2-Step Agent: How a Bayesian Decision Maker Learns from AI-Decision Support

Otto Nyberg[*1,2] **Davide Tugnoli**[*3] **Fausto Carcassi**[*4]

Giovanni Cinà[*1,2,4]

[1] Department of Medical Informatics, Amsterdam UMC, University of Amsterdam, the Netherlands
[2] Amsterdam Public Health Research Institute, the Netherlands
[3] Department of Mathematics, Informatics and Geosciences, University of Trieste, Italy
[4] Institute for Logic, Language and Computation, University of Amsterdam, the Netherlands
Correspondence to: Otto Nyberg <o.e.nyberg@amsterdamumc.nl>

Abstract

Prognostic predictions from ML models support human decision making in several fields, including high-stakes ones such as healthcare and the judiciary. Yet, we still lack a clear understanding of how decision makers learn from ML-based decision support (ML-DS). In this paper, we introduce a general computational framework, *the 2-Step Agent*, to capture how a prediction from an ML model can also be used as a source of useful information about the training data and the causal mechanism that generated it. Our framework models (i) how a prediction from an ML model affects the beliefs of a rational Bayesian agent, and (ii) how this change in beliefs affects the estimation of the causal effect, the downstream decision for a new observation, and the subsequent outcome. In addition to the framework itself, we make three contributions. First, for the multivariate linear-Gaussian setting, we derive a tractable solution for the Bayesian inference problem introduced, i.e. one in which the agent infers information about a population from an ML prediction. Second, we experimentally identify conditions under which ML-DS is beneficial compared to unassisted decision making. Third, we show that a single misaligned prior belief of the user can be sufficient for ML-DS to lead to worse downstream outcomes compared to no decision support *even when the prognostic model is well-specified and the agent is perfectly rational*. Hence, even under ideal conditions, ML-DS can do more harm than good.

1 Introduction

Background. Prediction models are often used to support decision making in the following fashion: a model trained on past data produces a forecast for a future outcome of interest, which an agent takes into account when deciding how to act. This is sometimes called 'predictive optimization' [Wang et al., 2024]. In contrast to full automation, in high-risk settings such an approach is sometimes perceived as a way to leverage the power of machine learning while retaining human oversight.[1] Our work focuses on models that are *prognostic* (i.e. the prediction is about a future state that can be influenced by the agent's actions) and *treatment-naive* (i.e. the action variable is not among the model's predictors). We refer to such decision support as Machine Learning Decision Support (ML-DS) henceforth.

[*] Equal contribution

[1] See for instance Article 14 of the EU AI Act [European Parliament and Council of the European Union, 2024].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Understanding the impact of adopting ML-DS is essential for at least two reasons. First, ML-DS is on the rise and has been deployed in a variety of fields including healthcare [Escobar et al., 2020], education [Smith et al., 2012], policing [Mohler et al., 2015], the judicial system [Imai et al., 2023], and child welfare [Chouldechova et al., 2018]. Second, allowing prediction models to influence decision making carries important risks, e.g. the incorporation of biased advice [Obermeyer et al., 2019]. Despite this, how agents incorporate ML predictions into their model of the world and how this influences their actions has remained underexplored. We next present two scenarios where ML-DS interacts with users' priors in subtle ways.

Motivating example 1: Effect of the agent's prior beliefs. Consider the situation of a clinician using ML-DS when deciding how to treat a cancer patient drawn from the same population the ML-DS model was trained on. The ML-DS model predicts that the patient has only two months left to live. We can imagine two lines of thinking, depending on the clinician's prior beliefs about the training data. If the model was trained on data from patients that received minimal treatment, a very heavy treatment may extend the survival time. On the other hand, if the model was trained on data from patients that received the heaviest treatment, it is best to put the patient on palliative care to avoid unnecessary suffering.

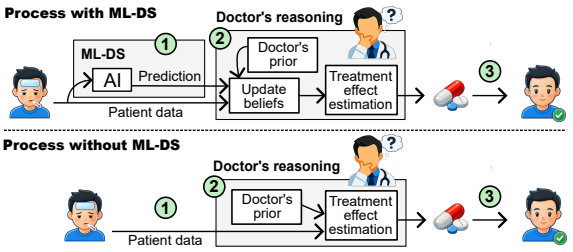


Figure 1: The flow of the framework for a clinical use case, with and without ML-DS. A new patient arrives for treatment, and the patient's data (1), potentially with the output of ML-DS, is observed by the agent, e.g. a doctor. Next, the agent estimates the effect of treatment and makes a treatment decision (2). Without ML-DS, this estimation is influenced by the agent's priors and patient data. With ML-DS, the agent uses these plus the ML prediction to update their beliefs about the population in a Bayesian fashion. The treatment and the patient's attributes determine the outcome (3).

This example shows that in the presence of the same information from ML-DS, *different prior assumptions about the model and the data it was trained on can lead to opposite actions*. This underscores the importance of an agent's priors in determining the effectiveness of ML-DS,[2] a point that was argued for in previous literature (see e.g., van Geloven et al. [2025]) but has not been studied quantitatively.

Motivating example 2: Agent learning from ML-DS. Consider a variation of the previous scenario involving a patient with a similar prognosis. Before seeing the model's prediction, the clinician believed (1) that similar patients ended up surviving 6 to 12 months when treated, and (2) that the training data of the model was from treated patients. Seeing a prediction of two months of survival, this set of beliefs is challenged. The clinician can then revise their beliefs in at least two different ways. First, they can conclude the training data in fact included some untreated patients, and that the current patient might be similar to one of the untreated patients. In this case, treatment might still be beneficial. Second, they might conclude they were wrong about the effect of the treatment. In this case, the treatment must be less beneficial than previously thought by the clinician. Therefore it might be best to abstain from treatment. Hence *the way in which ML-DS changes the agent's beliefs will influence the action of the agent*. Capturing this phenomenon means solving a challenging learning problem where the ML prediction is used as proxy information about the causal process the agent wants to intervene on.

Contributions. These examples showcase the importance of an agent's belief-updating process and prior beliefs for the impact of ML-DS. In this paper, we present a computational framework for ML-DS, *the 2-Step Agent*, which includes a Bayesian step to incorporate the prediction by the ML model into the agent's model of the world, and a causal inference step in which the agent draws the conclusion on which action is best to take. This is schematically depicted in Fig. 1 (top half) for the clinical example. In contrast to standard prediction-then-decision approaches, the distinctive feature

[2]Setting the agents' prior beliefs by proper training is indeed an important part of RCT protocols that is not always adhered to [Perepletchikova et al., 2007]. When it is not possible to create a common baseline of beliefs and behavior, a two-stage randomization might be required, both on the providers of the intervention and the subjects [Borm et al., 2005].

of our framework is that the agent treats the ML-DS prediction as a source of information about the training population and the latent causal mechanism, and uses this to update their beliefs about treatment effects before taking an action. Such an update is a challenging inference through a set of latent exchangeable variables in the predictive model's training data.

We make three further contributions. First, given the difficulty of solving the Bayesian inference problem we introduced, we prove that for the multivariate linear-Gaussian setting with a linear regression ML-DS model, it is possible to reduce the exchangeable variables to two auxiliary random variables, reducing computational cost and improving numerical stability. Second, we identify experimentally some settings in which learning from ML-DS leads to better effect estimates and better outcomes compared to not using ML-DS. Finally, we show that, even if the agent is fully rational and the model is well-specified, a single misaligned prior can be sufficient to derail learning from ML-DS and make the agent take suboptimal actions, resulting in worse outcomes.

While our running example is medical, the results are general and apply to every field where treatment-naive prognostic prediction models are used for decision support. Our simulations are only concerned with learning from non-causal prediction models. ML-DS with causal models is left for future work. The code and experiments are available open-source at `https://github.com/thelogicalgrammar/CapturingUsersPredictionModels`.

2 The 2-Step Agent framework

In this section, we give a formal description of the 2-Step Agent. We introduce the process that generates the training data for the predictive model in Sec. 2.1. We then formalize the first step consisting of a Bayesian agent updating their beliefs about the world given a prediction from the ML-DS model for a new patient, in Sec. 2.2. Sec. 2.3 defines the second step, where the Bayesian agent uses the updated beliefs about the population to decide on a treatment for the new patient. In Sec. 2.4 we combine these steps to determine the effect of introducing ML-DS, comparing outcomes with and without decision support. Sec. 2.5 reports our tractability result.

2.1 The setup: Historical data and predictive model

We first define a causal model of the domain. Let A be a variable denoting an intervention, such as the administration of a treatment. The treatment variable may be continuous or categorical, but to simplify the notation, we will present the formulas with two options for A. Let Y be a variable denoting a future outcome of interest affected by A. In our example, the treatment could be a fixed dose of chemotherapy and the outcome the months of survival, so in this case a higher outcome is better. We use $X \in \mathbb{R}^k$ to refer to the variables involved in the causal mechanisms governing the treatment and outcome.

Definition 2.1 (Historical data generation)**.** We assume $Hist = (\mathbf{S}, P_{\mathbf{N}})$ to be the minimal structural causal model (SCM) describing the causal mechanisms producing A and Y, where $\mathbf{S}$ is a set of assignments defining each variable in terms of its parents and $P_{\mathbf{N}}$ is a joint distribution over the noise variables. We use X to denote endogenous (non-noise) variables involved in the SCM that are different from A and Y.

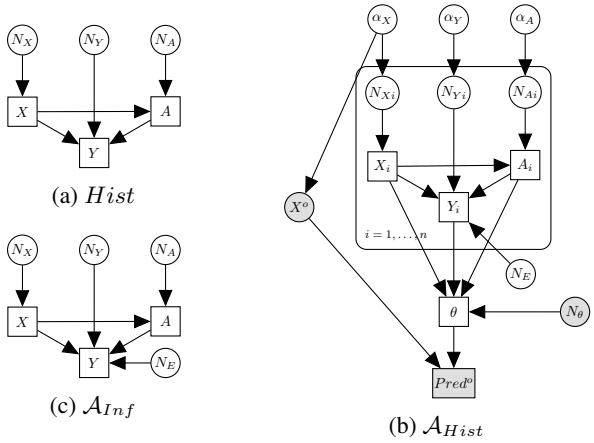
An example of this SCM is depicted in Fig. 2(a). The SCM $Hist$ stands for the process in the world that generated the training data. Note that the assignment determining A might be due to a specific intervention protocol, e.g., the guidelines for administering the treatment that were in use in that particular time window. The SCM determines a joint distribution P^{Hist} over variables Y, A, X.

Definition 2.2 (Training data and prediction model)**.** The dataset $\mathcal{D}_{Hist} = \{(x_1, a_1, y_1), \ldots, (x_n, a_n, y_n)\}$ that is generated from P^{Hist} is composed of tuples containing relevant characteristics, treatment, and outcome. The dataset contains n instances. This dataset is used to train a prediction model $f : X \to Y$ to estimate $\mathbb{E}^{Hist}[Y \mid X]$.

We call f *treatment-naive* since A is not among its inputs, and *well-specified* if its model class contains $\mathbb{E}^{Hist}[Y \mid X]$. Our framework does not currently extend to predictions of the treatment-dependent estimate $\mathbb{E}^{Hist}[Y \mid A, X]$ or the interventional estimate $\mathbb{E}^{Hist}[Y \mid do(A), X]$; these are left for future work.

2.2 The first step: Bayesian update of the agent's beliefs

We next introduce an agent $\mathcal{A}$ who decides on the intervention for new patients with the help of ML-DS. We assume this agent does not observe the training data or the model directly. This agent uses their own model of the world, capturing not only what they believe about the data-generating process $Hist$, but also what they believe concerning the prediction model f and the training data $\mathcal{D}_{Hist}$.[3] The agent updates their world model using the ML-DS prediction, exploiting the latter as a piece of compressed information about the causal mechanism. We model an agent who is aware of the model documentation containing the model *class*, the model *signature* (treatment-naive and included covariates), and the size of the training data.



(a) $Hist$

(c) $\mathcal{A}_{Inf}$

(b) $\mathcal{A}_{Hist}$

Figure 2: The graphical models of the framework with stochastic variables depicted as circles and deterministic variables as rectangles. Observed variables are shaded, unobserved ones are unshaded. Model (a) represents the true data-generating process, model (b) represents the agent's beliefs of how the ML-DS model was trained, and model (c) allows the agent to infer the treatment effect of new data points given the beliefs.

Definition 2.3 (Beliefs of the agent)**.** The Bayesian network defining the agent's joint distribution is denoted $\mathcal{A}_{Hist}$ (Fig. 2(b)). The distribution is over the following variables:

1. $\alpha_X, \alpha_Y, \alpha_A$: These are the variables fixing the population-level distributions, e.g. over the patient characteristics (α_X), the noise in the outcome (α_Y), and the noise in the treatment (α_A). Depending on how population-level uncertainty is parameterized, these can be vectors, for instance α_Y could be a pair of mean and variance if the noise on Y is Gaussian.

2. N_{Xi}, N_{Yi}, N_{Ai}: Within the Bayesian network, the historical data is represented via a plate. The α variables are outside the plate, capturing the fact that they are population-level distributions. N_{Xi}, N_{Yi}, N_{Ai} (with $i = 1, \ldots, n$) are historical samples from these population-level distributions, which $\mathcal{A}$ cannot observe directly. These encode the stochastic components of each patient that the predictive model was trained on.

3. A_i, Y_i, X_i: The stochastic components N_{Xi}, N_{Yi}, N_{Ai} are transformed deterministically into the specific patient characteristic (X_i), treatment (A_i) and outcome (Y_i), which become a training data point for the predictive model. All variables Y_i share the noise term N_E.

4. N_E: This variable denotes the treatment effect. Contrary to the other noise variables, this is not sampled for each plate repetition but only once, so it remains the same across the plate.

5. θ: The prediction model's parameters, obtained deterministically from the training data in the plate and possibly some additional known parameter N_θ which encodes e.g. the hyperparameter settings or initialization values. θ has as parents all the non-noise variables in the plate, representing the fact that it is trained on the whole data set.

6. X^o: Observed variables for a new patient, for which the agent will make a treatment decision. This vector is sampled from the same population distribution as X (for simplicity we skip the noise variable since it is identical to X^o).

7. $Pred^o$: The observed prediction for X^o. This is derived deterministically from X^o and θ.

[3] Recent guidelines ask authors to "give details of any treatments received, and how they were handled during model development or evaluation" [Collins et al., 2024], which would limit the agent's uncertainty about this aspect of the training data. Our framework can include such settings by restricting the uncertainty on this dimension.

The Bayesian network is shown in Fig. 2(b). To simplify presentation we depict the submodel of $\mathcal{A}_{Hist}$ in the plate as identical to $Hist$, meaning that the agent knows the DAG generating the data. This does not have to be the case and the framework is general enough to allow these two structures to diverge.

Definition 2.4 (Agent's priors)**.** The agent is assumed to have as prior a joint distribution over all variables in the model, which depends on the unconditional distribution $P^{\mathcal{A}_{Hist}}(\alpha_X, \alpha_Y, \alpha_A, N_E, N_\theta)$.

The agent's prior beliefs about the data- and model-generating processes are thus encoded in $\mathcal{A}_{Hist}$ and the parameter sets α_X, α_Y, α_A, N_E, and N_θ. The set α_X contains all parameters necessary for the noise distribution of N_X etc. We assume the agent has no uncertainty on N_θ since they have access to the model documentation, and therefore we omit this prior.

Definition 2.5 (Agent belief update)**.** Given X^o and $Pred^o$ the agent does a Bayesian update on the prior $P^{\mathcal{A}_{Hist}}(\alpha_X, \alpha_Y, \alpha_A, N_E)$, obtaining a posterior distribution $P^{\mathcal{A}_{Hist}}(\alpha_X, \alpha_Y, \alpha_A, N_E \mid X^o, Pred^o)$. To simplify exposition, we marginalize over uncertainty in the plate variables $(N_{Xi}, N_{Yi}, N_{Ai}, X_i, Y_i, A_i$ with $1 \leq i \leq n)$ and θ.

The uncertainty on N_E encodes the agent's beliefs concerning the effect of the intervention. By letting the mechanism be influenced by N_E, we can treat the posterior as encoding a distribution over SCMs, each with its own fixed assignment for Y (and thus each with its own treatment effect). Other uncertainties over the SCM's parameters can be added as variables outside the plate like N_E; we leave these out to simplify exposition.

2.3 The second step: deciding on the best action after belief update

In the second step, the agent estimates the effect of the treatment on the new patient using their updated beliefs from step 1.

Definition 2.6 (Agent model for treatment effect estimation, $\mathcal{A}_{Inf}$)**.** This model contains variables $N_E, N_X, N_A, N_Y, X, A, Y$, arranged as the similarly-named variables in the sub-model of $\mathcal{A}_{Hist}$. This model is used for inference on new instances and we denote it as $\mathcal{A}_{Inf}$.

This second agent model is depicted in Fig. 2(c). The crucial difference between this model and a regular SCM is that *the distributional parameters of the noise variables are* not *assumed to be independent but rather they are initialized with the joint posterior* obtained in the Bayesian update. The agent is thus drawing an inference over a distribution of possible SCMs.

Definition 2.7 (Decision rule)**.** To decide on treatment, the agent sets the joint distribution of the inference model drawing from the posterior learned in the previous step:

$$\psi \sim P^{\mathcal{A}_{Hist}}(\alpha_X, \alpha_Y, \alpha_A, N_E \mid X^o, Pred^o) := \Pi$$

Each realization of ψ fixes the distribution of the noises and structural equations, specifying a different SCM $\mathcal{A}^{\psi}_{Inf}$. Following this, the treatment a_1 is given if the Conditional Average Treatment Effect (CATE) for the new patient provides enough benefit over treatment a_2 in expectation across SCMs, i.e. $\mathbb{E}^{\Pi}[\text{CATE}^{\psi}] > \tau$ where τ is a threshold set by the agent and CATE^{ψ} is calculated via interventional expectations in $\mathcal{A}^{\psi}_{Inf}$

$$\text{CATE}^{\psi} = \mathbb{E}^{\mathcal{A}^{\psi}_{Inf}}\big[Y \mid X = X^o, do(A = a_1)\big] - \mathbb{E}^{\mathcal{A}^{\psi}_{Inf}}\big[Y \mid X = X^o, do(A = a_2)\big].$$

2.4 Quantifying the effect of introducing decision support

Given this machinery, for a given ML-DS model and a set of agent priors, the treatment A prescribed by the agent for a new observation X^o is deterministic both with and without ML-DS. In the former case (Fig. 1 top) the agent learns from ML-DS and chooses A according to the updated beliefs. In the latter case, the agent does not change beliefs and the treatment effect is estimated based on the prior beliefs (Fig. 1 bottom). We call the two treatments chosen with and without ML-DS a^{ML} and $a^{\neg ML}$ respectively. The effect of ML-DS is then defined as

$$\mathbb{E}^{Hist}[Y \mid X = X^o, do(A = a^{ML})] - \mathbb{E}^{Hist}[Y \mid X = X^o, do(A = a^{\neg ML})]$$

namely as a difference of expected outcomes obtained by setting the actions to those chosen with and without ML-DS, both for the new patient $X = X^o$ and under the true generating process $Hist$. In the experiments we report these two expected outcomes separately as distributions across simulation runs.

2.5 Theoretical results on the tractability of the Bayesian inference

This framework is very general and allows for the simulation of many interesting learning dynamics. Yet, its implementation faces a tractability problem: approximate Bayesian inference struggles to propagate through a plate of completely exchangeable variables. We prove that this problem can in simple settings be solved by replacing the plate with two auxiliary random variables.

Theorem 2.8 (Informal)**.** *Let $Hist$ be as depicted in Fig. 2(a), where X is a vector of k independent covariates. Assume that the structural equations are linear, $n \geq k+1$, all variables are continuous, all noises are Gaussian, the agent knows the DAG and the ML-DS model is fitted by ordinary least squares with an intercept. Then, conditional on the population-level variables (the variables outside the plate), for a new patient independent of the training data, the likelihood of a single prediction given $X^o = x$ has an exact representation using two scalar auxiliary random variables.*

We refer the reader to Theorem F.2 in Appendix F for the formal statement and the full proof. Corollary F.3 gives the corresponding plate reduction for the no-intercept (misspecified) model, used as a robustness check. By collapsing the plate, the Bayesian update becomes tractable and we can compute the updated beliefs of the agent over the population-level parameters given the prediction obtained from the ML-DS model. This unlocks the possibility to implement the framework (albeit in a simple setting): we can fully simulate what happens with and without decision support for a given agent $\mathcal{A}$ and an ML-DS model f following the recipe outlined in the previous section.

3 Experimental setup

The 2-Step Agent allows us to simulate the consequences of providing decision support. Using this framework, we investigate *how and how much decision support influences (i) agent's beliefs, (ii) treatment decisions, and (iii) downstream patient outcomes.* We run a series of simulations to see how beliefs, treatments, and outcomes are affected when a single prior is misaligned. We simulate all settings both with and without ML-DS to see the differences induced by ML-DS.

Case: univariate regression. We study what happens in the simple setting when the ML-DS model is a well-specified linear regression model f and the data-generating process is linear with one continuous covariate X, a continuous intervention A, and a continuous outcome Y. Higher values of Y correspond to better outcomes. One can think of the motivating example and interpret X as body weight, A as dosage of chemotherapy, and Y as months of survival. We use $n = 1000$. Despite the continuous treatment, we assume the agent compares two treatment options, $A = 10$ and $A = 20$ (intuitively ten and twenty dosage units). Let $Hist$ have the DAG depicted in Fig. 2(a) with a single X variable.

The model of the agent's beliefs, $\mathcal{A}_{Hist}$. The model of the agent $\mathcal{A}_{Hist}$ encodes the agent's uncertainty about *Hist*:

$$\left.\begin{aligned} N_{Xi} &\sim \mathcal{N}(\boldsymbol{\alpha_{X\mu}}, \boldsymbol{\alpha^2_{X\sigma}}) \\ N_{Ai} &\sim \mathcal{N}(\boldsymbol{\alpha_{A\mu}}, \boldsymbol{\alpha^2_{A\sigma}}) \\ N_{Yi} &\sim \mathcal{N}(\boldsymbol{\alpha_{Y\mu}}, \boldsymbol{\alpha^2_{Y\sigma}}) \\ X_i &= N_{Xi} \\ A_i &= 0.125\, X_i + N_{Ai} \\ Y_i &= 12 - 0.1\, X_i + \boldsymbol{N_E}\, A_i + N_{Yi} \end{aligned}\right\}_{\text{i=1,...,n}} \quad \left| \begin{aligned} \theta_0, \theta_1 &= \underset{\theta_0,\theta_1}{\operatorname{argmin}} \Big(\sum_{i=1}^{n} (\theta_0 + \theta_1 X_i - Y_i)^2\Big) \\ X^o &\sim \mathcal{N}(\boldsymbol{\alpha_{X\mu}}, \boldsymbol{\alpha^2_{X\sigma}}) \\ Pred^o &= \theta_0 + \theta_1 X^o \end{aligned}\right.$$

where θ_0 and θ_1 are the intercept and slope of a univariate regression model, and the variables in bold font are the agent's priors for the population-level parameters.

The agent has uncertainty on several dimensions. For instance, $\alpha_{A\mu}$ encodes the belief of the agent on how much historical dosages of treatment deviated from the standard protocol, $\alpha_{X\mu}$ and $\alpha_{X\sigma}$ encode the belief of the agent on how the feature X of the training data was distributed, and so forth. The distribution of N_E captures the belief of the agent about the effectiveness of treatment A. Note that in this scenario the agent knows some of the SCM's parameters; this is for ease of exposition. Those parameters could also be replaced with random variables that the agent has uncertainty over (like N_E); the plate-reduction result covers this case under the assumptions stated in Appendix F.

Defining the agent. We define a set of default priors with limited uncertainty to represent an agent that is "correct" in their beliefs.

$$\alpha_{X\mu} \sim \mathcal{N}(80, 0.1^2) \qquad \alpha_{X\sigma} \sim \mathcal{N}(10, 0.1^2) \mid \alpha_{X\sigma} \geq 0$$
$$\alpha_{A\mu} \sim \mathcal{N}(2, 1^2) \qquad \alpha_{A\sigma} \sim \mathcal{N}(1, 0.1^2) \mid \alpha_{A\sigma} \geq 0$$
$$\alpha_{Y\mu} \sim \mathcal{N}(0, 0.01^2) \qquad \alpha_{Y\sigma} \sim \mathcal{N}(0.1, 0.01^2) \mid \alpha_{Y\sigma} \geq 0$$
$$N_E \sim \mathcal{N}(1, 1^2)$$

This agent essentially has the correct beliefs about the data-generating distribution, the historical treatment policy, outcome variance, and treatment effect. They only have some uncertainty represented by some variance on these parameters. In our experiments, we vary each of the parameters colored in red, i.e. the prior means of $\alpha_{X\mu}$, $\alpha_{A\mu}$, $\alpha_{Y\mu}$, and N_E - one at a time - to simulate various ways in which the agent's beliefs may be incorrect. This only introduces misalignment in a mean of the agent's beliefs without affecting the beliefs on variability.

Reaching a decision. Finally, the model of the agent used for inference, $\mathcal{A}_{Inf}$, is defined as in Fig. 2(c). The agent applies the decision rule in Definition 2.7 with $a_1 = 20$, $a_2 = 10$, and $\tau = 5$. Hence, the agent administers a dosage of 20 if the posterior expectation of CATE exceeds 5 months, and a dosage of 10 otherwise.

This completes the decision pipeline: given a patient X^o with a prediction $Pred^o$ and the agent priors, we compute what the agent would do. For details of the implementation, see Appendix B. The same experiments are extended to a multivariate setup with k confounders, as well as to a univariate setup where the ML-DS model is a linear model without intercept (and is therefore *not* well-specified); full details are in Appendices E and D respectively.

4 Results

We begin by mentioning that our framework can capture the two motivating examples presented in the Introduction. What is clarified by this experiment is that the *learning from ML-DS manifests itself through a correlation in the posterior of the agent*: the posterior correlation encodes the fact that the agent is entertaining multiple competing hypotheses to reconcile beliefs and observations. We expand on this in Appendix A.

Univariate experiments with intercept. We report the experimental results on two dimensions: 1) the agent's belief of treatment effect and 2) the expected downstream outcomes Y under the true generating process $Hist$ after the agent has taken an action, visualized in the top and bottom rows of Fig. 3. In each of these plots, what happens using ML-DS (orange series) is contrasted with what happens without (blue series) for different types of agents. The plots sweep a wide range of options for the priors in order to reveal interesting dynamics of the agent's beliefs.

The N_E column in Fig. 3 shows what happens to the belief about CATE and the actual outcome Y when the agent's belief about N_E is misaligned. As expected, in the top row we can see that without decision support the prior mean of CATE (blue) is the prior mean of N_E multiplied by the difference in treatment dosage. In the orange series, however, we see an interesting behavior: after learning from the interaction with ML-DS, the agent corrects their belief about CATE re-aligning it with the correct value. In other words, *when all other beliefs are correct, the agent can learn from ML-DS to obtain better CATE estimates* and take better actions. The bottom row confirms that the orange series shows superior outcomes.

The column of Fig. 3 pertaining to $\alpha_{A\mu}$ shows what happens for the agents with different prior beliefs about the treatment protocol. In this scenario the prior CATE estimates are constant because the agent has a fixed belief in N_E. When interacting with ML-DS, the belief update changes CATE estimates quite a lot. This is because the agent is attempting to explain the gap between $Pred^o$ and their prior belief about it. This is reflected in the outcomes in the boxplots, where the orange series is for some priors below the blue one. Thus, incorrect beliefs in $\alpha_{A\mu}$ can result in CATE estimates that lead to suboptimal actions and worse outcomes Y. A similar pattern is observed in the column labeled $\alpha_{Y\mu}$, concerning beliefs about the outcome distribution: while prior CATE and Y estimates are relatively stable without ML-DS, CATE is affected when ML-DS is introduced, although in this specific example incorrect beliefs about $\alpha_{Y\mu}$ need to be very extreme to swing decisions.

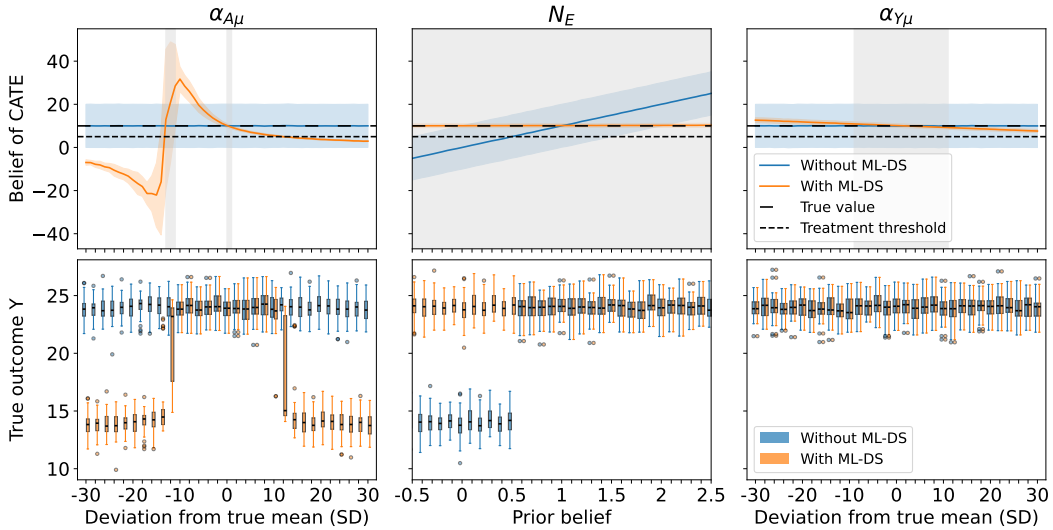


Figure 3: The effects of misaligned agent priors with ML-DS (in orange) and without (in blue) on the agent's beliefs (top row) and on the consequent downstream outcomes Y (bottom row). In the plots for $\alpha_{A\mu}$ and $\alpha_{Y\mu}$, the horizontal axes represent the deviation of the agent's prior belief from the true parameter in units of standard deviation, and the centerpoints in all plots correspond to aligned priors. The orange and blue bands show the mean ± 1 SD of the agent's CATE distributions pooled across the 50 runs per prior setting, with and without ML-DS, respectively. Gray bands mark settings where the true CATE lies within the orange band. Boxplots summarize the 50 expected outcomes under the selected treatments and the true generating process $Hist$, showing the median and interquartile range (IQR), with whiskers extending to the most extreme values within 1.5 IQR of the box. The leftmost panel shows that misaligned beliefs can lead to ML-DS being harmful, while in the central panel ML-DS corrects misaligned beliefs and ends up being beneficial.

Univariate experiment without intercept. To explore what happens when the ML-DS model is misspecified, we ran an otherwise identical experiment, but with an ML-DS model without intercept. Fig. 6 shows that the same conclusions we discussed also hold without intercept. The column pertaining to N_E serves as an example for how our framework does not require the ML-DS model to be well-specified for useful learning to take place. Intuitively this is because the agent is merely using $Pred^o$ to learn about the generating process, rather than trying to incorporate this information in some more direct way. Fig. 4, however, displays a new failure mode: when the ML-DS model is not well-specified, misaligned beliefs about $\alpha_{X\mu}$ can also lead to harmful actions even with moderate misalignment. See Appendices D and F.14 for more details.

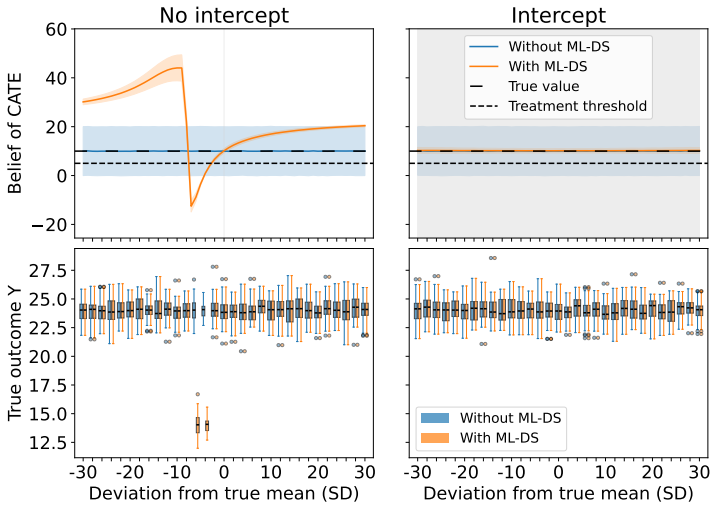


Figure 4: The effects of a misaligned agent prior $\alpha_{X\mu}$ about the distribution of covariate X, shown when the ML-DS model is misspecified (left) and well-specified (right), with bands and boxplots defined as in Fig. 3. The misspecified model introduces a new option for incorrect beliefs to result in harmful ML-DS.

Multivariate regression with intercept. Following the same pattern as for the well-specified univariate experiment, we ran an experiment where we varied the number of confounders k in X to see how it affects learning. We ran the experiment for $k \in \{2, 4, 8, 16, 32\}$, normalizing the confounders in a way that allowed us to isolate the effect of k from e.g. increased variance. More details and the results are reported in Appendix E. This experiment reproduced the qualitative conclusions of the main experiment for $k > 1$ and we found virtually no effect of k on learning.

5 Related work

A large body of literature is concerned with the analysis of ML-DS from different angles. A strand of literature focuses on how ML-DS is entangled with (changes in) the data-generating process. Works in this category study limitations of what can be learned from data generated from past policies [e.g. Lakkaraju et al., 2017] or the dynamics that occur with alternating cycles of ML-DS deployment and re-training, as in the performative prediction literature [Perdomo et al., 2020, Boeken et al., 2024]. Plecko and Bareinboim [2024] take a causal perspective on prediction-to-decision pipelines studying how biases in ML predictions amplify through downstream decisions. Their analysis treats the decision rule as a fixed mapping, where bias originates in the ML model. In contrast, our framework focuses on the decision maker, showing that a well-specified calibrated prediction paired with a misaligned prior can already be sufficient to worsen outcomes.

Another set of articles is concerned with the design and evaluation of ML-DS. Some lay out identifiability conditions for the effects of ML-DS on decisions [Imai et al., 2023], while others seek to optimize human-machine complementarity by design [Donahue et al., 2022, McLaughlin and Spiess, 2024]. Recent work in this direction studies prediction-set-based decision support [De Toni et al., 2024], or evaluates the counterfactual harm that such systems may cause [Straitouri et al., 2024]. Others evaluate the other side of the coin and ask what human input can add to artificial agents already reaching some level of optimality [Alur et al., 2024, Zhang and Bareinboim, 2022, Stensrud et al., 2024]. Guo et al. [2024] propose a decision-theoretic framework for measuring AI reliance, explicitly adding a Bayesian component to the decision making, though one whose beliefs concern only the reliability of the prediction. Our framework addresses a different mechanism from all of the above: we explicitly model how the agent learns about the data-generating process from the ML prediction, and show that this learning can make decision support harmful even under near-ideal conditions.

A related line of work asks when different notions of calibration suffice for robust downstream decisions [Foster and Vohra, 1997, Zhao et al., 2021, Kiyani et al., 2026]. These settings differ from ours in two crucial aspects. First, they are concerned with *diagnostic* problems where the prediction target Y is already set and not affected by actions (in contrast to our prognostic setup). Second, they assume a known payoff function for the available actions. We instead assume the user does *not* have such a function and wants to estimate treatment effect with the help of ML-DS. It follows that calibration cannot solve the problem we presented. Calibration concerns agreement between predictions and observed outcomes under the historical policy. Even a perfectly calibrated predictor equal to $\mathbb{E}^{Hist}[Y \mid X]$ does not by itself determine how outcomes would change under an intervention and can still lead to harmful decisions under the wrong priors (see the example in Appendix G).

Other papers study how information flows between agents influencing their actions. In Kamenica and Gentzkow [2011] a Sender of information influences a Receiver Bayesian agent in order to steer their action. What we add is that in our context the signal is an ML prediction produced from training data drawn from the same environment the agent wants to intervene on. This coupling yields the distinctive inference problem we study, i.e. updating through latent exchangeable training variables.

Another important strand of work focuses on inferring information about a population P given predictions from a model trained on samples from P in the context of confidentiality attacks. This literature studies whether it is possible to use models' predictions (and other behavior) to recover the presence of a specific data point in so-called *membership inference attacks* [Hu et al., 2022, Huang, 2025], or to infer statistical properties of training datasets (*property inference attacks*) [Ateniese et al., 2015, Wang et al., 2019]. In contrast, our goal is to directly infer information about the underlying population and about the causal mechanism itself, i.e. the treatment effect.

6 Discussion

We have presented an explicit framework unpacking the interaction between an agent and ML-DS. Our 2-Step Agent framework captures the effect of an agent's prior beliefs, modeling learning effects due to the interaction and tracking the agent's uncertainty. The framework allows for simulations comparing outcomes with and without ML-DS, quantifying the harm and benefit of ML-DS.

The framework defines a challenging Bayesian inference problem for which we derived a tractable solution in the linear-Gaussian setting. This result allowed us to simulate the end-to-end effects of decision support for different types of agents, making this the first quantitative study of harmful effects of ML-DS with misaligned beliefs. In addition, we found that i) an incorrect belief about treatment effect is in some settings rectified by the interaction with ML-DS; ii) incorrect beliefs about the ML-DS model's training data may lead to harm, confirming arguments put forward in previous literature [van Geloven et al., 2025]; and iii) incorrect beliefs about the distribution of covariates when the ML-DS model is not well-specified can lead to harm even in otherwise near-ideal conditions.

These findings highlight the sensitivity of ML-DS to prior beliefs when using a treatment-naive prognostic prediction model, and underscore the importance of providing full documentation and proper training to users of ML-DS. Where providers do not disclose such information, our results do not imply that ML-DS should be abandoned, but that its benefit cannot be taken for granted. Users' beliefs about the training data will determine whether it helps or harms. Finally, we remark that these results apply to *any* field deploying such systems and to any Bayesian agent interacting with a treatment-naive prognostic ML-DS, be it human or artificial.

Towards real-world use. The framework could be leveraged for pre-deployment audits, following these steps: (i) Elicit from prospective users, e.g. clinicians, priors over the quantities in $\mathcal{A}_{Hist}$: historical treatment policy, distributions of confounders, outcome and treatment effect. (ii) Simulate decisions and outcomes with and without ML-DS for each elicited prior. (iii) Identify users whose simulated outcomes worsen, and the specific belief responsible (as in Sec. 4). (iv) Adapt their training to fix said belief. This audit assumes approximately Bayesian updating and its validity for human users would need to be validated.

Limitations and future work. The experiments in this paper explored only part of the potential of the framework; they are limited to the multivariate linear-Gaussian case, with a treatment-naive ML-DS model, a single update and synthetic data. This was mostly due to the scope of Theorem 2.8; further work will investigate whether similar tractability results can be obtained for more complex cases.

We assumed the agent reasons in a Bayesian fashion. While this is not realistic with respect to human behavior, it is useful to make this assumption for three reasons. First, recent literature calls for formal computational cognitive models of the human decision maker as a necessary foundation for understanding human-AI complementarity, even when starting from a rational baseline [Steyvers and Kumar, 2024]. Second, this assumption is a necessary first step to isolate the learning mechanism: if we had harmful outcomes caused by an agent with limited rationality, we would not know if it is due to the learning mechanism or just bounded rationality. Finally, this assumption is also interesting when considering how an artificial agent might learn from predictions of other ML models. From another point of view, our results indicate that harmful interactions with ML-DS due to misaligned priors cannot be fixed by making the agent more rational.

Our theory describes what happens after a single interaction of an agent with ML-DS and does not yet cover repeated interactions. In future work this line of research will be expanded across different dimensions, including i) different types of agents (e.g. agent with incorrect model of the world or different kinds of uncertainties), ii) more complex ML models for ML-DS, iii) cases with heterogeneity of treatment effect, iv) repeated rounds of learning for the same agent and v) using causal models for ML-DS.

Acknowledgements

We would like to thank Wouter van Amsterdam, Nan van Geloven, Sagar N. Simha, Mohammad Azizmalayeri, and Juliette Ortholand for their feedback on draft versions of the paper. This publication was made possible through funding from the Sector Plan for Medicine and Health Sciences, supported by the Dutch Ministry of Education, Culture and Science (OCW). This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-16946. We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy). Davide Tugnoli's PhD scholarship at the University of Trieste is funded by InSilicoTrials Technologies S.p.A.

References

Oriol Abril-Pla, Virgile Andreani, Colin Carroll, Larry Dong, Christopher J. Fonnesbeck, Maxim Kochurov, Ravin Kumar, Junpeng Lao, Christian C. Luhmann, Osvaldo A. Martin, Michael Osthege, Ricardo Vieira, Thomas Wiecki, and Robert Zinkov. PyMC: A modern and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, 9:e1516, 2023. doi: 10.7717/peerj-cs.1516.

Rohan Alur, Manish Raghavan, and Devavrat Shah. Human expertise in algorithmic prediction. In *Advances in Neural Information Processing Systems*, volume 37, pages 138088–138129, 2024. doi: 10.52202/079017-4384.

Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015. doi: 10.1504/IJSN.2015.071829.

Philip Boeken, Onno Zoeter, and Joris Mooij. Evaluating and correcting performative effects of decision support systems via causal domain shift. In *Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 551–569. PMLR, 2024.

George F. Borm, René J. F. Melis, Steven Teerenstra, and Petronella G. Peer. Pseudo cluster randomization: a treatment allocation method to minimize contamination and selection bias. *Statistics in Medicine*, 24(23):3535–3547, 2005. doi: 10.1002/sim.2200.

Alexandra Chouldechova, Emily Putnam-Hornstein, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 134–148. PMLR, 2018.

Gary S. Collins, Karel G. M. Moons, Paula Dhiman, Richard D. Riley, Andrew L. Beam, Ben Van Calster, Marzyeh Ghassemi, Xiaoxuan Liu, Johannes B. Reitsma, Maarten Van Smeden, et al. TRIPOD+AI statement: updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ*, 385:e078378, 2024. doi: 10.1136/bmj-2023-078378.

Giovanni De Toni, Nastaran Okati, Suhas Thejaswi, Eleni Straitouri, and Manuel Gomez-Rodriguez. Towards human-AI complementarity with prediction sets. In *Advances in Neural Information Processing Systems*, volume 37, pages 31380–31409, 2024. doi: 10.52202/079017-0986.

Kate Donahue, Alexandra Chouldechova, and Krishnaram Kenthapadi. Human-algorithm collaboration: Achieving complementarity and avoiding unfairness. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1639–1656, 2022. doi: 10.1145/3531146.3533221.

Gabriel J. Escobar, Vincent X. Liu, Alejandro Schuler, Brian Lawson, John D. Greene, and Patricia Kipnis. Automated identification of adults at risk for in-hospital clinical deterioration. *New England Journal of Medicine*, 383(20):1951–1960, 2020. doi: 10.1056/NEJMsa2001090.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, L series, July 2024. URL `http://data.europa.eu/eli/reg/2024/1689/oj`.

Dean P. Foster and Rakesh V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1–2):40–55, 1997. doi: 10.1006/game.1997.0595.

Ziyang Guo, Yifan Wu, Jason D. Hartline, and Jessica Hullman. A decision theoretic framework for measuring AI reliance. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 221–236, 2024. doi: 10.1145/3630106.3658901.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys*, 54(11s): 235:1–235:37, 2022. doi: 10.1145/3523273.

Yongchao Huang. Bayesian inference of training dataset membership. *arXiv preprint arXiv:2506.00701*, 2025.

Kosuke Imai, Zhichao Jiang, D. James Greiner, Ryan Halen, and Sooahn Shin. Experimental evaluation of algorithm-assisted human decision-making: Application to pretrial public safety assessment. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 186(2):167–189, 2023. doi: 10.1093/jrsssa/qnad010.

Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6): 2590–2615, 2011. doi: 10.1257/aer.101.6.2590.

Shayan Kiyani, Hamed Hassani, George Pappas, and Aaron Roth. Robust decision-making with partially calibrated forecasters. In *International Conference on Learning Representations*, volume 2026, pages 113667–113686, 2026.

Himabindu Lakkaraju, Jon Kleinberg, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 275–284, 2017. doi: 10.1145/3097983.3098066.

Bryce McLaughlin and Jann Spiess. Designing algorithmic recommendations to achieve human-AI complementarity. *arXiv preprint arXiv:2405.01484*, 2024.

George O. Mohler, Martin B. Short, Sean Malinowski, Mark Johnson, George E. Tita, Andrea L. Bertozzi, and P. Jeffrey Brantingham. Randomized controlled field trials of predictive policing. *Journal of the American Statistical Association*, 110(512):1399–1411, 2015. doi: 10.1080/01621459.2015.1077710.

Robb J. Muirhead. *Aspects of Multivariate Statistical Theory*. John Wiley & Sons, New York, 1982. doi: 10.1002/9780470316559.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019. doi: 10.1126/science.aax2342.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7599–7609. PMLR, 2020.

Francheska Perepletchikova, Teresa A. Treat, and Alan E. Kazdin. Treatment integrity in psychotherapy research: analysis of the studies and examination of the associated factors. *Journal of Consulting and Clinical Psychology*, 75(6):829–841, 2007. doi: 10.1037/0022-006X.75.6.829.

Drago Plecko and Elias Bareinboim. Mind the gap: A causal perspective on bias amplification in prediction & decision-making. In *Advances in Neural Information Processing Systems*, volume 37, pages 84384–84408, 2024. doi: 10.52202/079017-2681.

Vernon C. Smith, Adam Lange, and Daniel R. Huston. Predictive modeling to forecast student outcomes and drive effective interventions in online community college courses. *Journal of Asynchronous Learning Networks*, 16(3):51–61, 2012. doi: 10.24059/olj.v16i3.275.

Mats J. Stensrud, J. D. Laurendeau, and Aaron L. Sarvet. Optimal regimes for algorithm-assisted human decision-making. *Biometrika*, 111(4):1089–1108, 2024. doi: 10.1093/biomet/asae016.

Mark Steyvers and Aakriti Kumar. Three challenges for AI-assisted decision-making. *Perspectives on Psychological Science*, 19(5):722–734, 2024. doi: 10.1177/17456916231181102.

Eleni Straitouri, Suhas Thejaswi, and Manuel G. Rodriguez. Controlling counterfactual harm in decision support systems based on prediction sets. In *Advances in Neural Information Processing Systems*, volume 37, pages 129443–129479, 2024. doi: 10.52202/079017-4113.

Nan van Geloven, Ruth H. Keogh, Wouter van Amsterdam, Giovanni Cinà, Jesse H. Krijthe, Niels Peek, Kim Luijken, Sara Magliacane, Paweł Morzywołek, Thijs van Ommen, et al. The risks of risk assessment: causal blind spots when using prediction models for treatment decisions. *Annals of Internal Medicine*, 178(9):1326–1333, 2025. doi: 10.7326/ANNALS-24-00279.

Angelina Wang, Sayash Kapoor, Solon Barocas, and Arvind Narayanan. Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy. *ACM Journal on Responsible Computing*, 1(1):9:1–9:45, 2024. doi: 10.1145/3636509.

Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Eavesdrop the composition proportion of training labels in federated learning. *arXiv preprint arXiv:1910.06044*, 2019.

Junzhe Zhang and Elias Bareinboim. Can humans be out of the loop? In *Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pages 1010–1025. PMLR, 2022.

Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. In *Advances in Neural Information Processing Systems*, volume 34, pages 22313–22324, 2021.

A Motivating examples

In this section we elaborate on how our framework and experimental setup can reconstruct and validate the intuition of the motivating examples. In the first example, the belief of the agent on the past policy determines how the prediction is interpreted and what action is taken. Indeed, in the simulation, when $X = 80$, with a bad prognosis under the assumption of $\alpha_{A\mu} = 0$, the agent chooses a high dosage to improve the outcome ($A = 20$). With a bad prognosis under the assumption of heavy treatment ($\alpha_{A\mu} = 20$), the agent chooses a low dosage to reduce over-treatment ($A = 10$). In the second motivating example, new information from the prediction forces an agent to consider different options. In the model, this means that either belief in treatment effect (N_E) or in historical policy ($\alpha_{A\mu}$) is revised, which appears as a correlation in the posterior belief in these two variables in Fig. 5.

To recap: after observing a prediction that did not match expectation, the agent revises their beliefs, entertaining multiple hypotheses on how their beliefs should change to account for the situation. In this case, either the treatment effect is lower than anticipated and the dosage higher than assumed, or the other way around.

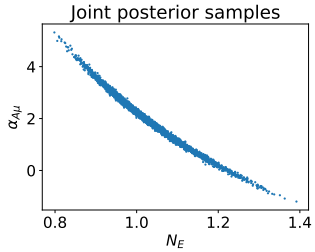


Figure 5: Plot showing joint posterior samples of variables N_E and $\alpha_{A\mu}$ in the posterior beliefs of the agent after update with ML-DS information. The negative correlation reflects that the agent entertains multiple possible explanations for the surprising prediction: either the treatment is less effective than expected, the historical dosages were higher than expected, or a scenario in between.

B Details on the experimental setup and implementation

We report here all the details about the experiments, complementing Sec. 3.

Historical SCM. Let the SCM $Hist$ be defined as:

$$N_X \sim \mathcal{N}(80, 10^2) \quad N_A \sim \mathcal{N}(2, 1^2) \quad N_Y \sim \mathcal{N}(0, 0.1^2)$$
$$X := N_X$$
$$A := 0.125\, X + N_A$$
$$Y := 12 - 0.1\, X + A + N_Y$$

In this model, the treatment protocol is simply to give a dosage of treatment A which is one eighth of the body weight X, and how well this dosing is followed is encoded by the noise term N_A. The mechanism of Y clarifies that the covariate X makes the outcome worse but the treatment improves the outcome. Of course, this equation is a great simplification of what happens in reality, but it is sufficient to demonstrate several important dynamics that emerge when introducing ML-DS. Average treatment effect is positive in this scenario, as witnessed by the positive coefficient of A in the equation for Y.

In general we denote the coefficient of X in the treatment equation by N_d and its coefficient in the outcome equation by N_b. In this univariate experiment they are fixed at 0.125 and -0.1, respectively. In the multivariate experiments, with k covariates, these coefficients are vectors (Appendix E). More generally, they may be population-level random variables like N_E, as allowed by the theoretical results in Appendix F.

The model of the agent's beliefs. The likelihood of the observed prediction $Pred^o$ is represented exactly via the auxiliary variables of Appendix F.

The model for the agent's effect estimation. Finally, the model of the agent used for inference, $\mathcal{A}_{Inf}$, is defined as:

$$
\begin{aligned}
N_X &\sim \mathcal{N}(\alpha_{X\mu}, \alpha^2_{X\sigma}) \quad N_A \sim \mathcal{N}(\alpha_{A\mu}, \alpha^2_{A\sigma}) \\
N_Y &\sim \mathcal{N}(\alpha_{Y\mu}, \alpha^2_{Y\sigma}) \\
X &:= N_X \\
A &:= 0.125\, X + N_A \\
Y &:= 12 - 0.1\, X + N_E\, A + N_Y
\end{aligned}
$$

B.1 Implementation details

The experiments were run on two high performance computing clusters, Snellius (SURF, NL) and Leonardo (CINECA, IT). The main experiments (univariate ML-DS model with and without intercept) were run on Leonardo, for a total of 24,400 runs. Each run used 4 cores and 8 GB of RAM and took between 2 and 10 minutes. The multivariate experiments were run on Snellius, for a total of 9,000 runs. Each run used 4 cores and 7 GB of RAM and took between 2 and 60 minutes. We estimate that we used 20,000 CPU-hours in total for the final runs of the experiments. The code for the simulations presented in this paper is based on PyMC [Abril-Pla et al., 2023]. We used the NUTS sampler to get 1000 samples for each of 4 chains, with 1000 tuning samples and a `target_accept` parameter of 0.85. Diagnostics and visual inspection of the traces showed good mixing and no convergence issues (max $\hat{R} = 1.01$, and a single divergence across all runs). For every value of a misaligned prior, we ran 50 such runs. Each run samples a new training set $\mathcal{D}_{Hist}$ from $Hist$, refits the ML-DS model on it, and samples a new patient X^o from the true distribution. Sanity checks on the Bayesian model are expanded upon in Appendix C. The plate reduction is derived in Appendix F, with implementation notes in Appendix F.13.

C Sanity checks on the Bayesian model

We have implemented a series of experiments to make sure that the framework is functioning as intended.

1. We compare the posterior distributions with the explicit plate and with the reduced formulation given the same observations. We check if the posteriors of the two agents end up the same. This is repeated for different sizes of plate to control for potential effects of size.
2. We run an update with many (1000) observations drawn from the real distribution of X and check that the posterior of the agent converges to the real parameter $\alpha_{X\mu}$ in the generating process.
3. We set one observation with X equal to the mean of the agent's prior, with the priors on A and N_E very peaked, but with $Pred^o$ set much higher than the value $Y = N_b X + N_E A$. We also set N_d, the coefficient of X in the structural equation of A, to zero. We observe in the posterior that there is now a correlation between the parameters of N_E and $\alpha_{A\mu}$.

D Results from univariate experiments with ML-DS model without intercept

We ran an experiment with an otherwise identical setup to the main experiment, but with an ML-DS model without intercept. In this setting, the ML-DS model is hence not well-specified. The results are presented in Fig. 6 and Fig. 4 (left). The results for misaligned $\alpha_{A\mu}$, N_E, and $\alpha_{Y\mu}$ are almost identical to the results for the main experiment where the ML-DS model was well-specified. For $\alpha_{X\mu}$, however, we see very different dynamics in the two settings (Fig. 4). While in the well-specified case the belief $\alpha_{X\mu}$ was inconsequential, in this misspecified setting having a poorly aligned belief $\alpha_{X\mu}$ can also lead to harm.

These experiments illustrate how our framework does not require the ML-DS model to be well-specified. Column N_E of Fig. 6 shows that useful learning can take place also when the ML-DS model is misspecified relative to both the true generating process $Hist$ and the generating process assumed by the agent. The intuitive explanation for this is that our framework uses the prediction only to infer what the training data for a model that produced the prediction should have looked like,

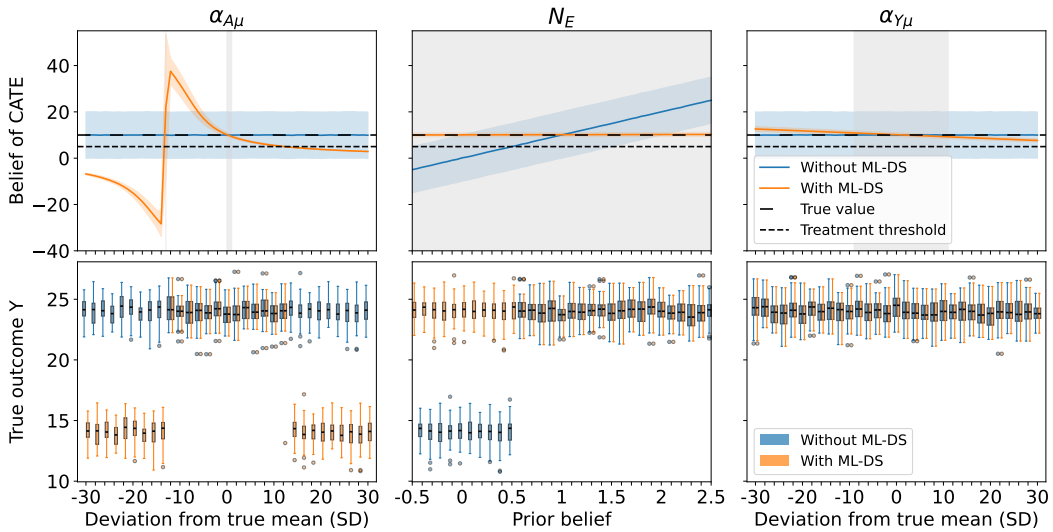


Figure 6: The effects of misaligned agent priors with ML-DS (in orange) and without (in blue) on the agent's beliefs (top row) and on the consequent downstream outcomes Y (bottom row). In contrast to Fig. 3, the ML-DS model here is misspecified so that the regression model lacks intercept (thus it diverges from the true generating process). In the plots for $\alpha_{A\mu}$ and $\alpha_{Y\mu}$, the horizontal axes represent the deviation of the agent's prior belief from the true parameter in units of standard deviation, and the centerpoints in all plots correspond to aligned priors. The orange and blue bands show the mean ± 1 SD of the agent's CATE distributions pooled across the 50 runs per prior setting, with and without ML-DS, respectively. Gray bands mark settings where the true CATE lies within the orange band. Boxplots summarize the 50 expected outcomes under the selected treatments and the true generating process $Hist$, showing the median and interquartile range (IQR), with whiskers extending to the most extreme values within 1.5 IQR of the box. Similarly to Fig. 3, the leftmost plots show how a misaligned belief $\alpha_{A\mu}$ can lead to the agent being more certain about flawed values of CATE with ML-DS, but this only leads to harmful treatment decisions when the belief drops below the treatment threshold. Similarly, the plots in the middle show how ML-DS can correct misaligned beliefs about the treatment effect over a wide range of priors which also results in improved treatment outcomes in some settings (where the blue line is below treatment threshold). The plot for misalignment of $\alpha_{X\mu}$ can be seen in Fig. 4.

rather than e.g. attempting to incorporate the prediction in some more direct way. This could in principle be done for any deterministic mathematical operation.

E Multivariate experiments

We investigated how the number of confounders k in X affects the belief updates, to see whether our findings from the univariate case extend also to the multivariate case. To isolate the effect of the number of confounders k, we kept certain properties of the model invariant. We specifically wanted the prior beliefs about CATE and its uncertainty to remain constant for all values of k and all values of misaligned priors. Further, we wanted the agent's prior beliefs about the interventional outcomes and its uncertainties to remain constant for all values of k and all values of misaligned priors. The first one would guarantee that the agent's prior belief about the treatment effect is correct, whereas the second would guarantee that the agent's prior beliefs about outcomes are correct. In a sense, these together would guarantee that the starting point of the agent is only different in terms of k.

We satisfied both invariances for misaligned beliefs $\alpha_{X\mu}$ and $\alpha_{A\mu}$, and invariant prior CATE for misaligned beliefs $\alpha_{Y\mu}$. For this experiment, X is a vector with k elements. We used the same settings as in the main experiment (univariate ML-DS model with intercept) but with the following

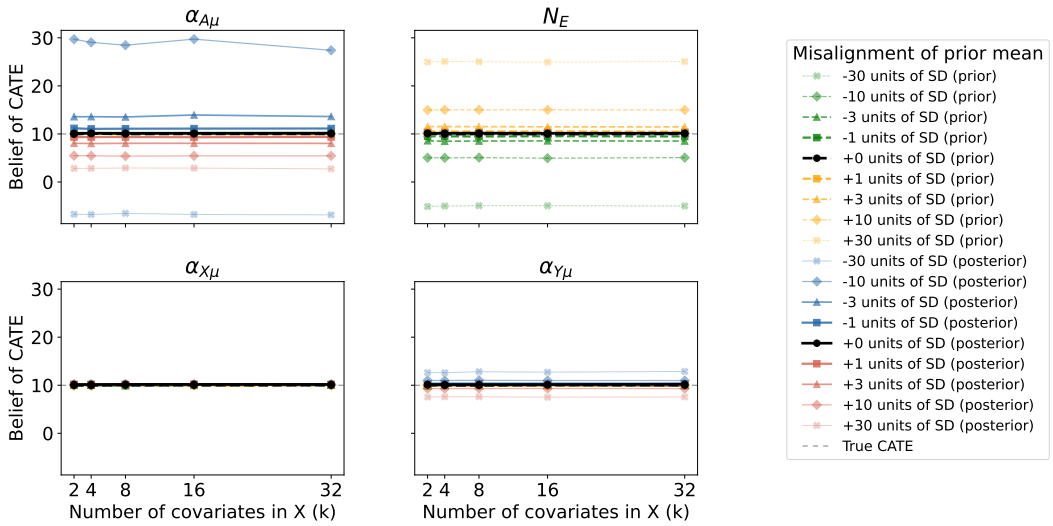


Figure 7: Number of covariates k vs. prior and posterior beliefs about CATE for different misalignments of priors. For N_E, deviations of the prior mean are expressed in units of 0.05. The experiment was constructed so that the prior expectation about CATE was correct despite the misaligned prior for $\alpha_{X\mu}$, $\alpha_{Y\mu}$, and $\alpha_{A\mu}$. For N_E this could not be done due to the structure of $\mathcal{A}_{Inf}$. The findings from the main experiment are repeated in this setting and all of the results are essentially unaffected by k. Misalignment of $\alpha_{A\mu}$ and $\alpha_{Y\mu}$ can lead to the agent having less correct posterior beliefs about CATE, and misaligned beliefs about N_E are largely corrected. Note that the lines that appear to be missing are simply masked by the thick black line.

changes:

$$(X_i)_1 \sim \mathcal{N}(\alpha_{X\mu}, \alpha_{X\sigma}^2) \tag{1}$$
$$(X_i)_j \sim \mathcal{N}(0, \alpha_{X\sigma}^2) \tag{2}$$
$$N_{b,1} = -0.1 \tag{3}$$
$$N_{b,j} = \frac{-0.1}{\sqrt{k-1}} \tag{4}$$
$$N_{d,1} = 0.125 \tag{5}$$
$$N_{d,j} = \frac{0.125}{\sqrt{k-1}} \tag{6}$$

for $j \in \{2, \dots, k\}$. This is equivalent to adjusting the variances of $(X_i)_j$ in a way that results in the variances of the inner products $N_b^\top X_i$ and $N_d^\top X_i$ being constant for all k. These experiments can only be run for $k \geq 2$, and we chose to run them for $k \in \{2, 4, 8, 16, 32\}$. We misaligned the beliefs about one variable at a time similarly to what was done in the main experiment, and of the confounders X_i we only misaligned the beliefs about $(X_i)_1$ always keeping $(X_i)_j$ for $j > 1$ aligned. Note that for N_E the invariances cannot be satisfied as that variable directly encodes CATE. We chose to run the experiments with the same pattern for N_E nevertheless. As a side note, a slightly more natural setting would be to investigate how introducing more confounders without adjusting for the additional variance affects posterior beliefs. In such a setting, however, we could not have separated the effect of the number of confounders from the effect of increased variance.

The results of these experiments are presented in Fig. 7 and Fig. 8.

Fig. 7 shows the posterior mean of CATE. Due to how we set the experiment up (prior belief about CATE invariant), the prior expectation of CATE matched the true value in all settings except those with a misaligned prior mean for N_E (dotted lines). Thus, to investigate how the beliefs were affected, it is sufficient to analyze the posterior values of CATE. Given that the plots are essentially horizontal lines, we observe no discernible effect of the number of confounders k on the belief update.

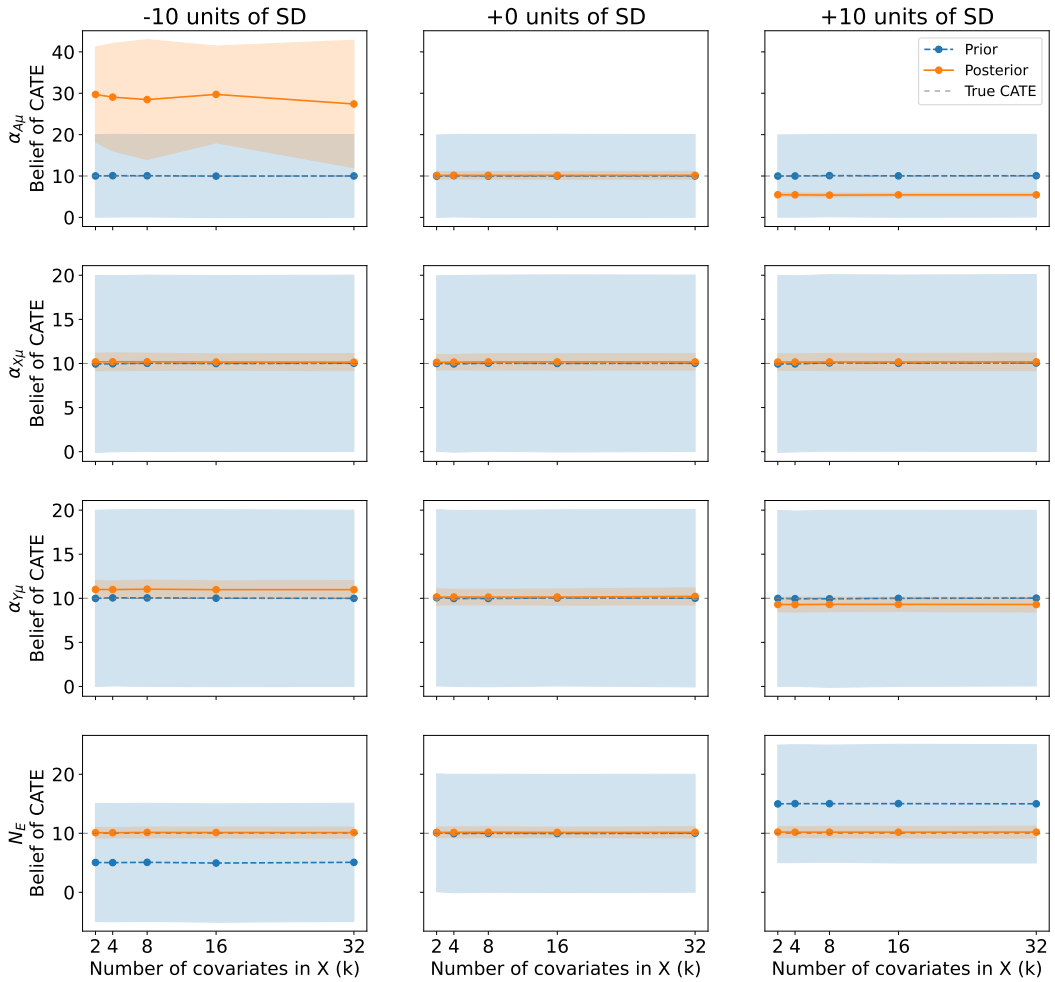


Figure 8: Prior (blue) and posterior (orange) beliefs about CATE vs. the number of covariates k. Lines show the mean and bands ± 1 SD of the agent's CATE distribution, pooled across the 50 runs per setting. For N_E, deviations of the prior mean are expressed in units of 0.05. Both the posterior expectation of CATE and its uncertainty seem largely unrelated to k. The same effect with N_E being corrected over a range of combinations seen in the main experiment is repeated here.

We further illustrate how the uncertainties about CATE were affected by k in Fig. 8. The plots indicate no relation between the number of confounders k and the posterior uncertainty of CATE. We suspect that this is due to the plate size (number of observations the ML-DS model was trained on, 1000) being much larger than k, which results in very little model uncertainty, but we have not tested this. Also for N_E, we see the main results repeated. The plot shows how ML-DS largely corrects a misaligned belief about N_E with k having virtually no effect on the posterior.

F Plate reduction

This appendix proves the plate-reduction result used for the Bayesian update in $\mathcal{A}_{Hist}$. We construct an exact representation of the conditional joint distribution of the fitted parameters θ, replacing the n latent training units in the plate with a small set of auxiliary variables. This representation preserves the likelihood of observations that depend on the training data only through θ. Examples include the fitted coefficients themselves and one or more predictions for new patients who are independent of the training data given the population-level variables.

Lemma F.1 (Reduction of the plate to auxiliary random variables)**.** *Let the historical DAG be as depicted in Fig. 2(a), where X is a vector of k independent covariates, and let n be the number of data points in the historical training data. Assume that:*

1. *the structural equations of the historical SCM are linear and without interaction terms;*
2. *all variables are continuous;*
3. *conditional on the population-level variables, patient-level noises are mutually independent across patients and variables and are Gaussian, with strictly positive covariate and outcome-noise variances;*
4. *the ML model is fitted by ordinary least squares with an intercept, predicting Y from X;*
5. *$n \geq k + 1$;*
6. *the agent knows the historical DAG, the structural functional forms, the training sample size, the model signature and the fitting procedure.*

Then, conditional on the population-level variables, the joint distribution of the fitted parameters θ in $\mathcal{A}_{Hist}$ has an exact representation using $\frac{k^2+5k+2}{2}$ scalar auxiliary variables.

Proof. Throughout the proof we condition on the population-level variables

$$N_b, \quad N_d, \quad N_E, \quad \vec{\alpha}_{X\mu}, \quad D_X, \quad \alpha_{A\mu}, \quad \alpha_{A\sigma}, \quad \alpha_{Y\mu}, \quad \alpha_{Y\sigma},$$

unless additional conditioning is stated explicitly. Uncertain population-level quantities are sampled once per realization of $\mathcal{A}_{Hist}$; known quantities are held fixed. In both cases they are shared by all patients in the plate.

The plate variables are the n patient-level latent variables and observations,

$$X_i, \quad A_i, \quad Y_i, \quad N_{Ai}, \quad N_{Yi}, \qquad i = 1, \ldots, n.$$

The variables introduced below,

$$s, \quad W, \quad g, \quad h,$$

are auxiliary variables for the proof and for the reduced implementation. They are not new SCM variables. They replace the distributional role of the plate when computing the OLS coefficients.

The proof first rewrites the design matrix in standardized coordinates and splits the covariate plate into its mean, captured by s, and its centered variation, captured by W. It then folds the treatment equation into a single regression of Y on X and writes the fitted slope and intercept as functions of (s, W) and two Gaussian noises g and h. Finally, it shows that s, W, g and h are mutually independent.

F.1 Multivariate SCM setup

For each historical patient $i = 1, \ldots, n$, with covariate vector $X_i \in \mathbb{R}^k$, the agent's historical SCM inside the plate is

$$X_i = N_{Xi}, \tag{7}$$

$$A_i = N_d^\top X_i + N_{Ai}, \tag{8}$$

$$Y_i = N_b^\top X_i + N_E A_i + N_{Yi}. \tag{9}$$

Here $N_b, N_d \in \mathbb{R}^k$ and $N_E \in \mathbb{R}$ are population-level variables, shared across the plate. The conditional derivation permits uncertainty about these coefficients, whereas the experiments fix

$N_b = b$ and $N_d = d$. The univariate experiment uses $b = -0.1$ and $d = 0.125$. In the multivariate experiment, the first covariate retains these coefficients, while the remaining covariates have the same coefficients scaled by $1/\sqrt{k-1}$. These fixed, coordinate-dependent coefficients are covered by Appendix F.12.

The noise distributions are

$$(N_{Xi})_j \sim \mathcal{N}(\alpha_{X\mu,j}, \alpha^2_{X\sigma,j}), \qquad j = 1, \ldots, k, \tag{10}$$

$$N_{Ai} \sim \mathcal{N}(\alpha_{A\mu}, \alpha^2_{A\sigma}), \tag{11}$$

$$N_{Yi} \sim \mathcal{N}(\alpha_{Y\mu}, \alpha^2_{Y\sigma}). \tag{12}$$

All patient-level noises are mutually independent across patients and across variables. We assume $\alpha_{X\sigma,j} > 0$, so the diagonal scaling matrix introduced below is invertible.

The explicit structural constant 12 in the experimental outcome equation is absorbed into the noise mean in (12): $\alpha^{\text{appendix}}_{Y\mu} = 12 + \alpha^{\text{main}}_{Y\mu}$. The prior mean is shifted by 12, with its variance unchanged. This structural constant is distinct from the fitted OLS intercept.

F.2 Standardization of the covariates

We express the patient-level exogenous noises as their population means plus scaled independent standard Gaussian variables. This exposes the standard-normal matrix Z from which the auxiliary variables s and W are built, and separates the population-level parameters $\vec{\alpha}_{X\mu}$ and D_X from the standardized noise matrix Z.

Let

$$\vec{\alpha}_{X\mu} = (\alpha_{X\mu,1}, \ldots, \alpha_{X\mu,k})^\top \in \mathbb{R}^k, \qquad D_X = \operatorname{diag}(\alpha_{X\sigma,1}, \ldots, \alpha_{X\sigma,k}) \in \mathbb{R}^{k\times k}.$$

Then each covariate vector can be written as

$$X_i = \vec{\alpha}_{X\mu} + D_X z_i, \qquad z_i \sim \mathcal{N}_k(0, I_k).$$

The vectors $z_1, \ldots, z_n$ are independent. The diagonal matrix D_X rescales the standardized covariates back to the population scale.

Similarly, write

$$N_{Ai} = \alpha_{A\mu} + \alpha_{A\sigma}\xi_{A_i}, \qquad N_{Yi} = \alpha_{Y\mu} + \alpha_{Y\sigma}\xi_{Y_i},$$

where ξ_{A_i} and ξ_{Y_i} are i.i.d. $\mathcal{N}(0,1)$, independently of all z_j.

Let $Z \in \mathbb{R}^{n\times k}$ be the matrix whose i-th row is $z_i^\top$, and let $X \in \mathbb{R}^{n\times k}$ be the design matrix whose i-th row is $X_i^\top$. Then

$$X = \mathbf{1}\vec{\alpha}_{X\mu}^\top + ZD_X, \tag{13}$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector of ones.

F.3 Centering and projection matrices

Since the model is fitted with an intercept, the slope depends on the covariates and outcome only through their deviations from the sample means. We introduce the two projections that carry out this split.

Define

$$P = \frac{1}{n}\mathbf{1}\mathbf{1}^\top, \qquad M = I_n - P.$$

The matrix P projects onto the span of $\mathbf{1}$ and extracts sample means. The matrix M projects onto the orthogonal complement of $\mathbf{1}$ and centers vectors. Both matrices are symmetric ($P^\top = P$ and $M^\top = M$) and idempotent ($P^2 = P$ and $M^2 = M$). They satisfy $P + M = I_n$, $PM = MP = 0$ and $M\mathbf{1} = 0$.

Let $Y = (Y_1, \ldots, Y_n)^\top \in \mathbb{R}^n$ be the outcome vector. For OLS with an intercept, the fitted slope vector $\theta_1 \in \mathbb{R}^k$ and intercept $\theta_0 \in \mathbb{R}$ are

$$\theta_1 = (X^\top MX)^{-1}X^\top MY, \qquad \theta_0 = \bar{Y} - \bar{X}^\top\theta_1, \tag{14}$$

where

$$\bar{X} = \frac{1}{n} X^\top \mathbf{1}, \qquad \bar{Y} = \frac{1}{n} \mathbf{1}^\top Y.$$

Using (13),

$$MX = M(\mathbf{1}\vec{\alpha}_{X\mu}^\top + ZD_X) = MZD_X,$$

because $M\mathbf{1} = 0$. Hence

$$\begin{aligned} X^\top MX &= (MX)^\top (MX) \\ &= (MZD_X)^\top (MZD_X) \\ &= D_X Z^\top MZD_X. \end{aligned} \tag{15}$$

The mean vector $\vec{\alpha}_{X\mu}$ disappears from the centered Gram matrix, so the slope denominator depends on the covariate plate only through its centered variation.

F.4 The auxiliary variable s

The population covariate mean affects the fitted model through its intercept. By (13),

$$\begin{aligned} \bar{X} &= \frac{1}{n} X^\top \mathbf{1} \\ &= \vec{\alpha}_{X\mu} + \frac{1}{n} D_X Z^\top \mathbf{1}. \end{aligned}$$

We therefore define the k-vector

$$\begin{aligned} s &= Z^\top \mathbf{1}, \\ s &\sim \mathcal{N}_k(0, nI_k). \end{aligned}$$

Its distribution follows because s is the sum of n independent $\mathcal{N}_k(0, I_k)$ vectors. Thus

$$\bar{X} = \vec{\alpha}_{X\mu} + \frac{1}{n} D_X s. \tag{16}$$

F.5 The auxiliary Wishart matrix W

By (15), the slope denominator depends on the covariate plate through the centered cross-products of the standardized covariates. Define the $k \times k$ matrix $W = Z^\top MZ$, so that

$$X^\top MX = D_X W D_X. \tag{17}$$

To identify the distribution of W, let $U \in \mathbb{R}^{n\times(n-1)}$ have orthonormal columns spanning the subspace orthogonal to $\mathbf{1}$. Then

$$\begin{aligned} M &= UU^\top, \\ U^\top U &= I_{n-1}. \end{aligned}$$

Therefore

$$\begin{aligned} W &= Z^\top MZ \\ &= Z^\top UU^\top Z \\ &= (U^\top Z)^\top (U^\top Z). \end{aligned}$$

The columns of Z are independent $\mathcal{N}_n(0, I_n)$ vectors, so the columns of $U^\top Z$ are independent $\mathcal{N}_{n-1}(0, I_{n-1})$ vectors, since $U^\top U = I_{n-1}$. Hence all entries of $U^\top Z$ are independent standard normals, and its rows are independent $\mathcal{N}_k(0, I_k)$ vectors. By the definition of the Wishart distribution [Muirhead, 1982, Definition 3.1.3],

$$W \sim \mathrm{Wishart}_k(I_k, n-1).$$

The assumption $n \geq k+1$ gives $n-1 \geq k$, so W is positive definite almost surely [Muirhead, 1982, Theorem 3.1.4].

F.6 Independence of s and W

To specify the joint distribution of (s, W), we now establish their independence.

The variable s depends only on the projection of Z onto the span of $\mathbf{1}$:

$$\begin{aligned} PZ &= \frac{1}{n}\mathbf{1}\mathbf{1}^\top Z \\ &= \frac{1}{n}\mathbf{1}s^\top. \end{aligned}$$

The matrix W depends only on the centered projection:

$$\begin{aligned} W &= Z^\top MZ \\ &= (MZ)^\top (MZ). \end{aligned}$$

For each column z of Z, the vectors Pz and Mz are jointly Gaussian with $\mathrm{Cov}(Pz, Mz) = PM^\top = PM = 0$, hence independent. Since the columns of Z are independent, PZ and MZ are independent. Therefore s, as a function of PZ, is independent of W, as a function of MZ:

$$s \perp W.$$

Together with the preceding subsection, this recovers the classical independence of the sample mean and the centered cross-product matrix of an i.i.d. Gaussian sample, the latter having a Wishart distribution with $n-1$ degrees of freedom [Muirhead, 1982, Theorem 3.1.2].

F.7 Condensing A into Y

The fitted model regresses Y on X, but in the SCM, part of Y's dependence on X is routed through A. We first collapse the two structural equations into one effective regression of Y on X, whose slope, intercept, and error variance absorb the treatment channel.

Substitute (8) into (9):

$$\begin{aligned} Y_i &= N_b^\top X_i + N_E(N_d^\top X_i + N_{Ai}) + N_{Yi} \\ &= N_b^\top X_i + N_E N_d^\top X_i + N_E(\alpha_{A\mu} + \alpha_{A\sigma}\xi_{A_i}) + \alpha_{Y\mu} + \alpha_{Y\sigma}\xi_{Y_i} \\ &= \underbrace{\alpha_{Y\mu} + N_E\alpha_{A\mu}}_{\beta_0} + \underbrace{(N_b + N_E N_d)^\top}_{\beta_1^\top} X_i + \underbrace{N_E\alpha_{A\sigma}\xi_{A_i} + \alpha_{Y\sigma}\xi_{Y_i}}_{\varepsilon_i}. \end{aligned}$$

Here ε_i is the regression error. Because ξ_{A_i} and ξ_{Y_i} are independent standard normal variables, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, where we define

$$\sigma^2 = N_E^2\alpha_{A\sigma}^2 + \alpha_{Y\sigma}^2.$$

Since $\alpha_{Y\sigma} > 0$ by assumption, also $\sigma > 0$. The regression errors are independent across i, and they are independent of all covariate noises z_j. In vector form,

$$Y = \beta_0\mathbf{1} + X\beta_1 + \varepsilon, \qquad \varepsilon \sim \mathcal{N}_n(0, \sigma^2 I_n), \qquad \varepsilon \perp Z. \tag{18}$$

Thus the treatment equation can be absorbed into an effective linear regression of Y on X with slope β_1, intercept β_0, and error variance σ^2.

F.8 OLS slope with intercept

Substitute (18) into the slope formula in (14):

$$\begin{aligned} \theta_1 &= (X^\top MX)^{-1}X^\top M(\beta_0\mathbf{1} + X\beta_1 + \varepsilon) \\ &= (X^\top MX)^{-1}\left[\beta_0 X^\top M\mathbf{1} + X^\top MX\,\beta_1 + X^\top M\varepsilon\right] \\ &= \beta_1 + (X^\top MX)^{-1}X^\top M\varepsilon. \end{aligned} \tag{19}$$

The term $\beta_0 X^\top M\mathbf{1}$ is zero because $M\mathbf{1} = 0$. This is where centering removes the intercept from the slope calculation.

Conditionally on Z, the vector $X^\top M\varepsilon$ is a linear transformation of the Gaussian error vector ε. Since $\varepsilon \mid Z \sim \mathcal{N}_n(0, \sigma^2 I_n)$,

$$\mathbb{E}[X^\top M\varepsilon \mid Z] = 0, \tag{20}$$

$$\begin{aligned}\mathrm{Cov}(X^\top M\varepsilon \mid Z) &= X^\top M(\sigma^2 I_n)M^\top X \\ &= \sigma^2 X^\top MX \\ &= \sigma^2 D_X W D_X. \end{aligned} \tag{21}$$

The conditional distribution depends on Z only through W. Hence

$$X^\top M\varepsilon \mid W \sim \mathcal{N}_k(0, \sigma^2 D_X W D_X). \tag{22}$$

Let $W^{1/2}$ denote the symmetric positive definite square root of W. A valid covariance factor for the covariance in (22) is $\sigma D_X W^{1/2}$, because

$$(\sigma D_X W^{1/2})(\sigma D_X W^{1/2})^\top = \sigma^2 D_X W D_X.$$

Define

$$g = \left(\sigma D_X W^{1/2}\right)^{-1} X^\top M\varepsilon,$$

so that

$$X^\top M\varepsilon = \sigma D_X W^{1/2} g. \tag{23}$$

Conditionally on Z, the vector g is a linear transformation of the Gaussian vector $X^\top M\varepsilon$, with mean 0 and covariance

$$\left(\sigma D_X W^{1/2}\right)^{-1} \sigma^2 D_X W D_X \left(\sigma D_X W^{1/2}\right)^{-\top} = I_k.$$

Hence $g \mid Z \sim \mathcal{N}_k(0, I_k)$ for almost every realization of Z. Because this conditional distribution does not depend on Z, the vector g is independent of Z, and therefore independent of (s, W), which are functions of Z:

$$g \sim \mathcal{N}_k(0, I_k), \qquad g \perp (s, W).$$

Substituting (23) and (17) into (19),

$$\begin{aligned}\theta_1 &= \beta_1 + (D_X W D_X)^{-1} \sigma D_X W^{1/2} g \\ &= \beta_1 + \sigma D_X^{-1} W^{-1} D_X^{-1} D_X W^{1/2} g \\ &= \beta_1 + \sigma D_X^{-1} W^{-1/2} g. \end{aligned}$$

F.9 OLS intercept

By (14), the intercept is $\theta_0 = \bar{Y} - \bar{X}^\top \theta_1$. Averaging (18) over the n rows gives

$$\bar{Y} = \beta_0 + \bar{X}^\top \beta_1 + \bar{\varepsilon},$$

where $\bar{\varepsilon} = \frac{1}{n}\mathbf{1}^\top \varepsilon$ is the mean regression error. Therefore

$$\begin{aligned}\theta_0 &= \beta_0 + \bar{X}^\top \beta_1 + \bar{\varepsilon} - \bar{X}^\top \theta_1 \\ &= \beta_0 + \bar{\varepsilon} - \bar{X}^\top (\theta_1 - \beta_1). \end{aligned} \tag{24}$$

Besides the slope noise g, the regression errors enter the fitted intercept through their mean $\bar{\varepsilon}$. As the mean of n independent $\mathcal{N}(0, \sigma^2)$ errors, $\bar{\varepsilon} \sim \mathcal{N}(0, \sigma^2/n)$. We obtain a fourth auxiliary variable, whose distribution, like that of g, does not depend on the population-level variables, by dividing $\bar{\varepsilon}$ by its standard deviation $\sigma/\sqrt{n}$,

$$h = \frac{\sqrt{n}}{\sigma}\bar{\varepsilon} \sim \mathcal{N}(0, 1).$$

Thus

$$\bar{\varepsilon} = \frac{\sigma}{\sqrt{n}} h. \tag{25}$$

The last term in (24) adjusts the fitted intercept for the error in the fitted slope. To preserve the joint distribution of (θ_0, θ_1), this term must use the same auxiliary vector g as the slope.

Using (16), (25), and $\theta_1 - \beta_1 = \sigma D_X^{-1} W^{-1/2} g$, equation (24) becomes

$$\begin{aligned}\theta_0 &= \beta_0 + \frac{\sigma}{\sqrt{n}} h - \sigma \left(\vec{\alpha}_{X\mu} + \frac{1}{n} D_X s \right)^\top D_X^{-1} W^{-1/2} g \\ &= \beta_0 + \frac{\sigma}{\sqrt{n}} h - \sigma \left(\vec{\alpha}_{X\mu}^\top D_X^{-1} + \frac{1}{n} s^\top \right) W^{-1/2} g.\end{aligned}$$

It remains to record the independence structure involving h. The variable h is a function of $P\varepsilon$, while g is generated from the centered component $M\varepsilon$ together with Z. Since P and M are orthogonal projections and ε is Gaussian, $P\varepsilon$ and $M\varepsilon$ are independent. Moreover, $\varepsilon \perp Z$ implies that the pair $(P\varepsilon, M\varepsilon)$ is independent of Z. Consequently, $P\varepsilon$ is independent of $(M\varepsilon, Z)$, and hence

$$h \perp (g, s, W).$$

Together with $g \perp (s, W)$ and $s \perp W$, the four auxiliary variables are mutually independent.

F.10 Final reduced representation

The full plate can be replaced by the following mutually independent auxiliary variables:

$$s \sim \mathcal{N}_k(0, nI_k), \qquad W \sim \mathrm{Wishart}_k(I_k, n-1), \qquad g \sim \mathcal{N}_k(0, I_k), \qquad h \sim \mathcal{N}(0,1).$$

Given these variables and the population-level variables, the fitted ML model parameters are

$$\theta_1 = \beta_1 + \sigma D_X^{-1} W^{-1/2} g, \tag{26}$$

$$\theta_0 = \beta_0 + \frac{\sigma}{\sqrt{n}} h - \sigma \left(\vec{\alpha}_{X\mu}^\top D_X^{-1} + \frac{1}{n} s^\top \right) W^{-1/2} g, \tag{27}$$

where

$$\beta_1 = N_b + N_E N_d, \qquad \beta_0 = \alpha_{Y\mu} + N_E \alpha_{A\mu}, \qquad \sigma^2 = N_E^2 \alpha_{A\sigma}^2 + \alpha_{Y\sigma}^2.$$

Equations (26)–(27) define the same conditional distribution for the OLS coefficients as the original n-row plate. Therefore the likelihood of $\theta = (\theta_0, \theta_1)$ in $\mathcal{A}_{Hist}$ can be computed from the population-level variables and the auxiliary tuple (s, W, g, h) instead of the full plate.

The number of scalar auxiliary variables is

$$\underbrace{k}_{s} + \underbrace{\frac{k(k+1)}{2}}_{W} + \underbrace{k}_{g} + \underbrace{1}_{h} = \frac{k^2 + 5k + 2}{2}.$$

This proves the lemma. □

F.11 Second-stage extension: single-prediction collapsed likelihood

Lemma F.1 preserves the full fitted linear model (θ_0, θ_1). In the special case where the agent observes only one scalar prediction

$$Pred^o = \theta_0 + x^\top \theta_1$$

for a fixed observed covariate vector $X^o = x$, a further likelihood-level collapse is possible. This second-stage collapse preserves the scalar likelihood of the observed prediction rather than the full fitted coefficient vector. The collapsed likelihood is obtained by substituting the reduced representation (26)–(27) into $Pred^o$.

Assume, for this subsection only, that the coefficient priors factorize identically across covariates:

$$(N_{b,j}, N_{d,j}) \overset{iid}{\sim} \mathcal{N}_2 \left(\begin{bmatrix} m_b \\ m_d \end{bmatrix}, \begin{bmatrix} v_b & v_{bd} \\ v_{bd} & v_d \end{bmatrix} \right), \qquad j = 1, \ldots, k,$$

independently of the plate noises and of the remaining population-level variables. This i.i.d. form is assumed only so that the marginalized coefficient prior enters through two scalar functions of x. The reduction itself does not depend on it (Appendix F.12, which also covers fixed N_b and N_d). Define

$$S^o = \mathbf{1}^\top x, \qquad Q^o = x^\top x, \qquad \Delta_X^2 = (x - \vec{\alpha}_{X\mu})^\top D_X^{-2}(x - \vec{\alpha}_{X\mu}).$$

We write $\Delta_X = \sqrt{\Delta_X^2} \geq 0$ and use $\chi_k'^2(\lambda)$ to denote the non-central chi-squared distribution with k degrees of freedom and non-centrality λ. Here S^o and Q^o will carry the marginalized coefficient prior, while Δ_X^2 is the squared standardized (Mahalanobis) distance of the query point x from the covariate mean $\vec{\alpha}_{X\mu}$, a measure of how far the prediction is extrapolated. Below, $n\Delta_X^2$ is the non-centrality parameter used to represent R. The symbol $\mathbf{1}$ in $S^o = \mathbf{1}^\top x$ denotes the all-ones vector in covariate space $\mathbb{R}^k$. The superscript o marks quantities evaluated at the new patient's covariate vector x, the observed value of X^o.

Theorem F.2 (Single-prediction collapsed likelihood)**.** *Under the assumptions of Lemma F.1 and the i.i.d. Gaussian coefficient prior above, consider a single observed prediction $Pred^o$ for a new patient who is independent of the training data given the population-level variables. Let*

$$R \sim \frac{1}{n}\chi_k'^2(n\Delta_X^2), \qquad T \sim \chi_{n-k}^2, \qquad R \perp T.$$

Then, after marginalizing N_b and N_d, the likelihood of $Pred^o$ under the full plate can be written as

$$p\left(Pred^o \mid N_E, X^o = x, \text{ remaining population-level variables}\right) = \mathbb{E}_{R,T}\left[\mathcal{N}\left(Pred^o; \mu_{\mathrm{red}}, V_{\mathrm{red}}\right)\right],$$

where

$$\mu_{\mathrm{red}} = \beta_0 + (m_b + N_E m_d)S^o$$

and

$$V_{\mathrm{red}} = Q^o(v_b + 2N_E v_{bd} + N_E^2 v_d) + \sigma^2\left(\frac{1}{n} + \frac{R}{T}\right).$$

The Gaussian density inside the expectation is the conditional density of $Pred^o$ given R and T,

$$\boxed{Pred^o \mid R, T, N_E, X^o = x, \text{ remaining population-level variables}} \sim \mathcal{N}(\mu_{\mathrm{red}}, V_{\mathrm{red}}).$$

The variable R can also be generated from central scalar primitives as

$$R = \left(\Delta_X + \frac{G_R}{\sqrt{n}}\right)^2 + \frac{C_R}{n}, \qquad G_R \sim \mathcal{N}(0,1), \qquad C_R \sim \chi_{k-1}^2,$$

where G_R, C_R and T are mutually independent, with the convention $\chi_0^2 = 0$ when $k = 1$.

Proof. Condition on N_E, the remaining population-level variables, and $X^o = x$. Keep

$$\beta_1 = N_b + N_E N_d$$

fixed until the last step, where N_b and N_d are integrated out. This lets us apply Lemma F.1, which holds conditionally on the population coefficients.

From Lemma F.1,

$$\theta_1 = \beta_1 + \sigma D_X^{-1} W^{-1/2} g$$

and

$$\theta_0 = \beta_0 + \frac{\sigma}{\sqrt{n}}h - \sigma\left(\vec{\alpha}_{X\mu}^\top D_X^{-1} + \frac{1}{n}s^\top\right)W^{-1/2}g.$$

Substituting into $Pred^o = \theta_0 + x^\top\theta_1$ gives

$$Pred^o = \beta_0 + x^\top\beta_1 + \sigma\left[\frac{1}{\sqrt{n}}h + \left(x^\top D_X^{-1} - \vec{\alpha}_{X\mu}^\top D_X^{-1} - \frac{1}{n}s^\top\right)W^{-1/2}g\right]. \tag{28}$$

In the coefficient of $W^{-1/2}g$, the first two terms combine into the standardized deviation of x from the covariate mean, the vector whose squared norm is Δ_X^2,

$$\delta = D_X^{-1}(x - \vec{\alpha}_{X\mu}).$$

Since D_X^{-1} is diagonal, hence symmetric, $x^\top D_X^{-1} - \vec{\alpha}_{X\mu}^\top D_X^{-1} = \delta^\top$. The whole coefficient of $W^{-1/2}g$ is then $r^\top$, with

$$r = \delta - \frac{1}{n}s.$$

By (16), $r = D_X^{-1}(x - \bar{X})$ is the standardized deviation of x from the sample mean. Then

$$Pred^o = \beta_0 + x^\top\beta_1 + \sigma\left(\frac{1}{\sqrt{n}}h + r^\top W^{-1/2}g\right). \tag{29}$$

Conditionally on r and W, the term $r^\top W^{-1/2}g = (W^{-1/2}r)^\top g$ is a linear functional of $g \sim \mathcal{N}_k(0, I_k)$, hence Gaussian with zero mean and conditional variance

$$\begin{aligned}\mathrm{Var}\left(r^\top W^{-1/2}g \mid r, W\right) &= (W^{-1/2}r)^\top(W^{-1/2}r)\\ &= r^\top W^{-1}r.\end{aligned}$$

Because h is independent of (g, r, W), the variances of the two zero-mean Gaussian terms add:

$$\frac{1}{\sqrt{n}}h + r^\top W^{-1/2}g \mid r, W \sim \mathcal{N}\left(0, \frac{1}{n} + r^\top W^{-1}r\right). \tag{30}$$

By (30), given (r, W) the prediction noise is Gaussian, and the training data enter its variance only through $r^\top W^{-1}r$. We now show that $r^\top W^{-1}r \overset{d}{=} R/T$, with $R = \|r\|^2$ and an independent $T \sim \chi^2_{n-k}$.

We first derive the distribution of $R = \|r\|^2$. Since $s \sim \mathcal{N}_k(0, nI_k)$,

$$r = \delta - \frac{1}{n}s \sim \mathcal{N}_k\left(\delta, \frac{1}{n}I_k\right).$$

The non-central χ^2 distribution describes the squared norm of a Gaussian vector with identity covariance, so we rescale r by $\sqrt{n}$,

$$\sqrt{n}\,r \sim \mathcal{N}_k(\sqrt{n}\,\delta, I_k).$$

Since $\|\sqrt{n}\,r\|^2 = nR$, the variable nR has a non-central χ^2_k distribution with non-centrality $\|\sqrt{n}\,\delta\|^2 = n\Delta_X^2$. Dividing by n,

$$R \sim \frac{1}{n}\chi'^2_k(n\Delta_X^2).$$

A non-central χ^2 generator is not always available, so we also give an equivalent construction from central primitives. It follows by rotation, and its $k = 1$ case is used in the implementation described below. Let

$$\eta = \sqrt{n}(r - \delta) \sim \mathcal{N}_k(0, I_k).$$

Then

$$nR = \|\sqrt{n}\,\delta + \eta\|^2.$$

Choose an orthogonal matrix O_δ that maps δ to $\Delta_X e_1$, where $e_1 = (1, 0, \ldots, 0)^\top \in \mathbb{R}^k$. If $\delta = 0$, any orthogonal O_δ may be used. By rotational invariance,

$$O_\delta\eta \sim \mathcal{N}_k(0, I_k).$$

Write

$$O_\delta\eta = (G_R, C_2, \ldots, C_k)^\top,$$

with independent standard normal components. Then

$$nR = (\sqrt{n}\,\Delta_X + G_R)^2 + \sum_{j=2}^{k} C_j^2.$$

Thus

$$R = \left(\Delta_X + \frac{G_R}{\sqrt{n}}\right)^2 + \frac{C_R}{n}, \qquad C_R = \sum_{j=2}^{k} C_j^2 \sim \chi^2_{k-1}.$$

We next reduce $r^\top W^{-1} r$. Since $W \perp s$, also $W \perp r$, so conditioning on r leaves the distribution of W unchanged. Conditionally on r, if $r = 0$ the quadratic form is zero. If $r \neq 0$, choose an orthogonal matrix O_r such that

$$O_r^\top r = \|r\| e_1.$$

Since the Wishart distribution with identity scale is invariant under orthogonal conjugation,

$$O_r^\top W O_r \sim \mathrm{Wishart}_k(I_k, n-1).$$

Therefore, conditionally on r,

$$r^\top W^{-1} r = \|r\|^2 e_1^\top (O_r^\top W O_r)^{-1} e_1 \overset{d}{=} \|r\|^2 e_1^\top W^{-1} e_1.$$

The distribution of the reciprocal $[(W^{-1})_{11}]^{-1}$ of the diagonal entry of an identity-scale inverse Wishart is classical [Muirhead, 1982, Theorem 3.2.12]; for completeness we give a short self-contained derivation. Let $\nu = n - 1$ and represent $W = V^\top V$, where $V \in \mathbb{R}^{\nu \times k}$ has independent standard normal entries. Partition

$$V = [v_1, V_2],$$

where $v_1 \in \mathbb{R}^\nu$ and $V_2 \in \mathbb{R}^{\nu \times (k-1)}$. The Schur complement identity gives

$$\left[(W^{-1})_{11}\right]^{-1} = v_1^\top \left[I_\nu - V_2 (V_2^\top V_2)^{-1} V_2^\top\right] v_1.$$

For $k = 1$, V_2 has no columns, and the projection onto its column span is zero, so the bracketed matrix is I_ν. In general, the bracketed matrix is the orthogonal projector onto the complement of the column span of V_2. Almost surely, V_2 has rank $k - 1$, so this projector has rank

$$\nu - (k-1) = \nu - k + 1.$$

Conditionally on V_2, the projector is fixed and $v_1 \sim \mathcal{N}_\nu(0, I_\nu)$. In an orthonormal basis adapted to this projector, the quadratic form is the sum of squares of $\nu - k + 1$ independent standard normal coordinates. Hence

$$\left[(W^{-1})_{11}\right]^{-1} \sim \chi^2_{\nu-k+1}.$$

Substituting $\nu = n - 1$ gives

$$\nu - k + 1 = n - k.$$

Therefore

$$e_1^\top W^{-1} e_1 \overset{d}{=} \frac{1}{T}, \qquad T \sim \chi^2_{n-k}.$$

Because $r \sim \mathcal{N}_k(\delta, \frac{1}{n} I_k)$, we have $r \neq 0$ almost surely, and the ratio

$$T = \frac{\|r\|^2}{r^\top W^{-1} r}$$

is well defined almost surely. Conditionally on r, $T \sim \chi^2_{n-k}$, and this distribution does not depend on r. Hence T is independent of r, and therefore of $R = \|r\|^2$. Thus

$$r^\top W^{-1} r \overset{d}{=} \frac{R}{T}, \qquad T \sim \chi^2_{n-k}, \qquad T \perp R.$$

By (29)–(30), the conditional prediction mean is $\beta_0 + x^\top \beta_1$, and its variance depends on (r, W) only through $r^\top W^{-1} r = R/T$. Averaging over (r, W) at fixed (R, T) therefore leaves the same Gaussian distribution. Conditional also on β_1, N_E, $X^o = x$ and the remaining population-level variables,

$$Pred^o \mid \beta_1, R, T \sim \mathcal{N}\left(\beta_0 + x^\top \beta_1,\, \sigma^2 \left(\frac{1}{n} + \frac{R}{T}\right)\right).$$

It remains to integrate out N_b and N_d. In the conditional distribution above the coefficients enter only through the mean $\beta_0 + x^\top \beta_1$, so marginalizing their prior adds an independent Gaussian term to the prediction and leaves the OLS-noise variance $\sigma^2(\frac{1}{n} + R/T)$ unchanged. Define

$$B^o = x^\top N_b, \qquad D^o = x^\top N_d.$$

Then

$$x^\top \beta_1 = B^o + N_E D^o.$$

Under the i.i.d. Gaussian coefficient prior,

$$(B^o, D^o)$$

is bivariate Gaussian with

$$\mathbb{E}[B^o] = m_b S^o, \qquad \mathbb{E}[D^o] = m_d S^o,$$

$$\mathrm{Var}(B^o) = v_b Q^o, \qquad \mathrm{Var}(D^o) = v_d Q^o, \qquad \mathrm{Cov}(B^o, D^o) = v_{bd} Q^o.$$

Therefore, conditionally on N_E,

$$B^o + N_E D^o \sim \mathcal{N}\left((m_b + N_E m_d) S^o, Q^o(v_b + 2N_E v_{bd} + N_E^2 v_d)\right).$$

This Gaussian term is independent of the OLS noise represented by (R, T), because the coefficient priors are independent of the auxiliary variables replacing the plate. Summing the two independent Gaussian components yields

$$Pred^o \mid R, T, N_E, X^o = x, \text{ remaining population-level variables} \sim \mathcal{N}(\mu_{\mathrm{red}}, V_{\mathrm{red}}),$$

with

$$\mu_{\mathrm{red}} = \beta_0 + (m_b + N_E m_d) S^o$$

and

$$V_{\mathrm{red}} = Q^o(v_b + 2N_E v_{bd} + N_E^2 v_d) + \sigma^2 \left(\frac{1}{n} + \frac{R}{T}\right).$$

Each step above preserves the distribution of $Pred^o$ given N_E, $X^o = x$ and the remaining population-level variables. Integrating over the distributions of R and T therefore gives the likelihood of $Pred^o$ under the full plate,

$$p\left(Pred^o \mid N_E, X^o = x, \text{ remaining population-level variables}\right) = \mathbb{E}_{R,T}\left[\mathcal{N}\left(Pred^o; \mu_{\mathrm{red}}, V_{\mathrm{red}}\right)\right].$$

This proves the single-prediction collapsed likelihood. □

The theorem above gives the conditional likelihood of the observed scalar prediction given the observed covariate vector. The full Bayesian update for the agent also contains the likelihood of the observed covariate vector itself. Under the diagonal covariate model,

$$p(X^o = x \mid \vec{\alpha}_{X\mu}, D_X) = \prod_{j=1}^{k} \mathcal{N}\big(x_j; \alpha_{X\mu,j}, \alpha^2_{X\sigma,j}\big).$$

This term updates the agent's beliefs about the covariate population parameters. It is not absorbed into the collapsed likelihood for $Pred^o$.

The collapse also retains the information needed for the treatment-effect decision rule. For a fixed realization ψ of the population-level parameters, the interventional model used by the agent gives

$$\mathbb{E}^{\mathcal{A}^\psi_{Inf}}[Y \mid X = x, do(A = a)] = B^o + N_E a + \alpha_{Y\mu},$$

where $B^o = x^\top N_b$. Therefore, for two treatment levels a_1 and a_2,

$$\mathrm{CATE}^\psi = \mathbb{E}^{\mathcal{A}^\psi_{Inf}}[Y \mid X = x, do(A = a_1)] - \mathbb{E}^{\mathcal{A}^\psi_{Inf}}[Y \mid X = x, do(A = a_2)] = N_E(a_1 - a_2).$$

This is the CATE at fixed parameters. The agent's estimate is its posterior mean, $(a_1 - a_2)\mathbb{E}[N_E \mid X^o = x, Pred^o]$. The projection B^o cancels in the contrast. Hence the reduced likelihood must preserve the posterior information about N_E for the decision rule, but it does not need to preserve the full posterior over the individual coefficient vectors N_b and N_d.

F.12 General prior across covariates

The compact form of Theorem F.2 uses the i.i.d. coefficient prior, but the reduction itself does not require it. With a jointly Gaussian prior on (N_b, N_d), independent of the other population-level variables and of the standardized plate noises, having mean vectors $m_b, m_d \in \mathbb{R}^k$ and covariance blocks $\Sigma_b, \Sigma_d, \Sigma_{bd} \in \mathbb{R}^{k\times k}$,

$$N_b \sim \mathcal{N}_k(m_b, \Sigma_b), \qquad N_d \sim \mathcal{N}_k(m_d, \Sigma_d), \qquad \mathrm{Cov}(N_b, N_d) = \Sigma_{bd},$$

the projections $B^o = x^\top N_b$ and $D^o = x^\top N_d$ used in the last step of the proof are still jointly Gaussian, with

$$\mathbb{E}[B^o] = x^\top m_b, \qquad \mathbb{E}[D^o] = x^\top m_d,$$
$$\mathrm{Var}(B^o) = x^\top \Sigma_b\, x, \qquad \mathrm{Var}(D^o) = x^\top \Sigma_d\, x, \qquad \mathrm{Cov}(B^o, D^o) = x^\top \Sigma_{bd}\, x.$$

The shorthands $S^o = \mathbf{1}^\top x$ and $Q^o = x^\top x$ of the theorem are replaced by these linear and quadratic forms in x, but the structure of the result is identical. More generally, the same Gaussian formulas apply if (N_b, N_d) is jointly Gaussian conditional on the remaining population-level variables, with the corresponding conditional moments, and remains conditionally independent of the standardized plate noises and the new patient. In particular, if the prior depends only on N_E, use its moments conditional on N_E.

For fixed coefficients $N_b = b$ and $N_d = d$, as in our experiments, set $m_b = b$, $m_d = d$ and $\Sigma_b = \Sigma_d = \Sigma_{bd} = 0$. Then B^o and D^o are constants, and the conditional Gaussian prediction likelihood has mean and variance

$$\mu_{\mathrm{red}} = \beta_0 + x^\top (b + N_E d),$$
$$V_{\mathrm{red}} = \sigma^2 \left(\frac{1}{n} + \frac{R}{T}\right).$$

For non-Gaussian coefficient priors, the reduction to (R, T) still holds under the same conditional independence assumptions. At fixed N_E, remaining population-level variables and $X^o = x$, the likelihood conditional on (R, T) is the Gaussian likelihood from the proof averaged over the conditional distribution of $B^o + N_E D^o$. This average need not have a closed form.

F.13 Variable count and implementation notes

Table 1 summarizes how many scalar sampling events are needed to represent the training-data plate under the three representations derived above. The full plate samples the n historical units explicitly, with $k + 2$ scalars per unit. The reduced plate of Lemma F.1 is constant in n but quadratic in k through the Wishart matrix W. The likelihood-level auxiliaries (R, T) of Theorem F.2 are constant in both. Population-level variables outside the plate are excluded from the count, since they are common to the three representations.

k	Full plate ($n = 1000$)	Reduced plate (Lemma F.1)	Collapsed (Thm F.2)
1	3000	4	2
2	4000	8	2
5	7000	26	2
10	12000	76	2

Table 1: Number of scalar sampling events needed to represent the training-data plate, as a function of the number of covariates k, for the three plate representations.

Probabilistic programming languages do not always provide a native non-central chi-squared distribution (PyMC, for instance, does not). At $k = 1$, our implementation samples $G_R \sim \mathcal{N}(0, 1)$ and sets $R = (\delta + G_R/\sqrt{n})^2$, where $\delta = (x - \alpha_{X\mu})/\alpha_{X\sigma}$ is the signed standardized offset. By the symmetry of G_R, this has the same distribution as the construction in Theorem F.2 with $C_R = 0$. At $k \geq 2$, it samples R from its scaled non-central χ^2 distribution, implemented in our code. In both cases, $T \sim \chi^2_{n-k}$ independently.

In addition, $\alpha_{A\mu}$, $\alpha_{Y\mu}$ and any uncertain coefficients entering the prediction linearly have Gaussian priors independent of one another and of the other population-level variables. These quantities

are marginalized analytically conditional on N_E and the remaining sampled variables, while $\vec{\alpha}_{X\mu}$ remains sampled. The two signs of N_E are then summed out exactly at fixed $|N_E|$ to improve NUTS mixing when the posterior has separated positive and negative modes. After sampling, the sign and marginalized variables are reconstructed from their conditional posterior, preserving the target posterior. The reported experiments fix N_b and N_d.

F.14 No-intercept addendum

For the no-intercept model, we reuse the standardized design, the auxiliary variables s and W, and the condensed outcome representation from the proof of Lemma F.1. We derive its single-prediction likelihood using (s, W), without the further reduction to (R, T).

Corollary F.3 (No-intercept case)**.** *Under the assumptions of Lemma F.1, except that the ML model is fitted by ordinary least squares without an intercept, the joint distribution of its fitted coefficients conditional on the population-level variables has an exact representation using $\frac{k^2+5k}{2}$ scalar auxiliary variables.*

Proof. The SCM, standardization, and condensed outcome representation are the same as in Lemma F.1. In particular, by (13) and (18),

$$X = \mathbf{1}\vec{\alpha}_{X\mu}^\top + ZD_X,$$
$$Y = \beta_0\mathbf{1} + X\beta_1 + \varepsilon,$$

with $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 I_n)$ and $\varepsilon \perp Z$. The no-intercept estimator, which plays the role of θ in this model, is

$$\phi = (X^\top X)^{-1} X^\top Y.$$

The uncentered covariate sum is

$$S_x := X^\top \mathbf{1} = n\vec{\alpha}_{X\mu} + D_X s.$$

Moreover,

$$Z^\top Z = Z^\top PZ + Z^\top MZ = \frac{1}{n} ss^\top + W.$$

Therefore

$$X^\top X = \frac{1}{n} S_x S_x^\top + D_X W D_X. \tag{31}$$

Since W is positive definite almost surely and D_X is invertible, $D_X W D_X$ is positive definite almost surely. The rank-one matrix $S_x S_x^\top / n$ is positive semidefinite, so $X^\top X$ is positive definite almost surely.

Substituting the condensed outcome equation,

$$\begin{aligned}\phi &= (X^\top X)^{-1} X^\top (\beta_0\mathbf{1} + X\beta_1 + \varepsilon) \\ &= \beta_1 + \beta_0 (X^\top X)^{-1} S_x + (X^\top X)^{-1} X^\top \varepsilon. \end{aligned} \tag{32}$$

Conditionally on Z,

$$X^\top \varepsilon \mid Z \sim \mathcal{N}_k(0, \sigma^2 X^\top X).$$

The covariance is a deterministic function of (s, W) through (31). As in the definition of g in the proof of Lemma F.1, define

$$\tilde{g} = \sigma^{-1} (X^\top X)^{-1/2} X^\top \varepsilon, \qquad \text{so that} \qquad X^\top \varepsilon = \sigma (X^\top X)^{1/2} \tilde{g}.$$

Conditionally on Z, $\tilde{g}$ is Gaussian with mean 0 and covariance I_k; since this conditional distribution does not depend on Z,

$$\tilde{g} \sim \mathcal{N}_k(0, I_k), \qquad \tilde{g} \perp (s, W).$$

Substituting into (32),

$$\phi = \beta_1 + \beta_0 (X^\top X)^{-1} S_x + \sigma (X^\top X)^{-1/2} \tilde{g},$$

with

$$S_x = n\vec{\alpha}_{X\mu} + D_X s, \qquad X^\top X = \frac{1}{n} S_x S_x^\top + D_X W D_X.$$

Thus the no-intercept plate is represented by

$$s \sim \mathcal{N}_k(0, nI_k), \qquad W \sim \mathrm{Wishart}_k(I_k, n-1), \qquad \tilde{g} \sim \mathcal{N}_k(0, I_k),$$

with $s \perp W$ and $\tilde{g} \perp (s, W)$. The scalar count is

$$\underbrace{k}_{s} + \underbrace{\frac{k(k+1)}{2}}_{W} + \underbrace{k}_{\tilde{g}} = \frac{k^2 + 5k}{2}.$$

No separate scalar h is needed here. The mean and centered components of the regression error enter ϕ only through $X^\top \varepsilon$ and are represented jointly by the Gaussian vector $\tilde{g}$. □

Single prediction without intercept. Our no-intercept experiment (Appendix D) observes a single prediction $Pred^o_{\mathrm{no}} = x^\top \phi$ at $X^o = x$, with $x \neq 0$. Its likelihood follows from the representation above by marginalizing $\tilde{g}$. Define

$$\lambda_x = x^\top (X^\top X)^{-1} S_x,$$
$$q_x = x^\top (X^\top X)^{-1} x.$$

Conditionally on the population-level variables and on (s, W), the prediction is linear in the standard Gaussian vector $\tilde{g}$. Hence

$$Pred^o_{\mathrm{no}} \mid s, W \sim \mathcal{N}\left(x^\top \beta_1 + \beta_0 \lambda_x,\ \sigma^2 q_x\right).$$

The likelihood is obtained by integrating this Gaussian distribution over (s, W). This (s, W) representation uses $k + \frac{k(k+1)}{2}$ scalar auxiliary variables. In the univariate experiment, $s \sim \mathcal{N}(0, n)$ and $W \sim \chi^2_{n-1}$ are independent, so this representation uses two scalar auxiliaries.

G Perfect predictive calibration does not identify treatment effects

This example separates predictive calibration from the causal information needed for a treatment decision. Consider

$$X \sim \mathcal{N}(0,1), \qquad A = \mu_A + \varepsilon_A, \qquad Y = X + N_E A + \varepsilon_Y,$$

where $\varepsilon_A \sim \mathcal{N}(0, \sigma_A^2)$ and $\varepsilon_Y \sim \mathcal{N}(0, \sigma_Y^2)$ are independent of each other and of X, and the noise variances are positive. The coefficient of X is known and fixed at one. Higher outcomes are preferred. Here μ_A denotes the historical treatment mean; the corresponding quantity in the agent's model is $\alpha_{A\mu}$.

The same calibrated predictor, opposite effects. The two SCMs $(\mu_A, N_E) = (1, 1)$ and $(\mu_A, N_E) = (-1, -1)$ induce the same observational distribution of (X, Y), since in both cases

$$Y \mid X = x \sim \mathcal{N}(x + 1, \sigma_A^2 + \sigma_Y^2).$$

The observational intercept is the product $N_E \mu_A$, which equals one in both SCMs, while the slope remains one. The noise variance $N_E^2 \sigma_A^2 + \sigma_Y^2$ is also unchanged. Their population predictor is $f(x) = x + 1$, which is perfectly calibrated in mean:

$$\mathbb{E}[Y \mid f(X)] = \mathbb{E}[\mathbb{E}[Y \mid X] \mid f(X)] = f(X).$$

Nevertheless, for $a_1 = 1$ and $a_2 = 0$, the CATE is N_E, hence $+1$ in the first SCM and -1 in the second. No property of the treatment-naive observational distribution $P(X, Y)$ alone distinguishes these effects. In particular, stronger predictive calibration does not resolve the ambiguity without information about the causal process.

A harmful Bayesian update. Take the first SCM to be true, and let the agent observe $X^o = 1$ and $Pred^o = f(1) = 2$. The agent's treatment-effect prior is $N_E \sim \mathcal{N}(1, 1)$, and it chooses a_1 when its posterior mean CATE exceeds $1/2$. Before seeing the prediction its mean CATE is one, so it chooses a_1.

For algebraic clarity, take the zero-variance limit of the agent's Gaussian prior on $\alpha_{A\mu}$, concentrated at the incorrect value -1, with the other structural quantities known. In the population-predictor limit of OLS training, the agent's model gives $Pred^o = X^o + N_E \alpha_{A\mu} = 1 - N_E$. Observing $Pred^o = 2$ therefore concentrates its posterior at $N_E = -1$ and changes the decision to a_2. Under the true SCM,

$$\mathbb{E}[Y \mid X = 1, do(A = 1)] = 2, \qquad \mathbb{E}[Y \mid X = 1, do(A = 0)] = 1.$$

The update lowers the expected outcome by one despite perfect predictive calibration. This exact calculation concerns the population predictor and the limiting prior; it does not assert that every finite-sample OLS fit is perfectly calibrated.

What disclosure changes. If the historical treatment mean is disclosed and the agent instead fixes $\alpha_{A\mu} = 1$, the same observation obeys $Pred^o = 1 + N_E$, giving $N_E = 1$ and the beneficial action a_1. The framework thus not only diagnoses the problem but also points to which prior should be corrected to steer the agent toward the beneficial action.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The four contributions of the paper, the framework, the tractability theorem and the two main experimental findings, are all accurately introduced in the Abstract and Introduction. The framework is described in Section 2, the theorem is outlined in Subsection 2.5 and described in full detail in Appendix F. The Experimental setup and results are described in Sections 3 and 4. Further details of experimental design are provided in Appendix B. The limitations of the contributions are discussed in the Discussion section, clarifying the scope of the claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are discussed in a dedicated part of the Discussion section. The assumptions of the theorem are stated both in the main text (informally) and detailed in the corresponding section of the Appendix. The scope of the experiments mirrors the scope of the Theorem, as mentioned explicitly in the text.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The full description of the assumptions and the proof of Theorem 2.8 are given in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experiments are described in Section 3 and all details about the design and implementation that did not fit in the main body are reported in Appendix B. Further, we provided access to the code used to run the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code used to run the experiments is provided in the submission (`https://github.com/thelogicalgrammar/CapturingUsersPredictionModels`). The paper provides a detailed account of the design and implementation.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and test details are provided in the paper in Section 3 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Fig. 3, Fig. 4 and the appendix figures summarize CATE distributions by means and bands of plus or minus one standard deviation, and downstream outcomes by boxplots. The CATE bands combine within-replicate uncertainty with variation between replicate means. They are not standard errors of the aggregate mean.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Information is provided in Section B.1 in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors reviewed the Code and confirm the paper complies.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Despite its formal nature, social impact is a core theme of the paper. The paper captures a previously underexplored learning mechanism for agents interacting with ML-driven decision support, and shows that–when specific misalignments of agent beliefs happen–this type of tool can lead to harmful consequences even in very ideal conditions (simple ML model, simple causal mechanism, fully rational agent, etc.). The paper clearly states that this failure mode applies to all treatment-naive prognostic decision support tools that fit our definition, in all fields of application, including high-stakes ones such as healthcare or law enforcement. Furthermore, the paper stresses the need to counter this failure mode by properly training the users of such decision support tools. The potential positive impact of the paper is a heightened attention to the themes above, as well as suggestions on how to conduct pre-deployment audits.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper is not releasing a new model or data.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The key packages used are properly referenced in the paper (e.g. PyMC), and the code is released under the MIT license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code used to generate the experiments is released alongside the paper, and it is properly documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only used for editing and formatting.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.