

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

Otto Nyberg*

Department of Medical Informatics
Amsterdam UMC
University of Amsterdam
the Netherlands
o.e.nyberg@amsterdamumc.nl

Fausto Carcassi*

Institute for Logic, Language and Computation
University of Amsterdam
the Netherlands

Davide Tugnoli

Department of Mathematics and Earth Sciences
University of Trieste
Italy

Giovanni Cinà*

Department of Medical Informatics
Amsterdam UMC
University of Amsterdam
the Netherlands

Abstract

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 this process. As a prediction from an ML model contains information about the training data, a prediction can also be used for inference. Our framework models (i) how a prediction for a new observation affects the beliefs of a rational Bayesian agent, and (ii) how this change in beliefs affects the estimation of causal effect, the downstream decision, and the subsequent outcome. In addition to the framework itself, we make three contributions. First, for the linear Gaussian setting, we derive a tractable solution for the challenging Bayesian inference problem we introduced, i.e. one in which the agent infers from an ML prediction. Second, we experimentally identify conditions under which ML-DS is beneficial. Third, we show that a single misaligned prior belief can be sufficient for ML-DS to lead to worse downstream outcomes compared to no decision support *even when the ML 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 can take into account when deciding how to act. We refer to such decision support as ML-DS (Machine Learning Decision Support). This is sometimes also called ‘predictive optimization’ [Wang et al., 2024]. Contrary to full automation, in high-risk settings ML-DS is sometimes perceived as a way to leverage the power of machine learning while retaining human oversight (see for instance Article 14 of the EU AI Act European Parliament and Council of the European Union [2024]).

Understanding the practical impact of adopting ML-DS is essential for at least two reasons. First, ML-DS is on the rise and currently deployed in a variety of fields including healthcare, education

*Equal contribution

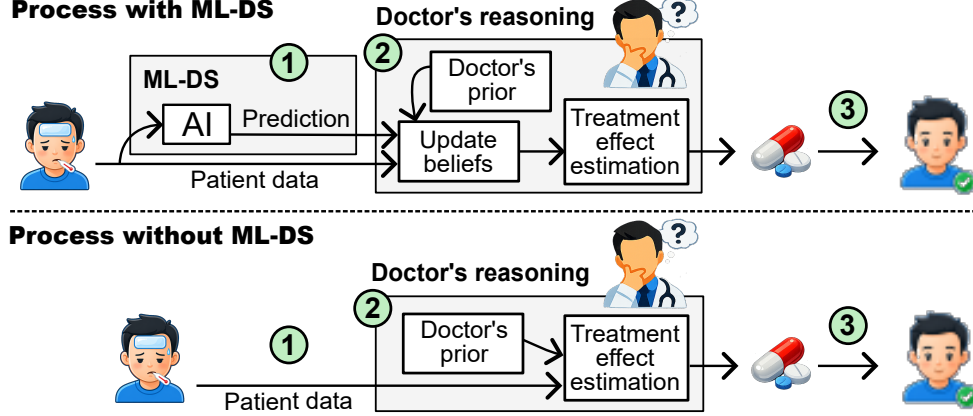


Figure 1: The flow of the framework for a clinical use case, with and without ML-DS (above and below, respectively). A new patient arrives for treatment, and the patient’s data (1), potentially with the output of ML-DS, is observed by the agent, e.g., a doctor. Next, the agent estimates the effect of 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 previous beliefs about the population in a Bayesian fashion. The treatment and the patient’s attributes determine the outcome (3).

[Smith et al., 2012], policing and judicial system [Kleinberg et al., 2018], as well as redistribution of welfare [Kube et al., 2019]. 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.

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 distribution the ML model was trained on. The ML 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 from side-effects.

This example showcases that, even in 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. 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. On the other hand, they might conclude they were wrong about the effect of treatment. In this case, the treatment must be

²Setting the agent prior beliefs by proper training is indeed an important part of RCTs’ protocols that is not always adhered to Pereplechikova 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].

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 of ML-DS, the *2-Step Agent* which includes a Bayesian step to incorporate the information given 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. To the best of our knowledge, this is the first treatment of how a Bayesian agent can update their beliefs about a population and treatment mechanism after observing a prediction from an ML model trained on data from that population. This update involves a challenging inference through a set of latent interchangeable 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 in a simple setting it is possible to reduce the interchangeable variables to a few sufficient statistics, reducing computational cost and improving numerical stability. Second, we identify experimentally some cases in which the learning from ML-DS always leads to better effect estimates and better outcomes compared to no decision support. 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 the learning from ML-DS and lead the agent to taking suboptimal actions, resulting in worse outcomes in comparison to not using decision support.

While our running example is medical, the results are general and apply for every field where prediction models are employed for decision support. Furthermore, 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://placeholder_link.

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 (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 an outcome of interest. 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_i to refer to covariates 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 $X = X_1, \dots, X_k$ 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 Figure 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 unique joint distribution P^{Hist} over variables Y, A, X .

Definition 2.2 (Training data and prediction model). The dataset $\mathcal{D}_{Hist} = \{(x_1, a_1, y_1), \dots, (x_n, a_n, y_n)\}$ 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 $f : X \rightarrow Y$ to estimate $E^{Hist}(Y | X)$.

Our framework does not currently extend to predictions of the treatment dependent estimate $E^{Hist}(Y | A, X)$ or the interventional estimate $E^{Hist}(Y | do(A), X)$; 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 they want to act on. We model an agent who is aware of the model documentation containing the model *class* (e.g. a linear model, a tree ensemble, etc.), the model’s *signature* (i.e. that it is treatment naive and what covariates are involved), and the size of the training data.

Definition 2.3 (Beliefs of the agent, $\mathcal{A}_{hist}$). The Bayesian network defining the agent’s joint distribution is denoted $\mathcal{A}_{hist}$. The distribution is over the following variables (depicted in Figure 2 (b)):

1. $\alpha_X, \alpha_Y, \alpha_A$: These are the variables fixing the population-level distributions over the various quantities of interest, respectively: the patient’s 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. $N_{X,i}, N_{Y,i}, N_{A,i}$: Within the Bayesian network, the historical data is represented via a plate. The α variables are fixed and outside the plate, capturing the fact that they are population-level distributions. These population-level distributions are sampled n times, producing $N_{X,i}, N_{Y,i}, N_{A,i}$ (with $i < n$). These encode the stochastic components of each participant that the predictive model was trained on.
3. A_i, Y_i, X_i : The stochastic components $N_{X,i}, N_{Y,i}, N_{A,i}$ are transformed deterministically into the specific patient characteristic (X_i), treatment (A_i) and outcome (Y_i), which become a training datapoint 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 variables in the plate, representing the fact that it is trained on the whole data set.
6. X^o : An observed data instance for a new patient, for which the agent will make a treatment decision. This array of variables 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 .

The Bayesian network is shown in Figure 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.

³Recent guidelines suggest to report the historical treatment policy 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.

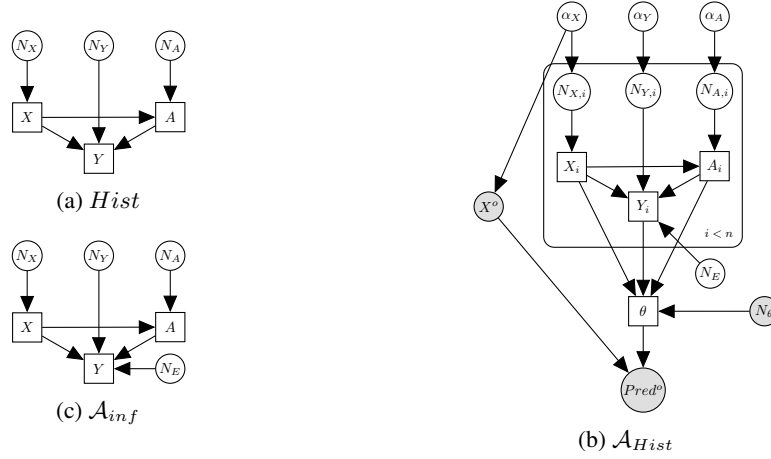


Figure 2: The SCMs involved in the paper with stochastic variables depicted as circles and deterministic variables as rectangles. Model (a) represents the SCM generating the data in the real world, model (b) represents the Agent’s beliefs of how the ML-DS model was trained, allowing the agent to perform a Bayesian belief update given new information from ML-DS, propagating information from observed variables (shaded) to variables on which there is uncertainty. Model (c) allows the agent to infer the treatment effect of new data points given the beliefs. Models (b) and (c) are the first and second steps of the 2-Step Agent.

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.

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 $\alpha_X, \alpha_Y, \alpha_A, N_E$, 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 ($N_{X,i}, N_{Y,i}, N_{A,i}, X_i, Y_i, A_i$ with $i < n$) and θ .

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.

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 Figure 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:

$$\alpha_X, \alpha_Y, \alpha_A, N_E \sim P^{\mathcal{A}_{hist}}(\alpha_X, \alpha_Y, \alpha_A, N_E \mid X^o, Pred^o)$$

Each draw generates a different SCM. 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. $\mathbb{E}_{P^{\mathcal{A}_{inf}}}(\text{CATE}) > \tau$ where τ is a threshold set by the agent and

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

Crucially, these interventional expectations are calculated based on the beliefs of the agent.

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). The effect of ML-DS is then defined as

$$\mathbb{E}^{Hist}(Y \mid do(A = \text{ML-DS})) - \mathbb{E}^{Hist}(Y \mid do(A = \text{no ML-DS}))$$

where $do(A = \text{ML-DS})$ is a shortcut to denote the action chosen with the intervention of providing ML-DS to the agent, and $do(A = \text{no ML-DS})$ the action chosen when ML-DS is not provided. This intervention affects the outcome through its effect on treatment selection.

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 in the first step: approximate Bayesian inference struggles to propagate through a plate of completely interchangeable variables. In what follows we show that in simple settings this problem can be solved reducing the plate to sufficient statistics.

Theorem 2.8 (Informal). *Let $Hist$ be as depicted in Fig. 2(a), where X is a vector of k independent confounders. 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 a linear regression model. Then the likelihood of θ in $\mathcal{A}_{hist}$ can be written as a function of the population-level variables (the variables outside the plate) plus $\frac{k^2 + 5k + 2}{2}$ sufficient statistics, where k is the dimensionality of X .*

We refer the reader to Appendix E for the full proof and Corollary E.2 covering the no-intercept case. 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 with no bias term We study what happens in the simplest of settings, when the ML-DS model is a linear regression model without bias term (see Corollary E.2) and consider a data-generating model with one continuous covariate X , a continuous intervention A , and a continuous outcome Y . Higher values of Y are better. 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 set $n = 1000$. 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 Figure 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*:

$$\left. \begin{array}{l} N_{X_i} \sim \mathcal{N}(\alpha_{X\mu}, \alpha_{X\sigma}) \\ N_{A_i} \sim \mathcal{N}(\alpha_{A\mu}, \alpha_{A\sigma}) \\ N_{Y_i} \sim \mathcal{N}(\alpha_{Y\mu}, \alpha_{Y\sigma}) \\ X_i = N_{X_i} \\ A_i = 0.125 X_i + N_{A_i} \\ Y_i = 12 - 0.1 X_i + N_E A_i + N_{Y_i} \end{array} \right\} i < n \quad \left| \quad \begin{array}{l} \theta = \underset{\theta}{\operatorname{argmin}}(MSE(\theta, \vec{X}, \vec{Y})) \\ X^o \sim \mathcal{N}(\alpha_{X\mu}, \alpha_{X\sigma}) \\ Pred^o = \theta X^o \end{array} \right.$$

where $\vec{Y} = (Y_1, \dots, Y_n)$ etc. are the respective vectors of n elements, θ is the single parameter of a least squares regression model without intercept, 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 RVs that the agent has uncertainty over (like N_E).

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

$$\begin{array}{ll} \alpha_{X\mu} \sim \mathcal{N}(\mathbf{80}, 0.1) & \alpha_{X\sigma} \sim \mathcal{N}(10, 0.1) \mid \alpha_{X\sigma} \geq 0 \\ \alpha_{A\mu} \sim \mathcal{N}(\mathbf{2}, 1) & \alpha_{A\sigma} \sim \mathcal{N}(1, 0.1) \mid \alpha_{A\sigma} \geq 0 \\ \alpha_{Y\mu} \sim \mathcal{N}(\mathbf{0}, 0.01) & \alpha_{Y\sigma} \sim \mathcal{N}(0.1, 0.01) \mid \alpha_{Y\sigma} \geq 0 \\ N_E \sim \mathcal{N}(\mathbf{1}, 1) & \end{array}$$

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 colored parameters, the means for $p(\alpha_{X\mu})$, $p(\alpha_{A\mu})$, $p(\alpha_{Y\mu})$, and $p(N_E)$, 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 Figure 2(c). The posterior defines a probability distribution over such SCMs. The agent is interested in choosing between dosages $A = 10$ and $A = 20$ for the patient with weight equal to X^o . To compute the desired quantity, the agent aggregates the CATE produced by every SCM according to the posterior

$$\widehat{CATE} = \mathbb{E}_{P^{\mathcal{A}_{inf}}} \left[\mathbb{E}^{\mathcal{A}_{inf}}(Y|X^o, do(A=20)) - \mathbb{E}^{\mathcal{A}_{inf}}(Y|X^o, do(A=10)) \right]$$

where the expectation is over the joint posterior of the parameters, which was obtained updating $\mathcal{A}_{hist}$. The agent will administer a dosage of 10 if the estimated CATE is below a threshold of 5, and a dosage of 20 otherwise. In the example, the higher dosage needs to improve the outcome by at least 5 months to be worth it. 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. We present the results in Figures 3 and 5. For more details of the setup see Appendix B.

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.

We report the experimental results on two dimensions: 1) the agent’s belief of treatment effect and 2) the downstream outcomes Y realized after the agent has taken an action, visualized in the top and bottom rows of Figure 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 CATE plots sweep a wide range of options for the priors in order to reveal interesting dynamics of the agent’s beliefs.

Figure 3a and Figure 3d show what happens to the belief of CATE and the actual outcome Y when the agent’s belief in N_E is misaligned. As expected, in Figure 3 (a) we can see that the prior predictive (blue) CATE has a slope that depends on the prior N_E : without decision support, the belief of CATE is equal to 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 in CATE re-aligning it to the correct value. In other words, *when all other beliefs are correct, the agent can learn from ML-DS to estimate better CATE* and take better actions. Figure 3d confirms that the orange series shows superior outcomes.

Figure 3b and Figure 3e show what happens when the agent’s prior belief in the treatment protocol, $\alpha_{A\mu}$, varies. In this scenario the prior CATE estimates are flat for the simple reason that the agent has a fixed belief in N_E and they are not updating it. When interacting with ML-DS however, the belief update changes CATE estimates quite a lot; this is because the agent is attempting to explain the gap between $Pred^o$ given by ML-DS and the expected outcome they had from their prior beliefs. This reflects on outcomes in 3e where the orange series is sometimes below the blue one: incorrect beliefs in $\alpha_{A\mu}$ can result in CATE estimates that lead to wrong actions and worse outcome Y .

A similar pattern is observed in Figures 3c, 3f, and 5a (Appendix D): while prior CATE and Y estimates are relatively stable without ML-DS, new dynamics emerge when ML-DS is introduced. For CATE, incorrect beliefs can have large effects both on the magnitude and sign. For Y , in both cases incorrect beliefs can lead to bad actions and worse outcomes.

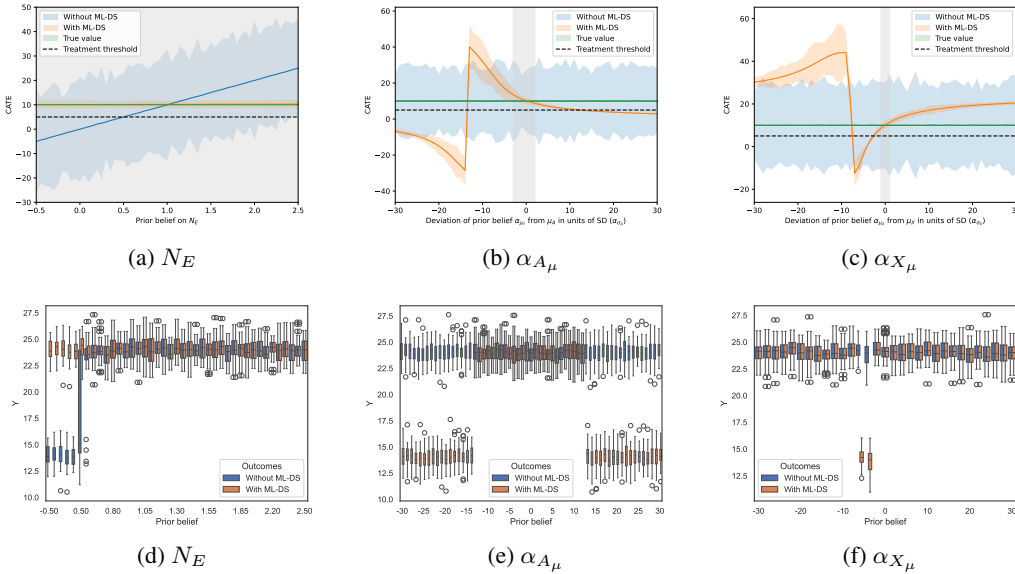


Figure 3: The effects of different agent priors when using ML-DS (in orange) and without (in blue). In all plots except for (a) and (d), the x-axis represents the deviation of the agent’s prior belief from the true parameters in $Hist$. The top row displays the agent’s prior and posterior (predictive) belief of CATE, while the bottom row shows the impact of the agent’s decisions on the downstream outcome Y . The left column pertains to the belief on treatment effect, N_E , the central column the belief about past treatment policy ($\alpha_{A\mu}$) and the right column about the belief on past covariate distribution ($\alpha_{X\mu}$). The gray areas represent the prior settings for which the true CATE is within one std from the CATE learned with ML-DS (orange band).

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 instead focuses on the decision-maker, showing that an accurate 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 modeling the decision-maker as a Bayesian agent, though one whose beliefs concern only the reliability of the prediction’s confidence intervals. 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.

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 interchangeable 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, so-called *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, e.g., treatment effect.

6 Discussion

We have presented the first 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 agent’s uncertainty. The framework allows for simulations comparing outcomes with and without ML-DS in an RCT-like fashion, quantifying the harm/benefit of ML-DS.

The framework defines a challenging Bayesian inference problem, for which we derived a tractable solution in a simple setting (Theorem 2.8). This result enabled us to simulate end-to-end the effect of decision support for different types of agents. With this implementation, we show that an incorrect belief about treatment effect is in some settings corrected by the interaction with ML-DS (Figure 3(a)). We also show that incorrect beliefs about the distribution of the predictive model’s training data may lead the agent to bad decisions, resulting in harmful outcomes even in near-ideal conditions. While previous literature had argued that this would happen with incorrect beliefs about past treatment policy ($\alpha_{A\mu}$ in our notation), e.g. in van Geloven et al. [2025], this is the first quantitative estimation of harmful effects of ML-DS due to misaligned beliefs. Additionally, our experiments show that ML-DS can be harmful if the agent has wrong beliefs not only about the past treatment policy, but also about the distribution of covariates and the distribution of the outcome.

These findings highlight the sensitivity of ML-DS to prior beliefs when using a treatment-naive prediction model, and underscore the importance of providing full documentation and proper training to users of ML-DS. Finally, we remark that these results apply to *any* field deploying such systems and to any agent interacting with ML-DS, be it human or artificial.

Limitations and future work. The experiments in this paper explored only part of the potential of the framework; they are limited to the case in which the ML-DS model is linear and treatment-naive and the data generating process is simple. 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 outcome 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, the Bayesian assumption and the simplicity of the use case *strengthen* our negative experimental results: harmful interactions with ML-DS due to misaligned priors cannot be fixed by making the agent more rational or streamlining the case.

Our theory describes what happens after a single interaction of an agent with ML-DS and does not yet cover repeated interactions. When running simulations, we reported the harm/benefit of ML-DS when used by multiple copies of the same agent on different observations: for each configuration of prior beliefs, we run the framework for different samples of X° . This would be analogous to, given a set of clinicians sharing the same starting beliefs (e.g. from the same department), tracking what happens to the first patients treated by these doctors with/without ML-DS.

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 (i.e. the effect of the action is not the same for all instances), iv) repeated rounds of learning for the same agent and v) using causal models for ML-DS.

Acknowledgements

We would like to thank Dr. Wouter van Amsterdam (UMC Utrecht) and Dr. Nan van Geloven (Leiden UMC) for their feedback on draft versions of the paper. This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-16946.

References

- Oriol Abril-Pla, Virgile Andreani, Colin Carroll, Larry Dong, Christopher J. Fonnesebeck, 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. *Advances in Neural Information Processing Systems*, 37:138088–138129, 2024.
- 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, September 2015. ISSN 1747-8405. 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 *Causal Learning and Reasoning*, pages 551–569. PMLR, 2024.

- George F Borm, René JF 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.
- W. G. Cochran. The distribution of quadratic forms in a normal system, with applications to the analysis of covariance. *Mathematical Proceedings of the Cambridge Philosophical Society*, 30(2): 178–191, 1934. doi: 10.1017/S0305004100016595.
- Gary S Collins, Karel GM 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, 2024.
- Giovanni De Toni, Nastaran Okati, Suhas Thejaswi, Eleni Straitouri, and Manuel Gomez-Rodriguez. Towards human-ai complementarity with prediction sets. *Advances in Neural Information Processing Systems*, 37:31380–31409, 2024.
- 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.
- European Parliament and Council of the European Union. Regulation (eu) 2024/1689 of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act), 2024. URL <https://artificialintelligenceact.eu/article/14/>. Published 12 July 2024; Article 14: Human oversight.
- 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.
- 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, September 2022. ISSN 0360-0300. 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.
- Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6): 2590–2615, 2011.
- Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293, 2018.
- Amanda Kube, Sanmay Das, and Patrick J Fowler. Allocating interventions based on predicted outcomes: A case study on homelessness services. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 622–629, 2019.
- 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.
- Bryce McLaughlin and Jann Spiess. Designing algorithmic recommendations to achieve human-ai complementarity. *arXiv preprint arXiv:2405.01484*, 2024.
- 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.

- Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, 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, 2007.
- Drago Plecko and Elias Bareinboim. Mind the gap: A causal perspective on bias amplification in prediction & decision-making. *Advances in Neural Information Processing Systems*, 37:84384–84408, 2024.
- 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.
- Mats J Stensrud, JD Laurendeau, and Aaron L Sarvet. Optimal regimes for algorithm-assisted human decision-making. *Biometrika*, 111(4):1089–1108, 2024.
- Mark Steyvers and Aakriti Kumar. Three challenges for ai-assisted decision-making. *Perspectives on Psychological Science*, 19(5):722–734, 2024.
- Eleni Straitouri, Suhas Thejaswi, and Manuel G Rodriguez. Controlling counterfactual harm in decision support systems based on prediction sets. *Advances in Neural Information Processing Systems*, 37:129443–129479, 2024.
- 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.
- Angelina Wang, Sayash Kapoor, Solon Barocas, and Arvind Narayanan. Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy. *ACM J. Responsib. Comput.*, 1(1):9:1–9:45, March 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.
- Chi Song Wong, Hua Cheng, and Joe Masaro. Multivariate versions of cochrane theorems. *Linear Algebra and its Applications*, 291(1–3):227–234, 1999. doi: 10.1016/S0024-3795(98)10239-2.
- Junzhe Zhang and Elias Bareinboim. Can humans be out of the loop? In *Conference on Causal Learning and Reasoning*, pages 1010–1025. PMLR, 2022.

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 no treatment ($\alpha_{A\mu} = 0$), 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 Figure 4.

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.

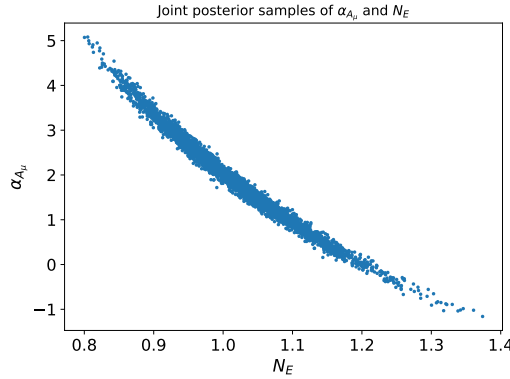


Figure 4: Plot showing the correlation of variables N_E and $\alpha_{A\mu}$ in the posterior beliefs of the agent after update with ML-DS information. The correlation reflects that the agent entertains multiple possible explanations for the surprising prediction: either treatment is less effective than expected, or 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 Section 2.1.

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

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

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 .

The model of the agent’s beliefs. In the PyMC implementation, we also use a small ($1e - 3$) observational noise around $Pred^o$ to allow us to sample from an observed deterministic transformation.

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

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

B.1 Implementation details

The experiments were run on a high performance computing cluster. Every node had 4 CPUs (no GPUs) and 7 GB of RAM. A single-node job took between 2 and 60 minutes to complete, and we ran 12,200 such jobs in total. We estimate that we used 40,000 CPU-hours on the cluster for the final run 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 only 16 out of 4000 runs had > 5 divergences, due to fixed ‘target_accept’ and extreme parameter values). For every value of a misaligned prior, we ran 50 such jobs with different X^o values sampled from the true distribution. Sanity checks on the Bayesian model are expanded upon in Appendix C. The exact translation of the theory in the code is detailed in Appendix F, after the main proofs of Appendix E.

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 sufficient statistics 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 a single observation $X^o = 200$ that is much higher than the prior mean of 80 (with standard deviation 10). We verify that after the update the posterior of $\alpha_{X\mu}$ has moved in the right direction. We also check if this creates correlations of $\alpha_{X\mu}$ with other parameters. This is repeated for different plate sizes.
3. We run an update with many (1000) observations drawn from the real distribution of X and check that the (incorrect) prior of the agent converges to the real parameter μ_X in the generating process.
4. 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 $Y = \text{coeff} * X + N_E * A$. We also set the coeff 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 obtained when varying prior $\alpha_{Y\mu}$

In Figure 5 we report the CATE and Y plots obtained when varying the belief of the agent concerning the mean of the outcome in the historical population, $\alpha_{Y\mu}$. It can be observed that CATE with ML-DS diverges from the true CATE as misalignment becomes more extreme (a), although in this range this is not enough to observe different outcome on the Y plot (b).

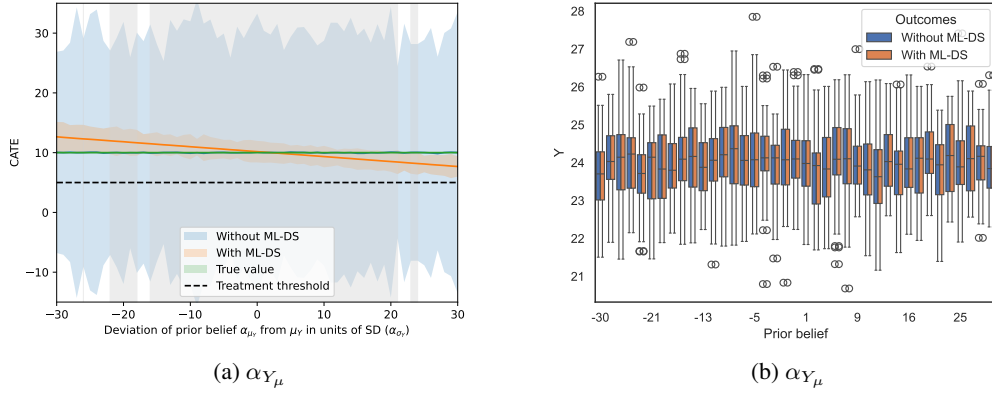


Figure 5: Figures displaying the effects of different agent's priors when using ML-DS (orange) and without it (blue). In all plots, the x-axis represents the deviation of the agent's prior belief from the historical SCM's parameter for $\alpha_{Y\mu}$. The left plot displays the agent's prior and posterior (predictive) belief of CATE, while the right one shows the impact of the agent's decisions on the downstream outcome Y . In a rather wide band is the posterior within 1 std of the true mean (gray area).

E Plate reduction

In this section we show how the plate for ordinary least squares regression was reduced, resulting in us having to sample only a handful of variables rather than the full plate. This makes our method both tractable and numerically more stable. We begin by re-stating Theorem 2.8; the proof follows.

Theorem E.1 (Reduction of the plate to sufficient statistics). *Let the historical DAG be as depicted in Fig. 2(a) and let X be a vector of k different independent variables (all confounders with respect to A and Y). 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. *all noises are Gaussian;*
4. *the ML model is a linear regression with intercept predicting Y from X ;*
5. *$n \geq k + 1$, i.e. there are at least $k + 1$ data points to train the linear model*
6. *the agent has perfect knowledge of the historical DAG and of the functional form of the structural equations;*
7. *the agent is a Bayesian reasoner who knows the number of data points in the training data, the model signature and the model class.*

Then the likelihood of θ (the parameters of the ML model) in $\mathcal{A}_{hist}$ can be written as a function of the population-level variables (those outside the plate) plus $\frac{k^2+5k+2}{2}$ sufficient statistics, where k is the number of confounders in the DAG.

We note that we do *not* assume the agent to know the exact coefficients of the SCM (see Subsection E.0.3 below for the exact formulation of the equations in $\mathcal{A}_{hist}$), we allow for k confounders and the ML model has intercept. This proof therefore covers a more general case than the use case presented in the paper for the experiments.

We also provide an analogous proof for the case without the intercept in Corollary E.2 in Section E.8 below.

E.0.1 Notation and setup

E.0.2 The vector $\mathbf{1}$ and the projection matrices

Throughout this proof we use the vector $\mathbf{1} \in \mathbb{R}^n$, a column vector with all entries equal to 1. We define two projection matrices:

$$P := \frac{1}{n} \mathbf{1} \mathbf{1}^\top, \quad M := I_n - P. \quad (\text{Proj})$$

P computes the mean: when applied to a vector, every component of the result equals the mean of the original vector. **M subtracts the mean** (centering): $Mv = v - Pv$, so every component of Mv is the original value minus the mean.

Properties of M (used throughout):

$$M\mathbf{1} = 0, \quad M^2 = M \text{ (idempotent)}, \quad M^\top = M \text{ (symmetric)}. \quad (\text{M-prop})$$

$M\mathbf{1} = 0$: a constant vector has mean equal to itself; subtracting the mean gives zero. $M^2 = M$: centering data that is already centered does nothing. M is the projection onto the orthogonal complement of $\text{span}(\mathbf{1})$. P projects onto $\text{span}(\mathbf{1})$. They are orthogonal: $PM = 0$.

E.0.3 The SCM with k covariates

For each patient $i \in \{1, \dots, n\}$, where X_i is now a vector of k covariates:

$$X_i \in \mathbb{R}^k, \quad X_i = N_{X_i} \quad (\text{SCM-0})$$

$$A_i = N_d^\top X_i + N_{A_i} \quad (\text{SCM-1})$$

$$Y_i = N_b^\top X_i + N_E A_i + N_{Y_i} \quad (\text{SCM-2})$$

where $N_b, N_d \in \mathbb{R}^k$ and $N_E \in \mathbb{R}$ are population-level random variables (sampled once, not per patient). In the agent's Bayesian model, each of them has a prior and is inferred from data.

The noise distributions are:

$$\begin{aligned} (N_{X_i})_j &\sim \mathcal{N}(\alpha_{X\mu_j}, \alpha_{X\sigma_j}^2) \quad \text{independently over } i \text{ and } j, \\ N_{A_i} &\sim \mathcal{N}(\alpha_{A\mu}, \alpha_{A\sigma}^2), \\ N_{Y_i} &\sim \mathcal{N}(\alpha_{Y\mu}, \alpha_{Y\sigma}^2), \end{aligned} \quad (\text{Noise})$$

all mutually independent across indices and across variables.

Remark (status of population-level parameters). The derivation below is written conditionally on the population-level parameters

$$(N_b, N_d, N_E, \{\alpha_{X\mu_j}, \alpha_{X\sigma_j}\}_{j=1}^k, \alpha_{A\mu}, \alpha_{A\sigma}, \alpha_{Y\mu}, \alpha_{Y\sigma}).$$

The auxiliary variables (s, W, g, h) and the formulas for $(\hat{\beta}_0, \hat{\beta}_1)$ are obtained for any fixed value of these parameters. In the agent's Bayesian model these parameters are themselves random variables with their own priors. The reduction therefore applies pointwise in the population-level parameters: for each posterior draw, the plate of n patients is replaced by a single draw of (s, W, g, h) .

E.0.4 Standardization of the noises

We define the vector of means and the diagonal matrix of standard deviations as

$$\alpha_{X_\mu} := \begin{pmatrix} \alpha_{X\mu_1} \\ \vdots \\ \alpha_{X\mu_k} \end{pmatrix} \in \mathbb{R}^k, \quad D_X := \text{diag}(\alpha_{X\sigma_1}, \dots, \alpha_{X\sigma_k}) \in \mathbb{R}^{k \times k}. \quad (\text{X-par})$$

We explicitly use vector notation for α_{X_μ} to reduce confusion. Each covariate vector can then be written as:

$$X_i = \alpha_{X_\mu} + D_X z_i, \quad z_i \stackrel{iid}{\sim} \mathcal{N}_k(0, I_k). \quad (\text{X-std})$$

The vector z_i is a k -dimensional standard normal. The matrix D_X scales each component by its standard deviation. Since the covariates are independent across components in this SCM, D_X is diagonal.

Similarly for the other noises:

$$N_{A_i} = \alpha_{A\mu} + \alpha_{A\sigma} \epsilon_{A_i}, \quad N_{Y_i} = \alpha_{Y\mu} + \alpha_{Y\sigma} \epsilon_{Y_i}, \quad \epsilon_{A_i}, \epsilon_{Y_i} \stackrel{iid}{\sim} \mathcal{N}(0, 1), \quad (\text{AY-std})$$

independent of all z_i .

E.0.5 Design matrix and OLS with intercept

Let $X \in \mathbb{R}^{n \times k}$ be the design matrix whose i -th row is $X_i^\top$, and $Y \in \mathbb{R}^n$ the outcome vector. The design matrix decomposes as:

$$X = \mathbf{1} \alpha_{X_\mu}^\top + Z D_X, \quad (\text{D1})$$

where $Z \in \mathbb{R}^{n \times k}$ has rows $z_i^\top$. Each row of $\mathbf{1} \alpha_{X_\mu}^\top$ is the same (the population mean); $Z D_X$ adds the patient-specific noise scaled by the standard deviations.

The OLS model with intercept is $Y \approx \beta_0 \mathbf{1} + X \beta_1$. The closed-form solution is:

$$\hat{\beta}_1 = (X^\top M X)^{-1} X^\top M Y, \quad \hat{\beta}_0 = \bar{Y} - \bar{X}^\top \hat{\beta}_1, \quad (\text{OLS})$$

where $\bar{X} := \frac{1}{n} X^\top \mathbf{1} \in \mathbb{R}^k$ and $\bar{Y} := \frac{1}{n} \mathbf{1}^\top Y \in \mathbb{R}$.

Given the assumption $n \geq k + 1$, $X^\top M X$ is invertible almost surely.

Our goal: express $(\hat{\beta}_0, \hat{\beta}_1)$ using a small set of auxiliary random variables, without sampling the n individual patients.

E.1 Why centering simplifies the problem

If one writes the block normal equations using the uncentered statistics $S_x = X^\top \mathbf{1}$, $S_{xx} = X^\top X$, $s_y = \mathbf{1}^\top Y$, $s_{xy} = X^\top Y$, the closed form is:

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = \begin{bmatrix} n & S_x^\top \\ S_x & S_{xx} \end{bmatrix}^{-1} \begin{bmatrix} s_y \\ s_{xy} \end{bmatrix}.$$

This is correct, but for our purposes it is *not* the cleanest route. The matrix $S_{xx} = X^\top X$ mixes the mean and the dispersion of the covariates. The centered formulation (OLS) avoids this by projecting out the mean direction with M . The key centered object is:

$$C_{xx} := X^\top M X. \quad (\text{B3})$$

We now show that C_{xx} factorizes cleanly.

E.2 Centering the design matrix

Apply M to the decomposition $X = \mathbf{1} \alpha_{X_\mu}^\top + Z D_X$:

$$M X = M(\mathbf{1} \alpha_{X_\mu}^\top + Z D_X) = \underbrace{M \mathbf{1}}_{=0} \alpha_{X_\mu}^\top + M Z \cdot D_X = M Z D_X. \quad (\text{D6})$$

Therefore:

$$\begin{aligned} C_{xx} &= X^\top M X = (M X)^\top (M X) \\ &= (M Z D_X)^\top (M Z D_X) = D_X Z^\top M^\top M Z D_X = D_X Z^\top M Z D_X. \end{aligned} \quad (\text{D7})$$

The step $(M X)^\top (M X) = X^\top M X$ uses three properties: transposing a product reverses the order ($(X^\top M^\top)$), M is symmetric ($M^\top = M$), and M is idempotent ($M^2 = M$). The step $D_X^\top = D_X$ holds because D_X is diagonal.

E.3 The auxiliary objects s and W

We next look into the auxiliary objects needed for the plate reduction and their properties.

E.3.1 First auxiliary object: s (sum of standardized noises)

Define:

$$s := Z^\top \mathbf{1} = \sum_{i=1}^n z_i \in \mathbb{R}^k. \quad (\text{D2})$$

Each column of Z contains n independent $\mathcal{N}(0, 1)$ entries. Summing n independent $\mathcal{N}(0, 1)$ variables gives $\mathcal{N}(0, n)$. The different columns are independent. Therefore:

$$s \sim \mathcal{N}_k(0, n I_k). \quad (\text{D3})$$

For $k = 1$, s reduces to the scalar $S_X = \sum \epsilon_{X_i}$. For later use, the sample mean of the covariates is:

$$\bar{X} = \alpha_{X_\mu} + \frac{1}{n} D_X s. \quad (\text{D11})$$

E.3.2 Second auxiliary object: W (centered Gram matrix)

Define:

$$W := Z^\top M Z \in \mathbb{R}^{k \times k}. \quad (\text{D8})$$

The matrix MZ is Z with column means subtracted (the centered noise table). $W = (MZ)^\top (MZ)$ is the Gram matrix of the centered noises:

- **Diagonal entries** W_{jj} : the centered sum of squares of the j -th noise column, $\sum_i (z_{ij} - \bar{z}_j)^2$. Measures how much variable j fluctuates around its mean.
- **Off-diagonal entries** W_{jl} : the centered cross-product between columns j and l , $\sum_i (z_{ij} - \bar{z}_j)(z_{il} - \bar{z}_l)$. Measures how variables j and l fluctuate together.

E.3.3 Distribution of W and independence from s

Standardization as a prerequisite. From (X-std) the rows of Z satisfy $z_i \stackrel{iid}{\sim} \mathcal{N}_k(0, I_k)$. This is the standard Gaussian setting required to apply the multivariate Cochran theorem. Recall that

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

Both P and M are symmetric idempotent projectors, they are complementary ($P + M = I_n$), and they are orthogonal to each other ($PM = 0$). This is exactly the projection setup underlying Cochran's theorem Cochran [1934] and its multivariate extension Wong et al. [1999]. By standard multivariate normal theory, orthogonal Gaussian projections are independent; therefore PZ and MZ are independent. Now observe that

$$PZ = \frac{1}{n} \mathbf{1}\mathbf{1}^\top Z = \frac{1}{n} \mathbf{1}s^\top, \quad W = Z^\top M Z = (MZ)^\top (MZ).$$

So s is a deterministic function of PZ , while W is a deterministic function of MZ . Since $PZ \perp MZ$, it follows that

$$s \perp W. \quad (\text{D9b})$$

Distribution of W . Let $U \in \mathbb{R}^{n \times (n-1)}$ be a matrix whose columns form an orthonormal basis of the orthogonal complement of $\text{span}(\mathbf{1})$, i.e. the subspace of vectors summing to zero. We denote this subspace by $\text{span}(\mathbf{1})^\perp$. Then

$$M = UU^\top, \quad U^\top U = I_{n-1}.$$

Hence

$$W = Z^\top M Z = Z^\top U U^\top Z = (U^\top Z)^\top (U^\top Z).$$

Because $U^\top$ is an orthonormal map onto the $n - 1$ centered directions and the rows of Z are i.i.d. $\mathcal{N}_k(0, I_k)$, the matrix $U^\top Z$ has $n - 1$ i.i.d. rows distributed as $\mathcal{N}_k(0, I_k)$. Therefore, by definition of the Wishart distribution,

$$W \sim \text{Wishart}_k(I_k, n - 1). \quad (\text{D9})$$

The Wishart distribution is the multivariate generalization of the chi-squared. If you take $n - 1$ independent draws $v_1, \dots, v_{n-1} \sim \mathcal{N}_k(0, I_k)$ and form $\sum_j v_j v_j^\top$, the result follows $\text{Wishart}_k(I_k, n - 1)$. For $k = 1$, each v_j is a scalar $\mathcal{N}(0, 1)$, and $\sum v_j^2 \sim \chi^2(n - 1)$.

Under the assumption $n \geq k + 1$, the degrees of freedom satisfy $n - 1 \geq k$, so W is symmetric positive definite almost surely. This is essential: if we tried to sample the diagonal (sums of squares) and off-diagonal (cross-products) entries independently, the resulting matrix could fail to be positive definite, which is mathematically impossible for a Gram matrix.

E.3.4 The centered Gram matrix of the covariates

Substituting (D8) into (D7):

$$\boxed{C_{xx} = D_X W D_X}. \quad (\text{D10})$$

This is the crucial simplification. The centered Gram matrix of the covariates factorizes into a product of known diagonal matrices (D_X) and a single random object (W) that we can sample directly.

The intercept *simplifies* the denominator: M eliminates the mean part, and only the Wishart part remains. This is why the off-diagonal cross-product issue does not arise.

E.4 Condensing A into Y

Substituting $A_i = N_d^\top X_i + \alpha_{A\mu} + \alpha_{A\sigma}\epsilon_{A_i}$ into SCM-2:

$$\begin{aligned} Y_i &= N_b^\top X_i + N_E(N_d^\top X_i + \alpha_{A\mu} + \alpha_{A\sigma}\epsilon_{A_i}) + \alpha_{Y\mu} + \alpha_{Y\sigma}\epsilon_{Y_i} \\ &= \underbrace{(\alpha_{Y\mu} + N_E\alpha_{A\mu})}_c + \underbrace{(N_b + N_EN_d)^\top}_{\beta^\top} X_i + \underbrace{N_E\alpha_{A\sigma}\epsilon_{A_i} + \alpha_{Y\sigma}\epsilon_{Y_i}}_{\varepsilon_i}. \end{aligned} \quad (C1)$$

The condensed noise ε_i is a linear combination of two independent $\mathcal{N}(0, 1)$ variables, hence

$$\begin{aligned} \varepsilon &\sim \mathcal{N}_n(0, \tau^2 I_n) \\ \tau^2 &= N_E^2 \alpha_{A\sigma}^2 + \alpha_{Y\sigma}^2 \\ Y &= c\mathbf{1} + X\beta + \varepsilon \end{aligned} \quad (C4)$$

Structural independence: ε_i depends on ϵ_{A_i} and ϵ_{Y_i} , which in the SCM are independent of all z_j (the covariate noises). Therefore:

$$\varepsilon \perp (Z, s, W, \text{and any function of } Z). \quad (\text{Indep})$$

E.5 Slope estimate

Substitute $Y = c\mathbf{1} + X\beta + \varepsilon$ into (OLS):

$$\begin{aligned} \hat{\beta}_1 &= (X^\top MX)^{-1} X^\top M(c\mathbf{1} + X\beta + \varepsilon) \\ &= (X^\top MX)^{-1} [c \underbrace{X^\top M\mathbf{1}}_{=0} + X^\top MX\beta + X^\top M\varepsilon] \\ &= \beta + C_{xx}^{-1} X^\top M\varepsilon. \end{aligned} \quad (S2)$$

The intercept c drops out because $M\mathbf{1} = 0$. The slope estimate equals the true slope β plus a noise term $C_{xx}^{-1} X^\top M\varepsilon$, which is a vector of k components. We need to reduce this noise term.

E.5.1 Step 1: conditional distribution of $X^\top M\varepsilon$

Given X , the matrix X is fixed. By (Indep), $\varepsilon \mid X \sim \mathcal{N}_n(0, \tau^2 I_n)$. The quantity $X^\top M\varepsilon$ is a linear transformation of a Gaussian vector, hence Gaussian:

$$\mathbb{E}[X^\top M\varepsilon \mid X] = 0, \quad (S4)$$

$$\text{Cov}(X^\top M\varepsilon \mid X) = X^\top M(\tau^2 I) M^\top X = \tau^2 X^\top MX = \tau^2 C_{xx}. \quad (S5)$$

The covariance uses the rule: if $u \sim \mathcal{N}(0, \Sigma)$ and A is a matrix, then $\text{Cov}(Au) = A\Sigma A^\top$. Here $u = \varepsilon$, $\Sigma = \tau^2 I$, $A = X^\top M$. Then $M^\top = M$ and $M^2 = M$ give $\tau^2 X^\top MX = \tau^2 C_{xx}$.

Therefore

$$X^\top M\varepsilon \mid X \sim \mathcal{N}_k(0, \tau^2 C_{xx}). \quad (S6)$$

The right-hand side depends on X *only through* $C_{xx} = D_X W D_X$, which is a deterministic function of W . So conditioning on W is sufficient:

$$X^\top M\varepsilon \mid W \sim \mathcal{N}_k(0, \tau^2 D_X W D_X). \quad (S7)$$

E.5.2 Step 2: standardize to obtain g (third auxiliary object)

Here $\Sigma = \tau^2 D_X W D_X$. The matrix square root is $\Sigma^{1/2} = \tau D_X W^{1/2}$:

Verification: $(\tau D_X W^{1/2})(\tau D_X W^{1/2})^\top = \tau^2 D_X W^{1/2}(W^{1/2})^\top D_X = \tau^2 D_X W D_X$, using $D_X^\top = D_X$ (diagonal) and $(W^{1/2})^\top = W^{1/2}$ (W is symmetric positive definite, so its square root is also symmetric).

Define $g \sim \mathcal{N}_k(0, I_k)$:

$$X^\top M\varepsilon = \tau D_X W^{1/2} g. \quad (S8)$$

$g \perp (s, W)$. Conditioning on Z and using $\varepsilon \perp Z$ from the SCM, we have $\varepsilon \mid Z \sim \mathcal{N}_n(0, \tau^2 I_n)$. The scaling $\tau D_X W^{1/2}$ in (S8) is Z -measurable, so its normalization yields $g \mid Z \sim \mathcal{N}_k(0, I_k)$. This conditional law does not depend on Z , hence $g \perp Z$, and in particular $g \perp (s, W)$.

E.5.3 Step 3: the slope formula

Substituting (S8) into (S2):

$$\begin{aligned}\hat{\beta}_1 &= \beta + (D_X W D_X)^{-1} \cdot \tau D_X W^{1/2} g \\ &= \beta + \tau D_X^{-1} W^{-1} D_X^{-1} \cdot D_X W^{1/2} g \\ &= \beta + \tau D_X^{-1} W^{-1/2} g.\end{aligned}\tag{S9}$$

The simplification: $(D_X W D_X)^{-1} = D_X^{-1} W^{-1} D_X^{-1}$. Then $D_X^{-1} D_X = I$ (cancel), and $W^{-1} W^{1/2} = W^{-1/2}$.

$$\boxed{\hat{\beta}_1 = \beta + \tau D_X^{-1} W^{-1/2} g}\tag{S10}$$

The slope depends only on W and g , but not on s (the sum of covariate noises, i.e. the mean).

E.6 Intercept estimate

From (OLS): $\hat{\beta}_0 = \bar{Y} - \bar{X}^\top \hat{\beta}_1$. Compute $\bar{Y}$ from (C4):

$$\bar{Y} = c + \bar{X}^\top \beta + \bar{\varepsilon}, \quad \bar{\varepsilon} := \frac{1}{n} \sum_i \varepsilon_i.$$

Substituting:

$$\begin{aligned}\hat{\beta}_0 &= c + \bar{X}^\top \beta + \bar{\varepsilon} - \bar{X}^\top \hat{\beta}_1 \\ &= c + \bar{\varepsilon} - \bar{X}^\top (\hat{\beta}_1 - \beta).\end{aligned}\tag{I3}$$

Using $\hat{\beta}_1 - \beta = \tau D_X^{-1} W^{-1/2} g$ from (S10):

$$\hat{\beta}_0 = c + \bar{\varepsilon} - \tau \bar{X}^\top D_X^{-1} W^{-1/2} g.\tag{I4}$$

E.6.1 Fourth auxiliary object: h (mean noise)

Each $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$ is independent. So $\sum_i \varepsilon_i \sim \mathcal{N}(0, n\tau^2)$. Standardize:

$$h := \frac{\sum_i \varepsilon_i}{\tau\sqrt{n}} \sim \mathcal{N}(0, 1), \quad \bar{\varepsilon} = \frac{\tau}{\sqrt{n}} h.\tag{I6}$$

$h \perp (s, W)$: h is a function of $P\varepsilon$, (s, W) are functions of Z , and $\varepsilon \perp Z$ in the SCM.

$h \perp g$: by Cochran on $\varepsilon \sim \mathcal{N}_n(0, \tau^2 I_n)$ with the complementary projectors P and M , $P\varepsilon \perp M\varepsilon$. By the SCM, $\varepsilon \perp Z$. Combining,

$$P\varepsilon \perp (M\varepsilon, Z).$$

Since h is a function of $P\varepsilon$ and both $X^\top M\varepsilon$ and W are functions of $(M\varepsilon, Z)$, this gives $h \perp (X^\top M\varepsilon, W)$. The representation (S8) determines g from $(X^\top M\varepsilon, W)$, so $h \perp g$.

E.6.2 The intercept formula

Using $\bar{X} = \alpha_{X_\mu} + \frac{1}{n} D_X s$ from (D11):

$$\begin{aligned}\hat{\beta}_0 &= c + \frac{\tau}{\sqrt{n}} h - \tau \left(\alpha_{X_\mu} + \frac{1}{n} D_X s \right)^\top D_X^{-1} W^{-1/2} g \\ &= c + \frac{\tau}{\sqrt{n}} h - \tau \left(\alpha_{X_\mu}^\top D_X^{-1} + \frac{1}{n} s^\top \right) W^{-1/2} g.\end{aligned}\tag{I7}$$

In the second line: $(D_X s)^\top D_X^{-1} = s^\top D_X^\top D_X^{-1} = s^\top D_X D_X^{-1} = s^\top$, using $D_X^\top = D_X$ (diagonal).

$$\boxed{\hat{\beta}_0 = c + \frac{\tau}{\sqrt{n}} h - \tau \left(\alpha_{X_\mu}^\top D_X^{-1} + \frac{1}{n} s^\top \right) W^{-1/2} g}\tag{I8}$$

Table 1: The sampling distributions of the reduced plate.

Object	Distribution	What it captures
$s \in \mathbb{R}^k$	$\mathcal{N}_k(0, nI_k)$	Sum of covariate noises
$W \in \mathbb{R}^{k \times k}$	$\text{Wishart}_k(I_k, n-1)$	Centered dispersion of covariates
$g \in \mathbb{R}^k$	$\mathcal{N}_k(0, I_k)$	Centered noise from A and Y
$h \in \mathbb{R}$	$\mathcal{N}(0, 1)$	Mean noise from A and Y

Table 2: The number of samples needed to generate one posterior sample using the reduced plate compared to using the full plate, without counting the population-level parameters outside the plate.

k	Reduced variables	Full plate ($n = 1000$)
1	4	3 000
2	8	4 000
5	26	7 000
10	76	12 000

E.7 Summary

E.7.1 The 4 auxiliary objects

The sampling distributions of the low-dimensional auxiliary variables are listed in Table 1. All four are **mutually independent**:

- $s \perp W$: Cochran (mean vs. centered part of $\{z_i\}$).
- $h \perp g$: Cochran (mean vs. centered part of $\{\varepsilon_i\}$).
- $(h, g) \perp (s, W)$: from the independence arguments established in Sections E.5.2 and E.6.1 (conditioning on Z , plus $\varepsilon \perp Z$ from the SCM).

E.7.2 Scalar degrees of freedom

$$\underbrace{k}_s + \underbrace{\frac{k(k+1)}{2}}_W + \underbrace{k}_g + \underbrace{1}_h = \frac{k^2 + 5k + 2}{2}.$$

Table 2 shows the number of sampling events needed to generate one posterior sample of $\hat{\beta}_1$ and $\hat{\beta}_0$. The number of samples needed using the reduced plate grows as $\mathcal{O}(k^2)$, whereas for the full plate it grows as $\mathcal{O}(kn)$. For $k \ll n$, the reduced plate is clearly preferable.

E.7.3 Final formulas

$$\hat{\beta}_1 = \beta + \tau D_X^{-1} W^{-1/2} g, \quad (1)$$

$$\hat{\beta}_0 = c + \frac{\tau}{\sqrt{n}} h - \tau \left(\alpha_{X_\mu}^\top D_X^{-1} + \frac{1}{n} s^\top \right) W^{-1/2} g, \quad (2)$$

where

$$\beta = N_b + N_E N_d, \quad c = \alpha_{Y_\mu} + N_E \alpha_{A_\mu}, \quad \tau^2 = N_E^2 \alpha_{A_\sigma}^2 + \alpha_{Y_\sigma}^2.$$

E.7.4 Connection with the uncentered sufficient statistics

The classical sufficient statistics can all be expressed from (s, W, g, h) :

$$S_x = n\alpha_{X_\mu} + D_X s, \quad (3)$$

$$S_{xx} = \frac{1}{n} S_x S_x^\top + D_X W D_X, \quad (4)$$

$$s_y = nc + S_x^\top \beta + \tau \sqrt{n} h, \quad (5)$$

$$s_{xy} = c S_x + S_{xx} \beta + \tau \left(\frac{1}{\sqrt{n}} S_x h + D_X W^{1/2} g \right). \quad (6)$$

E.8 Generalization to the no-intercept case

Corollary E.2 (No-intercept case). *Under the same assumptions as Theorem E.1, let the ML model be a linear model without intercept. Then the reduction to sufficient statistics involves $\frac{k^2+5k}{2}$ scalar sufficient statistics, where k is the number of independent confounders.*

The conclusion of Theorem E.1 holds with auxiliary objects (s, W, g, h) in the with-intercept case (as before) and $(s, W, \tilde{g})$ in the no-intercept case, where $\tilde{g} \sim \mathcal{N}_k(0, I_k)$ is defined in (S8') below. The total number of sufficient statistics is $\frac{k^2+5k+2}{2}$ in the first case and $\frac{k^2+5k}{2}$ in the second; the difference of one scalar is exactly the absent mean-noise term h .

E.8.1 Setup

The setup of Section E.0.3 and the design-matrix decomposition carry over verbatim. In particular, the SCM (SCM-0)–(SCM-2), the standardization (X-std)–(AY-std), the design-matrix decomposition $X = \mathbf{1}\alpha_{X_\mu}^\top + ZD_X$ from (D1), and the condensed form $Y = c\mathbf{1} + X\beta + \varepsilon$ with $\varepsilon \sim \mathcal{N}_n(0, \tau^2 I_n)$ and $\varepsilon \perp Z$ from (C4) and (Indep) are all unchanged.

Replace the OLS-with-intercept system of (OLS) by the slope-only OLS:

$$\hat{\phi} = (X^\top X)^{-1} X^\top Y \in \mathbb{R}^k. \quad (\text{OLS}')$$

Under Assumption 5 of Theorem E.1 ($n \geq k+1$), $X^\top X$ is invertible almost surely; this is verified explicitly in Step 1 below.

E.8.2 Step 1: the uncentered Gram and mean factor through (s, W)

The crucial observation is that, although the centering matrix M no longer appears, $X^\top X$ and $X^\top \mathbf{1}$ remain functions of (s, W) plus population-level parameters. Using $\mathbf{1}^\top \mathbf{1} = n$, $Z^\top \mathbf{1} = s$, and $Z^\top Z = Z^\top P Z + Z^\top M Z = \frac{1}{n} s s^\top + W$:

$$X^\top \mathbf{1} = n\alpha_{X_\mu} + D_X s =: S_x, \quad (\text{D1}')$$

$$\begin{aligned} X^\top X &= n\alpha_{X_\mu}\alpha_{X_\mu}^\top + \alpha_{X_\mu} s^\top D_X + D_X s \alpha_{X_\mu}^\top + D_X \left(\frac{1}{n} s s^\top + W \right) D_X \\ &= \frac{1}{n} (n\alpha_{X_\mu} + D_X s)(n\alpha_{X_\mu} + D_X s)^\top + D_X W D_X \\ &= \frac{1}{n} S_x S_x^\top + D_X W D_X. \end{aligned} \quad (\text{D7}')$$

The expressions (D1') and (D7') coincide with equations (3) and (4) of the ‘‘Connection with the uncentered sufficient statistics’’ subsection: the classical uncentered sufficient statistics (S_x, S_{xx}) are already expressed in terms of (s, W) and population-level parameters, so the no-intercept case requires no further reduction at this step.

Invertibility of $X^\top X$. Under Assumption 5, W is positive definite almost surely (as established for (D9)), so $D_X W D_X$ is positive definite. The rank-one term $\frac{1}{n} S_x S_x^\top$ is positive semidefinite. Their sum is therefore positive definite a.s., yielding invertibility of $X^\top X$.

E.8.3 Step 2: $X^\top \varepsilon$ via a fresh standard Gaussian

Substitute $Y = c\mathbf{1} + X\beta + \varepsilon$ into (OLS'):

$$\begin{aligned}\hat{\phi} &= (X^\top X)^{-1} X^\top (c\mathbf{1} + X\beta + \varepsilon) \\ &= \beta + c(X^\top X)^{-1} S_x + (X^\top X)^{-1} X^\top \varepsilon.\end{aligned}\tag{S2'}$$

For the noise term: by (Indep), $\varepsilon \perp Z$, hence $\varepsilon \mid Z \sim \mathcal{N}_n(0, \tau^2 I_n)$, and

$$X^\top \varepsilon \mid Z \sim \mathcal{N}_k(0, \tau^2 X^\top X).\tag{S6'}$$

The covariance is a deterministic function of (s, W) via (D7'), so conditioning on (s, W) is sufficient. Define

$$\tilde{g} := \tau^{-1} (X^\top X)^{-1/2} X^\top \varepsilon.\tag{S8'}$$

The conditional distribution $\tilde{g} \mid (s, W)$ is $\mathcal{N}_k(0, I_k)$ and does not depend on (s, W) . Hence $\tilde{g}$ is unconditionally $\mathcal{N}_k(0, I_k)$ and $\tilde{g} \perp (s, W)$. Equivalently:

$$X^\top \varepsilon = \tau (X^\top X)^{1/2} \tilde{g}, \quad \tilde{g} \sim \mathcal{N}_k(0, I_k), \quad \tilde{g} \perp (s, W).\tag{S9'}$$

The mean-noise object h of Section E.6.1 plays no role here: in the absence of an intercept estimate, there is no place for $\bar{\varepsilon}$ to enter.

E.8.4 Step 3: the slope-only formula

Substituting (S9') into (S2'):

$$\boxed{\hat{\phi} = \beta + c(X^\top X)^{-1} S_x + \tau (X^\top X)^{-1/2} \tilde{g}}\tag{S10'}$$

with

$$S_x = n \alpha_{X_\mu} + D_X s, \quad X^\top X = \frac{1}{n} S_x S_x^\top + D_X W D_X.$$

Sampling $(s, W, \tilde{g})$ once and applying (S10') reproduces a posterior draw of $\hat{\phi}$ without ever sampling the n -row plate.

E.8.5 Auxiliary objects, independence, degrees of freedom

The auxiliary objects in the no-intercept case are

$$s \in \mathbb{R}^k \sim \mathcal{N}_k(0, nI_k), \quad W \in \mathbb{R}^{k \times k} \sim \text{Wishart}_k(I_k, n-1), \quad \tilde{g} \in \mathbb{R}^k \sim \mathcal{N}_k(0, I_k).$$

The pairwise independence claims of the “4 auxiliary objects” subsection reduce to:

- $s \perp W$: Cochran on Z via the complementary projectors P, M , exactly as established for (D9b).
- $\tilde{g} \perp (s, W)$: established in Step 2; the conditional law of $\tilde{g}$ given (s, W) does not depend on (s, W) .

Total scalar degrees of freedom:

$$\underbrace{k}_s + \underbrace{\frac{k(k+1)}{2}}_W + \underbrace{k}_g = \frac{k^2 + 5k}{2}.$$

This is exactly $\frac{k^2 + 5k + 2}{2} - 1$: one fewer scalar than the with-intercept count, the missing scalar being h .

F Implementation of the theory in the code

We elaborate here on how the strategy to reduce the plate with the help of sufficient statistics is implemented in the code, since it employs a slightly different factorization of the OLS closed form with respect to what the Theorem and Corollary used in section E. As mentioned in Section 3, the experiment pertains to the use case with a single confounder X and a linear model without intercept.

As we are using PyMC for inference – which does not have an implementation for the non-central χ^2 -distribution – all random variables (RVs) will be reformulated using zero-centered normally distributed RVs (standard normals where needed) and (central) χ^2 distributed RVs.

We are specifically looking for a low-dimensional representation of the plate and treat a sample of the plate parameters $\{\alpha_{X\mu}, \alpha_{X\sigma}, \alpha_{A\mu}, \alpha_{A\sigma}, \alpha_{Y\mu}, \alpha_{Y\sigma}, N_E\}$ as given. The relevant parts of the SCM, reproduced here for the reader's convenience, are defined as

$$\begin{aligned} Y_i &= a + bX_i + N_E A_i + N_{Y_i} \\ A_i &= dX_i + N_{A_i} \\ X_i &= N_{X_i} \\ \forall i &\in [1, \dots, n] \end{aligned} \tag{7}$$

where n is the plate size, a , b , and d are constants and

$$\begin{aligned} N_{X_i} &\sim \mathcal{N}(\alpha_{X\mu}, \alpha_{X\sigma}^2) \\ N_{A_i} &\sim \mathcal{N}(\alpha_{A\mu}, \alpha_{A\sigma}^2) \\ N_{Y_i} &\sim \mathcal{N}(\alpha_{Y\mu}, \alpha_{Y\sigma}^2). \end{aligned} \tag{8}$$

We are modeling Y_i without intercept as

$$Y_i \approx \phi X_i \tag{9}$$

and looking for the ordinary least-squares estimate of ϕ , that is

$$\min_{\phi} \left(\sum_i (Y_i - \phi X_i)^2 \right). \tag{10}$$

The least-squares estimate for this case has a well-known closed-form solution given by

$$\phi = \frac{\sum_i X_i Y_i}{\sum_i X_i^2}. \tag{11}$$

Our goal is to express this equation in terms of RVs that summarize the sums thus removing the need to sample the full plate n times. The denominator of this equation can be expressed as

$$\sum_i X_i^2 = \sum_i N_{X_i}^2 \tag{12}$$

$\Leftrightarrow$

$$\sum_i X_i^2 = \sum_i \alpha_{X\mu}^2 + 2\alpha_{X\mu}\alpha_{X\sigma}\epsilon_{X_i} + \alpha_{X\sigma}^2\epsilon_{X_i}^2 \tag{13}$$

where

$$\epsilon_{X_i} \sim \mathcal{N}(0, 1). \tag{14}$$

The denominator of (11) rewritten in (13) consists of three parts:

$$S_1 = \sum_i \alpha_{X\mu}^2 \tag{15}$$

$$S_2 = \sum_i 2\alpha_{X\mu}\alpha_{X\sigma}\epsilon_{X_i} \tag{16}$$

$$S_3 = \sum_i \alpha_{X\sigma}^2\epsilon_{X_i}^2 \tag{17}$$

where

$$S_1 = n\alpha_{X\mu}^2 \quad (18)$$

$$S_2 = 2\alpha_{X\mu}\alpha_{X\sigma} \sum_i \epsilon_{X_i} \quad (19)$$

$$S_3 = \alpha_{X\sigma}^2 \sum_i \epsilon_{X_i}^2. \quad (20)$$

Given one set of parameters for the equations in (8), we see that S_1 is a constant, $S_2 = 2\alpha_{X\mu}\alpha_{X\sigma}S_X$ where $S_X \sim \mathcal{N}(0, n)$, and $S_3 \sim \alpha_{X\sigma}^2 Z_{S_3}$ where $Z_{S_3} \sim \chi^2(n)$. Here S_2 is the first low-dimensional auxiliary variable we need. Note that S_3 is dependent on S_2 and this has to be dealt with separately (see section F.1).

Next we decompose the numerator of equation (11) in a similar way. First we replace the RVs by the definitions from (7)

$$\sum_i X_i Y_i = \sum_i \left[aN_{X_i} + bN_{X_i}^2 + dN_E N_{X_i}^2 + N_E N_{A_i} N_{X_i} + N_{Y_i} N_{X_i} \right]. \quad (21)$$

As we are actually interested in equation (11), we can divide this expression by $\sum_i N_{X_i}^2$ (equation (12)) to cancel out some factors and get

$$\phi = b + dN_E + \frac{a \sum_i N_{X_i} + N_E \sum_i N_{X_i} N_{A_i} + \sum_i N_{X_i} N_{Y_i}}{\sum_i X_i^2} \quad (22)$$

Remember that a , b , and d are constants. The denominator of the last term is equivalent to equation (13). We currently lack suitable expressions only for the numerator of the last term. The numerator consists of the three parts

$$S_4 = a \sum_i N_{X_i} \quad (23)$$

$$S_5 = N_E \sum_i N_{X_i} N_{A_i} \quad (24)$$

$$S_6 = \sum_i N_{X_i} N_{Y_i}. \quad (25)$$

S_4 can be expressed similarly to S_2 using S_X , i.e. $S_4 = a n \alpha_{X\mu} + a \alpha_{X\sigma} S_X$. Both S_5 and S_6 are RVs following normal product distributions. We decompose S_6 as follows:

$$S_6 = \sum_i [(\alpha_{X\mu} + \alpha_{X\sigma} \epsilon_{X_i})(\alpha_{Y\mu} + \alpha_{Y\sigma} \epsilon_{Y_i})] \quad (26)$$

$\Leftrightarrow$

$$S_6 = \sum_i [\alpha_{X\mu} \alpha_{Y\mu} + \alpha_{X\mu} \alpha_{Y\sigma} \epsilon_{Y_i} + \alpha_{Y\mu} \alpha_{X\sigma} \epsilon_{X_i} + \alpha_{X\sigma} \alpha_{Y\sigma} \epsilon_{X_i} \epsilon_{Y_i}] \quad (27)$$

$\Leftrightarrow$

$$S_6 = n\alpha_{X\mu} \alpha_{Y\mu} + \alpha_{X\mu} \alpha_{Y\sigma} S_Y + \alpha_{Y\mu} \alpha_{X\sigma} S_X + \alpha_{X\sigma} \alpha_{Y\sigma} \sum_i \epsilon_{X_i} \epsilon_{Y_i} \quad (28)$$

where $\epsilon_{Y_i} \sim \mathcal{N}(0, 1)$ and $S_Y \sim \mathcal{N}(0, n)$. Note that S_X and S_Y are independent. We know that $\sum_i \epsilon_{X_i} \epsilon_{Y_i}$ follows a normal product distribution that is dependent on both S_X and S_Y and deal with this dependence in section F.2 below.

Following an almost identical derivation, we express

$$S_5 = N_E \left[n\alpha_{X\mu} \alpha_{A\mu} + \alpha_{X\mu} \alpha_{A\sigma} S_A + \alpha_{A\mu} \alpha_{X\sigma} S_X + \alpha_{X\sigma} \alpha_{A\sigma} \sum_i \epsilon_{X_i} \epsilon_{A_i} \right] \quad (29)$$

where

$$\epsilon_{A_i} \sim \mathcal{N}(0, 1) \quad (30)$$

$$S_A \sim \mathcal{N}(0, n). \quad (31)$$

F.1 Dependence of sum of squares and (square of) sum

Cochran's theorem Cochran [1934] states that the inner product $\vec{U}^\top \vec{W}$ of two vectors $\vec{U} \sim N(\vec{0}, I)$ and $\vec{W} \sim N(\vec{0}, I)$ can be separated into independent parts if there exists two $n \times n$ matrices B_1 and B_2 with eigenvalues taking only values 0 and 1 so that $\text{rank}(B_1) + \text{rank}(B_2) = n$. From the perspective of an eigenvalue decomposition, Cochran's theorem shows that the inner product can be simultaneously diagonalized, i.e.

$$\vec{U}^\top \vec{W} = \vec{U}^\top B_1 \vec{W} + \vec{U}^\top B_2 \vec{W} \quad (32)$$

$$= \vec{U}^\top V_1 \Lambda_1 V_1^{-1} \vec{W} + \vec{U}^\top V_2 \Lambda_2 V_2^{-1} \vec{W} \quad (33)$$

$$= \vec{U}^\top (V_1 + V_2) (\Lambda_1 + \Lambda_2) (V_1 + V_2)^{-1} \vec{W} \quad (34)$$

where $V_1 + V_2$ forms an orthonormal basis. Intuitively this means that there exists an orthonormal basis V that can be used to rotate $\vec{U}$ and $\vec{W}$ so that $\vec{U}^\top B_1 \vec{W}$ and $\vec{U}^\top B_2 \vec{W}$ are fully captured in mutually exclusive dimensions, and hence orthogonal. In practice it allows us to separate sums of squares into components that can be sampled separately to simplify our plate.

Next we show that the sum of squares can be split into two independent parts. Through algebraic manipulation the sum of squares (S_7) can be rewritten as

$$S_7 = n\bar{\epsilon}_X^2 + \sum_i (\epsilon_{X_i} - \bar{\epsilon}_X)^2. \quad (35)$$

In this case the vectors $\vec{U}$ and $\vec{W}$ are both equivalent to the vector ϵ_X where the element i takes the value ϵ_{X_i} , and bar denotes the sample average, i.e. $\bar{\epsilon}_X = \frac{\sum_i \epsilon_{X_i}}{n}$. We are looking for a matrix B_1 so that

$$n\bar{\epsilon}_X^2 = \epsilon_X^\top B_1 \epsilon_X. \quad (36)$$

The matrix

$$B_1 = \frac{1}{n} J \quad (37)$$

satisfies Equation (36), where J is a matrix with dimensionality $n \times n$ with every element set to 1. Because every row in B_1 is identical, $\text{rank}(B_1) = 1$ and hence there is one non-zero eigenvalue. Due to B_1 being idempotent, i.e. $B_1^2 = B_1$, it follows that $B_1 U = B_1^2 U$ and $\lambda_1 U = \lambda_1^2 U$. The only values for λ_1 that satisfies this criterion are 0 and 1, and hence the non-zero eigenvalue must take value 1. As corresponding eigenvector, we get $\vec{v}_1 = [\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \dots, \frac{1}{\sqrt{n}}]$. Through the properties of orthonormal projection, we get $\epsilon_X^\top \vec{v}_1 \sim \mathcal{N}(0, 1)$, and hence

$$\epsilon_X^\top \vec{v}_1 \vec{v}_1^\top \epsilon_X \sim \chi_1^2 \quad (38)$$

that is the square of sums follows a χ^2 -distribution with $\nu = 1$ degrees of freedom. Note, though that we will not be sampling this from a χ^2 -distribution as we are already sampling S_X , i.e.

$$n\bar{\epsilon}_X^2 = \frac{S_X^2}{n}. \quad (39)$$

Next, we set $B_2 = I - B_1$, i.e.

$$B_2 = \begin{bmatrix} 1 - \frac{1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & 1 - \frac{1}{n} & \dots & -\frac{1}{n} \\ \dots & \dots & \dots & \dots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & 1 - \frac{1}{n} \end{bmatrix}. \quad (40)$$

Through algebraic manipulation, we can express the second term of the right hand side of Equation (35) as

$$\sum_i (\epsilon_{X_i} - \bar{\epsilon}_X)^2 = \epsilon_X^\top B_2 \epsilon_X. \quad (41)$$

Every row in B_2 is now unique and only linearly dependent on all other rows because $\sum_i B_{2(i,:)} = \vec{0}$. From this follows that $\text{rank}(B_2) = n - 1$ and that there exists an eigenvalue decomposition of B_2 with $n - 1$ eigenvalues. It can be shown that B_2 is idempotent, and hence all non-zero

eigenvalues must take value 1. Given that B_2 has $n - 1$ eigenvalues, there exists $n - 1$ corresponding orthonormal eigenvectors $\vec{v}_j$ and due to properties of orthonormal projections, $\epsilon_X^\top \vec{v}_j \sim \mathcal{N}(0, 1)$. As a consequence $\sum_i (\epsilon_{X_i} - \bar{\epsilon}_X)^2 = \sum_{j=2}^n (\vec{v}_j^\top \epsilon_X)^2$ (note indices of sum!) and we can directly see that

$$\sum_i (\epsilon_{X_i} - \bar{\epsilon}_X)^2 \sim \chi_{n-1}^2. \quad (42)$$

We have now shown that the matrices B_1 and B_2 with the properties required by Cochran's theorem exist, and that the sum of squares can be sampled independently to generate one observation of S_7 .

F.2 Dependence between sum of normal product and sum of normals

For the plate reduction of S_8 , we follow a similar path based on Cochran's theorem Cochran [1934]. Through algebraic manipulation it can be shown that

$$S_8 = \sum_i \epsilon_{X_i} \epsilon_{Y_i} = n \bar{\epsilon}_X \bar{\epsilon}_Y + \sum_i (\epsilon_{X_i} - \bar{\epsilon}_X)(\epsilon_{Y_i} - \bar{\epsilon}_Y) \quad (43)$$

where $\bar{\epsilon}_Y = \frac{\sum_i \epsilon_{Y_i}}{n}$. We set $B_3 = \frac{1}{n} J_{(n)}$, i.e. B_3 is a $n \times n$ matrix with every element set to $1/n$. Using B_3 we can write the first term in Equation (43) as

$$n \bar{\epsilon}_X \bar{\epsilon}_Y = \epsilon_X^\top B_3 \epsilon_Y. \quad (44)$$

Now $\text{rank}(B_3) = 1$ as every row is identical and consequently there exists an eigenvalue decomposition of B_3 with exactly one non-zero eigenvalue and corresponding eigenvector. The eigenvector for B_3 is identical to the one for B_1 , i.e.

$$\vec{v}_1 = \left[\frac{1}{\sqrt{n}} \quad \frac{1}{\sqrt{n}} \quad \dots \quad \frac{1}{\sqrt{n}} \right]^\top \quad (45)$$

Due to B_3 being idempotent, its only non-zero eigenvalue has to be 1. Hence we can write

$$n \bar{\epsilon}_X \bar{\epsilon}_Y = \epsilon_X^\top \begin{bmatrix} \vec{v}_1 & \vec{0} & \dots & \vec{0} \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \vec{v}_1^\top \\ \vec{0}^\top \\ \dots \\ \vec{0}^\top \end{bmatrix} \epsilon_Y \quad (46)$$

Note that the product above is the inner product of the projections of ϵ_X and ϵ_Y onto $\vec{v}_1$, and due to the normality of the projection, $\epsilon_X^\top \vec{v}_1 \sim \mathcal{N}(0, 1)$ and $\epsilon_Y^\top \vec{v}_1 \sim \mathcal{N}(0, 1)$. The product of these two follows a normal product distribution. We will not, however, sample a normal product distribution but instead use $\frac{1}{n} S_X S_Y$ as these quantities are already sampled.

We next split the product $\epsilon_X^\top \epsilon_Y$ into components parallel and orthogonal to $\vec{v}_1$.

$$\epsilon_X = \epsilon_{X_\parallel} + \epsilon_{X_\perp} \quad (47)$$

$$\epsilon_Y = \epsilon_{Y_\parallel} + \epsilon_{Y_\perp} \quad (48)$$

and

$$\epsilon_{X_\parallel} = \vec{v}_1^\top \epsilon_X \vec{v}_1 \quad (49)$$

$$\epsilon_{Y_\parallel} = \vec{v}_1^\top \epsilon_Y \vec{v}_1. \quad (50)$$

Based on this

$$\epsilon_X^\top \epsilon_Y = \epsilon_{X_\parallel}^\top \epsilon_{Y_\parallel} + \epsilon_{X_\perp}^\top \epsilon_{Y_\perp}. \quad (51)$$

We already calculated the parallel component, i.e. $\epsilon_{X_\parallel}^\top \epsilon_{Y_\parallel} = n \bar{\epsilon}_X \bar{\epsilon}_Y$. As orthogonal components in inner products cancel out, we are left with

$$\epsilon_{X_\perp}^\top \epsilon_{Y_\perp} = \epsilon_{X_\perp}^\top \epsilon_{Y_\perp}. \quad (52)$$

Next, set $\alpha_k = \vec{v}_k^\top \epsilon_X$ and $\beta_k = \vec{v}_k^\top \epsilon_Y$ for $k \in [2, 3, \dots, n]$. Select all $\vec{v}_k$ so that they form an orthonormal basis together with $\vec{v}_1$. We get

$$\epsilon_{X_\perp}^\top \epsilon_{Y_\perp} = \sum_{k=2}^n \alpha_k \beta_k. \quad (53)$$

Due to $\vec{v}_k$ forming an orthonormal basis, $\alpha_k \sim \mathcal{N}(0, 1)$ and $\beta_k \sim \mathcal{N}(0, 1)$. We could settle for this and sample from a normal product distribution, but we use one last trick to rewrite this as follows:

$$\epsilon_{\vec{X}_\perp} \epsilon_{\vec{Y}_\perp} = \frac{1}{4} \sum_{k=2}^n (\alpha_k + \beta_k)^2 - \frac{1}{4} \sum_{k=2}^n (\alpha_k - \beta_k)^2. \quad (54)$$

Due to linearity of expectation, $(\alpha_k + \beta_k) \sim \mathcal{N}(0, 2)$ and $(\alpha_k - \beta_k) \sim \mathcal{N}(0, 2)$. We simplify to standard normals by dividing by $\sqrt{2}$, and rewrite this as

$$\epsilon_{\vec{X}_\perp} \epsilon_{\vec{Y}_\perp} = \frac{1}{2} (\gamma_{XY_a} - \gamma_{XY_b}) \quad (55)$$

where $\gamma_{XY_a} \sim \chi^2$ and $\gamma_{XY_b} \sim \chi^2$, both with $n - 1$ degrees of freedom.

We have now shown that an orthonormal projection exists that allows for decomposition of Equation (43). We note that the decomposition of sum of squares (Section F.1) is a special case of this, although we needed a slightly different path as the second term in Equation (43) is a sum of products rather than a sum or squares. The whole expression is

$$S_8 = \frac{S_X S_Y}{n} + \frac{1}{2} (\gamma_{XY_a} - \gamma_{XY_b}). \quad (56)$$

Following the same pattern, it can be shown that

$$S_9 = \sum_i \epsilon_{X_i} \epsilon_{A_i} = \frac{S_X S_A}{n} + \frac{1}{2} (\gamma_{XA_a} - \gamma_{XA_b}) \quad (57)$$

where $\gamma_{XA_a} \sim \chi^2(n - 1)$ and $\gamma_{XA_b} \sim \chi^2(n - 1)$.

F.3 Putting it all together

To generate one observation of ϕ from the agent's internal model, the code samples $\alpha_{X\mu}$, $\alpha_{X\sigma}$, $\alpha_{Y\mu}$, $\alpha_{Y\sigma}$, $\alpha_{A\mu}$, $\alpha_{A\sigma}$, and N_E . Then one observation is sampled for

$$S_X \sim \mathcal{N}(0, n) \quad (58)$$

$$S_Y \sim \mathcal{N}(0, n) \quad (59)$$

$$S_A \sim \mathcal{N}(0, n) \quad (60)$$

$$\gamma_{XX} \sim \chi^2(n - 1) \quad (61)$$

$$\gamma_{XY_a} \sim \chi^2(n - 1) \quad (62)$$

$$\gamma_{XY_b} \sim \chi^2(n - 1) \quad (63)$$

$$\gamma_{XA_a} \sim \chi^2(n - 1) \quad (64)$$

$$\gamma_{XA_b} \sim \chi^2(n - 1). \quad (65)$$

The sampled values are used to calculate

$$S_9 = \frac{S_X S_A}{n} + \frac{1}{2} (\gamma_{XA_a} - \gamma_{XA_b}) \quad (66)$$

$$S_8 = \frac{S_X S_Y}{n} + \frac{1}{2} (\gamma_{XY_a} - \gamma_{XY_b}) \quad (67)$$

$$S_7 = \gamma_{XX} + \frac{S_X^2}{n} \quad (68)$$

$$S_6 = n\alpha_{X\mu}\alpha_{Y\mu} + \alpha_{X\mu}\alpha_{Y\sigma}S_Y + \alpha_{Y\mu}\alpha_{X\sigma}S_X + \alpha_{X\sigma}\alpha_{Y\sigma}S_8 \quad (69)$$

$$S_5 = N_E(n\alpha_{X\mu}\alpha_{A\mu} + \alpha_{X\mu}\alpha_{A\sigma}S_A + \alpha_{A\mu}\alpha_{X\sigma}S_X + \alpha_{X\sigma}\alpha_{A\sigma}S_9) \quad (70)$$

$$S_4 = n\alpha_{X\mu} + \alpha_{X\sigma}S_X \quad (71)$$

$$S_3 = \alpha_{X\sigma}^2 S_7 \quad (72)$$

$$S_2 = 2\alpha_{X\mu}\alpha_{X\sigma}S_X \quad (73)$$

$$S_1 = n\alpha_{X\mu}^2. \quad (74)$$

With these, the code can finally generate one observation from the distribution of ϕ :

$$\phi = b + dN_E + \frac{S_4 + S_5 + S_6}{S_1 + S_2 + S_3}. \quad (75)$$