fixed; cond. W)	4.98×10^{3}	430.450	1.13×10^{5}	155.343	6.16×10^{3}	55.108	6.11×10^{5}	937.488
NMMR-U (fixed; ind. W)	4.90×10^{3}	408.021	1.14×10^{5}	148.985	5.94×10^{3}	55.162	5.97×10^{5}	1.36×10^{3}
NMMR-V (median; cond. W)	1.060	1.051	1.045	0.983	1.077	1.072	1.115	1.028
NMMR-V (median; ind. W)	1.061	1.051	1.044	0.983	1.077	1.073	1.115	1.029
NMMR-U (median; cond. W)	6.500	1.479	5.725	1.768	51.236	1.439	6.794	1.341
NMMR-U (median; ind. W)	6.940	1.460	5.725	1.763	49.715	1.420	7.024	1.324
ProximalFM (ours)	0.621	0.608	1.058	0.733	0.953	0.947	1.329	0.968

Table D.8: Exhaustive semisynthetic CATE nPEHE results at $n_{\mathrm{ctx}} = 512$. Each entry first averages the three replicates of each source dataset, then averages equally over the 12 sources; lower is better. The lowest mean in each column is bold. Other underlined methods are not significantly worse than that empirical best under one-sided exact paired sign-flip tests across source-level means, with Holm correction within the column ($\alpha = 0.05$).

	Linear				Nonlinear			
	Low confounding		High confounding		Low confounding		High confounding	
Method	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy
S-learner (ridge)	1.013	1.013	1.564	1.564	1.023	1.023	1.580	1.580
S-learner (RF)	0.669	0.669	1.423	1.423	0.849	0.849	1.546	1.546
S-learner (TabICL)	0.499	0.499	1.294	1.294	0.762	0.762	1.432	1.432
S-learner (TabICL) + kNN marg.	0.527	0.581	1.053	0.683	0.793	0.822	1.206	0.865
S-learner (TabICL) + kernel marg.	0.526	0.574	1.053	0.678	0.787	0.811	1.196	0.846
T-learner (ridge)	0.315	0.315	1.338	1.338	1.015	1.015	1.665	1.665
T-learner (RF)	0.690	0.690	1.506	1.506	0.921	0.921	1.605	1.605
T-learner (TabICL)	0.498	0.498	1.451	1.451	0.785	0.785	1.531	1.531
T-learner (ridge) + kNN marg.	**0.302**	0.282	1.041	**0.408**	0.978	0.926	1.441	1.008
T-learner (ridge) + kernel marg.	0.307	**0.279**	1.051	0.413	0.975	0.897	1.437	0.981
X-learner (RF)	0.530	0.530	1.381	1.381	0.814	0.814	1.508	1.508
X-learner (TabICL)	0.424	0.424	1.379	1.379	**0.726**	0.726	1.462	1.462
CausalFM	0.638	0.638	1.288	1.288	0.945	0.945	1.467	1.467
Do-PFN	0.609	0.609	1.362	1.362	1.043	1.043	1.516	1.516
CausalPFN	0.372	0.372	1.330	1.330	0.730	0.730	1.488	1.488
CausalPFN + kNN marg.	0.356	0.310	1.035	0.424	0.760	0.740	1.264	0.826
CausalPFN + kernel marg.	0.360	0.314	1.043	0.436	0.752	**0.713**	1.251	**0.783**
P-learner (RF)	0.606	0.569	1.273	0.741	0.830	0.808	1.406	0.896
P-learner (ridge)	0.410	0.379	1.164	0.567	0.922	0.910	1.443	0.991
P-learner (TabICL)	0.498	0.465	1.244	0.661	0.773	0.747	1.368	0.838
KPV (cond. W)	1.936	0.909	3.028	1.202	2.039	0.996	2.573	1.162
KPV (independent W)	1.222	0.844	2.172	0.898	1.228	0.903	2.013	0.956
PMMR (cond. W)	0.878	0.856	1.034	0.910	0.914	0.898	1.155	1.009
PMMR (independent W)	0.911	0.924	1.009	0.912	0.962	0.966	1.121	0.993
NMMR-V (fixed; cond. W)	0.727	0.695	1.187	0.781	0.883	0.892	1.368	0.973
NMMR-V (fixed; ind. W)	0.731	0.706	1.190	0.793	0.887	0.901	1.371	0.982
NMMR-U (fixed; cond. W)	14.122	0.929	3.879	0.955	2.371	0.998	33.761	1.009
NMMR-U (fixed; ind. W)	15.140	0.935	3.891	0.959	2.385	1.002	33.990	1.011
NMMR-V (median; cond. W)	1.044	1.036	1.015	0.977	1.040	1.045	**1.109**	0.997
NMMR-V (median; ind. W)	1.044	1.037	1.019	0.980	1.042	1.047	1.111	1.001
NMMR-U (median; cond. W)	1.079	1.081	1.100	1.023	1.072	1.067	1.121	1.014
NMMR-U (median; ind. W)	1.079	1.081	1.101	1.023	1.073	1.068	1.121	1.014
ProximalFM (ours)	0.392	0.370	**0.953**	0.468	0.780	0.758	1.151	0.789

Table D.9: Exhaustive semisynthetic CATE nPEHE results at $n_{\mathrm{ctx}} = 1,024$. Each entry first averages the three replicates of each source dataset, then averages equally over the 12 sources; lower is better. The lowest mean in each column is bold. Other underlined methods are not significantly worse than that empirical best under one-sided exact paired sign-flip tests across source-level means, with Holm correction within the column ($\alpha = 0.05$).

	Linear				Nonlinear			
	Low confounding		High confounding		Low confounding		High confounding	
Method	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy
S-learner (ridge)	1.011	1.011	1.567	1.567	1.024	1.024	1.593	1.593
S-learner (RF)	0.568	0.568	1.384	1.384	0.793	0.793	1.497	1.497
S-learner (TabICL)	0.379	0.379	1.256	1.256	0.653	0.653	1.393	1.393
S-learner (TabICL) + kNN marg.	0.393	0.426	0.994	0.552	0.694	0.732	1.169	0.783
S-learner (TabICL) + kernel marg.	0.396	0.425	0.995	0.550	0.686	0.715	1.159	0.765
T-learner (ridge)	0.242	0.242	1.284	1.284	0.998	0.998	1.627	1.627
T-learner (RF)	0.561	0.561	1.413	1.413	0.786	0.786	1.502	1.502
T-learner (TabICL)	0.382	0.382	1.357	1.357	0.670	0.670	1.425	1.425
T-learner (ridge) + kNN marg.	**0.220**	**0.194**	0.990	**0.312**	0.967	0.874	1.390	0.934
T-learner (ridge) + kernel marg.	0.225	0.197	0.992	0.318	0.962	0.855	1.386	0.913
X-learner (RF)	0.453	0.453	1.326	1.326	0.734	0.734	1.459	1.459
X-learner (TabICL)	0.348	0.348	1.303	1.303	**0.609**	0.609	1.396	1.396
CausalFM	0.616	0.616	1.272	1.272	0.927	0.927	1.453	1.453
Do-PFN	0.544	0.544	1.314	1.314	0.993	0.993	1.537	1.537
CausalPFN	0.297	0.297	1.277	1.277	0.629	0.629	1.412	1.412
CausalPFN + kNN marg.	0.273	0.224	0.977	0.333	0.646	0.621	1.175	0.675
CausalPFN + kernel marg.	0.275	0.225	0.979	0.338	0.637	**0.599**	1.168	**0.643**
P-learner (RF)	0.515	0.482	1.147	0.572	0.769	0.727	1.280	0.761
P-learner (ridge)	0.322	0.290	1.033	0.372	0.891	0.879	1.348	0.893
P-learner (TabICL)	0.416	0.387	1.094	0.481	0.664	0.636	1.218	0.669
KPV (cond. W)	1.499	0.926	2.598	1.208	1.433	0.825	2.208	0.960
KPV (independent W)	0.993	0.775	1.944	0.840	1.006	0.858	1.829	0.896
PMMR (cond. W)	0.878	0.846	1.062	0.903	0.918	0.897	1.188	0.995
PMMR (independent W)	0.903	0.910	1.033	0.905	0.956	0.961	1.156	0.991
NMMR-V (fixed; cond. W)	0.604	0.580	1.132	0.720	0.834	0.841	1.370	0.925
NMMR-V (fixed; ind. W)	0.609	0.602	1.135	0.745	0.840	0.859	1.375	0.937
NMMR-U (fixed; cond. W)	0.919	0.894	1.071	0.905	0.992	0.984	1.178	0.990
NMMR-U (fixed; ind. W)	0.920	0.903	1.076	0.920	0.994	0.989	1.180	0.995
NMMR-V (median; cond. W)	1.006	1.005	1.055	0.951	1.038	1.022	1.175	1.003
NMMR-V (median; ind. W)	1.006	1.009	1.058	0.956	1.041	1.027	1.178	1.006
NMMR-U (median; cond. W)	1.063	1.049	1.124	1.002	1.060	1.050	1.129	1.011
NMMR-U (median; ind. W)	1.063	1.050	1.122	1.007	1.060	1.051	1.132	1.010
ProximalFM (ours)	0.300	0.279	**0.884**	0.365	0.656	0.636	**1.061**	0.688

Table D.10: Exhaustive semisynthetic CATE nPEHE results at $n_{\mathrm{ctx}} = 2,048$. Each entry first averages the three replicates of each source dataset, then averages equally over the 12 sources; lower is better. The lowest mean in each column is bold. Other underlined methods are not significantly worse than that empirical best under one-sided exact paired sign-flip tests across source-level means, with Holm correction within the column ($\alpha = 0.05$).

	Linear				Nonlinear			
	Low confounding		High confounding		Low confounding		High confounding	
Method	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy	Low proxy	High proxy
S-learner (ridge)	1.011	1.011	1.560	1.560	1.022	1.022	1.606	1.606
S-learner (RF)	0.513	0.513	1.356	1.356	0.740	0.740	1.476	1.476
S-learner (TabICL)	0.310	0.310	1.239	1.239	0.564	0.564	1.371	1.371
S-learner (TabICL) + kNN marg.	0.291	0.298	0.953	0.441	0.580	0.624	1.111	0.695
S-learner (TabICL) + kernel marg.	0.293	0.297	0.953	0.437	0.585	0.608	1.111	0.679
T-learner (ridge)	0.205	0.205	1.264	1.264	1.112	1.112	1.754	1.754
T-learner (RF)	0.482	0.482	1.363	1.363	0.751	0.751	1.482	1.482
T-learner (TabICL)	0.306	0.306	1.303	1.303	0.604	0.604	1.420	1.420
T-learner (ridge) + kNN marg.	**0.179**	**0.132**	0.960	**0.271**	1.052	0.936	1.491	0.984
T-learner (ridge) + kernel marg.	0.181	0.134	0.962	0.272	1.046	0.913	1.486	0.961
X-learner (RF)	0.382	0.382	1.286	1.286	0.678	0.678	1.429	1.429
X-learner (TabICL)	0.299	0.299	1.262	1.262	**0.504**	**0.504**	1.352	1.352
CausalFM	0.600	0.600	1.262	1.262	0.917	0.917	1.460	1.460
Do-PFN	0.454	0.454	1.279	1.279	0.961	0.961	1.547	1.547
CausalPFN	0.246	0.246	1.255	1.255	0.545	0.545	1.372	1.372
CausalPFN + kNN marg.	0.217	0.167	0.946	0.285	0.571	0.540	1.127	0.595
CausalPFN + kernel marg.	0.218	0.168	0.948	0.288	0.567	0.521	1.121	**0.571**
P-learner (RF)	0.420	0.401	0.985	0.493	0.675	0.642	1.112	0.675
P-learner (ridge)	0.237	0.203	0.882	0.311	0.870	0.859	1.219	0.873
P-learner (TabICL)	0.334	0.308	0.939	0.420	0.575	0.553	1.037	0.589
KPV (cond. W)	1.745	0.715	2.778	0.934	1.752	0.781	2.591	0.948
KPV (independent W)	1.163	0.707	2.081	0.764	1.110	0.797	1.982	0.855
PMMR (cond. W)	0.881	0.843	1.084	0.905	0.924	0.892	1.217	1.005
PMMR (independent W)	0.897	0.905	1.050	0.902	0.954	0.957	1.182	0.994
NMMR-V (fixed; cond. W)	0.527	0.535	1.104	0.663	0.803	0.805	1.369	0.901
NMMR-V (fixed; ind. W)	0.531	0.557	1.109	0.694	0.807	0.818	1.372	0.910
NMMR-U (fixed; cond. W)	0.893	0.843	1.061	0.881	0.991	0.970	1.173	0.978
NMMR-U (fixed; ind. W)	0.897	0.858	1.063	0.902	0.992	0.979	1.175	0.984
NMMR-V (median; cond. W)	0.964	0.966	1.123	0.939	1.008	1.007	1.268	0.990
NMMR-V (median; ind. W)	0.969	0.972	1.127	0.950	1.008	1.009	1.271	0.997
NMMR-U (median; cond. W)	1.035	1.032	1.127	1.002	1.054	1.047	1.232	1.005
NMMR-U (median; ind. W)	1.035	1.032	1.134	1.003	1.054	1.048	1.228	1.006
ProximalFM (ours)	0.241	0.217	**0.834**	0.302	0.572	0.546	**1.029**	0.620

Table D.11: Extended-context semisynthetic CATE nPEHE results for nonlinear mechanisms with high confounding and low proxy reliability. Each entry first averages the 3 replicates of each source dataset, then averages equally over the 12 sources; lower is better. The lowest mean in each column is bold. Other underlined methods are not significantly worse than that empirical best under one-sided exact paired sign-flip tests across source-level means, with Holm correction within the column ($\alpha = 0.05$).

Method	Context size n_{ctx}						
	128	256	512	1,024	2,048	4,096	8,192
S-learner (ridge)	1.459	1.499	1.580	1.593	1.606	1.597	1.608
S-learner (RF)	1.542	1.536	1.546	1.497	1.476	1.602	1.473
S-learner (TabICL)	1.457	1.449	1.432	1.393	1.371	1.340	1.317
S-learner (TabICL) + kNN marg.	1.259	1.235	1.206	1.169	1.111	1.065	1.020
S-learner (TabICL) + kernel marg.	1.245	1.231	1.196	1.159	1.111	1.064	1.016
T-learner (ridge)	1.837	1.699	1.665	1.627	1.754	1.711	1.647
X-learner (RF)	1.605	1.559	1.508	1.459	1.429	1.504	1.363
X-learner (TabICL)	1.669	1.493	1.462	1.396	1.352	1.330	1.307
CausalPFN	1.687	1.541	1.488	1.412	1.372	1.371	1.376
CausalPFN + kNN marg.	1.460	1.338	1.264	1.175	1.127	1.121	1.117
CausalPFN + kernel marg.	1.466	1.349	1.251	1.168	1.121	1.140	1.127
P-learner (RF)	1.514	1.485	1.406	1.280	1.112	0.964	0.732
P-learner (ridge)	1.511	1.489	1.443	1.348	1.219	1.110	0.952
P-learner (TabICL)	1.535	1.467	1.368	1.218	1.037	**0.847**	**0.660**
NMMR-V (fixed; cond. W)	1.290	1.363	1.368	1.370	1.369	1.336	1.337
NMMR-V (fixed; ind. W)	1.292	1.366	1.371	1.375	1.372	1.338	1.339
NMMR-U (fixed; cond. W)	6.11×10^5	4.25×10^3	33.761	1.178	1.173	1.144	1.141
NMMR-U (fixed; ind. W)	5.97×10^5	4.29×10^3	33.990	1.180	1.175	1.149	1.144
NMMR-V (median; cond. W)	1.115	1.191	**1.109**	1.175	1.268	1.334	1.391
NMMR-V (median; ind. W)	**1.115**	**1.188**	1.111	1.178	1.271	1.335	1.396
NMMR-U (median; cond. W)	6.794	1.318	1.121	1.129	1.232	1.247	1.231
NMMR-U (median; ind. W)	7.024	1.311	1.121	1.132	1.228	1.242	1.230
ProximalFM (ours)	1.329	1.210	1.151	1.061	1.029	0.988	0.964
ProximalFM-FT (ours)	1.333	1.216	1.145	**1.049**	**1.007**	0.960	0.930

D.3.1 SPEED

We measured runtime in a separate run of the 36 semi-synthetic episodes with nonlinear mechanisms, strong confounding, and weak proxy information. Episodes were executed sequentially in one process to avoid contention between simultaneous benchmark jobs; methods retained the parallelism specified for their own hyperparameter searches. At each context size, the recorded wall-clock time covers estimator construction, fitting, and prediction on the 256 query rows. The filled points in Figure 5 show the median of these per-episode times at $n_{\mathrm{ctx}} = 2\,048$; hollow points subtract the measured hyperparameter-search time. Accuracy ranks come independently from the complete semi-synthetic evaluation matrix, using paired episode-level CATE nPEHE ranks.

D.4 SYNTHETIC STUDY OF PROXY INFORMATIVENESS

We vary proxy information within a fixed causal mechanism to examine when proximal estimation offers an advantage over regression adjustment using proxies. For each $d \in \{1, 3, 5\}$, we generate 20 information levels equally spaced from 0.01 to 0.99.

For each of the resulting 60 datasets, we evaluate methods using 128, 512, or $2\,048$ context rows and the same 256 held-out query rows. Each block X, U, W, Z has dimension d. Figure D.1(b) shows the causal structure and proxy-information control.

Causal mechanism and heterogeneity. For every observation, $X, U, \epsilon_W, \epsilon_Z$ are mutually independent $\mathcal{N}(0, I_d)$ vectors. The independent outcome disturbance is $\epsilon_Y \sim \mathcal{N}(0, 1)$. Independent uniform draws determine the Bernoulli treatments. Define

$$S_X = d^{-1/2} \sum_{j=1}^{d} X_j, \qquad S_U = d^{-1/2} \sum_{j=1}^{d} U_j.$$

We generate

$$p(X, U) = \mathrm{clip}\{\mathrm{sigmoid}(0.5S_X + 1.25S_U), 0.02, 0.98\}, \tag{59}$$

$$A \sim \mathrm{Bernoulli}\{p(X, U)\}, \tag{60}$$

$$Y(0) = 0.5S_X + 1.5S_U + \epsilon_Y, \tag{61}$$

$$Y(1) = Y(0) + 0.5 + S_X + 0.5S_U, \tag{62}$$

$$Y = (1 - A)Y(0) + AY(1). \tag{63}$$

The individual and conditional average effects satisfy

$$\mathrm{ITE} = 0.5 + S_X + 0.5S_U, \qquad Q(X) = \mathbb{E}[\mathrm{ITE} \mid X] = 0.5 + S_X.$$

Thus the CATE varies with X, while latent variation also modifies individual effects. In the population, $\mathrm{Var}(Q(X)) = 1$ and $\mathrm{Var}(\mathrm{ITE} - Q(X)) = 0.25$. Normalising the scores by $\sqrt{d}$ holds the population scales of confounding and heterogeneity fixed across dimensions.

Joint proxy information. For each dimension, we draw and fix independent orthogonal matrices R_W, R_Z using Gaussian matrices and QR decomposition with a nonnegative triangular diagonal. At target information s, define $q = s/(2 - s)$ and generate

$$W = \sqrt{q}\, R_W U + \sqrt{1 - q}\, \epsilon_W, \qquad Z = \sqrt{q}\, R_Z U + \sqrt{1 - q}\, \epsilon_Z.$$

Each proxy coordinate has unit variance. Each complete proxy block alone explains a fraction q of each latent coordinate's variance. The joint Gaussian distribution gives

$$\mathbb{E}[U \mid W, Z] = \frac{\sqrt{q}}{1 + q}(R_W^\top W + R_Z^\top Z), \tag{64}$$

$$\mathrm{Cov}(U \mid W, Z) = \frac{1 - q}{1 + q} I_d = (1 - s) I_d. \tag{65}$$

Consequently, the population R^2 for predicting any coordinate of U from (W, Z) jointly is exactly s. Independence of X preserves this value when conditioning additionally on X. The rotations mix latent coordinates without changing information.

Bridge and pairing. Define the unbiased proxy score $\widetilde{S}_U(W) = \mathbf{1}^\top R_W^\top W/\sqrt{qd}$. An explicit bridge is

$$h_0(W, a, X) = 0.5S_X + (1.5 + 0.5a)\widetilde{S}_U(W) + a(0.5 + S_X).$$

Independent proxy errors give $\mathbb{E}[Y - h_0(W, A, X) \mid A, Z, X] = 0$, and averaging the bridge contrast over $W \mid X$ gives $Q(X) = 0.5 + S_X$. The proxy-score error has conditional variance $(1-q)/q$. This quantifies the noise amplification as proxy information decreases, even though the bridge equation remains valid.

Estimator comparison. We fit an interacted proximal two-stage least-squares (2SLS) estimator (Tchetgen Tchetgen et al., 2020) with bridge and instrument designs

$$D_i = (1, X_i^\top, A_i, A_i X_i^\top, W_i^\top, A_i W_i^\top),$$
$$T_i = (1, X_i^\top, A_i, A_i X_i^\top, Z_i^\top, A_i Z_i^\top).$$

Writing P_T for projection onto the instrument column space, 2SLS minimises $\|P_T(Y - D\beta)\|_2^2$. The treatment interactions allow the bridge basis to contain h_0 and to represent effect modification by both X and the latent confounder. We compute the projection using economic QR followed by SVD, retaining singular values above $\epsilon_{\text{mach}} \max(n_{\text{ctx}}, p_T)\sigma_{\max}$, where $p_T = 2 + 4d$. The subsequent least-squares fit uses NumPy's `rcond=None` rank threshold.

We compare 2SLS with unpenalised OLS using the design $(1, X^\top, A, AX^\top, C^\top, AC^\top)$, with either no additional covariates (C empty) or proxy adjustment ($C = (W, Z)$). For these linear estimators, we average the fitted treatment contrast over the context distribution of the additional covariates:

$$\widehat{Q}(x) = \widehat{\beta}_A + x^\top \widehat{\beta}_{AX} + \overline{C}_{\text{ctx}}^\top \widehat{\beta}_{AC}.$$

For 2SLS, $C = W$; for X-only OLS, the final term is omitted. This marginalisation targets $Q(x)$ because X is independent of the additional covariates in the data-generating mechanism.

We also evaluate ProximalFM and proxy-adjusted CausalPFN. ProximalFM conditions on the observational context (X, W, Z, A, Y) and predicts CATE from query X. CausalPFN treats (W, Z) as adjustment covariates; we marginalise its treatment contrasts using 16 seeded, aligned proxy pairs sampled from the context. At each context size, all methods use the same training rows and are scored against the known $Q(X)$ on the same 256 held-out query rows. We report $\text{PEHE} = \{256^{-1} \sum_{i \in \text{query}} [\widehat{Q}(X_i) - Q(X_i)]^2\}^{1/2}$.

Discussion. Figure D.4 shows the advantage of a well-specified proximal estimator: the 2SLS bridge basis contains the true outcome bridge. Its large fluctuations at low proxy information reflect the difficulty of estimating that bridge with weak instruments, rather than model misspecification. ProximalFM improves more smoothly as information increases and, at $n_{\text{ctx}} = 2\,048$, outperforms proxy-marginalized CausalPFN throughout the sweep and proxy-adjusted OLS in most settings. OLS is a favourable comparator here: the treatment effect is linear, and independence of X and (W, Z) makes proxy marginalisation a simple average over context rows, without conditional-neighbour estimation. As the proxies nearly reveal U, OLS catches up with and surpasses PROXIMALFM. These curves come from one paired simulation per setting.

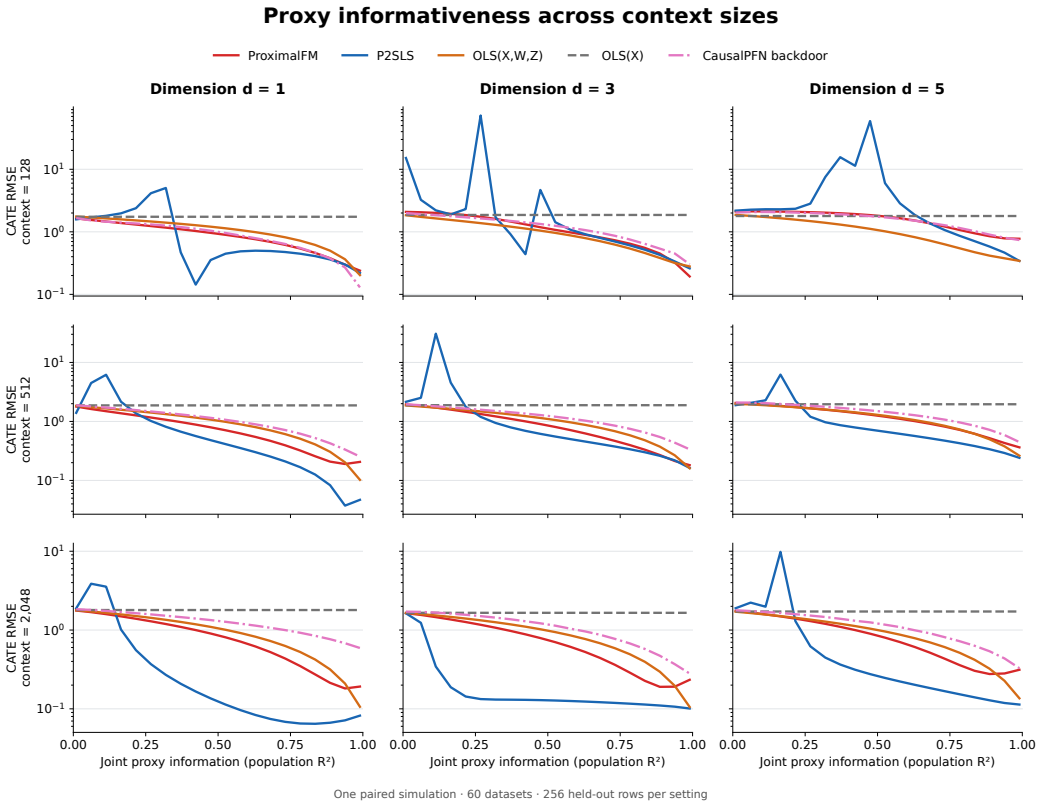


Figure D.4: CATE RMSE against joint proxy information, measured by the population R^2 for predicting U from (W, Z). Columns vary proxy dimension d; rows vary context size. All methods are evaluated on the same 256 held-out query rows at each setting. The vertical axis is logarithmic. Within each dimension, the underlying draws are shared across information levels; results come from one realization per setting.

E TOWARDS A REAL PHYSICAL SYSTEM: CAUSAL CHAMBERS

We use the Light Tunnel Mk2 on the Causal Chamber platform[6] (Gamella et al., 2025; Zeitler, 2025) to study average treatment effects in a controllable optical system. The experiment combines physical responses with software control to create dependence between a hidden confounder, proxy variables, treatment, and outcome. This provides an exploratory step towards proximal estimation in a real physical system, complementing the synthetic CATE benchmarks.

Physical setup and causal structure. Figure E.1 shows the intended pathways. The confounder U is an LED drive drawn uniformly from the integers $0, \dots, 255$, controlling its brightness. The observed covariate X controls a red LED, Z controls the ultraviolet LED 2, and W controls the position (in degrees) of the first polariser[7]. Treatment $A \in \{0, 1\}$ sets ultraviolet LED 3 to drive $a_1 A$, and the outcome Y is the infrared sensor 3 reading. Auxiliary measurements comprise the electrical current C drawn by the confounder LED, infrared readings I_1, I_2, and visible-light reading V_1.

The pathways $C \to X$, $I_1 \to Z$, $V_1 \to W$, and $I_2 \to A$—resulting from the software control functions—link successive physical measurements to subsequent interventions. In the working proximal structure, Z influences treatment assignment through I_2, while W influences the light reaching the outcome sensor. The green LED and crossed polarisers define the nominal weak-confounding setting. The blue LED and a second polariser fixed at zero degrees define the nominal strong-confounding setting.

Control functions and acquisition. Define

$$C_{[l,h]}(v) = \max\{l, \min(h, v)\}, \qquad S_{L,k,m,b}(v) = \frac{L}{1 + \exp\{-k(v - m)\}} + b.$$

For each observation, we set the selected confounder LED to U and measure C. We then set

$$X = C_{[0,255]}\{\text{round}(f_1(C))\}$$

and measure I_1, V_1. Independent standard-normal draws ϵ_Z, ϵ_W determine

$$Z = C_{[0,4095]}\{\text{round}(f_2(I_1) + k_Z \sigma_Z \epsilon_Z)\}, \tag{66}$$

$$W = C_{[-180,180]}\{f_4(V_1) + k_W \sigma_W \epsilon_W\}. \tag{67}$$

Polariser angles use degrees. In the weak setting, the second polariser is crossed with the first at $W + 90$, blocking most of the light reaching the third relative angle is zero. After setting these inputs, we measure I_2, assign treatment, and measure Y.

The linear family uses

$$f_1(c) = C_{[0,255]}(-0.085c - 1445), \qquad f_2(i) = C_{[0,4095]}(0.0624i).$$

The nonlinear family uses

$$f_1(c) = S_{255,0.002,-19000,0}(c), \qquad f_2(i) = S_{L,0.001,35000,0}(i),$$

with $L = 4095$ in the weak setting and $L = 1000$ in the strong setting. Every family uses $f_4(v) = S_{180,0.002,4000,-90}(v)$. Thus the family label specifies the form of f_1 and f_2. Figure E.2 shows their saturation, the common sigmoid polariser response, and the treatment-assignment rules. These mappings change which parts of the optical response the experiment explores. The reduced nonlinear amplitude in the strong setting limits the contribution of Z to the treatment-assignment sensor, preserving variation associated with the confounder.

Treatment and proxy-information settings. We collect 1 000 under the following experimental design. We have a Bernoulli assignment as follows

$$p(I_2) = \text{sigmoid}\{(c - I_2)/s_A\}, \qquad A = \mathbf{1}\{V_A < p(I_2)\}, \qquad V_A \sim \text{Unif}(0, 1),$$

[6] See `https://causalchamber.ai` for more details.

[7] For a complete diagram of the physical setup, see `https://docs.causalchamber.ai/the-chambers/light-tunnel-mk2`.

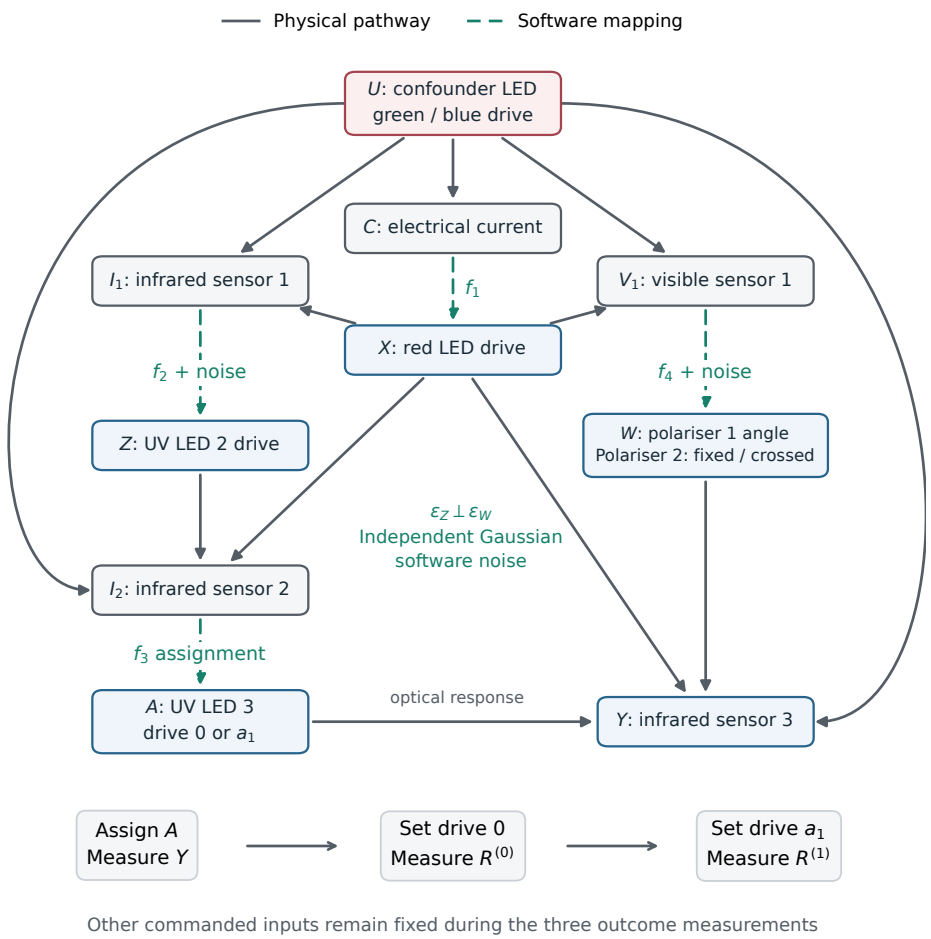


Figure E.1: Physical and software pathways in the optical experiment. Solid arrows represent the predetermined physical pathways and dashed green arrows represent effects resulting from the software control functions, i.e., *software mappings*. Independent Gaussian noise enters the two proxy mappings. The lower sequence shows acquisition of the factual outcome and the two intervention measurements. The graph describes the intended within-observation structure. Its causal interpretation requires stable device state and the proxy restrictions discussed in the text.

where $(c, s_A) = (5250, 800)$ for both linear settings, $(8789, 2000)$ for weak nonlinear, and $(3930, 800)$ for strong nonlinear. The Bernoulli rule supplies assignment variability at a given I_2. Its probabilities lie between 0.05 and 0.95 for 82.4%–99.7% of observations across the corresponding experimental conditions.

The noise multipliers (k_Z, k_W) equal $(0, 0)$ at high nominal informativeness, $(0.75, 0.75)$ at intermediate informativeness, and $(2, 2)$ at low informativeness. The strong-linear low-information condition uses $(0.6, 2.5)$ to retain a measurable confounding signal while reducing proxy information. The fixed noise scales are

Setting	σ_Z	σ_W (degrees)
Weak linear	325	62.2
Weak nonlinear	1789	64.5
Strong linear	383	66.2
Strong nonlinear	412	62.6

We quantify realised information by the R^2 of a linear regression of U on (Z, W) with an intercept. Confounding is summarised by $\mathbb{E}_n[R^{(0)} \mid A = 1] - \mathbb{E}_n[R^{(0)} \mid A = 0]$. Because the proxies are physical actuators, their noise also changes the treatment and outcome distributions. Each

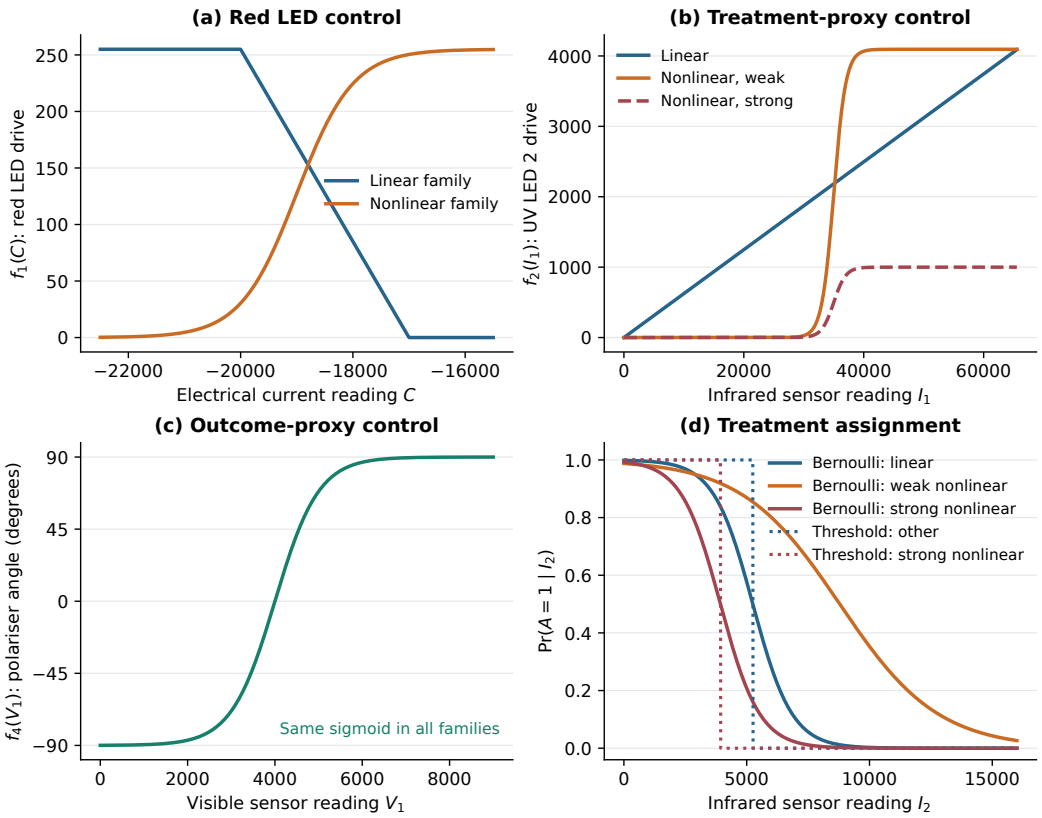


Figure E.2: Software control functions for the physical experiment. Panels (a)–(c) show the deterministic mappings before addition of proxy noise and final actuator clipping. Panel (d) shows the treatment probabilities for Bernoulli assignment and the corresponding hard-threshold protocols. The plots display the specified control functions over illustrative sensor ranges.

noise level therefore defines a distinct operating distribution. NumPy random streams use seeds 981264976135, 981264976142, and 981264976146 for U, proxy noises, and treatment uniforms, respectively. Restarting these streams for each condition pairs the commanded inputs and software randomness, while sensor values come from fresh physical measurements.

ATE reference and interpretation. After measuring Y, we apply drive zero and then drive a_1, retaining the other commanded inputs, to obtain $R^{(0)}$ and $R^{(1)}$. The empirical average intervention reference is

$$\widehat{\theta}_{\text{ref}} = \frac{1}{n}\sum_{i=1}^{n}\{R_i^{(1)} - R_i^{(0)}\}.$$

The sequence in Figure E.1 makes this average directly measurable over the sampled operating states. Its causal interpretation assumes stability during each sequence and negligible systematic effects of measurement order or carryover.

The mean contrast ranges from approximately 886 to 893 sensor counts at drive 500, and from 621 to 624 at drive 350. Across conditions, the standard deviation of individual contrasts is 3.55–65.64 counts. The root mean squared difference between Y and the separately remeasured factual arm is 3.10–65.00 counts. The overlap in these scales supports evaluating the average contrast, with individual contrasts interpreted in the presence of measurement variability. The physical estimand we obtain here is the ATE over the acquired states. The synthetic benchmarks in the other sections provide complementary evaluations against analytic CATE functions.

Adaptation of methods for ATE estimation. For each context prefix $\mathcal{D}_n = \{(X_i, W_i, Z_i, A_i, Y_i)\}_{i=1}^n$, we estimate the empirical ATE over the acquired operating states,

$$\theta_n = \frac{1}{n} \sum_{i=1}^{n} \{Y_i(1) - Y_i(0)\}.$$

All methods are fitted using the factual observations in $\mathcal{D}_n$ only. The paired intervention measurements $Y_i(0)$ and $Y_i(1)$ are reserved exclusively for the reference ATE. When a method produces a conditional effect estimate, we average its predictions over the same context prefix.

We consider a selection of methods from Appendix D.1. `X-learner (TabICL)` fits arm-specific outcome and effect models using (X, W, Z) and returns an estimated contrast at each observed (X_i, W_i, Z_i); its ATE estimate is the mean of these contrasts. `CausalPFN` directly predicts a conditional treatment effect from the factual context, which we likewise average over the observed covariate and proxy values.

The proximal methods use the proxy roles explicitly. `P-learner` estimates outcome and treatment bridge functions and forms cross-fitted doubly robust proximal scores; its ATE estimate is the mean of these scores. `KPV` and `PMMR` estimate an outcome bridge $h(a, w, x)$ and obtain a marginal ATE by integrating the contrast $h(1, W, X) - h(0, W, X)$ over the empirical joint distribution of (X, W) in the fitted context. `NMMR` uses the same empirical target, averaging its learned bridge contrast at the observed (X_i, W_i) pairs. `P2SLS` uses a linear two-stage specification: it first predicts W from (A, X, Z) and then regresses Y on $(A, X, \widehat{W})$; the fitted coefficient of A is its scalar ATE estimate.

Finally, PROXIMALFM treats the factual rows as its in-context dataset and predicts $\widehat{\tau}(X_i)$ for every context covariate value, with query-time proxies hidden. Its ATE estimate is the empirical mean $n^{-1} \sum_i \widehat{\tau}(X_i)$. This use of PROXIMALFM is not its primary design target: it uses (W, Z) to interpret the factual context but predicts effects from X alone, deliberately discarding query-time proxy information. It can therefore provide a useful ATE estimate by averaging its CATE predictions, but it is not designed to estimate the ATE directly or to integrate an outcome bridge over the empirical proxy distribution.

Results Figure E.3 shows no single method that performs best in every physical setting. The `X-learner (TabICL)` is the most consistently accurate method, especially when confounding is weak. `CausalPFN` is also competitive, particularly for nonlinear mechanisms and weak confounding. In contrast, the proximal methods, including PROXIMALFM, perform well only in some settings, and larger contexts do not lead to systematic improvements. Overall, these results illustrate the difficulty and variability of causal estimation in real physical systems.

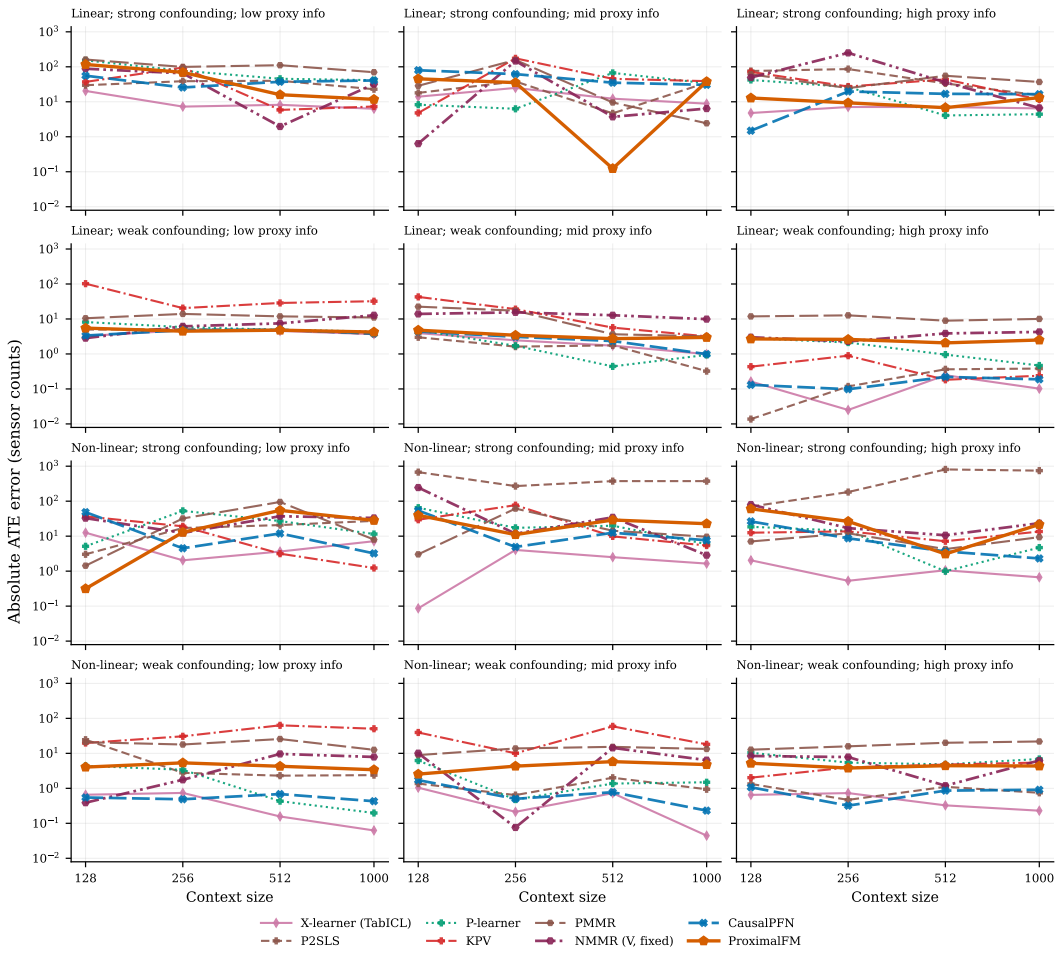


Figure E.3: Absolute error of estimated average treatment effects in the Causal Chambers Light Tunnel experiment as the factual context grows. Each panel is one physical operating condition, crossing control-function family (linear or non-linear), confounding strength, and nominal proxy informativeness. Curves show $|\widehat{\theta}_n - \widehat{\theta}_{\mathrm{ref},n}|$ in raw infrared-sensor counts, where both the estimate and the paired-intervention reference are evaluated on the same nested context prefix of size n. Both axes use logarithmic scales; lower is better.