Conservative Continuous-Time Treatment Optimization

Nora Schneider [*1 2 3] nora.schneider@tum.de
Georg Manten [*1 2 3] georg.manten@tum.de
Niki Kilbertus [1 2 3] niki.kilbertus@tum.de

[1] Technical University of Munich [2] Helmholtz Munich
[3] Munich Center for Machine Learning

Abstract

Finding optimal treatment plans in continuous time is crucial for personalized medicine. Existing methods, however, do not explicitly optimize treatment plans and instead focus solely on predicting treatment effects. Naive optimization over these treatment-response models can exploit model errors, resulting in plans that appear optimal under the learned model but perform poorly in practice. To mitigate model exploitation, we propose `SigCTO` (Signature-based Conservative Treatment Optimization), a conservative framework for continuous-time treatment optimization. We derive a path-space regularizer based on signature-kernel maximum mean discrepancy that penalizes treatment plans for which predicted and true responses differ. Under suitable assumptions, the regularizer allows for consistent estimation, and the regularized objective upper bounds the true treatment cost. Experiments on pharmacometric simulators show that `SigCTO` finds treatment plans with lower ground-truth costs than state-of-the-art baselines. On real-world ICU data, `SigCTO` predicts treatment cost more accurately than baselines, and the regularizer detects prediction errors.

1 Introduction

A key challenge in healthcare is designing optimal treatment plans that steer a patient's health state toward a desired clinical outcome, e.g., finding a chemotherapy dosing schedule that maximally reduces tumor volume. This task is inherently temporal: patient states evolve continuously, while a treatment plan specifies not only what kind of therapy to apply, but also specific dosing schedules. In such high-stakes settings, direct experimentation with alternative treatment plans is often infeasible or unethical. As a result, treatment optimization requires models that predict patient responses under candidate plans and enable optimization over unseen treatment plans.

Historically, treatment optimization has relied on mechanistic, pharmacometric models to describe treatment-response dynamics and inform treatment decisions [Minichmayr et al., 2024, Sheiner and Steimer, 2000]. These manually specified, hard-earned models are interpretable and formulate treatment-response dynamics in continuous time. Despite these strengths, they often fail to capture the complex, stochastic dynamics of heterogeneous patient populations. The limited flexibility has motivated data-driven causal inference methods that use longitudinal observational data to estimate patient responses under alternative treatment plans [Bica et al., 2020, Robins et al., 2000].

Most causal methods operate in discrete time and impose a fixed, regularly spaced time grid. Such a grid, however, does not match clinical data where measurements are typically made at irregular

[*]Equal Contribution

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

time points [Callahan et al., 2020]. Moreover, candidate plans are restricted to the grid, even though clinical treatment schedules are often irregular and patient-specific. More fundamentally, the grid discretizes continuously evolving treatment-response dynamics and can thereby compromise causal inference [Sun and Crawford, 2023]. Without imposing discretization grids, continuous-time methods naturally accommodate irregularly timed measurements and interventions [Sun and Crawford, 2022]. Recent causal inference methods have embraced this paradigm by modeling response dynamics with learned differential equations [Hess and Feuerriegel, 2025, Ying, 2025, Hess et al., 2024, Seedat et al., 2022]. Yet these causal methods stop at treatment response prediction – treatment optimization in continuous time remains largely unaddressed in this literature.

One key challenge of data-driven treatment optimization is that it must be conducted offline: we aim to find improved treatment plans using only historical data, without actively collecting new data under explorative alternative plans. Such novel plans may steer the system towards regions with limited historical data support, where predictive models are often inaccurate. Optimizing a predictive model can therefore exploit model errors, finding spuriously good-looking plans. This problem is well known in offline reinforcement learning [Yu et al., 2020, Kidambi et al., 2020] and relates in causal inference to violations of the commonly made overlap assumption, which is already fragile in static settings [D'Amour et al., 2021]. In continuous time, treatment plans live in an infinite-dimensional path space, and the overlap assumption becomes practically impossible to satisfy – making model exploitation a central challenge for reliable continuous-time treatment optimization.

In this work, we propose `SigCTO`, a framework for conservative, offline optimization of open-loop treatment plans in continuous time. We model patient-response dynamics with a controlled stochastic differential equation where treatment is the control input. To mitigate model exploitation, we augment the predicted cost with a path-space regularizer that penalizes differences between predicted and true responses. We show that, for any candidate treatment plan, the conservative objective upper-bounds the true cost and discourages plans that exploit overly optimistic predictions. We estimate the regularizer from data using a consistent, signature-kernel MMD estimator. Experiments on pharmacometric simulators show that `SigCTO` mitigates model exploitation and finds treatment plans with lower ground-truth costs than conservative and non-conservative baseline methods. On real-world ICU data, the regularizer detects the distribution mismatch it is designed to penalize.

2 Related Work

Continuous-time longitudinal causal inference estimates the potential outcome trajectory under a candidate treatment plan and does not address finding an optimal treatment plan. Some methods estimate population-level potential outcomes [Ying, 2025, Hızlı et al., 2023, Schulam and Saria, 2017, Lok, 2008] and leave patient-specific outcomes unaddressed. Other methods predict personalized outcomes by directly conditioning on patient history using neural ODEs [De Brouwer et al., 2022] or neural CDEs [Seedat et al., 2022, Hess et al., 2024], but only consider binary treatments applied at discrete event times. INSITE [Kacprzyk et al., 2024] allows for continuous-valued treatments by modeling them as control inputs in a governing ODE, but is also restricted to outcome prediction.

Treatment optimization is not a simple extension of outcome-prediction methods for two reasons. First, these methods do not specify nor readily support structured optimization, and a brute-force search over the full treatment plan space is often infeasible. When the dynamics are known and correctly specified, classical stochastic optimal control Bank et al. [2025], Berret and Jean [2020] provides structured optimization methods. Second, unconstrained search can, and typically will, exploit model errors. Thus, unconstrained search would only be reliable if the true dynamics were perfectly learned. This can only work in the infinite-data regime and if the dynamics are identifiable from observations in the assumed model class. Identifiability in turn requires the overlap assumption to be satisfied Lok [2008], Robins and Hernan [2008], Ying [2025], Sun and Crawford [2022]. Intuitively, overlap requires that for all observed patient histories, each treatment plan must be observed with a nonzero probability. Even for static settings, the overlap assumption is a strong one and often compromised [D'Amour et al., 2021]. In continuous time, observational data must support entire treatment paths across times and patient histories, making overlap even harder to satisfy. Even when one is willing to buy into the assumptions required for identifiability to hold, practical remedies, like `SigCTO`, are required for reliable plans. Thus, optimizing treatment plans directly with prediction-only methods may select treatment plans outside the data support, where identifiability guarantees do not hold, and predictions are unreliable.

Dynamic Treatment Regimes (DTR) methods learn a sequence of history-dependent decision rules that map patient information at each decision point to a treatment to optimize clinical outcomes [Murphy, 2003, Robins, 2004, Chakraborty and Murphy, 2014, Li et al., 2023, Tsirtsis and Rodriguez, 2023]. Most of this literature is formulated in discrete time and is often developed with sequential decision making (and sometimes adaptive data collection) in mind rather than personalized planning from a fixed observational dataset. Offline variants exist – for example, Zhou et al. [2023] propose a pessimistic Bayesian Q-learning approach for learning DTRs from observational data, and Wang et al. [2025] propose variational counterfactual intervention planning – but remain discrete-time frameworks. As a result, existing DTR methods do not address irregular observations typical in clinical time series, where measurements and treatments occur at non-uniform times. Xu and Zhang [2024] instead learn a joint model not only over dosage values, but also dosage times, while still fundamentally considering a discrete sequence of events rather than a continuous-time treatment path.

Offline Reinforcement Learning (RL) uses pessimism and uncertainty-aware regularization to mitigate model exploitation. For discrete-time Markov Decision Processes (MDPs), model-based RL approaches propose uncertainty-penalized rewards, conservative value estimates, or uncertainty-based rollout truncation [Yu et al., 2020, Kidambi et al., 2020, Yu et al., 2021, Zhang et al., 2023a, Koprulu et al., 2025]. In continuous time, existing methods are either model-free approaches [Zhang et al., 2023b, Burns et al., 2023] or introduce novel dynamics models and do not explicitly address model errors during optimization [Hoischen et al., 2025, Holt et al., 2023]. Moreover, RL typically optimizes closed-loop control; `SigCTO` optimizes open-loop plans and explicitly accounts for model errors.

3 Problem Statement

We formalize the problem of finding optimal treatment plans from irregular observational data following the causal perspective on treatment-effect estimation.

Setting. Let $I = [t_s, t_f] \subset \mathbb{R}$ be a compact time interval and let $(\Omega, \mathcal{F}, \mathbb{F}, P)$ be a complete filtered probability space with filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in I}$. On this space, we consider three observable stochastic processes: the *treatment process* $U : I \times \Omega \to A \subseteq \mathbb{R}^{d_u}$ (e.g., chemotherapy), the *outcome process* $Y : I \times \Omega \to \mathbb{R}^{d_y}$ (e.g., tumor volume), and the *covariate process* $Z : I \times \Omega \to \mathbb{R}^{d_z}$ (e.g., biomarkers). The treatment is the external intervention, while outcomes and covariates describe the patient's evolving condition. For convenience, we combine the outcome and covariate process into the *state process* $X = (Z, Y) : I \times \Omega \to \mathbb{R}^{d_x}$, where $d_x = d_y + d_z$. Following the potential outcome framework in causal inference Rubin [2005], for a deterministic treatment plan $u : I \to A \subseteq \mathbb{R}^{d_u}$, we write $X^{(u,\mathrm{pot})} : I \times \Omega \to \mathbb{R}^{d_x}$ for the *potential state process*, i.e., the patient state trajectory that would be observed if treatment were externally set to follow u over the full time interval I.

Throughout the paper, we use the following notation. Let $X = (X_t)_{t\in I}$ be a stochastic process indexed by I. The restriction of X to the interval $[a, b] \subseteq I$ is denoted by $X_{[a,b]}$. Its path law, $P_{X_{[a,b]}}$, is the probability distribution of the entire trajectory over this interval.[2] The natural filtration of X is denoted by $\mathcal{F}^X = (\mathcal{F}^X_t)_{t\in I}$, where $\mathcal{F}^X_t = \sigma(X_{[t']} \mid t_s \le t' \le t)$, that is, the sigma-algebra generated by the values of X up to time t. We denote the conditional path law of a process Y given $X_{[a,b]}$ by $P_{Y|X_{[a,b]}} = P_{Y|\mathcal{F}^X_{a,b}}$, where $\mathcal{F}^X_{a,b} = \sigma(X_t \mid a \le t \le b)$. We use lowercase letters to denote sample-path realizations, i.e., $x = (X_t(\omega))_{t\in I}$ for $\omega \in \Omega$.

Data. We observe a dataset $\mathcal{D} = \{(u^{(i)}, y^{(i)}, z^{(i)})\}_{i=1}^N$ of N i.i.d. patient trajectories. Here, $u^{(i)}$, $y^{(i)}$, and $z^{(i)}$ are the treatment, outcome, and covariate time series, respectively, recorded at patient-specific, potentially irregular times in $[t_s, t_f]$. The dataset only captures patient trajectories under historical treatment plans; potential state trajectories under alternative plans are generally unobserved.

Objective. Our goal is to find treatment plans that optimize patient outcomes according to a clinical objective. Specifically, at decision time $t_0 \in I$, given a patient's history available up to t_0 (i.e., $\mathcal{F}_{t_0}$),

[2] $P_{X_{[a,b]}} = P \circ X^{-1}_{[a,b]} \in \mathcal{M}_1((\mathbb{R}^{d_x})^{[a,b]})$, where $\mathcal{M}_1(\bullet)$ is the set of all probability measures on $\bullet$, $(\mathbb{R}^{d_x})^{[a,b]}$ the set of maps from set $[a, b]$ to set $\mathbb{R}^{d_x}$.

we seek an optimal treatment plan $u^* : [t_0, t_f] \to \mathbb{R}^{d_u}$ with

$$u^* = \arg\min_{u \in \mathcal{U}_{\text{adm}}} \underbrace{\mathbb{E}\Big[\int_{t_0}^{t_f} f\big(t, X_t^{(u,\text{pot})}, u_t\big) \mathrm{d}t + g\big(X_{t_f}^{(u,\text{pot})}\big) \Big| \mathcal{F}_{t_0} \Big]}_{=: \mathcal{J}(t_0, X^{(u,\text{pot})}, u)}. \tag{1}$$

The class of admissible treatment plans $\mathcal{U}_{\text{adm}}$ consists of deterministic, open-loop treatment plans. The running cost $f : I \times \mathbb{R}^{d_x} \times A \to \mathbb{R}$ and final cost $g : \mathbb{R}^{d_x} \to \mathbb{R}$ are prespecified and encode the clinical objective. For instance, the running cost f may penalize treatment intensity, whereas the final cost g may represent a target tumor volume at the end of the planning horizon. Intuitively, the state process X describes patient trajectories under historical treatment U, whereas $X^{(u,\text{pot})}$ describes the generally unobserved potential trajectory had treatment been set to a deterministic plan u. Equation (1) seeks a plan with the lowest expected cost under these potential trajectories, using only observational data $\mathcal{D}$.

4 Conservative Continuous-Time Treatment Optimization

In this section, we derive signature-based, conservative treatment optimization (`SigCTO`). The treatment optimization objective in Equation (1) poses two key difficulties: First, because the data is observational, the potential state process is generally unobserved [Imbens and Rubin, 2015]. We thus introduce additional assumptions under which its law is identifiable from the observational distribution (Section 4.1). Second, we have only access to a finite sample from the observational distribution. Thus, even when identifiability is given theoretically, estimates of the potential state distribution – and hence of the resulting cost – can be inaccurate. Minimizing the predicted cost therefore need not minimize the true cost and may favor treatment plans with overly optimistic cost estimates. We therefore derive a theoretical upper bound on the true cost (Section 4.2) and a consistent estimator of this bound (Section 4.3).

4.1 Structural Dynamics Model for Identifiability

Inspired by Kacprzyk et al. [2024], we establish identifiability of the potential state process from the observational distribution by assuming an underlying structural dynamics model. We use controlled stochastic differential equations (SDEs), which provide a well-established framework for modeling noisy dynamical systems and are particularly well-suited to pharmacology and disease modeling, where diseases and treatment responses evolve continuously and stochastically [Geng et al., 2017].

Assumption 4.1 (Structural Dynamics Model)**.** The observed state process $X = (X_t)_{t \in I}$ evolves under the, possibly stochastic, treatment process $U = (U_t)_{t \in I}$ according to

$$\mathrm{d}X_t = \mu(X_t, U_t)\, \mathrm{d}t + \sigma(X_t, U_t)\, \mathrm{d}W_t, \tag{2}$$

where $X_0 \sim \mu_0 \in \mathcal{M}_1(\mathbb{R}^{d_x})$, $W = (W_t)_{t \in I}$ is a standard Wiener process, and μ, σ are shared structural drift and diffusion functions, respectively. For a deterministic treatment plan $u : I \to A$, the potential state process $X^{(u,\text{pot})}$ is defined as the solution of Equation (2) with $U_t = u_t$ for all $t \in I$.

We further make standard regularity assumptions on (μ, σ), such as Lipschitz continuity and linear growth, to ensure the existence and path-wise uniqueness of the solution of the controlled SDE (see Appendix B for details). In the following, we write $X^{t_0,x_0,u}$ for the strong solution of Equation (2) on $[t_0, t_f]$ initialized at $X_{t_0} = x_0$ and driven by the deterministic plan u and denote its *solution path law* by $P_{X^{t_0,x_0,u}}$ or $P_{X^{t_0,x_0,u}_{[t_0,t_f]}}$. Under Assumption 4.1, the observed state process is recovered by the potential process under the realized treatment plan, i.e., $X = X^{(U,\text{pot})}$ a.s., similar to the classical consistency assumption in related treatment-effect estimation literature.

Assumption 4.2 (Observability and Structural Identifiability)**.** The dynamics (μ, σ) are identifiable from the observational distribution up to response-equivalent parameterizations on the admissible state-control domain.

Together, Assumptions 4.1 and 4.2 require the observational distribution to identify the system's response to every admissible treatment plan. In related treatment-effect estimation literature, this typically corresponds to the omnipresent consistency, overlap, and no-hidden-confounding assumptions.

Proposition 4.3 (Structural Identifiability)**.** *Under Assumptions 4.1 and 4.2 and the regularity conditions in Appendix B.2, the law of $X^{(u,\mathrm{pot})}$ and consequently the expected cost $\mathcal{J}(t_0, X^{(u,\mathrm{pot})}, u)$ in Equation* (1) *are identifiable for every $u \in \mathcal{U}_{\mathrm{adm}}$.*

A proof is provided in Appendix C.1. Identifiability means that the population observational distribution uniquely determines the potential state law for each admissible plan. Importantly, identifiability is an idealized, infinite-data statement: it does not ensure accurate estimation from finite data. Moreover, under Assumption 4.1 the dynamics are Markovian and we can rewrite the treatment optimization objective as the following stochastic open-loop control problem

$$u^* := \arg\min_{u \in \mathcal{U}_{\mathrm{adm}}} \underbrace{\mathbb{E}\left[\int_{t_0}^{t_f} f(t, X_t^{t_0,x_0,u}, u_t)\,\mathrm{d}t + g(X_{t_f}^{t_0,x_0,u})\right]}_{=:\mathcal{J}(t_0, X^{t_0,x_0,u}, u)}. \tag{3}$$

4.2 Conservative Treatment Optimization

Even under assumptions that allow for identifiability of the potential state process under a candidate treatment plan, and thus the associated ground-truth cost, finite-data estimates can still be inaccurate. To account for these estimation errors, we derive a tractable upper bound on the ground-truth cost in Equation (3). `SigCTO` minimizes the upper bound, resulting in conservative treatment optimization.

Approximated Dynamics. To construct a model-based surrogate for the unknown ground-truth cost of a treatment plan, we fit a model for the structural dynamics in Equation (2) from observational data using a well-specified model class. Our experiments use neural SDEs [Issa et al., 2023], but the construction also allows other model classes. Given a candidate plan u and initial state $\hat{X}_{t_0} = x_0$, the learned model induces an estimated state process $\hat{X}^{t_0,x_0,u}$. Because the cost function is known, substituting the process gives the *model-based surrogate cost* $\mathcal{J}(t_0, \hat{X}^{t_0,x_0,u}, u)$. Next, we relate the model-based surrogate to the ground-truth cost.

Decomposition of Estimation Error. We next express the difference between the ground-truth and surrogate costs in terms of local discrepancies between the true and learned dynamics. For any temporal partition of the treatment horizon, the following lemma decomposes this cost difference into a telescoping sum. Each term measures a conditional one-step discrepancy over one subinterval. For any interval $I_j := [t_j, t_{j+1}]$, we denote the true and learned solution path laws initialized at x at time t_j and driven by the deterministic treatment segment u_{I_j}, respectively, by $P_{\hat{X}_{I_j}^{t_j,x,u_{I_j}}}$ and $P_{X_{I_j}^{t_j,x,u_{I_j}}}$.

Lemma 4.4 (Estimation Error Telescope)**.** *For an admissible treatment plan $(u : I \to \mathbb{R}^{d_u}) \in \mathcal{U}_{\mathrm{adm}}$, initial state x_0, and any temporal partition $t_0 < t_1 < \cdots < t_K = t_f$,*

$$\mathcal{J}(t_0, X^{t_0,x_0,u}, u) - \mathcal{J}(t_0, \hat{X}^{t_0,x_0,u}, u) = \sum_{j=0}^{K-1} \mathbb{E}_{x \sim P_{\hat{X}_{t_j}}}[D_j(x, u)]$$

with one-step discrepancy on I_j between true and learned dynamics given by

$$D_j(x, u) = \mathbb{E}_{x_{I_j} \sim P_{X_{I_j}^{t_j,x,u_{I_j}}}}\left[h_j(x_{I_j})\right] - \mathbb{E}_{\hat{x}_{I_j} \sim P_{\hat{X}_{I_j}^{t_j,x,u_{I_j}}}}\left[h_j(\hat{x}_{I_j})\right]$$

and h_j is the deterministic path functional $x_{I_j} \mapsto \int_{t_j}^{t_{j+1}} f(s, x_s, u_s)\mathrm{d}s + \mathcal{J}(t_{j+1}, x_{t_{j+1}}, u)$ mapping a path segment x_{I_j} to a sum of the cumulative cost on I_j and the continuation value at t_{j+1}.

A proof is provided in Appendix C.2, and although it requires different techniques, it can be viewed as an extension of related discrete-time results [Luo et al., 2019]. Bounding each discrepancy by its absolute value gives the following upper bound on the ground-truth cost.

Corollary 4.5 (Estimation Gap Bound)**.** *In the setting of Lemma 4.4, we have*

$$\mathcal{J}(t_0, X^{t_0,x_0,u}, u) \le \mathcal{J}(t_0, \hat{X}^{t_0,x_0,u}, u) + \sum_{j=0}^{K-1} \mathbb{E}_{x \sim P_{\hat{X}_{t_j}}}[|D_j(x, u)|]$$

Thus, a treatment plan with low model-based surrogate cost and small accumulated discrepancies also has low ground-truth cost. The temporal partition is freely chosen, enabling flexible localization of model mismatch and improving empirical performance (see Section 6).

Path-space IPM. Each one-step discrepancy $D_j(x,u)$ is the difference between the expected value of the cost functional h_j under the true and learned solution path laws. Direct evaluation is infeasible because h_j contains the true continuation value, i.e., the true remaining cost from t_{j+1} onward. To bound $D_j(x,u)$ without evaluating h_j, we consider the worst-case difference between the true and learned solution path laws over a class $\mathcal{G}$ of path functionals for which $h_j \in \mathcal{G}$. The resulting integral probability metric (IPM) quantifies the discrepancy between the true and learned conditional path laws.

Lemma 4.6 (Increment Bound)**.** *For any function class $\mathcal{G} \subseteq \mathcal{L}^0\big((\mathbb{R}^{d_x})^{I_j}, \mathbb{R}\big)$ inside a normed space s.t. h_j can be normalized into $\mathcal{G}$ with normalization constant $c_{\mathcal{G}} \in \mathbb{R}_{\geq 0}$, we have*

$$|D_j(x,u)| \leq c_{\mathcal{G}} \cdot \mathrm{IPM}_{\mathcal{G}}[P_{X_{I_j}^{t_j,x,u_{I_j}}}, P_{\hat{X}_{I_j}^{t_j,x,u_{I_j}}}],$$

where $\mathrm{IPM}_{\mathcal{G}}(P,Q) \coloneqq \sup_{g\in\mathcal{G}} |\mathbb{E}_P[g] - \mathbb{E}_Q[g]|$ is the integral probability metric generated by $\mathcal{G}$ and $c_{\mathcal{G}}$ is a constant depending on $\mathcal{G}$. For path functionals $h_j \in \mathcal{G}$, the inequality holds for $c_{\mathcal{G}} = 1$.

IPMs encompass several familiar notions of distributional discrepancy. Depending on the assumptions imposed on h_j, different choices of $\mathcal{G}$ result in different discrepancy metrics (see Appendix D). We choose $\mathcal{G}$ as the unit disk of the signature kernel Reproducing Kernel Hilbert Space (RKHS), as discussed in detail in Section 4.3. There, we also explain the choice of $c_{\mathcal{G}}$ and how to accommodate cases where $h_j \notin \mathcal{G}$. Similar bounds have been developed for discrete time [Yu et al., 2020].

Conservative Objective. Combining Lemma 4.6 and Corollary 4.5 bounds the ground-truth cost of a treatment plan by the model-based cost plus a penalty for the accumulated discrepancy between the true and learned solution path laws. Based on this, we propose a *conservative*, "model-error-aware" optimization objective by augmenting the model-based cost with an IPM regularizer

$$\mathbb{E}\left[\int_{t_0}^{t_f} f\big(t, \hat{X}_t^{t_0,x_0,u}, u_t\big)\,\mathrm{d}t + g\big(\hat{X}_{t_f}^{t_0,x_0,u}\big)\right] + \lambda \underbrace{\sum_{j=0}^{K-1} \mathbb{E}_{x\sim P_{\hat{X}_{t_j}}}\left[\mathrm{IPM}_{\mathcal{G}}[P_{X_{I_j}^{t_j,x,u_{I_j}}}, P_{\hat{X}_{I_j}^{t_j,x,u_{I_j}}}]\right]}_{=:R(u)}. \tag{4}$$

Here, $\lambda > 0$ controls the degree of conservatism. For $\lambda \geq c_{\mathcal{G}}$, the objective upper bounds the true cost, and its minimizer satisfies the following optimality gap.

Lemma 4.7 (Optimality Gap of the Conservative Optimizer)**.** *Let $\hat{u}$ be the minimizer of the conservative objective in Equation* (4) *and u^* be the minimizer of the true objective in Equation* (3)*. If there exists a $c_{\mathcal{G}} \geq 0$ s.t. the increment bound in Lemma 4.6 holds for each $j \in \{0,\dots,K-1\}$ and choose $\lambda \geq c_{\mathcal{G}}$, then the following optimality gap holds:*

$$\mathcal{J}(t_0, X^{t_0,x_0,\hat{u}}, \hat{u}) - \mathcal{J}(t_0, X^{t_0,x_0,u^*}, u^*) \leq (c_{\mathcal{G}} + \lambda) R(u^*).$$

A proof is provided in Appendix C.3. The optimizer of the conservative objective is near-optimal whenever the true optimum u^* has a small discrepancy penalty $R(u^*)$. In the infinite-data limit of Assumption 4.2, assuming the learned solution path law converges to the true one, the IPM regularizer vanishes and Equation (4) reduces to the unregularized objective.

4.3 From IPM to Signature Kernel MMD and `SigCTO`

While the model-based cost can be evaluated using the learned dynamics, the regularizer $R(u)$ still depends on the choice of function class $\mathcal{G}$ and on the unknown true conditional path laws. In this section, we link the general IPM to the signature-kernel MMD and develop a tractable estimate of the conservative objective.

From IPM to MMD. When $\mathcal{G}$ is the unit ball of an RKHS, the IPM is called *maximum mean discrepancy* (MMD). It compares distributions through their kernel mean embeddings and can be estimated using kernel evaluations (see Appendix D). To initialize the IPMs in $R(u)$ with MMDs, we need a kernel for trajectories whose RKHS is rich enough to represent or approximate the cost functionals h_j. This motivates our choice of the signature kernel, which we introduce next.

Signature Kernel. The signature kernel is induced by the signature transform, which represents a continuous path x of bounded variation as a feature vector $\mathcal{S}(x)$ in a Hilbert space $T(\mathbb{R}^d)$ Cass and Salvi [2024]. These features are iterated integrals that capture the order and interaction of path increments. The representation is expressive: on suitable compact sets of paths, finite linear combinations of signature coordinates can uniformly approximate any continuous path functional. The inner product of these representations defines the signature kernel, $k_{\text{sig}}(x, y) := \langle \mathcal{S}(x), \mathcal{S}(y) \rangle_{T(\mathbb{R}^d)}$. Choosing $\mathcal{G}$ as the unit ball of the RKHS $H_{\mathcal{S}}(\mathbb{R}^d)$ induced by k_{sig} gives the *signature-kernel MMD*

$$\text{MMD}_{k_{\text{sig}}}(P_X, P_Y) = \|\mu_{P_X} - \mu_{P_Y}\|_{H_{\mathcal{S}}(\mathbb{R}^d)} = \text{IPM}_{D_1}[P_X, P_Y],$$

where $D_1 = \{f \in H_{\mathcal{S}}(\mathbb{R}^d) \mid \|f\|_{H_{\mathcal{S}}(\mathbb{R}^d)} \leq 1\}$. For a more formal background on signatures, see Appendix E.1. We use this MMD for the IPM terms in $R(u)$. When h_j belongs to the signature-kernel RKHS, its RKHS norm provides the scaling factor $c_{\mathcal{G}}$ in the bound on $|D_j(x, u)|$ and the upper bound holds exactly. For $h_j \notin H_{\mathcal{S},j}(\mathbb{R}^{d_x})$, signature universality yields a bound on the cost discrepancy in terms of the signature-kernel MMD, with an arbitrarily small additive remainder and an approximation-dependent multiplier $c_{\mathcal{G}}$, under suitable assumptions (see Appendix E.2).

Signature kernel MMD Estimator and `SigCTO`. Since the true solution path law for any treatment plan u is generally unknown, we estimate the signature kernel MMDs from observed and model-generated trajectories. Specifically, we use the maximum conditional mean discrepancy (MCMD) estimator [Park and Muandet, 2020]. It compares conditional mean embeddings in RKHSs, avoids explicit density estimation, and does not require repeated ground-truth trajectories for candidate plans [Park et al., 2021] (see Appendix D). To estimate the discrepancy between the solution path laws using MCMD, we require the solution path law to coincide with the corresponding conditional law given the initial state and treatment segment, as is the case, for example, when treatment on each subinterval is prescribed based on the initial state. The empirical MCMD consistently estimates the population conditional MMD under the assumptions of Park et al. [2021, Theorem 4.2]. We refer to that work for the detailed assumptions and consistency proof, and to Appendix E.2 for a discussion of signature-kernel universality and approximation properties relevant to these conditions. We provide the full construction of the MCMD estimator in Appendix E.3. Moreover, in Appendix F we detail the full end-to-end `SigCTO` algorithm that performs conservative treatment optimization from data.

5 Experimental Setup

We evaluate `SigCTO` on two state-of-the-art pharmacokinetic–pharmacodynamic (PK/PD) simulators and on one real-world clinical dataset[3]. The PK/PD simulators provide ground-truth outcomes under arbitrary (unseen) treatment plans, which allows us to benchmark the full `SigCTO` optimization pipeline. However, on the clinical dataset, true outcomes under newly optimized treatment plans are unobserved, so we focus on the components of `SigCTO` that can be evaluated on offline data: whether the approximated dynamics result in reliable cost predictions and whether our regularizer detects the distribution shifts it is designed to penalize.

5.1 Data and Treatment Optimization Tasks

The *cancer* simulator [Geng et al., 2017] models chemo- and radio-therapy effects on lung cancer and is commonly used in longitudinal treatment-effect estimation literature, e.g. Kacprzyk et al. [2024]. We sample observations from two clinically motivated regimes [Curran Jr et al., 2011]: sequential and concurrent therapy. The states are tumor volume and drug concentration, and the treatments are chemo- and radiotherapy ($d_x = d_u = 2$). We consider two treatment objectives: an *explicit target* objective minimizing final tumor volume, and a *relative target* objective aiming for a 70% reduction from initial volume. Both tasks also penalize treatment intensity.

The *Covid-19* simulator [Qian et al., 2021, Dai et al., 2021] models disease progression under dexamethasone treatment. We consider a single-intervention regime, i.e., dexamethasone is administered at a single time point with randomly sampled dosage. The treatment is the plasma dexamethasone concentration, and the state consists of viral load, innate immune response, adaptive immune response, and lung-tissue dexamethasone concentration ($d_x = 4$, $d_u = 1$). For treatment optimization, we define a *target trajectory* based on average disease progression under intensive treatment.

[3]Our code is available at `https://github.com/NoraSchneider/SigCTO`

MIMIC-IV [Johnson et al., 2024] is a publicly available intensive care unit (ICU) database. We follow the preprocessing protocol of Sivaraman et al. [2023], which applies Sepsis-3 inclusion criteria and aggregates measurements into 4-hour intervals. Restricting the cohort to septic shock patients (vasopressor use, $\geq$ 24h trajectory) results in $10{,}826$ trajectories with $d_x = 14$ state features and $d_u = 2$ treatment variables (IV fluid and vasopressor). The treatment optimization objective encourages *Sepsis-3 recovery*. Appendix G provides details on all datasets and optimization tasks.

5.2 Practical Implementation

`SigCTO`**.** We approximate the structural dynamics model with a neural SDE. For treatment optimization, we restrict to a parameterized family of deterministic open-loop controls, where the parameters describe treatment times and dosages. We optimize the parameters with gradient-based optimization: in each iteration, we sample trajectories from the learned neural SDE under the current control parameters and evaluate the objective corresponding to a direct single-shooting, optimal-control approach [Höglund et al., 2023]. We compute the `SigCTO`-regularizer in the conservative objective from a held-out dataset. `SigCTO` implementation details are in Appendix H.

Baselines. First, we benchmark `SigCTO` against state-of-the-art, continuous-time treatment-effect estimators: TE-CDE [Seedat et al., 2022] and INSITE [Kacprzyk et al., 2024] (see Appendix I for details on implementation and hyperparameter tuning). Note that both baselines are pure intervention-prediction and not treatment-optimization methods. Thus, neither provides a structured treatment-plan optimizer, and exhaustive search over continuous treatment plans is infeasible. Hence, we have to deploy a model-agnostic control-library protocol [Seedat et al., 2022, Wang et al., 2025] for evaluation. Specifically, we sample a shared library of 100 candidate treatment plans. For every candidate, each method predicts the response and cost and selects the plan with the lowest predicted cost. To ensure a fair, like-for-like comparison, we apply the same protocol to `SigCTO` (with $\lambda = 0$) and refer to this variant as $\texttt{SigCTO}^{\dagger}$. The control-library protocol assesses prediction-based treatment selection within the candidate set. It does not capture the full contribution of `SigCTO`: directly optimizing treatment plans while accounting for model error. We therefore separately evaluate `SigCTO` in its intended mode, namely end-to-end optimization of our conservative objective.

Second, we benchmark the `SigCTO` regularizer against a strong uncertainty-regularized RL baseline. A strong and fitting contender is MOPO [Yu et al., 2020]. MOPO is designed for discrete-time MDPs and closed-loop policies, so we do not get around adapting its uncertainty-penalization principle to continuous-time, open-loop treatment optimization for a meaningful comparison. We refer to this adaptation as *CT-MOPO* (see Appendix I). It uses an ensemble of neural SDEs and penalizes treatment plans with integrated uncertainty along the response trajectory, where the uncertainty at each time point is the maximum local diffusion covariance across the ensemble members (analogous to MOPO). CT-MOPO and `SigCTO` share the same model architecture, unregularized cost objective, control parameterization, and optimization procedure and thus only differ in the regularizer. Thus, benchmarking `SigCTO` against CT-MOPO is an isolated comparison of the proposed regularizers.

Evaluation Metrics. For control-library evaluation, we report the Spearman correlation between predicted and ground-truth costs across candidate plans and the ground-truth cost of the selected plan. Here, $\texttt{SigCTO}^{\dagger}$ uses $\lambda = 0$, as the regularizer would intentionally alter the cost-based ranking to account for model error. For full optimization, we evaluate the regularizer by reporting relative ground-truth costs: the optimized plan's ground-truth cost divided by that under naive optimization; values below 1 indicate improvement. As MIMIC-IV provides no true outcomes for newly optimized plans, we evaluate the dynamics model via cost-prediction correlation and absolute error on held-out trajectories, and the regularizer via its correlation with the discrepancy between observed and predicted trajectories. For further details, see Appendix I.2.

6 Experimental Results

`SigCTO` **finds better treatment plans than baselines.** Table 1 compares all methods on the PK/PD treatment optimization tasks. Among methods evaluated with the control-library protocol, $\texttt{SigCTO}^{\dagger}$ achieves the highest rank correlation on Cancer-Explicit and Covid-19 and the second-highest on Cancer-Relative (Table 1a). The fixed controls library may underestimate model exploitation in the full, continuous control space. When optimizing end-to-end and using the intended optimization mode, `SigCTO` achieves the best overall ground-truth costs, indicating that effective treatment optimization

Table 1: **Benchmarking `SigCTO` on PK/PD tasks.** All results are reported as mean±standard deviation over 15 task initializations. `SigCTO` uses a linear kernel and $K = 1$.

(a) `SigCTO` **compared to causal baselines.** Causal baselines and `SigCTO`† select treatments using the control library approach; `SigCTO` directly optimizes treatment plans using our proposed conservative objective.

	Spearman Correlation (↑)			**$\sqrt{\text{Ground-Truth Cost}}$** (↓)		
Method	Cancer-Exp	Cancer-Rel	Covid-19	Cancer-Exp	Cancer-Rel	Covid-19
TE-CDE	0.29±0.10	-0.30±0.10	0.35±0.51	16.78±14.11	13.18±10.71	4.07±0.70
INSITE	0.23±0.00	**0.23**±0.00	0.01±0.65	16.09±13.13	11.94±9.74	4.75±1.14
`SigCTO`† ($\lambda = 0$)	**0.39**±0.00	0.22±0.02	**0.76**±0.16	16.80±13.58	12.59±10.01	3.98±0.78
`SigCTO` ($\lambda = 100$)	-	-	-	**12.95**±8.22	**8.03**±6.31	**3.18**±0.94

(b) `SigCTO` **compared to CT-MOPO.** Relative Ground-Truth Cost (↓) for each optimization task.

	$\lambda = 100$			$\lambda = 1000$		
Method	Cancer-Exp	Cancer-Rel	Covid-19	Cancer-Exp	Cancer-Rel	Covid-19
CT-MOPO	0.39±0.41	0.66±0.20	1.93±1.65	0.10±0.06	0.62±0.31	11.90±33.16
`SigCTO`	**0.07**±0.11	**0.19**±0.11	**0.49**±0.31	**0.01**±0.03	**0.14**±0.01	**0.52**±0.48

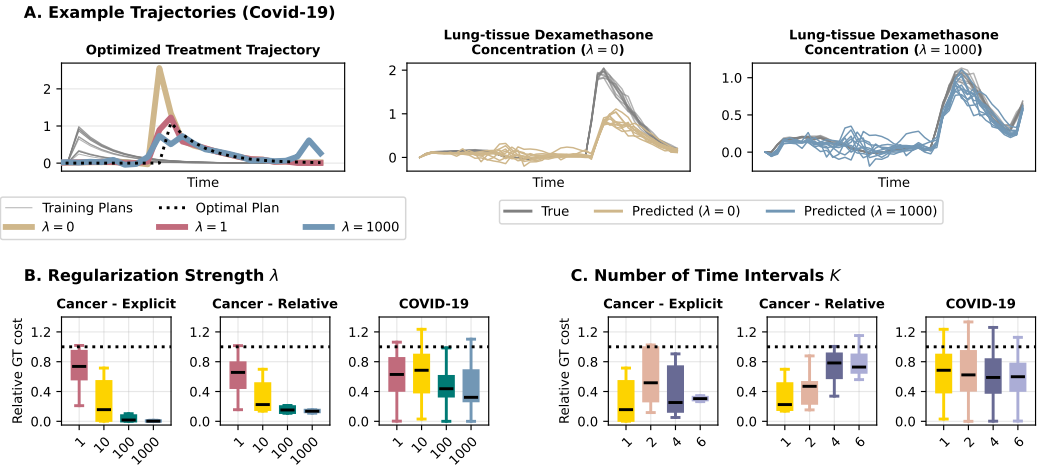

Figure 1: `SigCTO` **on PK/PD tasks. A.:** Optimized treatment plans (left) and corresponding predicted and true states (right). **B.** & **C.:** Relative ground-truth cost for different $\lambda \in \{1, 10, 100, 1000\}$ with $K = 1$, and $K \in \{1, 2, 4, 6\}$ with $\lambda = 10$ using a linear kernel. Values below 1 indicate improvement over naive optimization. Boxplots show medians and IQRs, with whiskers extending to the most extreme points within 1.5 IQRs over 15 optimization task initializations.

requires both searching beyond a fixed library and controlling for model errors. Table 1b shows that the `SigCTO` signature kernel MMD regularizer achieves lower relative ground-truth costs than uncertainty-based regularization with CT-MOPO. This indicates larger improvements over naive optimization and suggests that our `SigCTO` regularizer more effectively mitigates model exploitation than the uncertainty-based baseline.

`SigCTO` **regularization mitigates model exploitation and improves treatment plans.** Figure 1 evaluates the effectiveness of `SigCTO` on the three PK/PD-based treatment-optimization tasks. Panel **A** illustrates an example on Covid-19: under naive, unregularized optimization ($\lambda = 0$), the optimized treatment plan substantially deviates from the true optimum (left), and the predicted trajectories under this found plan do not match the true response (middle), indicating model exploitation. Stronger regularization ($\lambda = 1000$) pushes the treatment plan closer to the true optimum (left) and improves agreement between predicted and true trajectories (right).

When varying the regularization strength λ (Panel **B**), the relative ground-truth costs are clearly below 1 on all tasks and decrease as λ increases, indicating that our regularizer is effective and improves performance over naive optimization. Increasing the number of time intervals K (Panel **C**) leaves performance largely unchanged on Cancer-Explicit and Covid-19 but worsens it on Cancer-Relative, although regularization remains beneficial for all K compared to naive optimization. With an RBF lifting kernel (see Appendix K), relative ground-truth costs rise at $\lambda = 1000$ on both cancer tasks, suggesting overregularization. Increasing K improves performance on these tasks, possibly because conditioning on intermediate predicted states favors data-supported trajectories. Sensitivity analyses in Appendix K further examine violations of our modeling and identifiability assumptions. `SigCTO` remains beneficial on average under correlated process noise, but can worsen performance under partial observability or strong treatment-assignment confounding.

`SigCTO` **regularizer predicts distribution shifts in real-world clinical data.** Truly assessing newly optimized treatment plans on real patients is infeasible. However, with an offline dataset like MIMIC-IV, we can still evaluate two core components of `SigCTO`. First, Table 2 shows that `SigCTO` most reliably predicts observed costs on held-out patient trajectories compared to causal baselines. Second, we evaluate whether our regularizer predicts model mismatch by comparing the regularizer to the signature score between observed and predicted trajectories for held-out patients. The Spearman correlation of 0.26 ($p < 0.001$) indicates that larger regularizer values correspond to greater trajectory mismatch. Together, these results suggest that key components of `SigCTO` may transfer to real-world clinical data.

Table 2: **MIMIC-IV Results.** Correlations (↑) and absolute error (↓) of predicted costs on held-out data. All correlations are significant ($p < 0.001$) except TE-CDE Pearson ($p = 0.84$).

Method	Spearman	Pearson	Abs. Error
TE-CDE	-0.10	0.00	26.32±37.16
INSITE	0.66	0.41	19.98±102.24
`SigCTO`	**0.68**	**0.48**	**10.30**±38.35

7 Conclusion

We introduce a conservative continuous-time framework for treatment optimization from irregularly sampled observational data. By modeling patient dynamics as a controlled SDE and penalizing out-of-distribution treatment plans via a signature kernel MMD, our approach provably upper bounds the true treatment cost and empirically yields more reliable and better-performing plans than baselines.

Our framework has several limitations. Similar to other pessimistic approaches, restricting optimization to the data support may cause it to miss optimal treatment – particularly when it lies far outside the support. In clinical settings, selection bias and unmeasured confounding may further restrict the empirical support, risking that historic and potentially suboptimal treatment plans get reinforced rather than improved upon. Moreover, our MIMIC-IV evaluation suggests that our key components may transfer to clinical trajectories; however, it is best interpreted as an indirect, initial component-level validation, while clinical utility remains to be established in future work.

Our work further opens several directions for future work. Extending `SigCTO` to closed-loop policies or plans adapted at irregular decision points is a natural direction, since the regularizer already operates on a flexible time partition. Another important direction is translating `SigCTO` to real-world clinical settings. This includes relaxing modeling assumptions, such as full observability, or incorporating domain knowledge into the dynamics model. Practically, clinical deployment will require rigorous prospective evaluation, appropriate safety constraints, and human oversight.

Acknowledgments and Disclosure of Funding

We thank Junhyung Park for his time and for helpful discussions about the paper and the MCMD estimator. We thank Cristopher Salvi for his input on Signature Scores for neural SDE training. We thank Emilio Ferrucci for his input on causal identifiability, stochastic control, rough path theory, and for sharing useful pointers to the related literature. This work has been supported by the German Federal Ministry of Education and Research (Grant: 01IS24082). This work has been supported by the European Union (ERC, DYNAMICAUS, 101221985).

References

Peter Bank, Christian Bayer, Paul P Hager, Sebastian Riedel, and Tobias Nauen. Stochastic control with signatures. *SIAM Journal on Control and Optimization*, 63(5):3189–3218, 2025.

Bastien Berret and Frédéric Jean. Efficient computation of optimal open-loop controls for stochastic systems. *Automatica*, 115:108874, 2020.

Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations*, 2020.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.

Kaylee Burns, Tianhe Yu, Chelsea Finn, and Karol Hausman. Offline reinforcement learning at multiple frequencies. In *Conference on robot learning*, pages 2041–2051. PMLR, 2023.

Alison Callahan, Nigam H Shah, and Jonathan H Chen. Research and reporting considerations for observational studies using electronic health record data. *Annals of Internal Medicine*, 172 (11_Supplement):S79–S84, 2020.

Thomas Cass and Cristopher Salvi. Lecture notes on rough paths and applications to machine learning. *arXiv preprint arXiv:2404.06583*, 2024.

Bibhas Chakraborty and Susan A Murphy. Dynamic treatment regimes. *Annual review of statistics and its application*, 1(1):447–464, 2014.

Christa Cuchiero, Philipp Schmocker, and Josef Teichmann. Global universal approximation of functional input maps on weighted spaces. *Constructive Approximation*, pages 1–76, 2026.

Walter J Curran Jr, Rebecca Paulus, Corey J Langer, Ritsuko Komaki, Jin S Lee, Stephen Hauser, Benjamin Movsas, Todd Wasserman, Seth A Rosenthal, Elizabeth Gore, et al. Sequential vs concurrent chemoradiation for stage iii non–small cell lung cancer: randomized phase iii trial rtog 9410. *Journal of the National Cancer Institute*, 103(19):1452–1460, 2011.

Wei Dai, Rohit Rao, Anna Sher, Nessy Tania, Cynthia J Musante, and Richard Allen. A prototype qsp model of the immune response to sars-cov-2 for community development. *CPT: pharmacometrics & systems pharmacology*, 10(1):18–29, 2021.

Edward De Brouwer, Javier Gonzalez, and Stephanie Hyland. Predicting the impact of treatments over time with uncertainty aware neural differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 4705–4722. PMLR, 2022.

Alexander D'Amour, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021.

Changran Geng, Harald Paganetti, and Clemens Grassberger. Prediction of treatment response for combined chemo-and radiation therapy for non-small cell lung cancer patients using a biomathematical model. *Scientific reports*, 7(1):13542, 2017.

Sébastien Gros and Moritz Diehl. Numerical optimal control (draft), 2020.

Konstantin Hess and Stefan Feuerriegel. Stabilized neural prediction of potential outcomes in continuous time. In *The Thirteenth International Conference on Learning Representations*, 2025.

Konstantin Hess, Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Bayesian neural controlled differential equations for treatment effect estimation. In *The Twelfth International Conference on Learning Representations*, 2024.

Çağlar Hızlı, S. T. John, Anne Tuulikki Juuti, Tuure Tapani Saarinen, Kirsi Hannele Pietiläinen, and Pekka Marttinen. Causal modeling of policy interventions from treatment-outcome sequences. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 13050–13084. PMLR, 23–29 Jul 2023.

Melker Höglund, Emilio Ferrucci, Camilo Hernández, Aitor Muguruza Gonzalez, Cristopher Salvi, Leandro Sánchez-Betancourt, and Yufei Zhang. A neural RDE approach for continuous-time non-markovian stochastic control problems. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.

Nicolas Hoischen, Petar Bevanda, Max Beier, Stefan Sosnowski, Boris Houska, and Sandra Hirche. Operator models for continuous-time offline reinforcement learning. *arXiv preprint arXiv:2511.10383*, 2025.

Samuel Holt, Alihan Hüyük, Zhaozhi Qian, Hao Sun, and Mihaela van der Schaar. Neural laplace control for continuous-time delayed systems. In *International Conference on Artificial Intelligence and Statistics*, pages 1747–1778. PMLR, 2023.

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

Zacharia Issa, Blanka Horvath, Maud Lemercier, and Cristopher Salvi. Non-adversarial training of neural sdes with signature kernel scores. *Advances in Neural Information Processing Systems*, 36: 11102–11126, 2023.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV. *PhysioNet*, October 2024. doi: 10.13026/kpb9-mt58. Version 3.1.

Krzysztof Kacprzyk, Samuel Holt, Jeroen Berrevoets, Zhaozhi Qian, and Mihaela van der Schaar. ODE discovery for longitudinal heterogeneous treatment effects inference. In *The Twelfth International Conference on Learning Representations*, 2024.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Cevahir Koprulu, Franck Djeumou, and Ufuk Topcu. Neural stochastic differential equations for uncertainty-aware offline RL. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhen Li, Jie Chen, Eric Laber, Fang Liu, and Richard Baumgartner. Optimal treatment regimes: a review and empirical comparison. *International Statistical Review*, 91(3):427–463, 2023.

Judith J. Lok. Statistical modeling of causal effects in continuous time. *The Annals of Statistics*, 36 (3):1464–1507, June 2008. doi: 10.1214/009053607000000820.

Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *International Conference on Learning Representations*, 2019.

IK Minichmayr, E Dreesen, Maddalena Centanni, Z Wang, Y Hoffert, Lena E Friberg, and SG Wicha. Model-informed precision dosing: state of the art and future perspectives. *Advanced drug delivery reviews*, 215:115421, 2024.

Susan A Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(2):331–355, 2003.

Junhyung Park and Krikamol Muandet. A measure-theoretic approach to kernel conditional mean embeddings. In *Advances in Neural Information Processing Systems*, volume 33, pages 21247–21259. Curran Associates, Inc., 2020.

Junhyung Park, Uri Shalit, Bernhard Schölkopf, and Krikamol Muandet. Conditional distributional treatment effect with kernel conditional mean embeddings and u-statistic regression. In *International conference on machine learning*, pages 8401–8412. PMLR, 2021.

Huyên Pham. *Continuous-time Stochastic Control and Optimization with Financial Applications*, volume 61 of *Stochastic Modelling and Applied Probability*. Springer, Berlin, Heidelberg, 2009. ISBN 978-3-540-89499-5 978-3-540-89500-8. doi: 10.1007/978-3-540-89500-8.

Zhaozhi Qian, William Zame, Lucas Fleuren, Paul Elbers, and Mihaela van der Schaar. Integrating expert odes into neural odes: pharmacology and disease progression. *Advances in Neural Information Processing Systems*, 34:11364–11383, 2021.

James Robins and Miguel Hernan. Estimation of the causal effects of time-varying exposures. *Chapman & Hall/CRC Handbooks of Modern Statistical Methods*, pages 553–599, 2008.

James M Robins. Optimal structural nested models for optimal sequential decisions. In *Proceedings of the Second Seattle Symposium in Biostatistics: analysis of correlated data*, pages 189–326. Springer, 2004.

James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology, 2000.

Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American statistical Association*, 100(469):322–331, 2005.

Cristopher Salvi, Thomas Cass, James Foster, Terry Lyons, and Weixin Yang. The Signature Kernel is the solution of a Goursat PDE. *SIAM Journal on Mathematics of Data Science*, 3:873–899, January 2021. ISSN 2577-0187. doi: 10.1137/20M1366794. URL `http://arxiv.org/abs/2006.14794`. arXiv:2006.14794 [math].

Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. *Advances in neural information processing systems*, 30, 2017.

Nabeel Seedat, Fergus Imrie, Alexis Bellot, Zhaozhi Qian, and Mihaela van der Schaar. Continuous-time modeling of counterfactual outcomes using neural controlled differential equations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19497–19521. PMLR, 17–23 Jul 2022.

LB Sheiner and J-L Steimer. Pharmacokinetic/pharmacodynamic modeling in drug development. *Annual review of pharmacology and toxicology*, 40(1):67–95, 2000.

Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

Venkatesh Sivaraman, Leigh A Bukowski, Joel Levin, Jeremy M Kahn, and Adam Perer. Ignore, trust, or negotiate: understanding clinician acceptance of ai-based treatment recommendations in health care. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2023.

Jinghao Sun and Forrest W Crawford. Causal identification for continuous-time stochastic processes. *arXiv preprint arXiv:2211.15934*, 2022.

Jinghao Sun and Forrest W. Crawford. The role of discretization scales in causal inference with continuous-time treatment. *arXiv preprint arXiv:2306.08840*, 2023. URL `https://arxiv.org/abs/2306.08840`. Submitted; working paper.

Stratis Tsirtsis and Manuel Rodriguez. Finding counterfactually optimal action sequences in continuous state spaces. *Advances in Neural Information Processing Systems*, 36:3220–3247, 2023.

Toon Vanderschueren, Alicia Curth, Wouter Verbeke, and Mihaela Van Der Schaar. Accounting for informative sampling when learning to forecast treatment outcomes over time. In *International Conference on Machine Learning*, pages 34855–34874. PMLR, 2023.

Benjie Wang, Joel Jennings, and Wenbo Gong. Neural structure learning with stochastic differential equations. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=V1GM9xDvIY`.

Xin Wang, Shengfei Lyu, Chi Luo, Xiren Zhou, and Huanhuan Chen. Variational counterfactual intervention planning to achieve target outcomes. In *Forty-second International Conference on Machine Learning*, 2025.

Yanxun Xu and Zhiyue Zhang. Modeling and optimizing dynamic treatment regimens in continuous time. In *Statistics in Precision Health: Theory, Methods and Applications*, pages 513–535. Springer, 2024.

Andrew Ying. Causal identification for complex functional longitudinal studies. In *International Conference on Learning Representations*, volume 2025, pages 70816–70838, 2025.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Junjie Zhang, Jiafei Lyu, Xiaoteng Ma, Jiangpeng Yan, Jun Yang, Le Wan, and Xiu Li. Uncertainty-driven trajectory truncation for data augmentation in offline reinforcement learning. *arXiv preprint arXiv:2304.04660*, 2023a.

Zhiyue Zhang, Hongyuan Mei, and Yanxun Xu. Continuous-time decision transformer for healthcare applications. In *International conference on artificial intelligence and statistics*, pages 6245–6262. PMLR, 2023b.

Yunzhe Zhou, Zhengling Qi, Chengchun Shi, and Lexin Li. Optimizing pessimism in dynamic treatment regimes: A bayesian learning approach. In *International Conference on Artificial Intelligence and Statistics*, pages 6704–6721. PMLR, 2023.

A Summary of Notation

A summary of notation is provided in Table 3.

Table 3: Summary of notation

Symbol	Meaning
I	$I := [t_s, t_f]$, the considered finite time horizon
$(\Omega, \mathcal{F}, \mathbb{F}, P)$	the complete filtered probability space with filtration $\mathbb{F} = (\mathcal{F}_t)_{t \in I}$
A	the treatment space $A \subseteq \mathbb{R}^{d_u}$
N	the number of training samples
n_i	the number of measurements for patient i
$\mathcal{D}$	the dataset $\mathcal{D} := \{(u^{(i)}, y^{(i)}, z^{(i)})\}_{i \in [N]}$
d_z	dimension of the stochastic covariate process $Z : I \times \Omega \to \mathbb{R}^{d_z}$
d_y	dimension of the stochastic outcome process $Y : I \times \Omega \to \mathbb{R}^{d_y}$
d_u	dimension of the stochastic treatment process $U : I \times \Omega \to \mathbb{R}^{d_u}$
d_x	$d_x = d_z + d_y$ dimension of $X = (Z, Y)^\top$
U	the stochastic treatment process $U : I \times \Omega \to A$
Y	the stochastic outcome process $Y : I \times \Omega \to \mathbb{R}^{d_y}$
Z	the stochastic covariate process $Z : I \times \Omega \to \mathbb{R}^{d_z}$
X	the state process $X = (Z, Y) : I \times \Omega \to \mathbb{R}^{d_x}$
u	a deterministic treatment plan $u : I \to A$, or a sample of the stochastic process U
$X_{[a,b]}$	the restriction of the stochastic process X to $[a, b] \subseteq I$
$P_{X_{[a,b]}}$	the law of the restricted process $X_{[a,b]}$
$\mathcal{F}_t^X$	the sigma-algebra generated by the values of X up to time t
$P_{Y \mid X_{[a,b]}}$	the conditional law of a process Y given $X_{[a,b]}$
$\mathcal{M}_1(\Omega, \mathcal{A})$	the set of probability measures on measurable space $(\Omega, \mathcal{A})$
$\mathcal{Y}^{\mathcal{X}}$	the set of all functions $\mathrm{Funct}(\mathcal{X}, \mathcal{Y}) = \{f : \mathcal{X} \to \mathcal{Y}\}$
$\mathcal{L}^0((\mathcal{X}, \mathcal{A}_\mathcal{X}), (\mathcal{Y}, \mathcal{A}_\mathcal{Y}))$	the $\mathcal{A}_\mathcal{X} - \mathcal{A}_\mathcal{Y}$ measurable functions $f : \mathcal{X} \to \mathcal{Y}$
W	the standard Wiener process in the controlled SDE
μ, σ	the shared structural drift and diffusion functions of the controlled SDE
μ_0	the initial distribution of the state process X_0
$X^{(u,\mathrm{pot})}$	the potential state process that would be observed under treatment plan u
$X^{t_0,x_0,u}$	the true state process on $[t_0, t_f]$ initialized at $X_{t_0} = x_0$ and controlled by u
$\hat{X}^{t_0,x_0,u}$	the estimated state process induced by the learned dynamics model under initial condition x_0 and treatment plan u
$\mathcal{U}_{\mathrm{adm}}$	the class of admissible treatment plans
f, g	the running cost $f : I \times \mathbb{R}^{d_x} \times A \to \mathbb{R}$ and terminal cost $g : \mathbb{R}^{d_x} \to \mathbb{R}$
$\mathcal{J}(t_0, X, u)$	the expected cumulative cost of a state process X under treatment plan u from decision time t_0
t_0, x_0	the decision time and the corresponding decision-time state
K	the number of sub-intervals in the temporal partition of $[t_0, t_f]$
I_j	$I_j := [t_j, t_{j+1}]$, the j-th sub-interval of the temporal partition
$P_{X_{I_j}^{t_j,x,u_{I_j}}}$	the solution path law of the true dynamics on I_j starting at $X_{t_j} = x$ and $U_{I_j} = u_{I_j}$
$P_{\hat{X}_{I_j}^{t_j,x,u_{I_j}}}$	the solution path law of the learned dynamics on I_j starting at$\hat{X}_{t_j} = x$ and $U_{I_j} = u_{I_j}$
$D_j(x, u)$	the one-step discrepancy between true and learned dynamics on I_j given state x and plan u
h_j	the deterministic path functional that maps a path segment on I_j to its sub-interval cost plus continuation value
$\mathcal{G}$	the function class that defines the integral probability metric
$\mathrm{IPM}_\mathcal{G}$	the integral probability metric generated by function class $\mathcal{G}$
$c_\mathcal{G}$	the constant relating the one-step discrepancy to the chosen IPM
$R(u)$	the accumulated IPM regularizer for treatment plan u
λ	the regularization parameter controlling the degree of conservatism
$\mathcal{S}$	the signature transform of a path
$T(\mathbb{R}^d)$	the completed tensor algebra over $\mathbb{R}^d$
k_{sig}	the signature kernel $k_{\mathrm{sig}}(x, y) = \langle \mathcal{S}(x), \mathcal{S}(y) \rangle_{T(\mathbb{R}^d)}$
$H_\mathcal{S}(\mathbb{R}^d)$	the reproducing kernel Hilbert space induced by the signature kernel
μ_{P_X}	the kernel mean embedding of the law P_X
$\mu_{P_{X \mid U}}$	the conditional mean embedding of the conditional law $P_{X \mid U}$
$\mathrm{MMD}_{k_{\mathrm{sig}}}$	the maximum mean discrepancy induced by the signature kernel
$\mathrm{MCMD}_{k_{\mathrm{sig}}}$	the maximum conditional mean discrepancy induced by the signature kernel
B_1	the unit ball in $H_\mathcal{S}(\mathbb{R}^d)$

B Regularity Assumptions

Below, we detail regularity assumptions that ensure the controlled SDE admits a unique strong solution and that the SDE-governed optimization problem is well-defined.

B.1 Structural Dynamics Model

Let $I = [t_s, t_f]$, let $(\Omega, \mathcal{F}, \mathbb{F}, P)$ be a complete filtered probability space with filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in I}$ satisfying the usual conditions, $W = (W_t)_{t\in I}$ a $\mathbb{R}^{d_w}$-valued Wiener process adapted to $\mathbb{F}$, $U = (U_t)_{t\in I}$ the progressively $\mathbb{F}$-measurable treatment-process taking values in $A \subseteq \mathbb{R}^{d_u}$ and

$$\mu : \mathbb{R}^{d_x} \times A \to \mathbb{R}^{d_x}, \qquad \sigma : \mathbb{R}^{d_x} \times A \to \mathbb{R}^{d_x \times d_w}, \tag{5}$$

the measurable drift- and diffusion functions of the controlled SDE

$$\mathrm{d}X_t = \mu(X_t, U_t)\,\mathrm{d}t + \sigma(X_t, U_t)\,\mathrm{d}W_t, \qquad t \in I. \tag{6}$$

Assuming

(i) uniform Lipschitz condition in the state, uniformly over treatments, meaning there exists $K \geq 0$ such that for all $x, y \in \mathbb{R}^{d_x}$ and all $u \in A$,

$$\|\mu(x,u) - \mu(y,u)\| + \|\sigma(x,u) - \sigma(y,u)\| \leq K\|x - y\|.$$

(ii) the set of treatment processes are restricted such that

$$\mathbb{E}\left[\int_{t_0}^{t_f} \|\mu(0,u_t)\|^2 + \|\sigma(0,u_t)\|^2 \mathrm{d}t\right] < \infty$$

the controlled SDE Equation (6) admits a unique strong solution for every initial condition $(t_0, x_0) \in I \times \mathbb{R}^{d_x}$, for any progressively $\mathbb{F}$-measurable treatment process U, which we denote by $X^{(t_0,x_0,U)} = (X_t^{(t_0,x_0,U)})_{t\in I}$. For more details, see Pham [2009].

B.2 Class of Admissible Treatments

Assuming a controlled SDE (μ, σ, I) as in Equation (6) and measurable functions $f : I \times \mathbb{R}^{d_x} \times A \to \mathbb{R}$ and $g : \mathbb{R}^{d_x} \to \mathbb{R}$ s.t.

(i) g is lower-bounded

(ii) $\exists C > 0 : |g(x)| \leq C(1 + |x|^2) \quad \forall x \in \mathbb{R}^n$

the set of admissible controls for $(t_0, x_0) \in I \times \mathbb{R}^{d_x}$,

$$\mathcal{U}_{\mathrm{adm}} := \mathcal{U}^f_{(t_0,x_0)} := \left\{u \mid \mathbb{E}\left[\int_{t_0}^{t_f} |f(s, X_s^{(t_0,x_0,u)}, u_s)|\mathrm{d}s\right] < \infty\right\} \neq \emptyset$$

is non-empty, making the SDE-governed control problem

$$v(t_0,x_0) = \inf_{u\in\mathcal{U}_{\mathrm{adm}}} \underbrace{\mathbb{E}\left[\int_{t_0}^{t_f} f(s, X_s^{(t_0,x_0,u)}, u_s)\mathrm{d}s + g\left(X_{t_f}^{(t_0,x_0,u)}\right)\right]}_{=\mathcal{J}(t_0,x_0,u)}$$

well-defined, where $\mathcal{J}(t_0, x_0, u)$ is the cost function and $v(t_0, x_0)$ the associated value function.

B.3 On Identifiability Assumptions

Assumptions 4.1 and 4.2 are satisfied, for example, in the following fully observed setting:

$$\begin{aligned} \mathrm{d}X_t &= \mu(X_t, U_t)\,\mathrm{d}t + \sigma(X_t, U_t)\,\mathrm{d}W_t, \\ \mathrm{d}U_t &= f(U_t)\,\mathrm{d}t + g(U_t)\,\mathrm{d}B_t, \end{aligned}$$

where W and B are independent standard Wiener processes. Suppose (X, U) takes values in $\mathbb{R}^{d_x+d_u}$ with a known initial state, all coefficients are bounded and globally Lipschitz, and σ and g are diagonal with entries uniformly bounded away from zero in absolute value. Defining interventions by replacing U_t with $u(t)$ while preserving the state equation establishes Assumption 4.1. Under these conditions, the full joint trajectory law identifies the state drift and diffusion covariance through the generator of (X, U) [Wang et al., 2024, Appendix A.2]. These uniquely determine the response laws under admissible treatment plans, establishing Assumption 4.2.

C Proofs

C.1 Proof for Structural Identifiability Proposition

In this subsection, we prove Proposition 4.3, which we restate here:

Proposition (Structural Identifiability)**.** *Under Assumptions 4.1 and 4.2 and the regularity conditions in Appendix B.2, the law of $X^{(u,\mathrm{pot})}$ and consequently the expected cost $\mathcal{J}(t_0, X^{(u,\mathrm{pot})}, u)$ in Equation* (1) *are identifiable for every $u \in \mathcal{U}_{\mathrm{adm}}$.*

Proof. Fix $u \in \mathcal{U}_{\mathrm{adm}}$. According to Assumption 4.1 the law of X is governed by

$$\mathrm{d}X_t = \mu(X_t, U_t)dt + \sigma(X_t, U_t)\mathrm{d}W_t,$$

for all $u \in \mathcal{U}_{\mathrm{adm}}$, in the observational support or not. By Assumption 4.2 any learned $(\hat{\mu}, \hat{\sigma})$ that induces the observed law $P_{X,U}$ will yield a controlled response law for every deterministic $u \in \mathcal{U}_{\mathrm{adm}}$ that agrees with the true underlying dynamics. Hence, the law of $X^{(u,\mathrm{pot})}$ is uniquely determined by the observational law $P_{X,U}$. The regularity and integrability conditions in Appendix B.1 ensure existence and pathwise uniqueness of the controlled SDE as well as the existence of conditional expectations of the measurable functional

$$X^{(u,\mathrm{pot})} \mapsto \int_{t_0}^{t_f} f(t, X_t^{(u,\mathrm{pot})}, u_t)\,\mathrm{d}t + g(X_{t_f}^{(u,\mathrm{pot})}).$$

Hence identifiability of $X^{(u,\mathrm{pot})}$ implies identifiability of $\mathcal{J}(t_0, X^{(u,\mathrm{pot})}, u)$. □

C.2 Proof for Estimation Error Telescope Lemma

In this subsection, we prove Lemma 4.4, which we restate here:

Lemma (Estimation Error Telescope)**.** *For an admissible treatment plan $(u : I \to \mathbb{R}^{d_u}) \in \mathcal{U}_{\mathrm{adm}}$, initial state x_0, and any temporal partition $t_0 < t_1 < \cdots < t_K = t_f$,*

$$\mathcal{J}(t_0, X^{t_0,x_0,u}, u) - \mathcal{J}(t_0, \hat{X}^{t_0,x_0,u}, u) = \sum_{j=0}^{K-1} \mathbb{E}_{x \sim P_{\hat{X}_{t_j}}}[D_j(x, u)]$$

with one-step discrepancy on I_j between true and learned dynamics given by

$$D_j(x, u) = \mathbb{E}_{x_{I_j} \sim P_{X_{I_j}^{t_j,x,u_{I_j}}}}\left[h_j(x_{I_j})\right] - \mathbb{E}_{\hat{x}_{I_j} \sim P_{\hat{X}_{I_j}^{t_j,x,u_{I_j}}}}\left[h_j(\hat{x}_{I_j})\right]$$

and h_j is the deterministic path functional $x_{I_j} \mapsto \int_{t_j}^{t_{j+1}} f(s, x_s, u_s)\mathrm{d}s + \mathcal{J}(t_{j+1}, x_{t_{j+1}}, u)$ mapping a path segment x_{I_j} to a sum of the cumulative cost on I_j and the continuation value at t_{j+1}.

Without loss of generality, we assume $I = [0, 1]$ for notational convenience. Note the following.

Remark. Given an admissible $u \in \mathcal{U}$ we have by the path-wise uniqueness of the flow of the SDE for X, due to the Markovian structure of the SDE:

$$X_t^{t_0,x_0,u} = X_t^{(\tau, X_\tau^{t_0,x_0,u}, u)}, \quad t \geq \tau$$

for any stopping time τ valued in $[t_0, 1]$. Hence we obtain:

$$\mathcal{J}(t_0,x_0,u) = \mathbb{E}\left[\int_{t_0}^{\tau} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + \int_{\tau}^{1} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + g(X_1^{t_0,x_0,u})\right] \tag{7}$$

$$\overset{(*)}{=} \mathbb{E}\left[\mathbb{E}\left[\int_{t_0}^{\tau} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + \int_{\tau}^{1} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + g(X_1^{t_0,x_0,u}) \mid \mathcal{F}_\tau\right]\right] \tag{8}$$

$$\overset{(**)}{=} \mathbb{E}\left[\int_{t_0}^{\tau} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + \mathbb{E}\left[\int_{\tau}^{1} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + g(X_1^{t_0,x_0,u}) \mid \mathcal{F}_\tau\right]\right] \tag{9}$$

where in $(*)$ we used the tower property of the conditional expectation and in $(**)$ we used that τ a stopping time making $\int_{t_0}^{\tau} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s$ $\mathcal{F}_\tau$ measurable. Moreover, by shifting control and Brownian motion by $-\tau$, in the underlying controlled SDE, we obtain:

$$\mathbb{E}\left[\int_{\tau}^{1} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + g(X_1^{t_0,x_0,u}) \mid \mathcal{F}_\tau\right] = \mathcal{J}(\tau, X_\tau^{t_0,x_0,u}, u) \quad P - \text{a.s.}$$

such that

$$\mathcal{J}(t_0,x_0,u) = \mathbb{E}\left[\int_{t_0}^{\tau} f(s, X_s^{t_0,x_0,u}, u_s)\mathrm{d}s + \mathcal{J}(\tau, X_\tau^{t_0,x_0,u}, u)\right]$$

Proof. For $j \in \{0, \dots, K\}$, define the stochastic process $\tilde{X}^{(j)}$ that evolves up to time t_j according to the (estimated) SDE governed by $\mu_\theta, \sigma_\theta$ and starting from t_j according to the true SDE with μ, σ, meaning $\tilde{X}^{(j)} = (\tilde{X}_s^{(j)})_{s\in I}$ satisfies

$$\tilde{X}^{(j)}_{[0,t_j]} = \text{solution of} \quad x_0 + \int_0^{t_j} \mu_\theta(\tilde{X}_s^{(j)}, u_s)\mathrm{d}s + \int_0^{t_j} \sigma_\theta(\tilde{X}_s^{(j)}, u_s)dW_s, \tag{10}$$

$$\tilde{X}^{(j)}_{[t_j,1]} = \text{solution to} \quad \tilde{X}_{t_j}^{(j)} + \int_{t_j}^{1} \mu(\tilde{X}_s^{(j)}, u_s)\mathrm{d}s + \int_{t_j}^{1} \sigma(\tilde{X}_s^{(j)}, u_s)dW_s. \tag{11}$$

Note that $\tilde{X}^{(0)} := X := X^{(0,x_0,u)}$, $\tilde{X}^{(K)} = \hat{X} = \hat{X}^{(0,x_0,u)}$. Using the abbreviation

$$\tilde{\mathcal{J}}^{(j)} := \mathbb{E}\left[\int_0^{t_j} f(s, \tilde{X}_s^{(j)}, u_s)\mathrm{d}s + \mathcal{J}(t_j, \tilde{X}_{t_j}^{(j)}, u)\right]$$

and a "telescoping-sum" argument, we can write

$$\mathcal{J}(0, X^{(0,x_0,u)}, u) - \mathcal{J}(0, \hat{X}^{(0,x_0,u)}, u) = \tilde{\mathcal{J}}^{(0)} - \tilde{\mathcal{J}}^{(K)} = \sum_{j=0}^{K-1} \left(\tilde{\mathcal{J}}^{(j)} - \tilde{\mathcal{J}}^{(j+1)}\right).$$

Moreover

$$\begin{aligned}
\tilde{\mathcal{J}}^{(j)} &= \mathbb{E}\left[\int_0^{t_j} f(s, \tilde{X}_s^{(j)}, u_s)\,\mathrm{d}s + \mathbb{E}\Big[\int_{t_j}^{1} f(s, \tilde{X}_s^{(j)}, u_s)\,\mathrm{d}s + g(\tilde{X}_1^{(j)}) \;\Big|\; \tilde{\mathcal{F}}_{t_j}^{(j)}\Big]\right] \\
&= \mathbb{E}\Bigg[\int_0^{t_j} f(s, \tilde{X}_s^{(j)}, u_s)\,\mathrm{d}s + \mathbb{E}\Big[\int_{t_j}^{t_{j+1}} f(s, \tilde{X}_s^{(j)}, u_s)\,\mathrm{d}s \\
&\qquad + \underbrace{\mathbb{E}\Big[\int_{t_{j+1}}^{1} f(s, \tilde{X}_s^{(j)}, u_s)\,\mathrm{d}s + g(\tilde{X}_1^{(j)}) \;\Big|\; \tilde{\mathcal{F}}_{t_{j+1}}^{(j)}\Big]}_{\overset{\text{a.s.}}{=} \tilde{\mathcal{J}}\left(t_{j+1}, \tilde{X}_{t_{j+1}}^{(j)}, u\right)} \;\Big|\; \tilde{\mathcal{F}}_{t_j}^{(j)}\Big]\Bigg].
\end{aligned}$$

where $\tilde{\mathcal{F}}_t^{(j)} = \sigma\left(\tilde{X}_s^{(j)} \mid s \le t\right)$. Similarly

$$\tilde{\mathcal{J}}^{(j+1)} = \mathbb{E}\Bigg[\int_0^{t_j} f(s, \tilde{X}_s^{(j+1)}, u_s)\,\mathrm{d}s + \mathbb{E}\Big[\int_{t_j}^{t_{j+1}} f(s, \tilde{X}_s^{(j+1)}, u_s)\,\mathrm{d}s + \underbrace{\mathbb{E}\Big[\int_{t_{j+1}}^{1} f(s, \tilde{X}_s^{(j+1)}, u_s)\,\mathrm{d}s + g(\tilde{X}_1^{(j+1)}) \;\Big|\; \tilde{\mathcal{F}}_{t_{j+1}}^{(j+1)}\Big]}_{\overset{\text{a.s.}}{=} \mathcal{J}\left(t_{j+1}, \tilde{X}_{t_{j+1}}^{(j+1)}, u\right)} \;\Big|\; \tilde{\mathcal{F}}_{t_j}^{(j+1)}\Big]\Bigg].$$

Then, as $\tilde{X}_s^{(j)} = \tilde{X}_s^{(j+1)}\ \forall s \le t_j$, we have $\tilde{\mathcal{F}}_{t_j}^{(j)} = \tilde{\mathcal{F}}_{t_j}^{(j+1)}$ from which we can write

$$\begin{aligned}
&\tilde{\mathcal{J}}^{(j)} - \tilde{\mathcal{J}}^{(j+1)} \\
=&\mathbb{E}\Bigg[\mathbb{E}\Bigg[\left(\int_{t_j}^{t_{j+1}} f(s, \tilde{X}_s^{(j)}, u_s)\mathrm{d}s + \mathcal{J}(t_{j+1}, \tilde{X}_{t_{j+1}}^{(j)}, u)\right) \\
&\quad - \left(\int_{t_j}^{t_{j+1}} f(s, \tilde{X}_s^{(j+1)}, u_s)\mathrm{d}s + \mathcal{J}(t_{j+1}, \tilde{X}_{t_{j+1}}^{(j+1)}, u)\right) \mid \tilde{\mathcal{F}}_{t_j}^{(j)}\Bigg]\Bigg] \\
\overset{(a)}{=}&\mathbb{E}\left[\int_\Omega \left(h_j(\tilde{X}_{I_j}^{(j)}(\omega')) - h_j(\tilde{X}_{I_j}^{(j+1)}(\omega'))\right) dP^{(t_j)}(\omega, d\omega')\right] \\
\overset{(b)}{=}&\mathbb{E}_{y\sim P_{\tilde{X}_{t_j}^{(j)}}}\left[\underbrace{\mathbb{E}_{P_{X_{I_j}^{t_j,y,u_{I_j}}}}\left[h_j(x_{I_j})\right] - \mathbb{E}_{P_{\hat{X}_{I_j}^{t_j,y,u_{I_j}}}}\left[h_j(\tilde{x}_{I_j})\right]}_{D_j(y,u)}\right]
\end{aligned}$$

where in (a) we inserted the deterministic function h_j given by

$$(\gamma : I_j \mapsto \mathbb{R}^n) \to \int_{t_j}^{t_{j+1}} f(s, \gamma_s, u_s)\mathrm{d}s + \mathcal{J}(t_{j+1}, X^{(t_{j+1}, \gamma_{t_{j+1}}, u)}, u)$$

and in (b) we used a path notation for the push-forward measure. Note the dependence of h_j on $u_{[t_j, t_{j+1}]}$, thus the subscript j. Altogether we therefore obtain

$$\mathcal{J} - \hat{\mathcal{J}} = \sum_{j=0}^{K-1} \mathbb{E}_{y\sim P_{\tilde{X}_{t_j}^{(j)}}}\left[\mathbb{E}_{P_{X_{I_j}^{t_j,y,u_{I_j}}}}\left[h_j(x_{I_j})\right] - \mathbb{E}_{P_{\hat{X}_{I_j}^{t_j,y,u_{I_j}}}}\left[h_j(\tilde{x}_{I_j})\right]\right].$$

□

C.3 Proof for Optimality Gap Lemma

In this subsection, we prove Lemma 4.7, which we restate here:

Lemma (Optimality Gap of the Conservative Optimizer)**.** *Let $\hat{u}$ be the minimizer of the conservative objective in Equation* (4) *and u^* be the minimizer of the true objective in Equation* (3)*. If there exists a $c_{\mathcal{G}} \ge 0$ s.t. the increment bound in Lemma 4.6 holds for each $j \in \{0, \dots, K-1\}$ and choose $\lambda \ge c_{\mathcal{G}}$, then the following optimality gap holds:*

$$\mathcal{J}(t_0, X^{t_0,x_0,\hat{u}}, \hat{u}) - \mathcal{J}(t_0, X^{t_0,x_0,u^*}, u^*) \le (c_{\mathcal{G}} + \lambda)R(u^*).$$

For the following proof, we introduce the shorthand notation

$$\begin{aligned}
J(u) &:= \mathcal{J}(t_0, X^{t_0,x_0,u}, u), \\
\hat{J}(u) &:= \mathcal{J}(t_0, \hat{X}^{t_0,x_0,u}, u), \\
R(u) &:= \sum_{j=0}^{K-1} \mathbb{E}_{x\sim P_{\hat{X}_{t_j}}}\left[\mathrm{IPM}_{\mathcal{G}}[P_{X_{I_j}^{t_j,x,u_{I_j}}}, P_{\hat{X}_{I_j}^{t_j,x,u_{I_j}}}]\right].
\end{aligned}$$

Proof of Lemma 4.7. By Lemma 4.4,

$$J(u) - \hat{J}(u) = \sum_{j=0}^{K-1} \mathbb{E}_{x \sim P_{\hat{X}_{t_j}}} [D_j(x, u)].$$

Hence, for every $u \in \mathcal{U}_{\text{adm}}$,

$$\begin{aligned} |J(u) - \hat{J}(u)| &\leq \sum_{j=0}^{K-1} \mathbb{E}_{x \sim P_{\hat{X}_{t_j}}} [|D_j(x, u)|] \\ &\leq c_{\mathcal{G}} \sum_{j=0}^{K-1} \mathbb{E}_{x \sim P_{\hat{X}_{t_j}}} \left[\text{IPM}_{\mathcal{G}}[P_{X_{I_j}^{t_j, x, u_{I_j}}}, P_{\hat{X}_{I_j}^{t_j, x, u_{I_j}}}] \right] \\ &= c_{\mathcal{G}}\, R(u). \end{aligned}$$

Now let

$$\hat{u} \in \arg\min_{u \in \mathcal{U}_{\text{adm}}} \hat{J}(u) + \lambda R(u), \qquad u^* \in \arg\min_{u \in \mathcal{U}_{\text{adm}}} J(u),$$

and assume $\lambda \geq c_{\mathcal{G}}$. Then

$$\begin{aligned} J(\hat{u}) &\leq \hat{J}(\hat{u}) + c_{\mathcal{G}} R(\hat{u}) \\ &\leq \hat{J}(\hat{u}) + \lambda R(\hat{u}) \\ &\leq \hat{J}(u^*) + \lambda R(u^*) \\ &\leq J(u^*) + c_{\mathcal{G}} R(u^*) + \lambda R(u^*), \end{aligned}$$

where the third line uses the optimality of $\hat{u}$ for the conservative objective and the last line uses again $|J(u^*) - \hat{J}(u^*)| \leq c_{\mathcal{G}} R(u^*)$. Rearranging yields

$$J(\hat{u}) - J(u^*) \leq (c_{\mathcal{G}} + \lambda) R(u^*).$$

□

D Integral probability metrics and MMD

To quantify the distance between two probability distributions, a widely used approach is to express it in terms of integral probability metrics (IPM):

Definition (Integral probability metric (IPM))**.** Let $(\mathcal{X}, \mathcal{A}_{\mathcal{X}})$ be a measurable space, $P, Q \in \mathcal{M}^1(\mathcal{X}, \mathcal{A}_{\mathcal{X}})$ and $F \subseteq \mathcal{L}^0(\mathcal{X}, \mathbb{R}) \cap B(\mathcal{X}, \mathbb{R})$ a class of measurable and bounded test functions. The integral probability metric (IPM) generated by F is

$$\text{IPM}[F, P, Q] := \sup_{f \in F} \left| \mathbb{E}_P[f] - \mathbb{E}_Q[f] \right|$$

Remark. For the choice of F, we have the following tradeoff:

(i) rich enough s.t. $\text{IPM}[F, P, Q] = 0 \Leftrightarrow P = Q$

(ii) not too large to estimate $\text{IPM}[F, P, Q]$ quickly

Example. Notable function classes, used in machine learning:

(i) For $F_{TV} = \{f \in \mathcal{L}^0(\mathcal{X}, \mathbb{R}) \mid \sup_{x \in \mathcal{X}} |f(x)| \leq 1\}$, $\text{IPM}[F, P, Q] =: \|P - Q\|_{TV}$ is called the total variational distance

(ii) For $F_K = \{\chi_{(-\infty, t]} \mid t \in \mathbb{R}\}$ on $\mathcal{X} = \mathbb{R}$, $\text{IPM}[F, P, Q]$ is called the Kolmogorov / L^∞-distance, the max-norm of their cumulative distributions.

(iii) for metric space $(\mathcal{X}, d)$ ($\mathcal{A}_{\mathcal{X}} = \mathcal{B}(\mathcal{T}_d)$) and $F_W = \{f \in L^0(\mathcal{X}, \mathbb{R}) \cap B(\mathcal{X}, \mathbb{R}) \mid \|f\|_L := \sup_{x \neq y} \frac{|f(x) - f(y)|}{d(x, y)} \leq 1\}$, $\text{IPM}[F_W, P, Q]$ is called earth mover / Wasserstein / L^1-distance

Definition (Maximum Mean Discrepancy)**.** Let $(\mathcal{X}, \mathcal{A}_\mathcal{X})$ be a measurable space, $(H \subseteq \mathbb{C}^\mathcal{X}, \langle\cdot,\cdot\rangle_H, k : \mathcal{X} \times \mathcal{X} \to \mathbb{C})$ a RKHS, $P, Q \in \mathcal{M}^1(\mathcal{X}, \mathcal{A}_\mathcal{X})$ and $\mathcal{H} = \{f \in H \mid \|f\|_H \leq 1\}$. The integral probability metric $\mathrm{IPM}[\mathcal{H}, P, Q]$ is called maximum mean discrepancy (MMD), written

$$\begin{aligned}\mathrm{MMD}_k[P,Q] := \mathrm{IPM}[\mathcal{H},P,Q] &= \sup_{f\in\mathcal{H}} \left| \int_\mathcal{X} f(x)\, dP(x) - \int_\mathcal{X} f(x)\, dQ(x) \right| \\ &= \sup_{f\in\mathcal{H}} |\langle f, \mu_P - \mu_Q\rangle_H| = \|\mu_P - \mu_Q\|_H\end{aligned}$$

where we used $f(x) = \langle f, k(\cdot, x)\rangle_H$.

Remark (Expression of MMD, definiteness). Note:

(i) We can express the MMD in terms of kernel k as

$$\begin{aligned}\mathrm{MMD}_k^2[P,Q] &= \|\mu_P - \mu_Q\|_\mathcal{H}^2 \\ &= \mathbb{E}_{X,X'}[k(X,X')] - 2\mathbb{E}_{X,Y}[k(X,Y)] + \mathbb{E}_{Y,Y'}[k(Y,Y')]\end{aligned}$$

where $X, X' \sim P$, $Y, Y' \sim Q$ independent, and we use

$$\begin{aligned}\|\mu_P\|_H^2 &= \left\langle \int k(\cdot,x')\mathrm{d}P(x'), \int k(\cdot,x)dP(x) \right\rangle_H \\ &= \iint \langle k(\cdot,x'), k(\cdot,x)\rangle_H \mathrm{d}P(x')\mathrm{d}P(x) \\ &= \mathbb{E}_{X,X'}[k(X,X')]\end{aligned}$$

(ii) If H resp. k is characteristic, the MMD is definite, meaning

$$\mathrm{MMD}_k[P,Q] = 0 \Leftrightarrow P = Q$$

E Signature Kernel MMD

E.1 Signature Kernel MMD

A suitable tool for learning from streamed data is the signature transform, a feature map $\mathcal{S} : C_p(I, \mathbb{R}^d) \to T(\mathbb{R}^d)$ that converts a data stream (a continuous path $x \in C_p(\mathbb{R}^d)$ of bounded variation) into a graded sequence of iterated integrals

$$\mathcal{S}(x) = \Big(1, \mathcal{S}^{(1)}(x), \ldots, \mathcal{S}^{(k)}(x), \ldots\Big) \in \prod_{k=0}^\infty \big(\mathbb{R}^d\big)^{\otimes k},$$

$$\mathcal{S}^{(k)}(x) = \int_{t_0<t_1<\ldots<t_k<t_f} dx_{t_1} \otimes \ldots \otimes dx_{t_k} \in \big(\mathbb{R}^d\big)^{\otimes k}$$

that form an expressive representation of the path in the Hilbert space

$$T(\mathbb{R}^d) := \left\{ a = (a_k)_{k\in\mathbb{N}} \in \prod_{k=0}^\infty \big(\mathbb{R}^d\big)^{\otimes k} \mid \sum_{k=0}^\infty \|a_k\|^2 < \infty \right\}$$

and capture the order and interaction of its increments. An analogy is often drawn to monomials: just as polynomials are dense in $C(K)$ on compact $K \subset \mathbb{R}^d$ by Stone–Weierstraß, finite linear combinations of signature coordinates are dense in the continuous functionals on suitable compact sets of paths, so the signature gives a universal feature map for path-valued data. Equipped with the scalar product

$$\langle a, b\rangle_{T(\mathbb{R}^d)} := \sum_{k\geq 0} \langle a_k, b_k\rangle_{(\mathbb{R}^d)^{\otimes k}}$$

using the standard Euclidean inner products on $(\mathbb{R}^d)^{\otimes k}$, the signature feature map induces the signature kernel, $k_{\mathrm{sig}}(x,y) := \langle \mathcal{S}(x), \mathcal{S}(y)\rangle_{T(\mathbb{R}^d)}$, with the corresponding RKHS $H_\mathcal{S}(\mathbb{R}^d)$. This allows the embedding of laws on path-space by the kernel mean map $\mu_{P_X} := \mathbb{E}[k_{\mathrm{sig}}(X,\cdot)]$ (resp. the conditional mean embedding $\mu_{P_{X|U}} := \mathbb{E}[k_{\mathrm{sig}}(X,\cdot) \mid U]$), establishing the *signature kernel MMD*

$$\mathrm{MMD}_{k_{\mathrm{sig}}}(P_X, P_Y) = \|\mu_{P_X} - \mu_{P_Y}\|_{H_\mathcal{S}(\mathbb{R}^d)} = \mathrm{IPM}_{D_1}[P_X, P_Y]$$

where $D_1 := D_1(H_\mathcal{S}(\mathbb{R}^d)) := \{f \in H_\mathcal{S}(\mathbb{R}^d) \mid \|f\|_{H_\mathcal{S}(\mathbb{R}^d)} \leq 1\}$.

E.2 Signature Kernel MMD Properties

The IPM bound in Equation (4) is exact whenever the one-step continuation functional h_j belongs to the function class $\mathcal{G}$ that induces the IPM, for example $\mathcal{G}_{\mathrm{TV}} = \{g : \|g\|_\infty \leq 1\}$ for the total variation distance (TV) or $\mathcal{G} = D_1(H_{\mathcal{S},j}(\mathbb{R}^{d_x}))$ for the signature kernel MMD, where $D_1(H_{\mathcal{S},j}(\mathbb{R}^{d_x}))$ is the unit-ball in the RKHS $H_{\mathcal{S},j}(\mathbb{R}^{d_x})$ over the interval $[t_j, t_{j+1}]$ induced by the signature-kernel (c.f. Section 4.3).

Moreover, for the unit-ball classes considered here, if the path functional is not in $\mathcal{G}$, i.e. $h_j \notin \mathcal{G}$, but is contained in the larger ambient normed space of $\mathcal{G}$, e.g. inside L^∞ for the total variation distance (resp. inside the signature-kernel RKHS $H_{\mathcal{S},j}(\mathbb{R}^{d_x})$ for the signature kernel MMD), a rescaled version of the vector h_j is contained in $\mathcal{G}$, e.g. $\frac{h_j}{\|h_j\|_\infty} \in \mathcal{G}_{\mathrm{TV}}$ (resp. $\frac{h_j}{\|h_j\|_{H_{\mathcal{S},j}(\mathbb{R}^{d_x})}} \in D_1(H_{\mathcal{S},j}(\mathbb{R}^{d_x}))$). This gives the scaling factor $c_{\mathcal{G}}$ in Lemma 4.6, with $c_{\mathcal{G}} := \|h_j\|_\infty$ for the TV (resp. $c_{\mathcal{G}} := \|h_j\|_{H_{\mathcal{S},j}(\mathbb{R}^{d_x})}$ for the signature kernel MMD).

Moreover, for the signature kernel MMD, if the path functional is not even in the larger ambient normed space of $\mathcal{G}$, i.e., $h_j \notin H_{\mathcal{S},j}(\mathbb{R}^{d_x})$, one can provide an approximation-based justification for the discrepancy penalty using weighted signature approximation. Under suitable regularity and integrability conditions, h_j belongs to the weighted function space considered by Cuchiero et al. [2026] with the corresponding weight integrable under both solution path laws; see Cuchiero et al. [2026] for weighted signature approximation results. By the universality of the signature kernel on such weighted function spaces (see Cuchiero et al. [2026]), for every $\varepsilon > 0$ there exists $g_{j,\varepsilon} \in H_{\mathcal{S},j}(\mathbb{R}^{d_x})$ such that

$$\mathbb{E}_{P_j^y}[|h_j - g_{j,\varepsilon}|] \leq \varepsilon, \qquad \mathbb{E}_{\hat{P}_j^y}[|h_j - g_{j,\varepsilon}|] \leq \varepsilon.$$

We refer to Cuchiero et al. [2026] for details. Therefore

$$|D_j(y,u)| \leq \left|\mathbb{E}_{P_j^y}[g_{j,\varepsilon}] - \mathbb{E}_{\hat{P}_j^y}[g_{j,\varepsilon}]\right| + 2\varepsilon \leq \underbrace{\|g_{j,\varepsilon}\|_{H_{\mathcal{S},j}(\mathbb{R}^{d_x})}}_{c_{\mathcal{G},\varepsilon}} \mathrm{MMD}_{k_j}[P_j^y, \hat{P}_j^y] + 2\varepsilon.$$

Thus, under RKHS realizability, the scaled signature-kernel MMD bounds the cost discrepancy without an approximation remainder. Under the approximation assumptions above, it instead provides a bound with an arbitrarily small additive remainder and an approximation-dependent multiplier. This motivates its use as a computable discrepancy regularizer beyond RKHS realizability.

Under the assumptions of Park et al. [2021], including the standard boundedness and universality conditions on the kernels and suitable estimator regularization, the empirical MCMD is a consistent estimator of the corresponding population conditional MMD used in our regularizer.

E.3 Signature Kernel MMD Estimator

To estimate the conditional discrepancy between observed and model-predicted trajectories, we use samples $\{x^{(i)}, u^{(i)}, x_0^{(i)}\}_{i\in[N]}$ and $\{x'^{,(i)}, u'^{,(i)}, x_0'^{,(i)}\}_{i\in[M]}$ from joint distributions P_{X,U,X_0} and $P_{X',U',X_0'}$, respectively. For a fixed, but arbitrary query conditioning $(x_0, u) \in \mathbb{R}^{d_x} \times \mathcal{U}_{\mathrm{adm}}$, we have the plug-in estimate for the square of the MCMD function,

$$\begin{aligned}\hat{M}^2(P_{X|X_0=x_0,U=u}, P_{X'|X_0'=x_0,U'=u})(x_0,u) = {} & k_{X_0,U}^\top(x_0,u) W_{X_0,U} K_X W_{X_0,U}^\top k_{X_0,U}(x_0,u) \\ & - 2k_{X_0,U}^\top(x_0,u) W_{X_0,U} K_{X,X'} W_{X_0',U'}^\top k_{X_0',U'}(x_0,u) \\ & + k_{X_0',U'}^\top(x_0,u) W_{X_0',U'} K_{X'} W_{X_0',U'}^\top k_{X_0',U'}(x_0,u).\end{aligned}$$

where $(K_X)_{ij} = k_{sig}(x^{(i)}, x^{(j)})$ (analogous for $(K_{X'})_{ij}$ and $(K_{X,X'})_{ij} = k_{sig}(x^{(i)}, x'^{,(j)})$, $(K_{X_0,U})_{ij} = k_{X_0,U}((x_0^{(i)}, u^{(i)}), (x_0^{(j)}, u^{(j)}))$ (analogous for $(K_{X_0',U'})_{ij}$), $W_{X_0,U} = (K_{X_0,U} + N\varphi \mathbf{I}_N)^{-1}$ (analogous for $W_{X_0',U'}$), $(k_{X_0,U}(x_0,u))_i = k_{X_0,U}((x_0^{(i)}, u^{(i)}), (x_0,u))$ (analogous for $(k_{X_0',U'}(x_0',u'))_i$. φ is the ridge regression regularization parameter (see Park and Muandet [2020] for details). Although the target distributions are conditional on the same query (x_0, u), the estimator uses samples from the joint laws, whose initial conditions and treatment plans can vary across observations.

For estimation, we equip the conditioning space with the kernel

$$k_{X_0,U}\big((x_0,u),(\widetilde{x}_0,\widetilde{u})\big) = k_{X_0}(x_0,\widetilde{x}_0)\, k_U(u,\widetilde{u}),$$

where k_{X_0} is a Gaussian RBF kernel and k_U is a signature kernel on treatment paths. The corresponding RKHS is $\mathcal{H}_0 \otimes \mathcal{H}_U$.

In `SigCTO`, we set $N = M = N_{\text{val}}$ and use the held-out dataset $\mathcal{D}_{\text{val}} = \{(x^{(i)}, u^{(i)}, x_0^{(i)})\}_{i=1}^{N_{\text{val}}}$ of observed trajectories and their associated treatment plans and initial conditions. For each $i \in [N_{\text{val}}]$, we set $(x_0'^{(i)}, u'^{(i)}) = (x_0^{(i)}, u^{(i)})$ and generate a predicted trajectory $x'^{(i)}$ using the estimated dynamics model. This yields the prediction dataset $\mathcal{D}_{\text{pred}} = \{(x'^{(i)}, u'^{(i)}, x_0'^{(i)})\}_{i=1}^{N_{\text{val}}}$. We use these two datasets in the estimator above to estimate the signature MMD between the observed and model-predicted trajectory distributions conditional on a common query (x_0, u). The computational complexity of evaluating the signature kernel MMD estimator is as follows. The signature kernel evaluations via the Goursat-PDE solver scale as $O(dl^2)$, with path-dimension d and time-length l. Antidiagonal GPU parallelization reduces the wall-clock dependence on l to linear, with sufficient parallel resources [Salvi et al., 2021]. With N samples, computing $W_{X_0,U}$ is a one-time cubic cost in N, and each regularizer evaluation scales quadratically in N.

F `SigCTO` Algorithm

In Algorithm 1, we provide the full `SigCTO` algorithm. The regularizer $R(u)$ is defined for any partition $t_0 < t_1 < \cdots t_K = t_f$. In practice, we choose equidistant intervals. This does not require regularly sampled data: snippets may contain different numbers of observations at irregular times. We pool observed and predicted reference snippets separately across intervals and precompute the MCMD reference matrices once. During optimization, we reuse these matrices to evaluate the conditional discrepancy at each predicted boundary state and candidate treatment segment, then sum the discrepancies across intervals, as described in the algorithm.

G Datasets and Treatment Optimization Tasks

In this section, we provide details on the PK/PD simulators and on the MIMIC-IV dataset. Further, we provide for each dataset the treatment optimization objective.

G.1 Cancer

The cancer dataset is simulated using the lung cancer PK/PD model of Geng et al. [2017], which has been used extensively in longitudinal treatment-effect benchmarks.

Dynamics. The state is $X_t = (V_t, C_t) \in \mathbb{R}_+^2$, where V_t denotes tumor volume and C_t denotes the chemotherapy drug concentration. The control is $U_t = (U_{c,t}, U_{r,t}) \in \mathbb{R}_+^2$, where $U_{c,t}$ is the administered chemotherapy dose and $U_{r,t}$ is the radiotherapy dose.

Tumor volume evolves according to an ODE with multiplicative noise,

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \left(\rho \log\Big(\frac{K}{V(t)}\Big) - \beta_c\, C(t) - \big(\alpha_r\, U_r(t) + \beta_r\, U_r(t)^2\big) + e_t \right) V(t), \tag{12}$$

where $e_t \sim \mathcal{N}(0, \sigma^2)$. Equivalently, this can be written as the SDE

$$\mathrm{d}V_t = \Big[\rho \log\Big(\frac{K}{V_t}\Big) - \beta_c\, C_t - \alpha_r\, U_{r,t} - \beta_r\, U_{r,t}^2\Big] V_t\, \mathrm{d}t + \sigma\, V_t\, \mathrm{d}W_t, \tag{13}$$

with standard Brownian motion W_t. We set $(\rho, K, \beta_c, \alpha_r, \beta_r)$ to the mean values of the prior distribution reported by Geng et al. [2017] and fix the noise scale to $\sigma = 0.1$ in our simulations.

The chemotherapy concentration follows exponential decay with input from administered chemotherapy:

$$\frac{\mathrm{d}C}{\mathrm{d}t} = -k_C\, C(t) + U_c(t), \tag{14}$$

where $k_C = 0.5$.

Treatment Process. For each simulated patient trajectory, we sample one of two treatment protocols inspired by Curran Jr et al. [2011], Hess et al. [2024], Vanderschueren et al. [2023]: *sequential* therapy (three weeks of weekly chemotherapy followed by three weeks of radiotherapy) or *concurrent* therapy (biweekly chemo–radiotherapy administration over six weeks). Treatments are modeled as bang-bang controls: at each dosing time t, chemotherapy is either off or delivered at a fixed dose, $U_{c,t} \in \{0, 5\}$ (mg), and radiotherapy is either off or delivered at a fixed fraction dose, $U_{r,t} \in \{0, 2\}$ (Gy), where Gy denotes Gray (ionizing radiation dose).

Algorithm 1 Conservative Treatment Optimization with SigCTO

Require: Training and validation datasets $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$; objective function $\mathcal{J}(t_0, X, u)$; treatment horizon $[t_0, t_f]$; number of intervals $K \geq 1$; regularization strength λ; parameterized treatment family $\{u_\phi : \phi \in \Psi\}$; patient state x_0 at decision time t_0.
Ensure: Optimized treatment plan $u_{\phi_{\text{opt}}}$.

****Learn a dynamics model****
Initialize the parameters of the dynamics model θ.
Train θ on $\mathcal{D}_{\text{train}}$.
Fix the learned dynamics model.

****Conservative treatment optimization****
Initialize reference-sample collections for computing regularizer, $\mathcal{D}_1 \leftarrow \emptyset$ and $\mathcal{D}_2 \leftarrow \emptyset$.
Construct equally spaced interval boundaries $t_j \leftarrow t_0 + \frac{j}{K}(t_f - t_0), \qquad j = 0, \ldots, K$.
Set $I_j \leftarrow [t_j, t_{j+1}]$ for $j = 0, \ldots, K-1$.
for each validation trajectory $(x^{(i)}, u^{(i)}, x_0^{(i)}) \in \mathcal{D}_{\text{val}}$ **do**
    Predict $\hat{x}^{(i)}$ using the learned dynamics model with initial state $x_0^{(i)}$ and treatment plan $u^{(i)}$.
    for $j = 0, \ldots, K-1$ **do**
        Add the observed reference sample $\left(x_{t_j}^{(i)}, u_{I_j}^{(i)}, x_{I_j}^{(i)}\right)$ to $\mathcal{D}_1$.
        Add the predicted reference sample $\left(\hat{x}_{t_j}^{(i)}, u_{I_j}^{(i)}, \hat{x}_{I_j}^{(i)}\right)$ to $\mathcal{D}_2$.
    end for
end for

Using $\mathcal{D}_1$ and $\mathcal{D}_2$, compute and store K_X, $K_{X'}$, $K_{X,X'}$, $K_{X_0,U}$, and $K_{X_0',U'}$, (as in Appendix E.3).
Precompute and store $W_{X_0,U}$ and $W_{X_0',U'}$.

Initialize the treatment-plan parameters ϕ.
repeat
    Construct the candidate treatment plan u_ϕ.
    Predict patient trajectories under u_ϕ using the learned dynamics model: $\hat{X}^{t_0,x_0,u_\phi}$.
    Estimate the model-based surrogate cost $\hat{\mathcal{J}}(u_\phi) = \mathcal{J}\left(t_0, \hat{X}^{t_0,x_0,u_\phi}, u_\phi\right)$.
    Extract the predicted boundary states $\hat{x}_{\phi,t_0}, \ldots, \hat{x}_{\phi,t_{K-1}}$.
    $R(u_\phi) \leftarrow 0$
    for $j = 0, \ldots, K-1$ **do**
        Evaluate $M^2(\hat{x}_{\phi,t_j}, u_{\phi,I_j})$ using the precomputed reference matrices. ▷ See Appendix E.3
        Update $R(u_\phi) \leftarrow R(u_\phi) + \sqrt{M^2(\hat{x}_{\phi,t_j}, u_{\phi,I_j})}$.
    end for
    Update ϕ to minimize
$$\hat{\mathcal{J}}(u_\phi) + \lambda R(u_\phi).$$
until convergence
Set $\phi_{\text{opt}} \leftarrow \phi$.
return $u_{\phi_{\text{opt}}}$.

Initial State. Following Geng et al. [2017], we first sample a cancer stage and then sample the initial tumor volume V_0 from the corresponding stage-conditional prior defined in Geng et al. [2017]. The initial chemo concentration C_0 is set to zero.

Simulation and Preprocessing. We generate 800 patient trajectories over a 60-day time window ($t \in [0, 60]$) for training and sample 128 for validation purposes. To induce irregular observation patterns, we randomly mask 30% of observations uniformly at random. We apply a log-transform to the state variables and then standardize (zero mean, unit variance) using statistics computed on the training set only. Controls are min–max normalized to $[0, 1]$ using the known dose bounds.

Treatment Optimization Objective 1 (Explicit Target State). Let $X_t = (V_t, C_t)$ denote the cancer state, where V_t is tumor volume. Given an initial patient state $V_{t_0} = v_0$ and initial concentration $C_{t_0} = 0$, we seek a treatment plan that minimizes terminal tumor volume while regularizing treatment intensity:

$$\min_{u \in \mathcal{U}} \mathbb{E}\left[\int_{t_0}^{t_f} \lambda_c \, u_{c,t}^2 + \lambda_r \, u_{r,t}^2 \, \mathrm{d}t \; + \; V_{t_f}^2 \;\middle|\; X_{t_0} = (v_0, 0) \right], \qquad t_f = 60, \tag{15}$$

where $\lambda_c = \lambda_r = 10^{-3}$ in our experiments.

Treatment Optimization Objective 2 (Relative Target State). In the second cancer objective, given an initial patient state $V_{t_0} = v_0$ and $C_{t_0} = 0$, the goal is to achieve a prescribed *relative* reduction in tumor volume at the terminal time. We define the relative target $v^\star := 0.3\, v_0$, corresponding to a 70% reduction from the initial state. We minimize the deviation from this target while regularizing treatment intensity:

$$\min_{u \in \mathcal{U}} \mathbb{E}\left[\int_{t_0}^{t_f} \lambda_c \, u_{c,t}^2 + \lambda_r \, u_{r,t}^2 \, \mathrm{d}t \; + \; \left(V_{t_f} - v^\star\right)^2 \;\middle|\; X_{t_0} = (v_0, 0) \right], \qquad t_f = 60, \tag{16}$$

with $\lambda_c = \lambda_r = 10^{-3}$.

For both tasks, we solve the optimization problem 15 times, corresponding to 15 different initial conditions. In both tasks we sample from the same initial state prior distribution as the training dataset and restrict ourselves to initial tumors with a diameter between 2 and 5cm.

G.2 Covid-19

We use the COVID-19 disease progression simulator under dexamethasone treatment proposed by Qian et al. [2021] (adoption of Dai et al. [2021]). The simulator describes the interaction between disease progression and immune response, coupled with a treatment/exposure component.

Dynamics. The original simulator in Qian et al. [2021] is specified as a deterministic ODE system. To model stochasticity and unobserved heterogeneity, we add multiplicative diffusion terms to the dynamics, yielding the following SDE formulation with state $X_t = (X_{1,t}, X_{2,t}, X_{3,t}, X_{4,t})$. The state variables $X_{1,t}, X_{2,t}, X_{3,t}, X_{4,t}$ represent viral load, innate immune response to viral infection, adaptive immunity, and (effective) dexamethasone concentration in lung tissue, respectively:

$$\begin{aligned}
dX_{1,t} &= \left(k_{dp} X_{1,t} - k_{di} X_{1,t} X_{3,t}^{h_c} - k_{dr} X_{1,t} X_{2,t} \right) dt + \sigma_1 X_{1,t} dW_t^1, \\
dX_{2,t} &= \left(k_{id} X_{1,t} - k_{io} X_{2,t} + k_{if} X_{1,t} X_{2,t} + \frac{k_{ep} X_{2,t}^{h_p}}{k_{cp}^{h_p} + X_{2,t}^{h_p}} - k_d X_{4,t} X_{2,t} \right) dt + \sigma_2 X_{2,t} dW_t^2, \\
dX_{3,t} &= \left(k_{im} X_{2,t} \right) dt + \sigma_3 X_{3,t} dW_t^3, \\
dX_{4,t} &= \left(k_{kel} U_{c,t} - k_{kel} X_{4,t} \right) dt + \sigma_4 X_{4,t} dW_t^4.
\end{aligned}$$

Here, W_t^i denote independent standard Wiener processes and we set $\sigma = \sigma_i = 0.1$ for $i \in \{1, 2, 3, 4\}$. The dynamics parameters are positive real numbers and described in Qian et al. [2021], Dai et al. [2021]. Following Qian et al. [2021] we set the coefficients $h_p = h_c = 2$ and the rest to one.

The signal $U_{c,t}$ denotes our control variable and represents the exogenous dexamethasone administration/exposure input.

Treatment Process. We consider a single-intervention setting over a 14-day horizon. Dexamethasone is administered after 7 days at different dosages. Concretely, for each simulated trajectory, we sample a dosage $d \sim \mathrm{Unif}(0.5, 5)$ mg and define the control signal as an exponential decay

$$U_c(t) = d \cdot \mathbb{I}(t \geq t^\star) \exp\big(- k_{kel}(t - t^\star) \big).$$

Initial State. We sample initial conditions X_0 from an exponential distribution with rate $\lambda = 100$, as described in Qian et al. [2021].

Simulation and Preprocessing. In total we sample 100 trajectories over a 14-day horizon. State trajectories are min–max normalized to $[0, 1]$ using statistics computed on the training set. Treatment inputs are min–max normalized to $[0, 1]$ using the known dose bounds $d \in [0, 10]$. Validation trajectories consist of 128 trajectories sampled in the same way.

Treatment Optimization Objective (Target Trajectory). Let $X_t \in \mathbb{R}^{d_x}$ denote the COVID-19 simulator state and let $u_t \in \mathbb{R}^{d_u}$ denote the (one-dimensional) dexamethasone exposure control. Given a randomly sampled initial condition $X_{t_0} = x_0$, we define a target trajectory $X_t^\star$ over $[t_0, t_f]$ by sampling $M = 100$ trajectories $\{X^{(m)}\}_{m=1}^M$ under a sampled control curve. Specifically, we sample a random timepoint in $[t_0, t_f]$ and a random dosage in $[5, 10]$ and define the control using a similar exponential decay relationship. The target trajectory is then obtained by averaging pointwise in time $X_t^\star := \frac{1}{M}\sum_{m=1}^M X_t^{(m)}, \qquad t \in [t_0, t_f]$. We then optimize treatment to track $X^\star$:

$$\min_{u \in \mathcal{U}} \mathbb{E}\left[\int_{t_0}^{t_f} \lambda_x \left\|X_t - X_t^\star\right\|_2^2 \mathrm{d}t \;\middle|\; X_{t_0} = x_0\right]. \tag{17}$$

We choose a factor of $\lambda_x = 1$ so that the regularization strength of our regularizer can live on the same scale as for the Cancer task. We sample 15 initial conditions from the same prior distribution as the training dataset and then 15 treatment trajectories resulting in 15 optimization tasks.

G.3 MIMIC-IV

MIMIC-IV [Johnson et al., 2024] is a publicly available critical care database containing EHR data from ICU admissions at Beth Israel Deaconess Medical Center. We build on the preprocessing pipeline of Sivaraman et al. [2023], which focuses on sepsis patients using Sepsis-3 criteria and aggregates clinical variables at 4-hour intervals. Specifically, we run the provided preprocessing pipeline provided in `https://github.com/cmudig/AI-Clinician-MIMICIV`.

Further Preprocessing. We further restrict to the septic shock subgroup by filtering patients according to two additional criteria: (1) vasopressor administration at any point during the ICU stay and (2) a minimum trajectory length of 24 hours (6 timesteps), resulting in $10,826$ patients. The dataset is then randomly split 80/20 at the patient level into training ($8,660$ patients) and test ($2,166$ patients) sets. Then, we select 14 relevant features as state variables: hemodynamic indicators (MAP, systolic blood pressure, heart rate), tissue perfusion markers (arterial lactate, cumulative fluid balance, urine output), SIRS criteria (temperature, respiratory rate, white blood cell count), and SOFA organ dysfunction components (creatinine, platelet count, PaO2_2, GCS), and the previous vasopressor dose is appended to the state. Together these variables capture the key physiological dimensions used to assess severity and guide treatment decisions in septic shock. The two treatment variables (IV fluid volume and vasopressor dose) are log1p-transformed before modeling. All state features are z-standardized using training set statistics only.

Treatment Optimization Objective (Sepsis-3 recovery). Let $X_t \in \mathbb{R}^{d_x}$ denote the patient's physiological state and let $u_t \in \mathbb{R}^{d_u}$ denote the treatment control (IV fluid volume and vasopressor dose). Given an initial condition $X_{t_0} = x_0$, we optimize treatment to restore the two Sepsis-3 recovery criteria [Singer et al., 2016] (MAP ≥ 65 mmHg and lactate ≤ 2 mmol/L) while minimizing treatment intensity:

$$\min_{u \in \mathcal{U}} \mathbb{E}\left[\int_{t_0}^{t_f} \lambda_{\text{MAP}} \,\text{ReLU}(\bar{m} - X_t^{\text{MAP}})^2 + \lambda_{\text{lac}} \,\text{ReLU}(X_t^{\text{lac}} - \bar{l})^2 + \lambda_u \left\|u_t\right\|_2^2 \mathrm{d}t \;\middle|\; X_{t_0} = x_0\right], \tag{18}$$

where $\bar{m} = 65$ mmHg and $\bar{l} = 2$ mmol/L are the Sepsis-3 clinical thresholds (converted to z-scored units), and $\lambda_{\text{MAP}} = \lambda_{\text{lac}} = 1$ and $\lambda_u = 0.01$ are weighting coefficients.

H Implementation Details of `SigCTO`

Neural SDEs as Structural Dynamics Model

We model the drift and diffusion of the SDE with neural networks $\mu_\theta : \mathbb{R}^{d_x} \times A \to \mathbb{R}^{d_x}$, $\sigma_\theta : \mathbb{R}^{d_x} \times A \to \mathbb{R}^{d_x \times d_w}$ to model the strong solution $\hat{X} : I \to \mathbb{R}^{d_x}$ of

$$\mathrm{d}\hat{X}_t = \mu_\theta(\hat{X}_t, U_t)\mathrm{d}t + \sigma_\theta(\hat{X}_t, U_t)\mathrm{d}W_t \tag{19}$$

$W : I \to \mathbb{R}^{d_w}$ is a d_w-dimensional Brownian motion and $U = (U_t)_{t \in I}$ our control.

In all experiments, we parameterized the drift and diffusion so that each state has its own independent neural network function, but still takes all state variables as inputs. This allows for independently learned functions for each state. Each drift neural network consists of a three-layer fully-connected neural network with 64 hidden units, while the diffusion networks consist of a one-layer fully-connected neural network with 8 hidden units. We use LipSwish-activation function for intermediate layers and tanh-activation for the final layers.

Neural SDE Training Parameters. We optimize the neural SDE using a signature-kernel score objective (see below, based on Issa et al. [2023]). For training we use the RMSProp optimizer with a learning rate of 0.0001, 0.0005 and 0.0005 for the Cancer, Covid and MIMIC-IV dataset, respectively. We optimize over 15000 gradient steps, where each gradient step has a batch size of 16 trajectories for Cancer and 32 trajectories for Covid and MIMIC-IV.

Signature Score Loss Function. Here, we detail how we adapt the signature score training method by Issa et al. [2023] to our conditional training as we have samples from the treatment process and the state process. Issa et al. [2023] propose a non-adversarial training objective for neural SDEs based on the signature kernel: The signature kernel score of $P \in \mathcal{M}^1(\mathcal{X}, \mathcal{B}(\mathcal{T}_\mathcal{X}))$ and $y \in \mathcal{X}$ is the map $\Phi_{\text{sig}} \colon \mathcal{M}^1(\mathcal{X}, \mathcal{B}(\mathcal{T}_\mathcal{X})) \times \mathcal{X} \longrightarrow \mathbb{R}$ defined by

$$\Phi_{\text{sig}}(P, y) := \mathbb{E}_{x,x' \sim P}\left[k_{\text{sig}}(x, x')\right] - 2\,\mathbb{E}_{x \sim P}\left[k_{\text{sig}}(x, y)\right].$$

Φ_{sig} is a strictly proper score, meaning it assigns the lowest expected score when the proposed prediction is realized by the true probability distribution. Let $(\Omega, \mathcal{F}, P)$ be a probability space, $T > 0$, $d_x \in \mathbb{N}$. Define path-space $\mathcal{X} := \left\{\gamma \in C_{\text{BV}}([0,T], \mathbb{R}^{d_x}) \;\middle|\; \exists i \in [d] : \gamma^i \text{ monotone}\right\}$. Let $\mathcal{T}$ be a topology such that the signature kernel $k_{\text{sig}} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is continuous. Let $X^{\text{true}} \colon \Omega \to \mathcal{X}$ be a random variable with sample $\{x^{(i)}\}_{i=1}^N \overset{iid}{\sim} P_{X^{\text{true}}} \in \mathcal{M}^1(\mathcal{X}, \mathcal{B}(\mathcal{T}_\mathcal{X}))$.

In the unconditional setting (as described in Issa et al. [2023]), one can learn a neural SDE X^θ into $\mathcal{X}$ such that its law $P_{X^\theta} \approx P_{X^{\text{true}}}$, use the training objective $\min_\theta \mathcal{L}(\theta)$, $\mathcal{L}(\theta) := \mathbb{E}_{y \sim P_{X^{\text{true}}}}\left[\Phi_{\text{sig}}(P_{X^\theta}, y)\right]$, which is equivalent to optimization using $\text{MMD}^2_{k_{\text{sig}}}[P_{X^\theta}, P_{X^{\text{true}}}]$ as the last term $\mathbb{E}_{y,y' \sim P_{X^{\text{true}}}}[k_{\text{sig}}(y, y')]$ is constant w.r.t. θ.

However, our neural SDE is a conditional setting, since we observe both a treatment process and a state process. Given a distribution P_Z to condition on, $P_{X^{\text{true}}|Z}(\cdot \mid z)$, $z \sim P_Z$ the target conditional distribution with sample $\{(z^{(i)}, x^{(i)})\}_{i \in [N]}$, $z^{(i)} \sim P_Z$, $x^{(i)} \sim P_{X^{\text{true}}|Z}(\cdot \mid z^{(i)})$. To learn a neural SDE X^θ into $\mathcal{X}$ such that $P_{X^\theta}(\cdot \mid z) \approx P_{X^{\text{true}}}(\cdot \mid z)$, train with the modified objective:

$$\min_\theta \mathcal{L}'(\theta), \quad \text{where} \quad \mathcal{L}'(\theta) := \mathbb{E}_{z \sim P_Z}\left[\mathbb{E}_{x \sim P_{X^{\text{true}}}(\cdot|z)}\left[\Phi_{\text{sig}}(P_{X^\theta}(\cdot \mid z), x)\right]\right].$$

Thus, in our controlled SDE setting, we have to adapt the above conditional objective in the following way. Given a control signal u to condition on, the target conditional distribution $P_{X^{\text{true}}|U}(\cdot \mid u)$ with sample $\{(u^{(i)}, x^{(i)})\}_{i \in [N]}$, $u^{(i)} \sim P_U$, $x^{(i)} \sim P_{X^{\text{true}}|U}(\cdot \mid u^{(i)})$. To learn a neural SDE X^θ into $\mathcal{X}$ such that $P_{X^\theta}(\cdot \mid u, x_0) \approx P_{X^{\text{true}}}(\cdot \mid u, x_0)$, train with the modified objective:

$$\min_\theta \mathcal{L}'(\theta), \quad \text{where} \quad \mathcal{L}'(\theta) := \mathbb{E}_{u \sim P_U, x_0 \sim P_{X_0}}\left[\mathbb{E}_{x \sim P_{X^{\text{true}}}(\cdot|u,x_0)}\left[\Phi_{\text{sig}}(P_{X^\theta}(\cdot \mid u, x_0), x)\right]\right].$$

In all of our experiments, we use an RBF-lifting kernel with a smoothing parameter $\sigma = 1$ to compute the above loss. We set the order of dyadic refinement associated to the PDE solver for computing the signature kernel to 1. Following Issa et al. [2023], we apply a time-normalization transform and augment true and predicted paths with the normalized-time component before passing them through the loss function.

`SigCTO` regularizer

We compute the Signature-Kernel MCMD regularizer using the held-out validation dataset. Specifically, the validation sample are $\{x^{(i)}, u^{(i)}, x_0^{(i)}\}_{i \in [N]}$ and we use our trained neural SDE to obtain $\{\hat{x}^{(i)}, u^{(i)}, x_0^{(i)}\}_{i \in [N]}$ using the same initial states and controls. We use a linear lifting kernel with smoothing parameter $\sigma = 1$ for the trajectories. Again, we use a dyadic refinement factor 1 for the PDE solver and apply a time-normalization transform and augment true and predicted paths with the normalized-time component before evaluating the path. For the initial states, we use an RBF-kernel with smoothing parameter $\sigma = 1$.

Libraries

For all signature-kernel implementations (for the regularizer) or for training the neural SDE, we build on the following JAX implementation: `https://github.com/crispitagorico/sigkerax`.

Treatment Optimization

We solve the treatment optimization problem using a direct single-shooting approach. For details on numeric approaches to solving continuous-time optimal control problems, we refer to Gros and Diehl [2020].

Treatment Parameterization

We parameterize the control function $u(t)$ (denoted by $u(t, \phi)$) with the parameter ϕ. Specifically, for the Cancer tasks where we have a two-dimensional control consisting of the chemo- and radiotherapy treatment, we parameterize the chemo parameters $\phi_c = [\phi_{c,timepoints}, \phi_{c,dosage}]$, with $\phi_{c,timepoints}$ denoting the timepoints of treatment administration, and $\phi_{c,dosage}$ the respective dosages. Similarly we parameterize the radiotherapy parameters ϕ_r. We restrict the controls to a maximum of K treatment administrations for chemo- and radiotherapy, respectively. Now, to obtain a control function from our parameters, we model the control similarly to the data generating process and model the control function $u_{chemo}(t)$ by modeling a bang-bang control and setting $u_{chemo}(t) = \sum_{i=1}^{K} \mathbb{I}_{(t \in [\phi_{c,timepoints,i}, \phi_{c,timepoints,i+1}])} \cdot \phi_{c,dosage;i}$, where $\mathbb{I}$ is the indicator function. The function $u_{radio}(t)$ is constructed similarly.

For the COVID-19 task, we have a one-dimensional control corresponding to the Dexamethasone administration. As parameters, we directly parameterize the time series of Dexamethasone administration with 42 timesteps, and the continuous control function is given by the linear interpolation between the 42 parameters.

Treatment Optimization

We use a parameterized representation of the control functions (as described above). This allows for direct single-shooting control optimization. For cancer, we initialize the timepoint parameters $\phi_{timepoints}$ by uniformly sampling over the time interval I. The dosage parameters ϕ_{dosage} are initialized by uniformly sampling between $[0.1, 0.3]$. For Covid-19, we initialize the timeseries parameters by uniformly sampling between $[0.0, 1.0]$. We use gradient-descent-based methods (RMSProp optimizer) with a learning rate of 0.001 and 5000 gradient-update steps. Moreover, we use Monte Carlo sampling with a sample size $= 100$ to approximate the expectation in our objective function.

I Baseline Implementation

I.1 Control Library Protocol

The library of candidate controls consists of 100 randomly sampled control candidates. Importantly, every method evaluates the exact same set of candidate controls. We randomly sample the parameters ϕ, which parameterize the continuous-time control function. For the respective problems, we use the parameterization described in Appendix H. For cancer, we initialize the treatment timepoints $\phi_{timepoints}$ by uniformly sampling over the time interval of interest $[t_0, t_f]$. The dosages are obtained by randomly sampling over the interval $[0.1, 0.3]$ (in the min-max-normalized control space). For the Cancer tasks, we choose a maximum of $K = 5$ treatment administrations. For Covid-19, we randomly sample the control for each of the 42 timepoints in the time-interval randomly from $[0, 1]$ (in the min-max-normalized control space). Each method, including `SigCTO`†, predicts the patient outcome for each candidate control and selects the control with the lowest predicted costs as optimal.

I.2 Evaluation Metrics

Control-Library Metrics. For each method, we report the Spearman rank correlation between the predicted costs and the true costs. Let $\mathcal{J}$ be the cost function (see Appendix G for our choices). Moreover, let $\{u_i\}_{i=1}M$ be the control library. Each method predicts the costs for the control candidates, $\{\hat{\mathcal{J}}_i\}_{i=1}^{M}$, and we can use the simulators to obtain the true costs for the candidates $\{\mathcal{J}_i\}_{i=1}^{M}$. The Spearman rank correlation is then computed between $\{\hat{\mathcal{J}}_i\}_{i=1}^{M}$ and $\{\mathcal{J}_i\}_{i=1}^{M}$. Thus, higher values indicate that a method keeps the same ordering among candidate controls. Further, we report the ground-truth costs directly, i.e., $\mathcal{J}$, for the control from the control library that a method predicts as optimal.

Ground-Truth Costs. For an optimized treatment plan u, the ground-truth costs are $\mathcal{J}(t_0, X^{t_0,x_0,u}, u)$, where the expectations of X are computed using MC sampling using the true, underlying SDE. Moreover, we have $\sqrt{\text{Ground-Truth Cost}} = \sqrt{\mathcal{J}(t_0, X^{t_0,x_0,u}, u)}$.

Relative Ground-Truth Costs. The relative ground-truth costs aim at evaluating how well a method performs compared to naive optimization. Specifically, let u_{naive} be the treatment plan obtained via naive optimization ($\lambda = 0$) with corresponding ground truth-costs $\mathcal{J}(t_0, X^{t_0,x_0,u_{\text{naive}}}, u_{\text{naive})}$ and let u_{method} be the treatment plan obtained by a method (e.g., choosing $\lambda > 0$ in `SigCTO`) with corresponding ground-truth costs $\mathcal{J}(t_0, X^{t_0,x_0,u_{\text{method}}}, u_{\text{method})}$. The relative ground-truth costs are defined as:

$$\frac{\mathcal{J}(t_0, X^{t_0,x_0,u_{\text{method}}}, u_{\text{method}})}{\mathcal{J}(t_0, X^{t_0,x_0,u_{\text{naive}}}, u_{\text{naive}})}.$$

Thus, if the relative ground-truth costs are 1 both treatments perform equally. Values < 1 suggest improvement by u_{method} and values > 1 suggest worse performance of u_{method} compared to naively optimized treatment plans. We evaluate the expectations in the ground-truth costs using a monte carlo sample size of 200.

I.3 Baselines

TE-CDE Seedat et al. [2022]. In our experiments, we use the same model architecture as proposed in the original paper, meaning a 2-layer neural network with hidden state dimension 128 for both the neural CDE encoder and decoder. We train the decoder already to predict the same time horizon as needed in our control tasks. We also adjust the balancing-representation loss objective from a cross-entropy loss to a mean-squared-error loss as it naturally extends to our general formulation of continuous-valued treatments. We perform a grid-search for selecting the learning rate $\in \{0.0001, 0.001, 0.01\}$, number of epochs $\in \{50, 100\}$ and dimensions of the latent state $\in \{8, 16\}$. The optimal hyperparameters are determined by choosing the set that results in the lowest mean squared error on the validation dataset, for the Cancer and Covid datasets, respectively. For TE-CDE we build on the implementation provided in `https://github.com/konstantinhess/Bayesian-Neural-CDE.git`.

INSITE/SINDy Kacprzyk et al. [2024]. INSITE is a framework for bridging treatment effect estimation and dynamical system discovery. Following Kacprzyk et al. [2024], we also instantiate the INSITE framework using SINDy [Brunton et al., 2016]. Importantly, since we do not have patient heterogeneity in our experiments (so where each patient has their "own" dynamics parameters) and instead model patient-conditioned effects by conditioning on the observed history, we only obtain the population-level dynamics equation and not further fine-tune it on a specific trajectory. Moreover following Kacprzyk et al. [2024], we incorporate the continuous-valued treatment directly as a control input and use SINDy with control. In order to establish a fair comparison with the SINDy baseline, similar to Kacprzyk et al. [2024], we used the STLSQ optimizer together with the polynomial library and performed hyperparameter-tuning over the polynomial degree $\deg \in \{1, 2, 3, 4\}$, threshold $\{0.1, 0.2, 0.5\}$ and the ridge regularization parameter $\lambda \in \{0.1, 0.2, 0.5, 1, 2, 3\}$. We implement INSITE based on the original repository.

CT-MOPO Among model-based offline RL baselines, which are built for discrete-time dynamics and closed-loop policies, making them poorly suited to irregular continuous-time open-loop treatment optimization, MOPO ([Yu et al., 2020]) is the most natural candidate for a reasonable continuous-time adaptation. To obtain the strongest possible MOPO-inspired baseline, which is as close as possible to the original pessimistic design, we use the same neural-SDE backbone and gradient-based open-loop optimization pipeline as our method. More precisely we

(i) replace the closed-loop policy by a trainable open loop treatment path $u_\phi : [t_0, t_f] \to \mathbb{R}^{d_u}$, trained via gradient descent

(ii) replace the probabilistic ensemble of discrete-time dynamics models by an ensemble of $M = 5$ independently initialized and trained neural SDEs with drift and diffusion functions $(\mu_{\theta_m}, \sigma_{\theta_m})$

and after each ensemble member is rolled out with control u_ϕ to give state trajectory $\widehat{X}^{m,u_\phi}$ we

(iii) compute the covariance at each time t for each member m

$$\Sigma_m(t, u_\phi) = \sigma_{\theta_m}(\widehat{X}_t^{m,u_\phi}, u_\phi(t))\sigma_{\theta_m}(\widehat{X}_t^{m,u_\phi}, u_\phi(t))^\top.$$

(iv) calculate the conservative uncertainty signal across ensemble members (for diagonal diffusions) via

$$r(t, u_\phi) = \max_{m=1,\dots,M} \sqrt{\text{tr}(\Sigma_m(t, u_\phi))}.$$

Note that in the original MOPO implementation, a diagonal Gaussian output model is used.

The resulting open-loop MOPO objective is

$$\min_\phi \ \widehat{\mathcal{J}}(u_\phi) + \lambda \int_{t_0}^{t_f} r(t, u_\phi)\, \mathrm{d}t,$$

where $\widehat{\mathcal{J}}(u_\phi)$ denotes the cost evaluated under the learned neural-SDE rollout.

J Experiments Compute Resources

All experiments were conducted on a high-performance computing (HPC) cluster equipped with NVIDIA A100 and V100 GPUs and Intel Xeon Gold 6142 CPUs. Each model training run, including those for our method and the baselines, was run with a wall-clock time limit of 4 hours on a single GPU. Each optimization-task initialization was also run with a wall-clock time limit of 4 hours.

K Additional Experimental Results

K.1 Additional Evaluation Metrics for `SigCTO`

In Figure 2 **A**, we report the squared signature-kernel MMD between the true and predicted trajectories under optimized treatment plans, relative to naive optimization. For increasing regularization strengths, the relative MMD decreases, suggesting that the predicted and true trajectories are more aligned than in the naive optimization. In Figure 2 **B**, we show the value of our `SigCTO` regularizer relative to the value obtained using naive optimization. The ratio is consistently below 1.

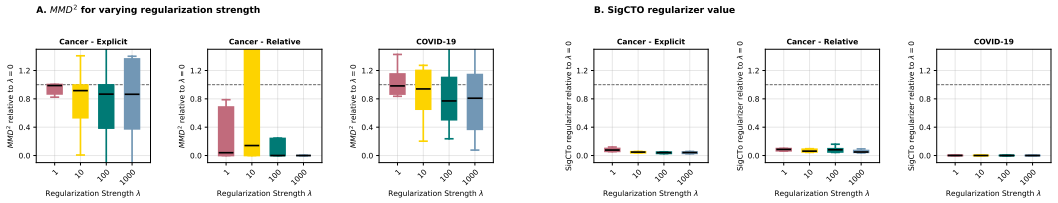

Figure 2: **A:** $\mathrm{MMD}^2_{k_{\mathrm{sig}}}$ between the true and predicted patient trajectory distributions for the optimal treatment plan obtained with different levels of regularization strength λ. **B:** $\mathrm{MCMD}_{k_{\mathrm{sig}}}$ estimator for the optimal control obtained with different levels of regularization strength λ. Results are reported using the rbf kernel and $K = 1$. Boxplots summarize results over 15 task initializations; boxes show medians and interquartile ranges (IQR), and whiskers extend to the farthest point within $1.5 \times$ IQR.

K.2 Ablations

Figure 3 shows an ablation in which we replace the linear kernel in the `SigCTO` signature kernel MMD regularizer with an RBF kernel. As with the linear kernel, increasing the regularization strength λ generally reduces relative ground-truth cost. However, increasing λ from 100 to 1000 raises the relative ground-truth cost on both cancer tasks. This suggests that excessive regularization may dominate the treatment-cost term during optimization and limit further improvements in the treatment optimization task. Increasing the number of time intervals K results in greater improvements with the RBF kernel than with the linear kernel on both cancer tasks, whereas there is no consistent improvement for the Covid-19 task.

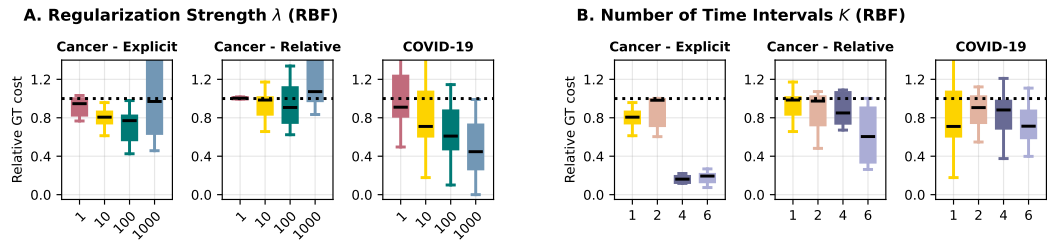

Figure 3: `SigCTO` **on PK/PD simulation tasks using an RBF kernel in the signature kernel MMD regularizer. A:** & **B.** Relative ground-truth cost for varying regularization strength λ and number of time intervals K, respectively. Values below 1 indicate improvement over naive optimization. Boxplots summarize results over 15 task initializations, boxes show medians and interquartile ranges (IQR), and whiskers extend to the farthest point within $1.5 \times$ IQR.

Table 4: Sensitivity results on the Cancer-Explicit task. We evaluate SigCTO using a Linear kernel with $\lambda = 100$ and $K = 1$.All values are reported as mean ± standard deviation over 15 task initializations.

(a) Sensitivity to Structural Variations

	SDE	**ODE**	**Non-Markovian SDE**
$\sqrt{\text{Ground Truth Cost}}(\downarrow)$	9.91 ± 13.66	1.45 ± 2.43	2.81 ± 2.21
Relative ground-truth costs ($\downarrow$)	0.15 ± 0.20	0.06 ± 0.13	10.11 ± 9.00

(b) Overlap violations/ Confounded Treatment

	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$
$\sqrt{\text{Ground Truth Cost}}(\downarrow)$	0.57 ± 0.01	5.15 ± 4.95	4.57 ± 7.07
Relative ground-truth costs ($\downarrow$)	0.03 ± 0.03	0.42 ± 0.46	1.84 ± 3.96

(c) Confounded Noise

	$\sigma = 0$	$\sigma = 0.1$	$\sigma = 0.4$
$\sqrt{\text{GT-Cost}}$	9.91 ± 13.66	6.91 ± 16.40	26.18 ± 25.93
Relative ground-truth costs ($\downarrow$)	0.15 ± 0.20	0.42 ± 1.24	0.40 ± 0.42

K.3 Sensitivity to violations of structural model assumption

In Table 4, we evaluate the sensitivity of SigCTO on the Cancer-Explicit task to variations in the data-generating dynamics and violations of structural model assumptions. Specifically, Table 4a considers two modifications to the cancer simulator (Appendix G): (1) setting the diffusion coefficients to zero to obtain an ODE model and (2) withholding the chemotherapy concentration from the observed state, resulting in partially observed, non-Markovian dynamics. The regularizer remains effective in the ODE setting but substantially increases relative ground-truth costs in the partially observed setting. A potential way to address this limitation is to augment the state with past observations and treatments to better capture history dependence.

We also assess sensitivity to time-dependent confounding in treatment assignment by increasing its strength γ (Table 4b), using a setup similar to the study of overlap violations in Kacprzyk et al. [2024]. Following the setup of Kacprzyk et al. [2024], we use a different treatment assignment process for this experiment. Consequently, the results at $\gamma = 0$ are not directly comparable to those in Table 4a. Nevertheless, SigCTO remains effective in this setting, reducing the mean ground-truth cost relative to naive optimization. Compared with the unconfounded setting, both confounded settings yield higher ground-truth costs, suggesting that treatment optimization becomes more challenging. Nevertheless, the mean relative ground-truth costs remain well below 1 at $\gamma = 0$ and $\gamma = 1$, demonstrating a continued benefit from our SigCTO regularization to prevent model exploitation. At $\gamma = 2$, the mean relative cost rises to 1.84 ± 3.96, while the median remains low at 0.25. Thus, regularization still improves performance in most task initializations under strong confounding, although substantial cost increases in some cases outweigh these gains on average.

Finally, we assess sensitivity to correlated process noise by introducing nonzero off-diagonal entries in the simulator's diffusion matrix. At the strongest tested noise coupling, $\sigma = 0.4$, the ground-truth costs are higher than in the baseline setting, suggesting increased difficulty of the treatment optimization task. Despite this increase, the mean relative ground-truth costs remain well below 1 across all tested noise levels: 0.15, 0.42, and 0.40 for $\sigma = 0$, 0.1, and 0.4, respectively. These results show that the regularizer retains a substantial average benefit over naive optimization even under correlated process noise, consistent with its role in limiting model exploitation.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims in the abstract and introduction highlight the contributions of our paper. Specifically, we highlight problem setting, theoretical contribution, and empirical results.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In Section 4 we highlight the assumptions of our work, which also correspond to the limitations of our theoretical results. Using simulators we empirically evaluate relaxing the underlying assumptions Section 4.1 and Appendix K and the real-world dataset MIMIC-IV may also not strictly satisfy these assumptions. In Section 7, we further highlight limitations of our work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We list full assumptions for all theoretical results. In Appendices B, C and E we provide additional formal regularity assumptions, definitions, and proofs for all theoretical results.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have fully disclosed all the information needed to reproduce the main experimental results of the paper in Section 5 and Appendices G to I. Moreover, the Python implementation is included in the supplementary materials.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: The datasets are either publicly available or synthetic and can thus be generated as detailed in Appendix G. The code to reproduce our results is also linked in Section 5.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: In Section 5 we describe the general experimental setup at a high-level. In Appendices G to I, we further detail data splits, model architectures, and training hyperparameters. Moreover, we provide a Python implementation in the supplementary materials for documentary purposes.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: In Figure 1, boxplots summarize variability across 15 treatment optimization tasks with different initial conditions, showing medians and interquartile ranges, with whiskers extending to the most extreme observations within 1.5 times the interquartile range. Furthermore, the benchmarking results in Tables 1a and 1b include standard deviations over 15 initialized treatment optimization tasks and in Table 2 we report p-values to show statistical significance.

   Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: We provide the information on the compute resources in Appendix J.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: Our research adheres fully to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

Answer: [Yes]

Justification: The paper discusses positive impacts through more robust treatment optimization from observational clinical data, as well as risks from biased data, unmeasured confounding, and unsupported clinical deployment in Section 7. In Section 7 we also state that real-world usability would require prospective validation, safety constraints, and human oversight.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: This paper does not include any data or models with a high risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We credit all reused code and datasets with explicit citations.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: With the released codebase, we will provide accompanying documentation.

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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