PATH-INTEGRAL VALUE-CONTROL MATCHING WITH APPLICATIONS TO SMOOTHING AND GAME

Anonymous authors
Paper under double-blind review

ABSTRACT

Stochastic optimal control (SOC) provides a computational framework for a broad range of machine-learning problems. Policy-based solvers learn feedback controls directly, but often require costly full-horizon rollouts and may suffer from unstable likelihood-ratio weights during off-policy training. Local path-integral value learning enables short-trajectory updates and experience replay, but its scalar objective does not directly supervise the value gradient that determines the control. This limitation is particularly relevant to control-sensitive applications such as Bayesian smoothing and mean-field games. Building on Path-Integral Value Matching (PI-VM), we propose Path-Integral Value-Control Matching (PI-VCM), which adds direct control supervision while retaining local value learning and replay. We spatially differentiate the local path-integral value recursion to derive a corresponding control recursion, allowing the same short trajectories to provide both value and control targets for a shared value network. Affine sensitivity caching and local likelihood correction further enable off-policy experience replay without repeated adjoint propagation. Experiments demonstrate improved control accuracy over baselines on SOC benchmarks and higher-dimensional instances, together with competitive performance on Bayesian smoothing and mean-field games.

1 INTRODUCTION

Stochastic optimal control (SOC) studies how to steer stochastic dynamical systems through feedback controls to minimize expected costs (Domingo-Enrich, 2024). Beyond classical control, SOC has become a useful computational framework for a range of machine-learning problems, including generative fine-tuning (Domingo-Enrich et al., 2025), energy-based sampling (Liu et al., 2026), transition-path sampling (Holdijk et al., 2023), and rare-event simulation (Du et al., 2026).

Learning-based SOC solvers broadly follow policy-based and value-based approaches. Policy-based methods optimize controls directly (Nüsken & Richter, 2021), with matching-based methods providing explicit control targets (Domingo i Enrich et al., 2024; Domingo-Enrich et al., 2025). However, many require costly full-horizon rollouts, and off-policy reuse can suffer from unstable pathwise likelihood ratios. Value-based methods instead learn the optimal cost-to-go and recover controls from its spatial gradient (Liao et al., 2026; Jensen et al., 2026; Du et al., 2026). Path-integral value learning further enables short-trajectory updates and experience replay (Liao et al., 2026).

However, a scalar value objective does not directly supervise the gradient used to compute the control, so accurate value prediction need not imply accurate feedback. This distinction matters in Bayesian smoothing, where feedback shapes posterior path dynamics, and in mean-field games, where best-response controls drive population updates. These applications motivate direct control supervision alongside local value learning, rather than relying on value accuracy alone.

Building on Path-Integral Value Matching (PI-VM) (Liao et al., 2026), we propose Path-Integral Value-Control Matching (PI-VCM), which adds direct control supervision while retaining local value learning and replay. We spatially differentiate the same local path-integral recursion to obtain a corresponding control recursion. The resulting control targets combine endpoint value gradients with running-cost sensitivities transported along each path. The same short trajectories thus provide both value and control targets for one value network, jointly supervising its values and the gradient-derived controls used at deployment. Affine sensitivity caching and local likelihood correction allow us to refresh targets from off-policy experience without repeated adjoint propagation. Experiments show

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

improved control accuracy over PI-VM across SOC benchmarks and increasing state dimensions, together with competitive performance on Bayesian smoothing and mean-field games.

Our main contributions are summarized as follows:

- We propose PI-VCM, deriving control targets from the local value recursion to jointly supervise a single value network.
- We develop affine sensitivity caching with local likelihood correction, enabling value-control replay without repeated adjoint propagation.
- We demonstrate improved control accuracy over PI-VM on SOC benchmarks and evaluate PI-VCM on Bayesian smoothing and mean-field games.

2 Related Work

Learning-based stochastic optimal control. Policy-based solvers optimize feedback controls through path-space objectives (Nüsken & Richter, 2021; Holdijk et al., 2023) or supervised matching objectives (Domingo i Enrich et al., 2024; Domingo-Enrich et al., 2025). Adjoint Matching (AM), in particular, uses backward cost sensitivities to construct control labels along full-horizon trajectories. Value-based solvers instead approximate the cost-to-go and derive a feedback policy from its spatial gradient. In the control–noise matched setting, path-integral control represents the value through an exponential path expectation (Kappen, 2005; Todorov, 2006). PI-VM exploits the local recursion of this expectation to support short rollouts and experience replay (Liao et al., 2026). Value Matching (VM) learns a lightweight value function for reward-guided flow adaptation without backpropagating through the base generator (Jensen et al., 2026). Our starting point is PI-VM's local SOC recursion. We differentiate it to construct a control target and train one value network against both targets, using affine sensitivity caching to refresh them during replay.

Smoothing. Gaussian smoothers use linearization or sigma-point approximations (Särkkä, 2008; Yaghoobi et al., 2025), while particle methods approximate non-Gaussian posteriors (Godsill et al., 2004; Lindsten et al., 2014). Control-based inference optimizes path measures or observation-informed proposals (Archambeau et al., 2007; Kim & Mehta, 2020; Ruiz & Kappen, 2017; Heng et al., 2020), with related constructions in backward filtering and guided smoothing (van der Meulen & Sommer, 2025; Eklund et al., 2026). We build on this established smoothing–control connection, applying joint value-control learning to the resulting SOC problem.

Mean-field games. Mean-field games couple individual optimality with population consistency (Lasry & Lions, 2007; Huang et al., 2007). Numerical methods include HJB–Fokker–Planck discretizations (Achdou & Capuzzo-Dolcetta, 2010), neural PDE and adversarial solvers (Lin et al., 2021; Assouli & Missaoui, 2023; 2024), and learned best responses with population updates (Magnino et al., 2025). For fixed population flows, quadratic control costs and matched control and noise channels admit a path-integral representation, related to the Cole–Hopf formulation of linearly solvable games (Guéant, 2012). Within a best-response framework, we use PI-VCM as the inner SOC solver, coupled with relaxed population updates.

3 Background

We introduce the stochastic optimal control problem and its path-integral representation, then review local path-integral value matching. These formulations establish the notation for the joint value-control method in Section 4.

3.1 Stochastic Optimal Control

We consider the following finite-horizon stochastic optimal control (SOC) problem:

$$\begin{aligned} \mathrm{d}X_t &= \big[b_t(X_t) + \sigma_t u_t(X_t)\big]\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t, \qquad X_s = x, \\ V(s,x) &= \inf_u \mathbb{E}^u_{s,x}\left[\int_s^T \left(f_t(X_t) + \frac{\lambda}{2}\|u_t(X_t)\|^2\right)\mathrm{d}t + g(X_T)\right], \end{aligned} \tag{1}$$

Here $X_t \in \mathbb{R}^d$, and W is a d_u-dimensional standard Brownian motion. For each $t \in [0, T]$, $b_t : \mathbb{R}^d \to \mathbb{R}^d$ is the drift, $u_t : \mathbb{R}^d \to \mathbb{R}^{d_u}$ is the control, and $\sigma_t \in \mathbb{R}^{d \times d_u}$ is the state-independent diffusion matrix. The functions $f_t, g : \mathbb{R}^d \to \mathbb{R}$ are the running and terminal costs, and $\lambda > 0$ scales the control penalty.

Minimizing the Hamiltonian in the Hamilton–Jacobi–Bellman (HJB) (Bellman, 1966) equation gives the optimal control

$$u^*(s, x) = -\lambda^{-1} \sigma_s^\top \nabla_x V(s, x). \tag{2}$$

An exponential transformation of the HJB equation followed by the Feynman–Kac formula yields the path-integral representation (Kappen, 2005; see also Todorov, 2006):

$$V(s, x) = -\lambda \log \mathbb{E}^{\text{base}}_{s,x} \left[\exp\left(-\frac{\mathcal{W}_{s:T} + g(X_T)}{\lambda} \right) \right], \tag{3}$$

where $\mathbb{P}^{\text{base}}$ denotes the reference law with $u = 0$, and $\mathcal{W}_{s:T} = \int_s^T f_t(X_t)\, \mathrm{d}t$ is the accumulated running cost. For an admissible sampling control $\bar{u}$, Girsanov's theorem gives the off-policy form

$$V(s, x) = -\lambda \log \mathbb{E}^{\bar{u}}_{s,x} \left[\Omega^{\bar{u}}_{s:T} \right], \tag{4}$$

where the full-path weight and likelihood correction are given by:

$$\Omega^{\bar{u}}_{s:T} = \exp\left(-\frac{\mathcal{W}_{s:T} + g(X_T)}{\lambda} + \Gamma^{\bar{u}}_{s:T} \right)$$

$$\Gamma^{\bar{u}}_{s:T} = \log \frac{\mathrm{d}\mathbb{P}^{\text{base}}}{\mathrm{d}\mathbb{P}^{\bar{u}}}(X_{s:T}) = -\int_s^T \bar{u}_t(X_t)^\top \, \mathrm{d}W^{\bar{u}}_t - \frac{1}{2} \int_s^T \|\bar{u}_t(X_t)\|^2 \, \mathrm{d}t.$$

Here $W^{\bar{u}}$ is a d_u-dimensional Brownian motion under the sampling law $\mathbb{P}^{\bar{u}}$. For a segment $[s, \tau]$, $\Gamma^{\bar{u}}_{s:\tau}$ denotes the same correction restricted to that interval. Appendix B.1 states the assumptions and derives the HJB equation and its exponential transformation.

3.2 Path-Integral Value Matching

For a segment $[s, \tau]$ with $s < \tau \leq T$ and an admissible sampling control $\bar{u}$, the path-integral representation in equation 4 gives the off-policy value recursion

$$V(s, x) = -\lambda \log \mathbb{E}^{\bar{u}}_{s,x} \left[\exp\left(-\frac{\mathcal{W}_{s:\tau} + V(\tau, X_\tau)}{\lambda} + \Gamma^{\bar{u}}_{s:\tau} \right) \right]. \tag{5}$$

Retaining the correction associated with the collection control allows historical segments to be reused with updated continuation values. Appendix B.3 derives the local recursion and its off-policy form.

Path-Integral Value Matching (PI-VM; Liao et al., 2026) learns a scalar value network V_θ from this recursion. Let $\bar{V}_\tau$ be a fixed estimate of the continuation value $V(\tau, \cdot)$. Given M trajectory branches sampled under $\bar{u}$ from the same starting pair (s, x), replacing $V(\tau, \cdot)$ by $\bar{V}_\tau$ and approximating the conditional expectation by a sample mean yields the target

$$\widehat{V}(s, x) = -\lambda \log \left[\frac{1}{M} \sum_{j=1}^{M} \exp\left(-\frac{\mathcal{W}^{(j)}_{s:\tau} + \bar{V}_\tau(X^{(j)}_\tau)}{\lambda} + \Gamma^{\bar{u},(j)}_{s:\tau} \right) \right]. \tag{6a}$$

Regressing the value network onto this fixed target gives the PI-VM loss

$$\mathcal{L}_{\text{PI-VM}}(\theta; s, x) = \left| V_\theta(s, x) - \text{sg}[\widehat{V}(s, x)] \right|^2. \tag{6b}$$

For segments ending at T, we set $\bar{V}_T = g$. The operator sg holds the target fixed. The value loss does not directly supervise the gradient used to compute the control.

4 Path-Integral Value-Control Matching

Path-Integral Value-Control Matching (PI-VCM) jointly supervises a value network and its induced control using short off-policy segments. We first derive the optimal control representation using a lean adjoint, then derive a local value-control recursion and its variance reduction property. Finally, we construct the joint loss and an affine sensitivity cache for replay. We use the notation and scaling of Section 3.

4.1 Optimal Control and Its Off-Policy Representation

Full-horizon optimal control. Along trajectories sampled under a frozen collection control $\bar{u}$, define the lean adjoint $\widetilde{a}_t \in \mathbb{R}^d$ by the backward equation

$$\begin{aligned} \frac{\mathrm{d}\widetilde{a}_t}{\mathrm{d}t} &= -(D_x b_t(X_t))^\top \widetilde{a}_t - \nabla_x f_t(X_t), \\ \widetilde{a}_T &= \nabla_x g(X_T). \end{aligned} \tag{7}$$

Here $D_x b_t(X_t) \in \mathbb{R}^{d\times d}$ is the Jacobian of the reference drift. Under the reference dynamics, it gives the pathwise gradient $\nabla_x(\mathcal{W}_{s:T} + g(X_T)) = \widetilde{a}_s$. Differentiating the path-integral value in equation 3 therefore yields

$$u^*(s,x) = -\lambda^{-1}\sigma_s^\top \frac{\mathbb{E}^{\mathrm{base}}_{s,x}\left[e^{-(\mathcal{W}_{s:T}+g(X_T))/\lambda}\widetilde{a}_s\right]}{\mathbb{E}^{\mathrm{base}}_{s,x}\left[e^{-(\mathcal{W}_{s:T}+g(X_T))/\lambda}\right]}. \tag{8}$$

Thus the optimal control is an exponentially weighted conditional mean of the adjoint.

Off-policy form. Using the likelihood correction $\Gamma^{\bar{u}}_{s:T}$ from Section 3.1, Girsanov's theorem gives

$$u^*(s,x) = -\lambda^{-1}\sigma_s^\top \frac{\mathbb{E}^{\bar{u}}_{s,x}\left[e^{-(\mathcal{W}_{s:T}+g(X_T))/\lambda+\Gamma^{\bar{u}}_{s:T}}\widetilde{a}_s\right]}{\mathbb{E}^{\bar{u}}_{s,x}\left[e^{-(\mathcal{W}_{s:T}+g(X_T))/\lambda+\Gamma^{\bar{u}}_{s:T}}\right]}. \tag{9}$$

This identity holds for any frozen admissible Markov collection control $\bar{u}$. The adjoint still uses $D_x b_r$ along the collected path: we differentiate under the reference dynamics before changing measure. No derivative of the collection control is needed. The assumptions and derivation are given in Appendix B.1.

Why recursion matters. Directly evaluating equation 9 requires simulating to T and propagating the terminal gradient through the entire remaining path. Both the exponential weight and the adjoint depend on the sampled future. Replacing the future after τ by $V(\tau,\cdot)$ and $\nabla V(\tau,\cdot)$ gives a recursive target from $[s,\tau]$. Repeated updates propagate terminal information through these short segments, and the stored likelihood correction permits their reuse.

4.2 Value-Control Recursion and Variance Reduction

Deriving the recursion. Conditioning the reference path integral on the state at $s < \tau \le T$ gives

$$e^{-V(s,x)/\lambda} = \mathbb{E}^{\mathrm{base}}_{s,x}\left[e^{-(\mathcal{W}_{s:\tau}+V(\tau,X_\tau))/\lambda}\right]. \tag{10}$$

Run the same adjoint equation on $[s,\tau]$, now with the continuation gradient at its endpoint:

$$\frac{\mathrm{d}\widetilde{a}^\tau_r}{\mathrm{d}r} = -(D_x b_r(X_r))^\top \widetilde{a}^\tau_r - \nabla_x f_r(X_r), \qquad \widetilde{a}^\tau_\tau = \nabla_x V(\tau, X_\tau). \tag{11}$$

The superscript indicates the endpoint of the adjoint; at $\tau = T$, it agrees with $\widetilde{a}_r$ in equation 7.

Proposition 1 (Off-policy value-control recursion)**.** *Under the assumptions in Appendix B.1,*

$$V(s,x) = -\lambda \log \mathbb{E}^{\bar{u}}_{s,x}\left[e^{-(\mathcal{W}_{s:\tau}+V(\tau,X_\tau))/\lambda+\Gamma^{\bar{u}}_{s:\tau}}\right], \tag{12a}$$

$$u^*(s,x) = -\lambda^{-1}\sigma_s^\top \frac{\mathbb{E}^{\bar{u}}_{s,x}\left[e^{-(\mathcal{W}_{s:\tau}+V(\tau,X_\tau))/\lambda+\Gamma^{\bar{u}}_{s:\tau}}\widetilde{a}^\tau_s\right]}{\mathbb{E}^{\bar{u}}_{s,x}\left[e^{-(\mathcal{W}_{s:\tau}+V(\tau,X_\tau))/\lambda+\Gamma^{\bar{u}}_{s:\tau}}\right]}. \tag{12b}$$

Proof. Under the reference dynamics, the adjoint gives $\nabla_x(\mathcal{W}_{s:\tau} + V(\tau, X_\tau)) = \widetilde{a}^\tau_s$. Taking the logarithmic derivative of equation 10 yields

$$u^*(s,x) = -\lambda^{-1}\sigma_s^\top \frac{\mathbb{E}^{\mathrm{base}}_{s,x}\left[e^{-(\mathcal{W}_{s:\tau}+V(\tau,X_\tau))/\lambda}\widetilde{a}^\tau_s\right]}{\mathbb{E}^{\mathrm{base}}_{s,x}\left[e^{-(\mathcal{W}_{s:\tau}+V(\tau,X_\tau))/\lambda}\right]}. \tag{13}$$

Changing measure in both expectations proves the control identity; the same change of measure proves the value identity. □

The first identity is the PI-VM recursion from Section 3.2. The second propagates the endpoint gradient with the same exponential weighting. Setting $\tau = T$ recovers the full-horizon representation, while $\tau < T$ replaces the omitted future by its value and gradient.

Variance reduction. With an exact continuation, recursion integrates out the sampled future. To quantify this benefit, consider the exponential weight shared by the two identities in Proposition 1:

$$Z_{s:\tau} = \exp\left[-\frac{\mathcal{W}_{s:\tau} + V(\tau, X_\tau)}{\lambda} + \Gamma^{\bar{u}}_{s:\tau}\right], \qquad s < \tau \le T. \tag{14}$$

Here $V(T, X_T) = g(X_T)$, so $Z_{s:T}$ is the full-horizon weight. Fix (s, x) and the collection policy $\bar{u}$, and let $\mathcal{F}_\tau$ denote the simulated history up to τ.

Proposition 2 (Variance reduction of the path weight)**.** *With exact continuation $V(\tau, \cdot)$ and $\mathbb{E}^{\bar{u}}_{s,x}[Z^2_{s:T}] < \infty$,*

$$Z_{s:\tau} = \mathbb{E}^{\bar{u}}_{s,x}[Z_{s:T} \mid \mathcal{F}_\tau]. \tag{15}$$

Consequently, under $\mathbb{P}^{\bar{u}}_{s,x}$,

$$\mathrm{Var}(Z_{s:T}) = \mathrm{Var}(Z_{s:\tau}) + \mathbb{E}[\mathrm{Var}(Z_{s:T} \mid \mathcal{F}_\tau)] \ge \mathrm{Var}(Z_{s:\tau}). \tag{16}$$

Proof. The conditional expectation of the exponential continuation factor over $[\tau, T]$ is $e^{-V(\tau, X_\tau)/\lambda}$. Multiplying by the observed prefix factor gives equation 15. The law of total variance then gives equation 16. □

Both weights have mean $e^{-V(s,x)/\lambda}$, but the short-segment weight removes the conditional variance of the future. For M independent branches, the variance reduction of their sample averages is the last expectation in equation 16 divided by M. This establishes a sampling benefit for the shared exponential weight. Appendix B.4 extends the argument to the weighted adjoint and the asymptotic covariance of the final value and control targets. With a learned continuation, the segment length balances this reduction in sampling noise against bootstrap error.

4.3 Joint Value-Control Learning

Joint loss. PI-VCM augments value matching with direct supervision of the control induced by V_θ. For a minibatch $\mathcal{I}$ of B starting pairs and a subset $\mathcal{I}_u$ of B_u pairs used for control matching, we minimize

$$\begin{aligned}
\mathcal{L}_k(\theta) &= \mathcal{L}_{\mathrm{V}}(\theta) + \beta_k \mathcal{L}_{\mathrm{u}}(\theta), \\
\mathcal{L}_{\mathrm{V}}(\theta) &= \frac{1}{B s_V^2} \sum_{i \in \mathcal{I}} \left| V_\theta(s_i, x_i) - \mathrm{sg}[\widehat{V}(s_i, x_i)] \right|^2, \\
\mathcal{L}_{\mathrm{u}}(\theta) &= \frac{1}{B_u d_u s_u^2} \sum_{i \in \mathcal{I}_u} \left\| u_\theta(s_i, x_i) - \mathrm{sg}[\widehat{u}(s_i, x_i)] \right\|^2.
\end{aligned} \tag{17}$$

The predictions share a single value network:

$$u_\theta(s, x) = -\lambda^{-1} \sigma_s^\top \nabla_x V_\theta(s, x). \tag{18}$$

Here $\widehat{V}$ and $\widehat{u}$ are fixed Monte Carlo targets defined below, $\beta_k \ge 0$ balances the two terms, and $s_V, s_u > 0$ are detached loss scales. The value loss learns the continuation cost. The control loss supervises the component of the state gradient that determines the policy.

Constructing the targets. At iteration k, freeze the endpoint function as

$$\bar{V}_\tau(z) = \begin{cases} V_{\theta_k}(\tau, z), & \tau < T, \\ g(z), & \tau = T. \end{cases} \tag{19}$$

For each start (s, x), collect M independent branches on $[s, \tau]$ under a frozen policy $\bar{u}$. On each branch, solve the lean adjoint with endpoint $\nabla \bar{V}_\tau(X^{(j)}_\tau)$, obtaining $\widetilde{a}^{(j)}_s$. The empirical versions of the two recursions are

$$\widehat{V}(s, x) = -\lambda \log \left[\frac{1}{M} \sum_{j=1}^{M} \exp\left(-\frac{\mathcal{W}^{(j)}_{s:\tau} + \bar{V}_\tau(X^{(j)}_\tau)}{\lambda} + \Gamma^{\bar{u},(j)}_{s:\tau} \right) \right], \tag{20a}$$

$$\widehat{u}(s,x) = -\lambda^{-1}\sigma_s^\top \frac{\sum_{j=1}^{M} \exp\left(-\dfrac{\mathcal{W}_{s:\tau}^{(j)} + \bar{V}_\tau(X_\tau^{(j)})}{\lambda} + \Gamma_{s:\tau}^{\bar{u},(j)}\right) \widetilde{a}_s^{(j)}}{\sum_{j=1}^{M} \exp\left(-\dfrac{\mathcal{W}_{s:\tau}^{(j)} + \bar{V}_\tau(X_\tau^{(j)})}{\lambda} + \Gamma_{s:\tau}^{\bar{u},(j)}\right)}. \tag{20b}$$

Both targets use the same branches and exponential weights, pairing each continuation value with its endpoint gradient. The control target is their normalized weighted adjoint, projected into the control coordinates. The segment schedule includes terminal-ending branches to anchor the recursion to the prescribed terminal cost. Evaluate endpoint values and gradients before the update, then detach the resulting targets. For numerical stability, use log-sum-exp for the value target and subtract a common maximum exponent in the control ratio.

Affine sensitivity caching and replay. Reusing a trajectory segment requires updating its adjoint when the continuation gradient changes. Along a fixed branch $X^{(j)}$, the adjoint solves the continuous-time backward equation

$$\begin{aligned} \frac{\mathrm{d}\widetilde{a}_r^{(j)}}{\mathrm{d}r} &= -(D_x b_r(X_r^{(j)}))^\top \widetilde{a}_r^{(j)} - \nabla f_r(X_r^{(j)}), \qquad r \in [s,\tau], \\ \widetilde{a}_\tau^{(j)} &= \nabla \bar{V}_\tau(X_\tau^{(j)}). \end{aligned} \tag{21}$$

Since its solution is affine in the endpoint gradient, the path-dependent coefficients can be computed once by integrating backward from τ to s:

$$\begin{aligned} \frac{\mathrm{d}K_{r:\tau}^{(j)}}{\mathrm{d}r} &= -(D_x b_r(X_r^{(j)}))^\top K_{r:\tau}^{(j)}, & K_{\tau:\tau}^{(j)} &= I, \\ \frac{\mathrm{d}c_{r:\tau}^{(j)}}{\mathrm{d}r} &= -(D_x b_r(X_r^{(j)}))^\top c_{r:\tau}^{(j)} - \nabla f_r(X_r^{(j)}), & c_{\tau:\tau}^{(j)} &= 0. \end{aligned} \tag{22}$$

The starting adjoint for any refreshed continuation is then

$$\widetilde{a}_s^{(j)} = K_{s:\tau}^{(j)} \nabla \bar{V}_\tau(X_\tau^{(j)}) + c_{s:\tau}^{(j)}. \tag{23}$$

We store complete branch bundles in the replay buffer $\mathcal{D}$:

$$\left(s, x, \tau, \left\{X_\tau^{(j)}, \mathcal{W}_{s:\tau}^{(j)}, \Gamma_{s:\tau}^{\bar{u},(j)}, K_{s:\tau}^{(j)}, c_{s:\tau}^{(j)}\right\}_{j=1}^{M}\right). \tag{24}$$

The cost and likelihood correction retain their continuous-time definitions from the background. On replay, we refresh endpoint values and gradients and apply the cached affine map, without solving another backward equation. The original collection correction is retained, and branches stay grouped by their common start. The affine representation is exact along each fixed trajectory; its proof is given in Appendix B.5. Updating the continuation network or the current collection policy leaves existing bundles valid with their original likelihood corrections. If the reference dynamics or running cost changes, we clear and repopulate the affected buffer; a change only to the terminal cost can be handled by refreshing terminal values and gradients. Appendix D details cache validity and numerical implementation. Dense $K_{s:\tau}$ requires $O(d^2)$ storage and replay work per branch, with lower costs for structured drift Jacobians. Algorithm 1 in Appendix D summarizes the complete training procedure.

Applications. The same joint value-control objective applies to Bayesian smoothing and mean-field games after specifying the corresponding SOC dynamics and costs. In smoothing, a fixed observation path determines the control problem; the learned value is used to reweight the initial distribution, and its induced control defines the approximate posterior dynamics. In mean-field games, each fixed population flow defines an SOC best-response problem. Simulating the learned control yields an estimated response flow for the outer population update. Appendix A gives the full formulations and application-specific training procedures.

5 Experiments

We first evaluate PI-VCM on analytic stochastic optimal control (SOC) benchmarks, then study two applications: Bayesian smoothing and mean-field games. Detailed experimental settings, metric formulas, and reporting conventions are collected in Appendix G.

Table 1: Control $\mathcal{L}_2$ ($\downarrow$) error on three OU tasks.

METHOD	OU LINEAR	OU QUADRATIC (EASY)	OU QUADRATIC (HARD)
RE	0.000030 ± 0.000005	0.000190 ± 0.000015	0.004538 ± 0.000387
CE	0.000324 ± 0.000076	0.002156 ± 0.000216	1.469716 ± 0.013633
VAR	0.000330 ± 0.000181	0.092349 ± 0.000648	1.469716 ± 0.013633
LVAR	0.000148 ± 0.000031	0.000612 ± 0.000038	0.015214 ± 0.001026
REINFORCE	0.000938 ± 0.000425	0.016159 ± 0.000410	0.208436 ± 0.020531
SOCM	$\mathbf{0.000000 \pm 0.000000}$	0.000441 ± 0.000011	1.469716 ± 0.013633
SOCM-A	0.000002 ± 0.000005	0.000734 ± 0.000061	1.469716 ± 0.013633
AM	0.000002 ± 0.000000	0.000286 ± 0.000022	0.006749 ± 0.000425
VM	0.000348 ± 0.000038	0.000986 ± 0.000116	0.011803 ± 0.002636
PI-VM	0.000008 ± 0.000001	0.000107 ± 0.000010	0.000788 ± 0.000065
PI-VCM (ours)	0.000002 ± 0.000000	$\mathbf{0.000033 \pm 0.000004}$	$\mathbf{0.000612 \pm 0.000080}$

Table 2: Control $\mathcal{L}_2$ ($\downarrow$) error across state dimensions on Quadratic OU Easy.

DIM	AM	VM	PI-VM	PI-VCM (ours)
10	0.000265 ± 0.000011	0.000580 ± 0.000043	0.000060 ± 0.000004	$\mathbf{0.000017 \pm 0.000004}$
20	0.000286 ± 0.000022	0.000986 ± 0.000116	0.000107 ± 0.000010	$\mathbf{0.000033 \pm 0.000004}$
80	0.029590 ± 0.000281	0.003545 ± 0.000392	0.000435 ± 0.000041	$\mathbf{0.000102 \pm 0.000008}$
150	0.076204 ± 0.000540	0.008990 ± 0.000373	0.000790 ± 0.000084	$\mathbf{0.000171 \pm 0.000012}$
200	0.089832 ± 0.000227	0.022904 ± 0.000518	0.002557 ± 0.000617	$\mathbf{0.000230 \pm 0.000004}$

5.1 STOCHASTIC OPTIMAL CONTROL

OU benchmarks.

We evaluate control learning on three OU problems (Domingo i Enrich et al., 2024) with analytic optimal controls: Linear, Quadratic Easy, and Quadratic Hard. We compare against RE, CE, VAR, LVAR, and REINFORCE; control-matching methods SOCM, SOCM-A, and AM; and value-matching methods VM and PI-VM (Nüsken & Richter, 2021; Domingo i Enrich et al., 2024; Domingo-Enrich et al., 2025; Jensen et al., 2026; Liao et al., 2026). The metric, denoted Control $\mathcal{L}_2$, is mean squared control error per coordinate at preterminal states along optimal-policy trajectories, directly assessing the learned control.

Table 1 shows the clearest gains on the quadratic tasks, where PI-VCM achieves the lowest mean control errors, improving on PI-VM by approximately 69% and 22% on Easy and Hard, respectively. These tasks require state-dependent control, making accurate value gradients essential. The results are consistent with the benefit of directly supervising these gradients. On OU Linear, SOCM leads and several matching methods attain similarly small errors, leaving less room for improvement.

Dimensional scaling. To examine dimensional scaling, we vary Quadratic OU Easy from $d = 10$ to 200, comparing with AM, VM, and PI-VM using the same control-error metric. Table 2 shows that PI-VCM achieves the lowest mean control error among the compared methods at every tested dimension , with PI-VM's error reaching $11.1\times$ that of PI-VCM at $d = 200$.

Ablation studies. Table 7 supports our choice of $M = 8, H = 8$, which achieves the lowest mean control error among the tested settings.

5.2 BAYESIAN SMOOTHING

We reconstruct posterior state distributions from noisy observations for reaction–diffusion(Eklund et al., 2026), double-well, and Lorenz–63 models, using the formulation in Section A.1. We compare with optimized variational guides (Variational) and ad-hoc guides (Ad-hoc) (Eklund et al., 2026), as well as control-based methods trained with log-variance (LV) and relative-entropy (RE) objectives (Nüsken & Richter, 2021). For reaction–diffusion, global and standard-deviation RMSE measure errors in the posterior mean and spread, respectively. For double-well smoothing, marginal

Table 3: Performance comparison on Bayesian smoothing benchmarks.

	Reaction–Diffusion		Double-Well Smoothing		Lorenz–63	
Method	Global RMSE ↓	Std. RMSE ↓	Marginal W_1 ↓	Mode-prob. MAE ↓	Norm. marginal W_1 ↓	Unobserved RMSE ↓
Variational	0.00980 ± 0.00088	0.00625 ± 0.00034	0.05653 ± 0.00170	$\mathbf{0.00977 \pm 0.00717}$	$\mathbf{0.00311 \pm 0.00062}$	$\mathbf{0.09938 \pm 0.03912}$
LV	0.02778 ± 0.00541	0.01227 ± 0.00311	0.08277 ± 0.03624	0.02943 ± 0.02021	0.00689 ± 0.00353	0.26661 ± 0.16768
RE	0.02187 ± 0.00499	0.00972 ± 0.00344	0.09268 ± 0.02486	0.05557 ± 0.01858	0.00608 ± 0.00123	0.22191 ± 0.04214
Ad-hoc	0.02627 ± 0.00019	0.01216 ± 0.00028	0.05678 ± 0.00103	0.01231 ± 0.00299	0.00474 ± 0.00011	0.14541 ± 0.00600
PI-VCM (ours)	$\mathbf{0.00601 \pm 0.00094}$	$\mathbf{0.00211 \pm 0.00017}$	$\mathbf{0.01652 \pm 0.00397}$	0.01601 ± 0.00604	0.00483 ± 0.00112	0.17635 ± 0.05014

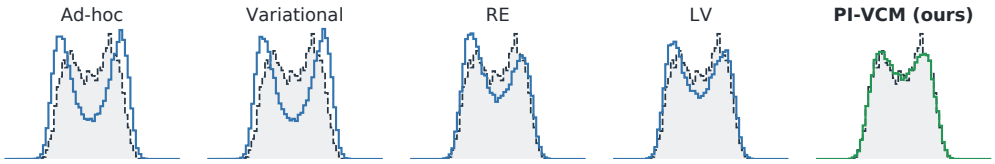

Figure 1: Double-well posterior marginals. Dashed curves show the reference.

Wasserstein-1 (W_1) distance measures distributional discrepancy, while mode-probability MAE measures error in the probability assigned to each well. For Lorenz–63, normalized marginal W_1 measures distributional discrepancy on a normalized scale, while unobserved-component RMSE measures reconstruction error in states that are not directly observed.

Table 3 highlights the strengths of PI-VCM in both reconstruction and distributional accuracy. It achieves the lowest mean errors for posterior reconstruction and spread on reaction–diffusion, and substantially reduces marginal W_1 on double-well smoothing. On Lorenz–63, PI-VCM improves on LV and RE on both metrics. These results demonstrate the applicability of a common control-based solver across spatially coupled, multimodal, and chaotic smoothing problems, with particularly strong gains on reaction–diffusion and double-well models.

Figure 1 compares terminal double-well marginals. Ad-hoc and Variational overconcentrate near the two peaks, underestimating the mass between them. RE and LV better capture the central region but show imbalanced peak heights relative to the reference. PI-VCM more closely follows the reference across both peaks and the intervening region.

5.3 Mean-field games

Synthetic benchmarks. We evaluate our framework on three mean-field game benchmarks: two-exit routing with terminal crowding, transport–congestion, and non-potential linear–quadratic (LQ). These tasks examine destination choice under terminal crowding, target-directed transport under congestion, and control under asymmetric coupling with the population mean, respectively.

We compare with APAC-Net, which alternates population and control networks (Lin et al., 2021); deep policy iteration (DPI) (Assouli & Missaoui, 2024); MFDGM, which minimizes coupled HJB–FP residuals (Assouli & Missaoui, 2023); and DEDA-FP, which combines reinforcement learning with fictitious play (Magnino et al., 2025), where results are available. We also include the value-only PI-VM baseline (Liao et al., 2026). For non-potential LQ, we implement a task-specific DPI variant, denoted DPI-MGF, using a Gaussian population representation and a log-MGF-based weak Fokker–Planck loss. Its results therefore correspond to this modified implementation. We evaluate control optimality while holding each method's predicted population distribution fixed. For two-exit routing and transport–congestion, the best-response (BR) gap measures the difference in cost between the learned control and a numerically computed best response. For non-potential LQ, Bellman advantage measures the accumulated loss from suboptimal control choices using the exact discrete-time LQ best response.

Table 4 shows that PI-VCM achieves the lowest mean optimality error across all three MFG benchmarks. Its clearest gain over PI-VM occurs in two-exit routing, where the BR gap is approximately 18 times smaller. The advantage also holds under congestion and non-potential interactions, supporting joint value-control learning for computing individual responses across these settings.

Table 4: Evaluation on mean-field game benchmarks and real-world pedestrian-density prediction.
Control optimality on three mean-field game benchmarks.

	TWO-EXIT ROUTING	TRANSPORT–CONGESTION	NON-POTENTIAL LQ
METHOD	BR gap ↓	BR gap ↓	Bellman advantage ↓
APAC-Net	3.3712 ± 0.6516	0.0253 ± 0.0030	–
DPI / DPI-MGF	6.5258 ± 0.3025	0.6512 ± 0.3643	0.00208 ± 0.00015
MFDGM	7.0869 ± 0.0380	0.0243 ± 0.0041	0.00540 ± 0.00067
DEDA-FP	2.1418 ± 0.4839	0.1922 ± 0.0489	0.0355 ± 0.0219
PI-VM	0.2576 ± 0.0043	0.0214 ± 0.0014	0.00031 ± 0.00002
PI-VCM (ours)	$\mathbf{0.0145 \pm 0.0101}$	$\mathbf{0.00176 \pm 0.00152}$	$\mathbf{0.00024 \pm 0.00014}$

Pedestrian-density prediction on Lyon measured by relative $\mathcal{L}_2$ (↓)

METHOD	1B	2D	2E	AVERAGE
MFDGM	0.5795 ± 0.0018	0.6932 ± 0.0027	$\mathbf{0.5248 \pm 0.0023}$	0.5992 ± 0.0012
DEDA-FP	0.5840 ± 0.0027	0.7087 ± 0.0169	0.5262 ± 0.0083	0.6063 ± 0.0060
PI-VCM (ours)	$\mathbf{0.5650 \pm 0.0017}$	$\mathbf{0.6306 \pm 0.0040}$	0.5400 ± 0.0025	$\mathbf{0.5786 \pm 0.0002}$

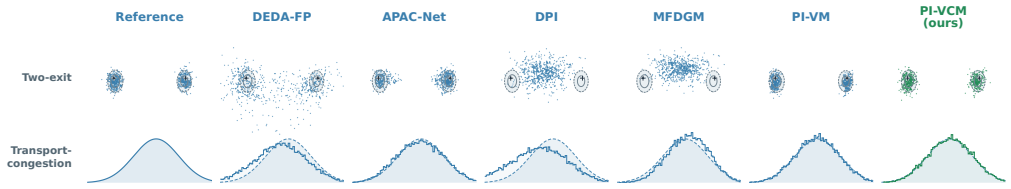

Figure 2: Terminal distributions for two-exit routing (top) and transport–congestion (bottom).

Figure 2 compares terminal population distributions. In two-exit routing, both PI-VCM and PI-VM recover the bimodal structure, while PI-VCM better matches the reference dispersion and PI-VM produces more concentrated modes. Other baselines show excessive dispersion, interior bias, or substantial mass between the exits. In transport–congestion, PI-VCM and PI-VM both closely match the reference density, including its peak and tails, whereas the remaining methods exhibit larger location or density-shape deviations.

Lyon density prediction. We predict pedestrian densities 2 seconds ahead from a 3-second observation window on sequences 1B, 2D, and 2E of the Lyon pedestrian dataset (Dufour et al., 2024). We compare PI-VCM with MFDGM and DEDA-FP using density relative L_2 error against future observations, reporting each sequence and their equally weighted average. Table 4 shows that PI-VCM achieves a modest average advantage on Lyon, while all three methods leave room for improvement in density prediction. The results illustrate both the applicability of the framework to observed pedestrian dynamics and the limitations of its current forecasting accuracy. Cross-sequence density forecasts and the time evolution of sequence 2D are shown in Appendix H.

6 CONCLUSION

We introduced PI-VCM, which augments local path-integral value learning with direct supervision of the gradient-derived control. Differentiating the local value recursion yields a control target from the same short trajectories, and affine sensitivity caching makes these targets reusable as the continuation network changes. A single value network is trained against both targets and supplies the deployed feedback. The same construction applies to observation-dependent smoothing problems and to best-response solves in mean-field games. Experiments demonstrate that our method consistently improves control accuracy over exsiting baselines across SOC benchmarks, including high-dimensional settings, while achieving strong performance on Bayesian smoothing and mean-field games.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experimental design, assist with method implementation, assist with translation, and support the interpretation of experimental results.

We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, propose or refine hypotheses, clean or reformat datasets, or conduct qualitative or thematic data analysis.

Additionally, we used generative AI tools to assist with code editing, manuscript drafting and revision, literature search and summarization, and improving readability and presentation.

All AI-assisted outputs were reviewed, verified, and revised by the authors. Code and experimental suggestions were validated through implementation and experiments, and suggested references were checked against the original sources. We take responsibility for the final content of this work.

REFERENCES

Yves Achdou and Italo Capuzzo-Dolcetta. Mean Field Games: Numerical Methods. *SIAM Journal on Numerical Analysis*, 2010. doi: 10.1137/090758477. URL https://doi.org/10.1137/090758477.

Cédric Archambeau, Manfred Opper, Yuan Shen, Dan Cornford, and John Shawe-Taylor. Variational Inference for Diffusion Processes. *Advances in Neural Information Processing Systems*, 2007. URL https://papers.neurips.cc/paper_files/paper/2007/hash/818f4654ed39a1c147d1e51a00ffb4cb-Abstract.html.

Mouhcine Assouli and Badr Missaoui. Deep Learning for Mean Field Games with non-separable Hamiltonians. *Chaos, Solitons & Fractals*, 2023. doi: 10.1016/j.chaos.2023.113802. URL https://doi.org/10.1016/j.chaos.2023.113802.

Mouhcine Assouli and Badr Missaoui. Deep Policy Iteration for High-Dimensional Mean Field Games. *Applied Mathematics and Computation*, 2024. URL https://arxiv.org/abs/2310.10827.

Richard Bellman. Dynamic programming. *science*, 153(3731):34–37, 1966.

Carles Domingo-Enrich. A taxonomy of loss functions for stochastic optimal control. *arXiv preprint arXiv:2410.00345*, 2024.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

Carles Domingo i Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky TQ Chen. Stochastic optimal control matching. *Advances in Neural Information Processing Systems*, 37:112459–112504, 2024.

Yuanqi Du, Jiajun He, Dinghuai Zhang, Eric Vanden-Eijnden, and Carles Domingo-Enrich. Rare event analysis via stochastic optimal control. *arXiv preprint arXiv:2604.13213*, 2026.

Oscar Dufour, Huu-Tu Dang, Jakob Cordes, Raphael Korbmacher, Gaudou Benoit, Mohcine Chraibi, Alexandre Nicolas, and Antoine Tordeux. Dense crowd dynamics and pedestrian trajectories: A multiscale field study at the Fête des Lumières in Lyon. Zenodo, 2024. URL https://doi.org/10.5281/zenodo.13830435. Dataset, version v1.

Oskar Eklund, Annika Lang, and Moritz Schauer. Guided smoothing and control for diffusion processes. *Stochastic Processes and their Applications*, 2026. doi: 10.1016/j.spa.2025.104806. URL https://doi.org/10.1016/j.spa.2025.104806.

Simon J. Godsill, Arnaud Doucet, and Mike West. Monte Carlo Smoothing for Nonlinear Time Series. *Journal of the American Statistical Association*, 2004. URL https://scholars.duke.edu/publication/705667.

Olivier Guéant. Mean field games equations with quadratic Hamiltonian: a specific approach. *Mathematical Models and Methods in Applied Sciences*, 2012. URL https://arxiv.org/abs/1106.3269.

Jeremy Heng, Adrian N. Bishop, George Deligiannidis, and Arnaud Doucet. Controlled Sequential Monte Carlo. *The Annals of Statistics*, 2020. doi: 10.1214/19-AOS1914. URL https://doi.org/10.1214/19-AOS1914.

Lars Holdijk, Yuanqi Du, Ferry Hooft, Priyank Jaini, Berend Ensing, and Max Welling. Stochastic optimal control for collective variable free sampling of molecular transition paths. *Advances in Neural Information Processing Systems*, 36:79540–79556, 2023.

Minyi Huang, Peter E. Caines, and Roland P. Malhamé. Large-Population Cost-Coupled LQG Problems With Nonuniform Agents: Individual-Mass Behavior and Decentralized epsilon-Nash Equilibria. *IEEE Transactions on Automatic Control*, 2007. URL https://people.math.carleton.ca/~mhuang/tac07_lqg.pdf.

Cristian Jensen, Luca Schaufelberger, Riccardo De Santi, Kjell Jorner, and Andreas Krause. Value matching: Scalable and gradient-free reward-guided flow adaptation. In *International Conference on Learning Representations*, volume 2026, pp. 144926–144958, 2026.

Hilbert J Kappen. Linear theory for control of nonlinear stochastic systems. *Physical review letters*, 95(20):200201, 2005.

Jin Won Kim and Prashant G. Mehta. An Optimal Control Derivation of Nonlinear Smoothing Equations. In *Advances in Dynamics, Optimization and Computation*. 2020. doi: 10.1007/978-3-030-51264-4_12. URL https://doi.org/10.1007/978-3-030-51264-4_12.

Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japanese Journal of Mathematics*, 2007. URL https://journal.ms.u-tokyo.ac.jp/~toshi/jjm/JJM_HP/contents/abstract/2-1/JJM0657.html.

Bangyan Liao, Chenglei Yu, Yuchen Yang, Chuanrui Wang, Zhisheng Song, Peidong Liu, and Tailin Wu. Path integral value matching for linear quadratic stochastic optimal control. *arXiv preprint arXiv:2608.10777*, 2026.

Alex Tong Lin, Samy Wu Fung, Wuchen Li, Levon Nurbekyan, and Stanley J. Osher. Alternating the Population and Control Neural Networks to Solve High-Dimensional Stochastic Mean-Field Games. *Proceedings of the National Academy of Sciences*, 2021. URL https://arxiv.org/abs/2002.10113.

Fredrik Lindsten, Michael I. Jordan, and Thomas B. Schön. Particle Gibbs with Ancestor Sampling. *Journal of Machine Learning Research*, 2014. URL https://www.jmlr.org/beta/papers/v15/lindsten14a.html.

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin K Miller, and Ricky TQ Chen. Adjoint schrödinger bridge sampler. *Advances in Neural Information Processing Systems*, 38:15673–15708, 2026.

Lorenzo Magnino, Kai Shao, Zida Wu, Jiacheng Shen, and Mathieu Lauriere. Solving Continuous Mean Field Games: Deep Reinforcement Learning for Non-Stationary Dynamics. *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/968582fc27a1f2213c3e3d96e92f008a-Abstract-Conference.html.

Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2(4):48, 2021.

Hans-Christian Ruiz and Hilbert J. Kappen. Particle Smoothing for Hidden Diffusion Processes: Adaptive Path Integral Smoother. *IEEE Transactions on Signal Processing*, 2017. URL https://arxiv.org/abs/1605.00278.

Simo Särkkä. Unscented Rauch–Tung–Striebel Smoother. *IEEE Transactions on Automatic Control*, 2008. URL https://users.aalto.fi/~ssarkka/pub/uks-preprint.pdf.

Emanuel Todorov. Linearly-solvable markov decision problems. *Advances in neural information processing systems*, 19, 2006.

Frank H. van der Meulen and Stefan Sommer. Backward Filtering Forward Guiding. *Journal of Machine Learning Research*, 2025. URL https://jmlr.org/papers/v26/25-1130.html.

Fatemeh Yaghoobi, Adrien Corenflos, Sakira Hassan, and Simo Särkkä. Parallel Square-Root Statistical Linear Regression for Inference in Nonlinear State Space Models. *SIAM Journal on Scientific Computing*, 2025. doi: 10.1137/23M156121X. URL https://doi.org/10.1137/23M156121X.

A Applications to Bayesian Smoothing and Mean-Field Games

We apply PI-VCM to smoothing and mean-field games through observation-dependent control problems and population-dependent best responses, respectively. Both use the joint loss in Section 4.3; their dynamics and costs determine the training targets. The learned value and control define a posterior sampler for smoothing or a best-response control and its induced population flow for mean-field games.

A.1 Bayesian Smoothing as a Control Problem

Inference problem. Consider the latent state and observation processes

$$\mathrm{d}X_t = b_t(X_t)\,\mathrm{d}t + \sigma_t\,\mathrm{d}W_t, \qquad X_0 \sim \mu_0,$$
$$\mathrm{d}Y_t = h_t(X_t)\,\mathrm{d}t + R^{1/2}\,\mathrm{d}B_t, \qquad W \perp B,$$

where W and B are Brownian motions, $R \succ 0$, and σ_t is independent of the state. Given an observation path $y_{0:T}$, the smoothing problem seeks the posterior path law $\Pi^y = \mathrm{Law}(X_{0:T} \mid Y_{0:T} = y_{0:T})$.

Control reformulation. Let $a_t = \sigma_t\sigma_t^\top$, define the observation potential $\Phi_t^y(x) = y_t^\top R^{-1} h_t(x)$, and write

$$\mathcal{L}_t\varphi = b_t^\top \nabla\varphi + \frac{1}{2}\,\mathrm{Tr}(a_t\nabla^2\varphi)$$

for the generator of the latent process. A pathwise transformation of the observation likelihood gives

$$\bar{b}_t^y(x) = b_t(x) - a_t\nabla\Phi_t^y(x), \tag{25a}$$
$$c_t^y(x) = \frac{1}{2}\|h_t(x)\|_{R^{-1}}^2 + \partial_t^{\exp}\Phi_t^y(x) + \mathcal{L}_t\Phi_t^y(x) - \frac{1}{2}\|\sigma_t^\top\nabla\Phi_t^y(x)\|^2, \tag{25b}$$
$$g^y(x) = -\Phi_T^y(x). \tag{25c}$$

The derivative $\partial_t^{\exp}$ acts only on the explicit time dependence of h_t; the construction requires no time derivative of y_t.

The resulting conditional control problem has objective J^y and controlled dynamics

$$J^y(s,x;u) = \mathbb{E}_{s,x}^{u,y}\left[\int_s^T \left(c_r^y(X_r) + \frac{1}{2}\|u_r\|^2\right)\mathrm{d}r + g^y(X_T)\right], \tag{26a}$$
$$\mathrm{d}X_r = \left[\bar{b}_r^y(X_r) + \sigma_r u_r\right]\mathrm{d}r + \sigma_r\,\mathrm{d}W_r, \qquad X_s = x. \tag{26b}$$

Its value function is $V^y(s,x) = \inf_{u\in\mathcal{U}} J^y(s,x;u)$. This is an instance of the SOC problem in Section 3.1, with coefficients $(\bar{b}^y, \sigma, c^y, g^y, \lambda = 1)$. The derivation of the transformation and its connection to the posterior path law are given in Appendix E.

Using PI-VCM. For the fixed observation path, we apply the joint loss equation 17 with targets constructed from the transformed coefficients in equation 25. The path weights use c^y, g^y and the likelihood correction to the transformed reference law; the control-target adjoint uses $D_x\bar{b}^y$, ∇c^y, and the endpoint value gradient. Thus the observations enter both training targets.

Let V_θ^y denote the value approximation for the given observation path. With $\lambda = 1$, its induced control is

$$u_\theta^y(t,x) = -\sigma_t^\top\nabla_x V_\theta^y(t,x).$$

We train the value and its induced control with

$$\mathcal{L}_{\mathrm{smooth}}(\theta;y) = \mathcal{L}_{\mathrm{V}}(\theta;y) + \beta\mathcal{L}_{\mathrm{u}}(\theta;y). \tag{27}$$

The two terms are the normalized matching losses in equation 17, with observation-dependent targets.

To construct the posterior sampler, combine the learned control with the transformed drift and reweight the initial distribution:

$$b_{\mathrm{post},\theta}^y(t,x) = \bar{b}_t^y(x) + \sigma_t u_\theta^y(t,x)$$
$$= b_t(x) - a_t\nabla_x\left[V_\theta^y(t,x) + \Phi_t^y(x)\right], \tag{28a}$$
$$\pi_{0,\theta}^y(\mathrm{d}x) \propto \exp\left[-V_\theta^y(0,x) - \Phi_0^y(x)\right]\mu_0(\mathrm{d}x). \tag{28b}$$

Simulation from $\pi^y_{0,\theta}$, with drift $b^y_{\mathrm{post},\theta}$ and diffusion coefficient σ_t, defines an approximate posterior sampler. In continuous time, replacing V^y_θ by the exact value V^y recovers Π^y under the assumptions stated in Appendix E.

Joint matching supervises both components: the value determines initial posterior reweighting, and its gradient determines the posterior dynamics.

A.2 Mean-Field Games through SOC Best Responses

Mean-field game formulation. Mean-field games model interactions among a large population of players, each optimizing its own control with dynamics and costs that depend on the population distribution. The collective evolution of these players determines that distribution.

Let $m = (m_t)_{t\in[0,T]}$ be a flow of population state distributions. For a fixed m, consider the control problem with drift $b_t(x, m_t)$, running cost $f_t(x, m_t)$, terminal cost $g(x, m_T)$, and initial distribution μ_0. An optimal control $u^{m,\star}$ induces a response flow

$$\Gamma(m)_t = \mathrm{Law}(X_t^{u^{m,\star}}), \qquad m^\star = \Gamma(m^\star).$$

The fixed-point condition defines a mean-field equilibrium: the control is optimal against the population flow it generates.

Control reformulation. The fixed-population problem has objective J^m and controlled dynamics

$$J^m(s,x;u) = \mathbb{E}^{u,m}_{s,x}\left[\int_s^T \left(f_r(X_r,m_r) + \frac{\lambda}{2}\|u_r\|^2\right)\mathrm{d}r + g(X_T,m_T)\right], \tag{29a}$$

$$\mathrm{d}X_r = \big[b_r(X_r,m_r) + \sigma_r u_r\big]\mathrm{d}r + \sigma_r\,\mathrm{d}W_r, \qquad X_s = x. \tag{29b}$$

Its value function is $V^m(s,x) = \inf_{u\in\mathcal{U}} J^m(s,x;u)$. This has the SOC coefficients $(b^m, \sigma, f^m, g^m, \lambda)$, where $b^m_t(x) = b_t(x,m_t)$, $f^m_t(x) = f_t(x,m_t)$, and $g^m(x) = g(x,m_T)$.

Using PI-VCM. At outer iteration k, hold m^k fixed and minimize the joint loss equation 17, constructing its targets using $(b^{m^k}, \sigma, f^{m^k}, g^{m^k}, \lambda)$. Spatial derivatives retain the state dependence of these coefficients while keeping m^k fixed.

The control induced by the current value network is

$$u^{m^k}_\theta(t,x) = -\frac{1}{\lambda}\sigma_t^\top \nabla_x V^{m^k}_\theta(t,x).$$

For the fixed population, the training loss is

$$\mathcal{L}_{\mathrm{MFG}}(\theta; m^k) = \mathcal{L}_{\mathrm{V}}(\theta; m^k) + \beta\mathcal{L}_{\mathrm{u}}(\theta; m^k), \tag{30}$$

with both matching terms in equation 17 using targets computed at m^k.

Let θ_k denote the parameters after the current inner solve. Simulating $u^{m^k}_{\theta_k}$ from μ_0 under the fixed flow m^k yields an estimated response flow $\widehat{\Gamma}_k(m^k)$. We then apply the relaxed update

$$m^{k+1} = (1-\gamma_k)m^k + \gamma_k\widehat{\Gamma}_k(m^k), \qquad 0 < \gamma_k \le 1. \tag{31}$$

We initialize each inner solve with the previously learned parameters. When the population update changes the reference dynamics or running cost, we clear and repopulate the affected replay buffer as described in Section 4.3.

The inner solve seeks an optimal control for the fixed population, while the outer update seeks population consistency. Together, these define the two requirements for a mean-field equilibrium. Appendix F presents the coupled optimality system and a sufficient condition for convergence of the outer iteration.

B Derivations and Guarantees for PI-VCM

B.1 Assumptions and the Path-Integral Representation

We use the SOC convention of equation 1: the diffusion is σ_t, the controlled drift is $b_t + \sigma_t u_t$, and the control cost is $\lambda\|u_t\|^2/2$. Gradients are column vectors, and $(D_x b_t)_{ij} = \partial b_{t,i}/\partial x_j$. We assume the following.

1. The reference and controlled SDEs have nonexplosive solutions. The diffusion is deterministic and state independent, and the reference stochastic flow is differentiable in its starting state. The drift, costs, and continuation functions have the state derivatives used below.
2. The value is a classical HJB solution for which verification applies. The exponential expectations are finite and positive; differentiation under expectation is valid. Weighted adjoints are integrable, and the weighted least-squares objectives are finite.
3. The collection control is a frozen Markov policy. Its Girsanov exponential is a true martingale on every interval considered; Novikov's condition is sufficient.
4. For covariance comparisons, both $e^{-(\mathcal{W}_{s:T}+g(X_T))/\lambda+\Gamma^{\bar{u}}_{s:T}}$ and its product with $\widetilde{a}_s$ are square integrable. For Monte Carlo limits, branches are independent and identically distributed conditional on their common start and collection policy.

The second-moment condition is additional to the first-moment condition needed for importance sampling. All statements below are conditional on a fixed starting pair (s, x). The collection policy and continuation are held fixed when taking the expectations; they may change between training iterations. The filtration is that of the driving process and state, augmented in the usual way. The square-integrability assumptions used for a particular covariance comparison apply to both the full-horizon and conditional moments in that comparison. They do not follow from Novikov's condition alone. For redundant noise coordinates, Girsanov is interpreted on the driving-noise probability space. Its density still gives a valid importance weight for state-path functionals, even when it differs from the marginal state-path density. No inverse of σ_t is required.

Minimizing the Hamiltonian gives

$$\begin{aligned} 0 &= \partial_t V + b_t^\top \nabla V + \frac{1}{2}\operatorname{tr}(\sigma_t\sigma_t^\top \nabla^2 V) + f_t - \frac{1}{2\lambda}\|\sigma_t^\top \nabla V\|^2, \\ V(T,x) &= g(x), \qquad u^* = -\lambda^{-1}\sigma_t^\top \nabla V. \end{aligned} \tag{32}$$

The exponential transformation cancels the gradient-square term:

$$\partial_t e^{-V/\lambda} + b_t^\top \nabla e^{-V/\lambda} + \frac{1}{2}\operatorname{tr}\Big(\sigma_t\sigma_t^\top \nabla^2 e^{-V/\lambda}\Big) - \frac{f_t}{\lambda}e^{-V/\lambda} = 0. \tag{33}$$

To see the cancellation explicitly, put $Z = e^{-V/\lambda}$. Then

$$\nabla Z = -\lambda^{-1} Z\nabla V, \qquad \nabla^2 Z = Z\big[\lambda^{-2}\nabla V\nabla V^\top - \lambda^{-1}\nabla^2 V\big].$$

Substitution of the HJB equation into $\partial_t Z + \mathcal{L}_t Z$ leaves $(f_t/\lambda)Z$, where $\mathcal{L}_t$ is the reference generator. This calculation uses the same diffusion channel σ_t in the noise and controlled drift and the penalty $\lambda\|u\|^2/2$; changing that convention requires rescaling the control and diffusion together. With terminal condition $e^{-g/\lambda}$, Feynman–Kac yields equation 3. The Markov property at τ gives equation 10.

B.2 Optimal Control and the Adjoint Representation

Because the noise is state independent, varying the initial state only differentiates the reference drift in the linearized dynamics. More explicitly, for a reference trajectory starting at x, let $J_{r,s} = D_x X_r$. For almost every reference realization,

$$\frac{\mathrm{d}J_{r,s}}{\mathrm{d}r} = D_x b_r(X_r) J_{r,s}, \qquad J_{s,s} = I. \tag{34}$$

For any differentiable endpoint cost v at τ, the pathwise chain rule gives

$$\nabla_x\left[\int_s^\tau f_r(X_r)\,\mathrm{d}r + v(X_\tau)\right] = \int_s^\tau J_{r,s}^\top \nabla f_r(X_r)\,\mathrm{d}r + J_{\tau,s}^\top \nabla v(X_\tau). \tag{35}$$

If $\widetilde{a}_r^v$ is the lean adjoint with endpoint $\nabla v(X_\tau)$, then $\mathrm{d}(J_{r,s}^\top \widetilde{a}_r^v)/\mathrm{d}r = -J_{r,s}^\top \nabla f_r(X_r)$. Integrating this identity from s to τ identifies the right-hand side of Eq. equation 35 with $\widetilde{a}_s^v$. Taking $\tau = T$ and $v = g$ proves the full-horizon identity below. Integration by parts between this linearized state equation and the lean adjoint gives

$$\nabla_x \big(\mathcal{W}_{s:T} + g(X_T)\big) = \widetilde{a}_s. \tag{36}$$

Indeed, in that product the two terms containing $D_x b_r$ cancel, leaving the running-cost gradient; the adjoint endpoint supplies $\nabla g(X_T)$. Differentiation under expectation then gives

$$\begin{aligned} &\nabla_x \mathbb{E}_{s,x}^{\mathrm{base}} \left[e^{-(\mathcal{W}_{s:T} + g(X_T))/\lambda} \right] \\ &\quad = -\lambda^{-1} \mathbb{E}_{s,x}^{\mathrm{base}} \left[e^{-(\mathcal{W}_{s:T} + g(X_T))/\lambda} \widetilde{a}_s \right]. \end{aligned} \tag{37}$$

Taking the logarithmic derivative in equation 3 and multiplying by $-\lambda^{-1}\sigma_s^\top$ proves equation 8. Girsanov changes the measure in its numerator and denominator by multiplying their integrands by $e^{\Gamma_{s:T}^{\bar{u}}}$, which proves equation 9. The adjoint is a path functional defined by the reference coefficients, so evaluating it on a controlled path leaves its equation unchanged. In particular, it is not the derivative of a rollout of $b + \sigma\bar{u}$ with its proposal Brownian increments held fixed. Differentiating that different map would introduce $D_x\bar{u}$. The order used here is to differentiate the reference expectation first and then change measure for the resulting path functionals.

For SOCM-Adjoint, condition the weighted regression on $X_s = x$. The Markov property separates the past from the future. The conditional factor from the past is

$$\mathbb{E}^{\bar{u}} \left[e^{-\mathcal{W}_{0:s}/\lambda + \Gamma_{0:s}^{\bar{u}}} \,\middle|\, X_s = x \right],$$

which is positive and independent of the fitted control at this start. After canceling it, the first-order condition is

$$\mathbb{E}_{s,x}^{\bar{u}} \left[e^{-(\mathcal{W}_{s:T} + g(X_T))/\lambda + \Gamma_{s:T}^{\bar{u}}} \left(u_\psi(s,x) + \lambda^{-1}\sigma_s^\top \widetilde{a}_s \right) \right] = 0. \tag{38}$$

Solving for $u_\psi(s,x)$ yields equation 9. The Hessian of this conditional quadratic objective is a positive scalar multiple of the identity, so the minimizer is unique. This is a population statement on the sampled support; a restricted network class and finite sampling can change the fitted result.

B.3 Local Recursion and Affine Dependence on the Endpoint

The reference Markov identity is

$$\begin{aligned} &\mathbb{E}_{s,x}^{\mathrm{base}} \left[e^{-(\mathcal{W}_{s:T} + g(X_T))/\lambda} \right] \\ &\quad = \mathbb{E}_{s,x}^{\mathrm{base}} \left[e^{-\mathcal{W}_{s:\tau}/\lambda} \mathbb{E}_{\tau,X_\tau}^{\mathrm{base}} \left[e^{-(\mathcal{W}_{\tau:T} + g(X_T))/\lambda} \right] \right] \\ &\quad = \mathbb{E}_{s,x}^{\mathrm{base}} \left[e^{-(\mathcal{W}_{s:\tau} + V(\tau,X_\tau))/\lambda} \right]. \end{aligned} \tag{39}$$

The same pathwise differentiation as above, with endpoint $V(\tau,\cdot)$, gives

$$\nabla_x \big(\mathcal{W}_{s:\tau} + V(\tau, X_\tau)\big) = \widetilde{a}_s^\tau. \tag{40}$$

The logarithmic derivative and change of measure prove Proposition 1.

For the variance proof and replay, use the coefficients K, c of the affine adjoint map. Their continuous-time equations along a fixed prefix are

$$\begin{aligned} \frac{\mathrm{d}K_{r:\tau}}{\mathrm{d}r} &= -(D_x b_r(X_r))^\top K_{r:\tau}, & K_{\tau:\tau} &= I, \\ \frac{\mathrm{d}c_{r:\tau}}{\mathrm{d}r} &= -(D_x b_r(X_r))^\top c_{r:\tau} - \nabla f_r(X_r), & c_{\tau:\tau} &= 0. \end{aligned} \tag{41}$$

Substitution into the adjoint equation verifies

$$\widetilde{a}_s = K_{s:\tau}\widetilde{a}_\tau + c_{s:\tau}, \qquad \widetilde{a}_s^\tau = K_{s:\tau}\nabla V(\tau, X_\tau) + c_{s:\tau}. \tag{42}$$

Both coefficients depend only on the prefix path and hence are $\mathcal{F}_\tau$-measurable. These are the coefficients stored by the affine replay buffer.

B.4 Proof of Variance Reduction

Shared exponential weight. Fix (s, x) and $\bar{u}$, and use $Z_{s:\tau}$ from equation 14. All expectations in this subsection are under $\mathbb{P}^{\bar{u}}_{s,x}$. Additivity of $\mathcal{W}$ and Γ, followed by the off-policy value identity at τ, gives

$$\begin{aligned}\mathbb{E}[Z_{s:T} \mid \mathcal{F}_\tau] &= e^{-\mathcal{W}_{s:\tau}/\lambda + \Gamma^{\bar{u}}_{s:\tau}} \mathbb{E}^{\bar{u}}_{\tau, X_\tau}\left[e^{-(\mathcal{W}_{\tau:T} + g(X_T))/\lambda + \Gamma^{\bar{u}}_{\tau:T}}\right] \\ &= e^{-(\mathcal{W}_{s:\tau} + V(\tau, X_\tau))/\lambda + \Gamma^{\bar{u}}_{s:\tau}} = Z_{s:\tau}.\end{aligned} \tag{43}$$

Taking expectations shows that both weights have mean $e^{-V(s,x)/\lambda}$. The law of total variance proves equation 16. The reduction is strict when $\mathrm{Var}(Z_{s:T} \mid \mathcal{F}_\tau) > 0$ with positive probability. For independent branch averages, both variances are divided by M. The tower property also gives $\mathrm{Var}(Z_{s:\tau_1}) \le \mathrm{Var}(Z_{s:\tau_2})$ for $s < \tau_1 \le \tau_2 \le T$.

Extension to the weighted adjoint. To connect this scalar result to control estimation, we also use the weighted-adjoint second-moment assumption in Appendix B.1. The gradient representation at the endpoint implies

$$\mathbb{E}\left[e^{-(\mathcal{W}_{\tau:T} + g(X_T))/\lambda + \Gamma^{\bar{u}}_{\tau:T}} \widetilde{a}_\tau \,\middle|\, \mathcal{F}_\tau\right] = e^{-V(\tau, X_\tau)/\lambda} \nabla V(\tau, X_\tau). \tag{44}$$

This identity precedes projection by $\sigma_\tau^\top$, so it also holds for rank-deficient diffusion. Using the affine decomposition in equation 42, whose coefficients are measurable with respect to the prefix, yields

$$\begin{aligned}\mathbb{E}[Z_{s:T} \widetilde{a}_s \mid \mathcal{F}_\tau] &= Z_{s:\tau}\left[K_{s:\tau} \nabla V(\tau, X_\tau) + c_{s:\tau}\right] \\ &= Z_{s:\tau} \widetilde{a}^\tau_s.\end{aligned} \tag{45}$$

Together with equation 15, this conditions both the denominator and the adjoint numerator on the same prefix. The law of total covariance therefore gives

$$\begin{aligned}\mathrm{Cov}\left[Z_{s:T}\begin{pmatrix}1\\ \widetilde{a}_s\end{pmatrix}\right] = \mathrm{Cov}\left[Z_{s:\tau}\begin{pmatrix}1\\ \widetilde{a}^\tau_s\end{pmatrix}\right] \\ + \mathbb{E}\left[\mathrm{Cov}\left[Z_{s:T}\begin{pmatrix}1\\ \widetilde{a}_s\end{pmatrix}\,\middle|\, \mathcal{F}_\tau\right]\right].\end{aligned} \tag{46}$$

The last term is positive semidefinite, and the same ordering holds for independent branch averages.

The final value and control targets. Use the targets in equation 20 with exact continuation $V(\tau, \cdot)$. The population means of the empirical denominator and adjoint numerator, each written as a branch average, are $e^{-V(s,x)/\lambda}$ and $e^{-V(s,x)/\lambda}\nabla V(s, x)$, respectively. The multivariate central limit theorem and delta method give

$$\begin{aligned}&\sqrt{M}\begin{pmatrix}\widehat{V}(s,x) - V(s,x)\\ \widehat{u}(s,x) - u^*(s,x)\end{pmatrix} \\ &\quad\Longrightarrow \mathcal{N}\left(0,\ \mathrm{Cov}\left[e^{(V(s,x) - \mathcal{W}_{s:\tau} - V(\tau, X_\tau))/\lambda + \Gamma^{\bar{u}}_{s:\tau}}\begin{pmatrix}-\lambda\\ -\lambda^{-1}\sigma_s^\top \widetilde{a}^\tau_s - u^*(s,x)\end{pmatrix}\right]\right).\end{aligned} \tag{47}$$

For full-horizon sampling, replace τ by T, $V(T, X_T)$ by $g(X_T)$, and $\widetilde{a}^\tau_s$ by $\widetilde{a}_s$. The vector inside this covariance is a fixed linear transformation of the joint weight and weighted adjoint. For clarity, if $R = Z_{s:\tau}/\mathbb{E}[Z_{s:\tau}]$, a centered influence vector for the same limit is

$$\begin{pmatrix}-\lambda(R-1)\\ R\big(-\lambda^{-1}\sigma_s^\top \widetilde{a}^\tau_s - u^*(s,x)\big)\end{pmatrix}. \tag{48}$$

Its mean is zero. The first component differs from the expression inside Eq. equation 47 by a constant and therefore has the same covariance. The delta method is applicable because the population denominator is strictly positive. It is applied jointly to the numerator and denominator, rather than treating their errors as independent. The conditional identities above are preserved by that transformation, so the limiting covariance for the shorter segment cannot increase. This accounts for the dependence between numerator and denominator.

This covariance belongs to the limiting normal law. Identifying it with a limit of M times the actual finite-sample covariance requires uniform integrability. The log and ratio targets do not in general inherit a finite-sample variance ordering from their input moments.

B.5 Continuous-Time Affine Sensitivity Replay

Fix a branch $X^{(j)}$ on $[s,\tau]$, and let $K^{(j)}_{r:\tau}, c^{(j)}_{r:\tau}$ solve equation 22. The endpoint gradient $\nabla\bar{V}_\tau(X^{(j)}_\tau)$ is constant with respect to the backward integration variable r. Differentiating the affine expression gives

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}r}\left[K^{(j)}_{r:\tau}\nabla\bar{V}_\tau(X^{(j)}_\tau)+c^{(j)}_{r:\tau}\right]\\&\quad=-(D_x b_r(X^{(j)}_r))^\top\left[K^{(j)}_{r:\tau}\nabla\bar{V}_\tau(X^{(j)}_\tau)+c^{(j)}_{r:\tau}\right]-\nabla f_r(X^{(j)}_r).\end{aligned}\tag{49}$$

At $r=\tau$, the same expression equals $\nabla\bar{V}_\tau(X^{(j)}_\tau)$, since $K^{(j)}_{\tau:\tau}=I$ and $c^{(j)}_{\tau:\tau}=0$. It therefore satisfies the adjoint equation and terminal condition in equation 21. Uniqueness of the linear terminal-value ODE implies

$$\widetilde{a}^{(j)}_r=K^{(j)}_{r:\tau}\nabla\bar{V}_\tau(X^{(j)}_\tau)+c^{(j)}_{r:\tau},\qquad r\in[s,\tau].\tag{50}$$

Evaluating at s proves equation 23.

The coefficients depend only on the fixed path and the reference drift and running cost, so they are unchanged when the continuation network is updated. Once they have been integrated backward during collection, replay requires only the new endpoint gradient and a matrix-vector product. The endpoint value is also refreshed in the exponential weights, while the stored $\mathcal{W}^{(j)}_{s:\tau}$ and $\Gamma^{\bar{u},(j)}_{s:\tau}$ are retained. The latter remains the likelihood correction for the policy that generated the branch.

The identity is pathwise and continuous in time. A dense $K^{(j)}_{s:\tau}$ needs $O(d^2)$ storage and replay work. Evaluating its matrix ODE uses matrix-matrix products, with $O(d^3)$ cost per evaluation under ordinary dense multiplication; structured Jacobians can reduce this cost.

B.6 Approximation and Replay

The continuous-time identities assume exact expectations and continuation functions. Finite branch counts introduce sampling error, and replacing $V(\tau,\cdot)$ by $\bar{V}_\tau$ introduces bootstrap error. Appendix C discusses finite-sample bias and continuation error. Numerical integration of the state and sensitivity equations introduces additional solver error; Appendix D describes the implementation conventions. The affine representation itself is exact for each fixed continuous trajectory.

The value and gradient errors of the continuation have distinct effects. On the same path, their exponential weight ratio is

$$\frac{e^{-(\mathcal{W}_{s:\tau}+\bar{V}_\tau(X_\tau))/\lambda+\Gamma^{\bar{u}}_{s:\tau}}}{e^{-(\mathcal{W}_{s:\tau}+V(\tau,X_\tau))/\lambda+\Gamma^{\bar{u}}_{s:\tau}}}=e^{-(\bar{V}_\tau(X_\tau)-V(\tau,X_\tau))/\lambda}.\tag{51}$$

The corresponding change in the starting adjoint is $K_{s:\tau}[\nabla\bar{V}_\tau(X_\tau)-\nabla V(\tau,X_\tau)]$. Thus small value error alone does not control the gradient error. Proposition 2 applies to the exact continuation, not to an arbitrary learned endpoint.

Each historical bundle retains its own collection-time $\Gamma^{\bar{u},(j)}_{s:\tau}$, so the corrected conditional expectations always refer to the same reference process. There is no need to change measure to the current policy. Replaying changes the distribution of starting states; coverage of the relevant time-state region is still needed. Branches must stay grouped by their common start when forming the log and ratio targets. Repeated updates on a finite buffer do not supply new independent branches, and the variance theorem does not describe the uncertainty of the whole training process.

C Finite-Sample Targets and Continuation Error

The identities in Appendix B describe population expectations with an exact continuation. Here we distinguish finite sampling, continuation approximation, and numerical integration. We also clarify how the likelihood correction and the connection to Adjoint Matching apply to these targets.

C.1 A Fixed Continuation and Monte Carlo Consistency

Fix a starting pair (s, x), an endpoint $s < \tau \le T$, and a differentiable continuation function v. Under the regularity and integrability assumptions in Appendix B.1, define

$$\mathcal{B}_{s:\tau} v(x) = -\lambda \log \mathbb{E}^{\text{base}}_{s,x}\left[e^{-(\mathcal{W}_{s:\tau}+v(X_\tau))/\lambda}\right]. \tag{52}$$

Let $\widetilde{a}^v_s$ be the starting lean adjoint with terminal condition $\nabla v(X_\tau)$. For a frozen collection policy, write

$$w^v = \exp\left[-\frac{\mathcal{W}_{s:\tau} + v(X_\tau)}{\lambda} + \Gamma^{\bar{u}}_{s:\tau}\right].$$

Assume independent, identically distributed branches conditional on the start and policy, $0 < \mathbb{E}[w^v] < \infty$, and $\mathbb{E}[w^v \|\widetilde{a}^v_s\|] < \infty$. The strong law gives

$$\begin{aligned} \frac{1}{M}\sum_{j=1}^{M} w^v_j &\longrightarrow e^{-\mathcal{B}_{s:\tau} v(x)/\lambda}, \\ \frac{\sum_{j=1}^{M} w^v_j \widetilde{a}^{v,(j)}_s}{\sum_{j=1}^{M} w^v_j} &\longrightarrow \nabla_x \mathcal{B}_{s:\tau} v(x) \qquad \text{almost surely.} \end{aligned} \tag{53}$$

Thus the empirical targets converge to the value and starting optimal control of the continuous-time problem on $[s, \tau]$ with terminal cost v, provided HJB verification applies to that problem. Taking $v = V(\tau, \cdot)$ recovers the full-horizon value and control in Proposition 1.

This statement concerns an increasing number of independent branches. Repeated optimizer updates on a fixed replay buffer do not increase that number. A continuation fitted adaptively using the same branches also requires additional statistical assumptions for a corresponding consistency statement.

C.2 Finite-Sample Bias and Continuation Error

For finite M, Jensen's inequality gives

$$\mathbb{E}[\widehat{V}(s, x)] \ge -\lambda \log \mathbb{E}[w^v] = \mathcal{B}_{s:\tau} v(x),$$

whenever the expected logarithm is well defined. The self-normalized control target is generally biased as well, with no universal sign. These biases are separate from continuation error and numerical integration error. With normalized weights $\omega_j = w^v_j / \sum_{\ell=1}^{M} w^v_\ell$, the diagnostic $\text{ESS} = 1/\sum_j \omega_j^2$ measures weight concentration. Short segments alone do not ensure a large effective sample size.

The value operator is nonexpansive in the uniform norm:

$$\|\mathcal{B}_{s:\tau} v_1 - \mathcal{B}_{s:\tau} v_2\|_\infty \le \|v_1 - v_2\|_\infty. \tag{54}$$

Indeed, if $|v_1 - v_2| \le \varepsilon$, the exponential integrands differ by factors between $e^{-\varepsilon/\lambda}$ and $e^{\varepsilon/\lambda}$; taking logarithms proves the bound. The operator is not a strict contraction: adding a constant to a continuation adds the same constant to its propagated value.

Uniform accuracy of a learned scalar function does not by itself control its spatial derivative. In one dimension, $e_\varepsilon(x) = \varepsilon \sin(x/\varepsilon^2)$ has amplitude at most ε and derivative amplitude $1/\varepsilon$. This distinction motivates direct supervision of the gradient-derived control. It is not a convergence guarantee for joint training. Appendix B.6 describes separately how continuation values affect path weights and continuation gradients affect adjoints.

C.3 A Conditional Bound for Continuation Perturbations

The two effects can be separated without making a claim about the optimization dynamics. Fix (s, x, τ) and use the reference path law on this interval. For a continuation v, let Q_v be the normalized tilted law with density proportional to $\exp[-(\mathcal{W}_{s:\tau} + v(X_\tau))/\lambda]$. Write $a^v = K_{s:\tau}\nabla v(X_\tau) + c_{s:\tau}$. Then $\nabla \mathcal{B}_{s:\tau} v = \mathbb{E}_{Q_v}[a^v]$. For another continuation $\bar{v} = v + \delta$,

$$\frac{\mathrm{d}Q_{\bar{v}}}{\mathrm{d}Q_v} = \frac{e^{-\delta(X_\tau)/\lambda}}{\mathbb{E}_{Q_v}[e^{-\delta(X_\tau)/\lambda}]}. \tag{55}$$

Subtracting the two gradient representations gives the exact identity

$$\nabla \mathcal{B}_{s:\tau} \bar{v} - \nabla \mathcal{B}_{s:\tau} v = \mathbb{E}_{Q_{\bar{v}}}[K_{s:\tau} \nabla \delta(X_\tau)] + \mathbb{E}_{Q_v}\left[\left(\frac{\mathrm{d}Q_{\bar{v}}}{\mathrm{d}Q_v} - 1\right) a^v\right]. \tag{56}$$

For example, suppose additionally that, on the relevant endpoint support, $|\delta| \le \varepsilon$ and $\|\nabla\delta\| \le \eta$, and that $\mathbb{E}_{Q_{\bar{v}}}\|K_{s:\tau}\|_{\mathrm{op}} \le C_K$ and $\mathbb{E}_{Q_v}\|a^v\| \le C_a$. Equation equation 55 lies between $e^{-2\varepsilon/\lambda}$ and $e^{2\varepsilon/\lambda}$. Consequently,

$$\|u^{\bar{v}}(s,x) - u^v(s,x)\| \le \frac{\|\sigma_s^\top\|_{\mathrm{op}}}{\lambda}\left[C_K \eta + C_a\left(e^{2\varepsilon/\lambda} - 1\right)\right], \tag{57}$$

where $u^v = -\lambda^{-1}\sigma_s^\top \nabla \mathcal{B}_{s:\tau} v$. This is a population bound under the stated additional assumptions, not a finite-sample or neural-training guarantee. The first term is transported gradient error; the second is the change in the tilted path distribution. A constant δ changes the value by that constant but leaves the control unchanged, so the bound need not be sharp. Small scalar value error alone does not bound η.

Compatibility of the two targets. For a fixed continuation, the population targets are the value $\mathcal{B}_{s:\tau} v$ and its gradient-derived control. This is the compatibility used by a shared scalar network. A finite bundle does not in general make the empirical control target the derivative of a differentiated off-policy Monte Carlo value estimator: changing the start can change the sampling paths and likelihood factors. Training instead treats both constructed targets as fixed labels. The compatibility statement is therefore about the conditional population operator and its Monte Carlo limit.

C.4 Likelihood Corrections and Adjoint Matching

For any integrable path functional F, the valid collection-time correction satisfies

$$\mathbb{E}^{\mathrm{base}}_{s,x}[F] = \mathbb{E}^{\bar{u}}_{s,x}\left[e^{\Gamma^{\bar{u}}_{s:\tau}} F\right].$$

Applying this identity to the exponential path weight and its product with the lean adjoint gives the two moments used by PI-VCM. The sensitivity is defined using the reference drift before the change of measure. Evaluating it on a collected path therefore uses the reference drift Jacobian without differentiating the collection policy. For redundant noise coordinates, we use the driving-noise interpretation in Appendix B.1.

AM and PI-VCM share this adjoint structure. AM uses full-horizon trajectories, terminal gradient ∇g, and unweighted control regression under the frozen current policy. PI-VCM uses a continuation gradient at the segment endpoint, exponential cost weights with a local likelihood correction, and a value loss in addition to the control loss. Their finite-iteration targets therefore differ, even though the population control target of PI-VCM agrees with that of SOCM-Adjoint under exact continuation.

Table 5: Supervision and target construction. AM refers to its original on-policy update; importance correction can be introduced in extensions.

	AM	PI-VM	PI-VCM
Value supervision	No	Yes	Yes
Direct control supervision	Yes	No	Yes
Label horizon	To terminal time	Local	Local
Local likelihood-corrected replay	Not in base update	Yes	Yes
Learned output	Control	Value	Value

For a constant collection control q, the driving-noise likelihood ratio $R = e^{\Gamma^q_{s:\tau}}$ satisfies

$$\mathbb{E}^q[R] = 1, \qquad \mathrm{Var}_{\mathbb{P}^q}(R) = e^{\|q\|^2(\tau - s)} - 1.$$

This follows from the Gaussian moment-generating function and illustrates why a shorter correction interval can help. It does not give a universal variance ordering when costs, continuations, or collection policies differ. Proposition 2 provides the precise comparison for a fixed policy and exact continuation. Importance correction also requires adequate path coverage; it cannot reconstruct modes absent from a finite buffer.

D TRAINING AND IMPLEMENTATION CONVENTIONS

Algorithm 1 specifies the continuous-time training procedure. This appendix explains target differentiation, normalization, cache validity, and the distinction between the continuous-time identities and their numerical implementation.

Algorithm 1 Path-Integral Value-Control Matching with affine replay

Input: SOC problem; branch count M; batch sizes B, B_u; weights β_k; a rule for choosing $s < \tau \le T$.
Initialize: Network V_θ; collect initial branch bundles, including terminal-ending segments, under $\bar{u} = 0$ and cache them in $\mathcal{D}$.
for each training iteration k **do**
  if collection is scheduled **then**
    Freeze $\bar{u} = u_{\theta_k}$; select starting pairs (s, x) and endpoints $\tau \in (s, T]$.
    for each selected (s, x, τ) **do**
      Sample M independent trajectories of equation 1 under $\bar{u}$ on $[s, \tau]$.
      Evaluate $\mathcal{W}^{(j)}_{s:\tau}$ and $\Gamma^{\bar{u},(j)}_{s:\tau}$ along each branch.
      Integrate equation 22 backward to obtain $K^{(j)}_{s:\tau}, c^{(j)}_{s:\tau}$.
      Store the bundle equation 24; evict the oldest bundles if capacity is exceeded.
    end for
  end if
  Sample B complete bundles from $\mathcal{D}$; select B_u starts for control matching.
  Refresh endpoint values and gradients using V_{θ_k}; use $g, \nabla g$ at T.
  Refresh adjoints by equation 23, retaining the stored collection corrections.
  Form and detach the targets in equation 20 and the loss scales.
  Update θ by minimizing equation 17.
end for
Output: V_θ and $u_\theta = -\lambda^{-1}\sigma^\top \nabla_x V_\theta$.

Target construction and differentiation. Every replay bundle contains branches with a common start (s, x). Endpoint values and gradients are evaluated using the parameters before the optimizer step. At a nonterminal endpoint, automatic differentiation computes $\nabla_x V_{\theta_k}(\tau, X_\tau)$; the endpoint value and gradient are then detached. Terminal endpoints use g and ∇g. The prediction $u_\theta = -\lambda^{-1}\sigma^\top \nabla_x V_\theta$ at the start retains the graph needed to differentiate the control loss with respect to θ. Detaching this prediction's spatial gradient would remove the intended control supervision. Both targets are formed using the same branch bundle and the same refreshed exponential weights.

Stable evaluation of one bundle. Let v_j and p_j be the detached endpoint value and gradient, and abbreviate the stored cost and correction by W_j, Γ_j. For each branch compute

$$\ell_j = -(W_j + v_j)/\lambda + \Gamma_j, \qquad q = \max_j \ell_j, \qquad r_j = e^{\ell_j - q}.$$

Then the numerically stable formulas are

$$\begin{aligned} \widehat{V} &= -\lambda \left[q + \log\left(\sum_j r_j \right) - \log M \right], \\ \widehat{u} &= -\lambda^{-1} \sigma_s^\top \sum_j \frac{r_j}{\sum_k r_k} (K_j p_j + c_j). \end{aligned} \tag{58}$$

The division by M belongs inside the value logarithm; the control uses normalized weights. The stabilization subtracts the same constant on every branch and changes neither mathematical target. It does not correct poor path coverage or weight concentration. Nonfinite costs, corrections, or endpoint evaluations indicate an invalid numerical target and should be diagnosed before an update.

Terminal anchoring. Terminal-ending bundles use the prescribed g and ∇g, independently of the current network. Such bundles provide boundary information to the recursive targets. They

do not alone enforce $V_\theta(T,x) = g(x)$ for every x. If an exact network boundary is desired, a compatible parameterization is $V_\theta(t,x) = g(x) + (T-t)F_\theta(t,x)$, for differentiable g, F_θ. This is an implementation option, not an assumption needed for the population recursion or a claim that all experiments use that parameterization.

Normalization and optimization. One target-RMS normalization compatible with equation 17 is

$$\begin{aligned} s_V &= \mathrm{sg}\left[\max\left\{\varepsilon_V, \sqrt{B^{-1}\sum_{i\in\mathcal{I}} \widehat{V}_i^2}\right\}\right], \\ s_u &= \mathrm{sg}\left[\max\left\{\varepsilon_u, \sqrt{(B_u d_u)^{-1}\sum_{i\in\mathcal{I}_u} \|\widehat{u}_i\|^2}\right\}\right], \end{aligned} \tag{59}$$

with positive floors. Fixed scales are also possible; their choice determines the relative weighting of the two terms together with β_k. These scales are separate from network input and output normalization. A value-only warm-up sets $\beta_k = 0$ initially; the control term and its targets can then be omitted rather than evaluated with an empty control batch. For control microbatches, accumulating correctly normalized losses before one optimizer step preserves the objective when targets and scales remain fixed across the microbatches. Any gradient clipping is applied to the accumulated joint gradient.

Cache validity and computational cost. Appendix B.5 proves the continuous-time affine identity used to refresh adjoints. Dense coefficients require $O(Md^2)$ storage per bundle and $O(Md^2)$ work to apply them to new endpoint gradients. With ordinary dense multiplication, evaluating the matrix ODE during collection costs $O(Md^3)$ per solver evaluation. Structured Jacobians can reduce these costs. Caching trades storage and collection-time computation for cheaper target refreshes.

Updating the continuation network or the current collection policy leaves an existing bundle valid with its original correction. Changing the running cost invalidates the stored accumulated cost and the inhomogeneous sensitivity term. Changing the reference dynamics invalidates the stored likelihood correction or sensitivities, or both. We therefore clear and repopulate affected buffers after such changes. If complete trajectories and the necessary sampling information are retained, explicit recomputation may be possible. A terminal-cost-only change can be handled by refreshing terminal values and gradients.

For smoothing, the observation path determines the transformed coefficients, so a cache belongs to that observation-dependent problem. For mean-field games, each outer update may change the coefficients of the inner SOC problem. Parameters can initialize the next inner solve, while affected replay records must be recollected.

Numerical integration. The state SDE, accumulated cost, likelihood correction, and affine sensitivity ODEs require numerical integration. For example, an Euler–Maruyama implementation uses steps of length Δt, with the endpoint chosen on the grid. For square nonsingular diffusion, the proposal and its discrete reference likelihood satisfy

$$\begin{aligned} X_{n+1} &= X_n + [b_{t_n}(X_n) + \sigma_{t_n}\bar{u}_{t_n}(X_n)]\Delta t + \sigma_{t_n}\sqrt{\Delta t}\,\xi_n, \qquad \xi_n \sim \mathcal{N}(0, I), \\ \Gamma^{\Delta} &= \sum_{n=0}^{N-1}\left[-\sqrt{\Delta t}\,\bar{u}_{t_n}(X_n)^\top \xi_n - \frac{\Delta t}{2}\|\bar{u}_{t_n}(X_n)\|^2\right]. \end{aligned} \tag{60}$$

The likelihood ratio follows directly from the Gaussian transition densities. It is exact for the discrete reference chain. With redundant noise coordinates, the corresponding expression is an importance weight on the augmented noise space. Changing the diffusion, clipping increments, or changing the transition rule requires the correction for the actual sampling law.

For the same Euler reference map and left-endpoint cost quadrature, put $A_n = I + \Delta t\, D_x b_{t_n}(X_n)^\top$. A compatible discrete sensitivity cache satisfies

$$\begin{aligned} K_N &= I, & c_N &= 0, \\ K_n &= A_n K_{n+1}, & c_n &= A_n c_{n+1} + \Delta t\, \nabla f_{t_n}(X_n), \quad n = N-1, \dots, 0. \end{aligned} \tag{61}$$

Indeed, the discrete reference adjoint obeys $a_n = A_n a_{n+1} + \Delta t \nabla f_{t_n}(X_n)$, so backward induction gives $a_n = K_n p_N + c_n$ for any endpoint gradient p_N. This is exact algebra for the chosen discrete reference map and quadrature. Other discretizations require their own consistent sensitivity update. The Jacobian is still that of b, evaluated on the stored path; it is not the Jacobian of the collection drift.

At a fixed step size, discretized costs and sensitivities approximate the continuous-time targets. The gradient-derived control of a discretized path-integral value is not generally the optimal action of a finite-step control problem restricted to shifted Gaussian transitions. Recovering the continuous-time control requires convergence of the simulated costs and sensitivities as the step size decreases. The pathwise affine identity in the method remains a continuous-time statement; numerical integration introduces a separate solver error.

Equivalent control conventions. Some implementations write the controlled SDE as $\mathrm{d}X = (b + \Sigma v)\,\mathrm{d}t + \sqrt{\lambda}\,\Sigma\,\mathrm{d}W$ with action cost $\|v\|^2/2$. It agrees with the convention of this paper under $\sigma = \sqrt{\lambda}\,\Sigma$ and $u = v/\sqrt{\lambda}$, leaving the value function unchanged. Then $v^\star = -\Sigma^\top \nabla V$, and a collection-time likelihood correction uses $v/\sqrt{\lambda}$ in place of $\bar{u}$. Per-coordinate squared control errors in the v coordinates are λ times the corresponding errors in the u coordinates. Comparisons must use the same convention for the predicted and reference actions; equal values alone do not fix the units of a reported control error.

Experimental settings and evaluation quantities are given in Appendix G.

E Derivation of the Smoothing Reformulation

We derive the coefficients and initial reweighting used in Section A.1. Assume sufficient temporal and spatial regularity of h_t, regularity of the latent diffusion, and finite normalizing constants. Assume also that the exponential martingales below define valid changes of measure. Smooth bounded coefficients and suitable bounded derivatives provide convenient sufficient conditions; unbounded models require separate integrability checks. To construct control sensitivities, the transformed drift and cost must additionally have the spatial derivatives used by the adjoint recursion. The claims are conditional on one observation path in a full-measure set for which the likelihood and transformed expectations are defined. They do not assert a pathwise likelihood for every arbitrary continuous input. The transformed reference drift and likelihood correction must be used together; the original latent reference cannot be substituted in only one of the two training targets.

E.1 A pathwise observation likelihood

Relative to an independent observation Wiener law, the likelihood of an observation path has the formal stochastic-integral representation

$$\mathcal{L}^y(X) = \exp\left[\int_0^T h_t(X_t)^\top R^{-1}\,\mathrm{d}y_t - \frac{1}{2}\int_0^T \|h_t(X_t)\|^2_{R^{-1}}\,\mathrm{d}t\right]. \tag{62}$$

This expression is interpreted via the joint reference construction and disintegration for almost every observation path, rather than as an ordinary Riemann–Stieltjes integral against arbitrary continuous y. Independence of the latent and observation noises gives zero cross variation. Integration by parts and Itô's formula then give

$$\begin{aligned}\int_0^T h_t(X_t)^\top R^{-1}\,\mathrm{d}y_t = \Phi^y_T(X_T) - \Phi^y_0(X_0) \\ - \int_0^T [\partial_t^{\exp}\Phi^y_t + \mathcal{L}_t\Phi^y_t](X_t)\,\mathrm{d}t - \int_0^T q_t(X_t)^\top \mathrm{d}W_t,\end{aligned} \tag{63}$$

where $q_t = \sigma_t^\top \nabla \Phi^y_t$. Only explicit time derivatives of h_t occur. No classical derivative of the noisy observation path is taken.

Let P^{base} be the latent law with initial distribution μ_0, and define a transformed reference law Q^y by

$$\frac{\mathrm{d}Q^y}{\mathrm{d}P^{\mathrm{base}}} = \exp\left[-\int_0^T q_t(X_t)^\top \mathrm{d}W_t - \frac{1}{2}\int_0^T \|q_t(X_t)\|^2\,\mathrm{d}t\right]. \tag{64}$$

The change of measure is required to be normalized conditional on each initial state, so Q^y retains initial law μ_0. Its drift is $b - a\nabla\Phi^y = \bar{b}^y$. Combining Eqs. equation 62–equation 64 gives

$$\mathcal{L}^y(X)\,\mathrm{d}P^{\mathrm{base}}(X) = e^{-\Phi_0^y(X_0)}\exp\left[-\int_0^T c_t^y(X_t)\,\mathrm{d}t - g^y(X_T)\right]\,\mathrm{d}Q^y(X), \tag{65}$$

with c^y, g^y from Eq. equation 25. In particular, the negative quadratic term in c^y comes from the Girsanov exponential and is necessary for the identity.

E.2 Posterior drift and initial law

Define the future normalizer under the transformed reference,

$$Z^y(s,x) = \mathbb{E}_{s,x}^{Q^y}\left[\exp\left(-\int_s^T c_r^y(X_r)\,\mathrm{d}r - g^y(X_T)\right)\right], \qquad V^y = -\log Z^y.$$

The normalized future tilt has drift

$$\bar{b}^y + a\nabla\log Z^y = \bar{b}^y - a\nabla V^y = b - a\nabla(V^y + \Phi^y).$$

Integrating Eq. equation 65 over future trajectories conditional on $X_0 = x$ gives

$$\pi_0^y(\mathrm{d}x) = \frac{e^{-\Phi_0^y(x)}Z^y(0,x)\,\mu_0(\mathrm{d}x)}{\int e^{-\Phi_0^y(z)}Z^y(0,z)\,\mu_0(\mathrm{d}z)}.$$

These are exactly the drift and initial law in Eq. equation 28 with $V_\theta^y = V^y$. Their combination recovers the posterior path law. For deterministic X_0, the initial factor is constant on its support; otherwise it changes the initial posterior distribution. Replacing V^y with a learned approximation and using a finite-step simulator introduces approximation errors in addition to those of the SOC training targets.

Verification of the conditional path law. When Z^y is positive and $C^{1,2}$, its backward equation implies, under Q_x^y,

$$M_t = \frac{e^{-\int_0^t c_r^y(X_r)\,\mathrm{d}r}Z^y(t,X_t)}{Z^y(0,x)}, \qquad \frac{\mathrm{d}M_t}{M_t} = \sigma_t^\top\nabla\log Z^y(t,X_t)\,\mathrm{d}W_t.$$

By the Markov property, M_t is the conditional expectation of the normalized terminal cost tilt, so it is a true martingale. Girsanov's formula therefore adds $a\nabla\log Z^y$ to the reference drift and produces precisely the conditional posterior tilt. This proves the drift assertion at the path-law level.

The same posterior has a variational characterization. If P is the original latent law, $\Pi^y = \mathcal{L}^y P/\mathcal{Z}^y$, and $Q \ll P$ has finite terms below, then

$$\mathrm{KL}(Q\|P) - \mathbb{E}_Q\log\mathcal{L}^y = -\log\mathcal{Z}^y + \mathrm{KL}(Q\|\Pi^y).$$

This follows by taking the logarithm of the Radon–Nikodym densities and integrating under Q. Nonnegativity of KL gives the unique minimizing path law Π^y. For random initial states, this optimization includes the initial marginal; constraining it to μ_0 would generally solve a different problem.

Terminal observations and cost centering. An additional independent terminal observation with negative log-likelihood $\ell_T(x)$ replaces g^y by $\ell_T - \Phi_T^y$; the preceding derivation is unchanged. Reaction–diffusion uses $\ell_T(x) = \|x - z_T\|^2/(2\,0.1)$. Also, replacing $c_t^y(x)$ by $c_t^y(x) + k_t^y$, where k^y is time dependent but state independent, gives

$$\widetilde{Z}^y(s,x) = e^{-\int_s^T k_r^y\,\mathrm{d}r}Z^y(s,x), \qquad \widetilde{V}^y(s,x) = V^y(s,x) + \int_s^T k_r^y\,\mathrm{d}r.$$

It leaves the posterior drift and normalized initial weights unchanged. This justifies removing the observation-only running-cost term in the reaction–diffusion implementation while accounting for the corresponding value offset.

Joint role of the initial law and the controlled drift. For a fixed initial state x, the conditional posterior density relative to the transformed reference is

$$\frac{\mathrm{d}\Pi_x^y}{\mathrm{d}Q_x^y}(X_{0:T}) = \frac{\exp[-\int_0^T c_t^y(X_t)\,\mathrm{d}t - g^y(X_T)]}{Z^y(0,x)}. \tag{66}$$

Multiplying this conditional density by $\pi_0^y(\mathrm{d}x)$ cancels $Z^y(0,x)$ and recovers the normalized joint tilt in Eq. equation 65. This verifies the entire posterior path law, including its initial marginal. Using the exact controlled drift but starting from μ_0 instead of π_0^y would generally leave the initial marginal incorrect.

If $h_t \equiv 0$ and there is no terminal observation, then $\Phi^y = c^y = g^y = 0$, $Z^y = 1$, and $V^y = 0$. The formulas reduce to the original latent drift b and initial distribution μ_0, as required when the observations convey no state information. These identities also explain why a constant offset in a learned value cancels in the normalized initial weights, whereas a state-dependent value error can change the initial posterior and its gradient can change the drift.

F Mean-Field Optimality and Population Consistency

For a fixed population flow m, the HJB equation is

$$\partial_t V^m + (b^m)^\top \nabla V^m + \frac{1}{2}\operatorname{Tr}(a_t \nabla^2 V^m) + f^m - \frac{1}{2\lambda}\|\sigma_t^\top \nabla V^m\|^2 = 0, \qquad V^m(T,x) = g(x, m_T). \tag{67}$$

The optimal control is $u^{m,\star} = -\lambda^{-1}\sigma^\top \nabla V^m$. If the response has a density ρ_t, its forward equation is

$$\partial_t \rho_t = -\nabla\cdot\left[\left(b_t(\cdot, m_t) - \lambda^{-1} a_t \nabla V^m(t,\cdot)\right)\rho_t\right] + \frac{1}{2}\sum_{i,j}\partial^2_{x_i x_j}(a_{t,ij}\rho_t), \qquad \rho_0 = \mu_0. \tag{68}$$

For general measures this equation is interpreted weakly. At equilibrium, $\rho = m$, giving the coupled HJB–Fokker–Planck system. The finite-horizon initial and terminal conditions are both required. The best-response map is taken to be single valued; otherwise an explicit selection rule is needed. Specifically, for $\varphi \in C_c^2(\mathbb{R}^d)$, the weak forward equation is

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int \varphi(x)\,\rho_t(\mathrm{d}x) = &\int (b_t(x, m_t) - \lambda^{-1} a_t \nabla V^m(t,x))^\top \nabla\varphi(x)\,\rho_t(\mathrm{d}x) \\ &+ \tfrac{1}{2}\int \operatorname{Tr}(a_t \nabla^2\varphi(x))\,\rho_t(\mathrm{d}x),\end{aligned} \tag{69}$$

in the distributional sense in time, and pointwise whenever the derivative exists, under the required integrability. It follows by Itô's formula under the best-response dynamics. The candidate m determines the coefficients while ρ is the resulting law; only a fixed point permits identifying them.

Verification of individual optimality. Suppose V^m is a classical HJB solution and the admissible control u has the integrability needed for Itô's formula and for its stochastic integral to have zero expectation. Completing the square in the Hamiltonian gives, along its controlled trajectory,

$$\partial_t V^m + (b^m + \sigma u)^\top \nabla V^m + \tfrac{1}{2}\operatorname{Tr}(a\nabla^2 V^m) + f^m + \tfrac{\lambda}{2}\|u\|^2 = \tfrac{\lambda}{2}\|u - u^{m,\star}\|^2.$$

Integration, the terminal condition, and averaging over μ_0 yield

$$J^m(0,\mu_0;u) - \int V^m(0,x)\,\mu_0(\mathrm{d}x) = \frac{\lambda}{2}\mathbb{E}^{u,m}_{\mu_0}\int_0^T \|u_t(X_t) - u^{m,\star}(t,X_t)\|^2\,\mathrm{d}t. \tag{70}$$

When $u^{m,\star}$ is admissible, its right-hand side vanishes, establishing the best-response value. This identity fixes m throughout. It does not compare m with the population generated by u, nor does it identify the OU evaluation error under optimal trajectories with an objective gap under learned trajectories.

A sufficient condition for the outer iteration. The response map in the following bound is the best-response distribution map defined in Section A.2. Suppose the exact response map Γ is a contraction with factor $q < 1$ on a complete invariant set of flows in a norm $\|\cdot\|$, and the set is closed under the mixtures in Eq. equation 31. For a constant relaxation $\gamma \in (0, 1]$, the exact update satisfies

$$\|m^{k+1} - m^\star\| \le [1 - \gamma(1 - q)]\|m^k - m^\star\|.$$

If the estimated response obeys $\|\widehat{\Gamma}_k(m^k) - \Gamma(m^k)\| \le \delta_k$, then

$$\|m^{k+1} - m^\star\| \le [1 - \gamma(1 - q)]\|m^k - m^\star\| + \gamma\delta_k. \tag{71}$$

To obtain the bound, subtract $m^\star = (1 - \gamma)m^\star + \gamma\Gamma(m^\star)$, add and subtract $\gamma\Gamma(m^k)$, and apply the triangle inequality and contraction assumption. For the inexact iteration, assume also that the iterates remain in the set where these hypotheses hold. Writing $\alpha = 1 - \gamma(1 - q) < 1$, induction gives

$$\|m^k - m^\star\| \le \alpha^k\|m^0 - m^\star\| + \gamma \sum_{j=0}^{k-1} \alpha^{k-1-j}\delta_j. \tag{72}$$

Thus $\delta_k \to 0$ implies convergence under these contraction assumptions, while $\delta_k \le \bar{\delta}$ yields an asymptotic error bound $\bar{\delta}/(1 - q)$. Neural fitting error and population Monte Carlo error both contribute to δ_k. The contraction assumption is not automatic for mean-field games, especially for non-potential interactions, and is not established by a small inner training loss. Relaxation alone does not supply it.

Interpreting equilibrium diagnostics. For a candidate control u and population m, an ideal best-response gap is

$$\mathrm{Gap}(u; m) = J^m(0, \mu_0; u) - \inf_v J^m(0, \mu_0; v), \qquad J^m(0, \mu_0; u) = \int J^m(0, x; u)\, \mu_0(\mathrm{d}x).$$

Here $J^m(0, x; u)$ is the expected cost in Eq. equation 29 before minimizing over the control. Population consistency instead compares m to the law induced by u. These quantities test different conditions: a best response to an inconsistent population is not an equilibrium, and a self-consistent population can be generated by suboptimal controls. Empirical gaps, Bellman diagnostics, and distribution errors are approximations to these requirements, and their definitions and evaluation references must be specified for each benchmark.

G Experimental Setup and Evaluation

This appendix gives the basic experimental settings and defines the quantities reported in Section 5.

G.1 Reporting Convention

All reported errors are minimized. Bold entries mark the lowest mean, without implying a significance test. Dashes indicate unavailable entries; displayed zeros can result from rounding. For repeated-run scores $e_1, \ldots, e_R$, we use

$$\bar{e} = \frac{1}{R}\sum_{r=1}^{R} e_r, \qquad s_e = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}(e_r - \bar{e})^2}. \tag{73}$$

The sample standard deviation s_e describes variation across runs and differs from Monte Carlo uncertainty within a run.

G.2 Basic Experimental Settings

Table 6 gives the principal simulation settings and PI-VCM budgets. An outer-by-inner budget denotes population iterations and optimizer updates per response.

Table 6: Basic simulation settings and PI-VCM training budgets. N denotes simulation steps over T. Two-exit evaluation uses 144 steps instead of the 36-step training grid.

Task	Dimension	T	N	PI-VCM updates
OU Linear	10	1	100	60,000
Quadratic OU Easy / Hard	20	1	50	60,000 / 80,000
Reaction–diffusion	20	1	1000	10,000
Double well	2	2.5	250	6,000
Lorenz–63	3	1	200	10,000
Transport–congestion	1	1	20	8×256
Non-potential LQ	2	1	16	40×500
Two-exit routing	2	1.8	36	24×512

OU tasks. All OU tasks use $\lambda = 1$ and $X_0 \sim \mathcal{N}(0, 0.25I)$. OU Linear has $A = -I + 0.1G$, $\sigma = I + 0.1G$, $f = 0$, and $g(x) = \mathbf{1}^\top x$, with one common matrix of independent standard Gaussian entries G. Quadratic OU uses $\sigma = I$, $f = x^\top Px$, and $g = x^\top Qx$, with $(A, P, Q) = (0.2I, 0.2I, 0.1I)$ for Easy and $(I, I, 0.5I)$ for Hard. The dimension comparison keeps the Easy coefficients and varies $d \in \{10, 20, 80, 150, 200\}$.

Smoothing tasks. Methods share the observation path and reference within each task. Reaction–diffusion observes all 20 components and has an additional terminal observation. Double well observes a quadratic function of its first component and a linear function of its second. Lorenz–63 uses the physical parameters $(10, 28, 8/3)$, observes only its first component, and normalizes physical coordinates by $(20, 30, 40)$. The endpoints use 10,000 posterior paths per run.

Local sampling and population updates. For the three OU tasks and the reaction–diffusion, transport–congestion, and two-exit settings in Table 6, PI-VCM uses $M = 8$ branches and maximum segment length $H = 8$; double well and Lorenz–63 use $(M, H) = (8, 16)$. These settings use normalized control-loss weight $\beta = 3$, except for reaction–diffusion, which uses $\beta = 30$. The two-exit and transport–congestion population iterations use relaxation 0.15. Their physical noise is $\sqrt{2\nu}I$, with $\nu = 0.12$ and 0.5, respectively; non-potential LQ uses $\nu = 0.08$. Implementation-specific optimizer, network, and replay settings are supplied with the code.

Reaction–diffusion uses five repetitions; double well, Lorenz–63, transport–congestion, and non-potential LQ use four; two-exit uses three. The corresponding error bars are sample standard deviations.

Synthetic MFG models. The three synthetic tasks use physical actions a_t with dynamics

$$\mathrm{d}X_t = [b(X_t) + a_t]\,\mathrm{d}t + \sqrt{2\nu}\,\mathrm{d}W_t$$

and running cost $\frac{1}{2}\|a_t\|^2 + f_t(X_t, m_t)$, followed by terminal cost $g(X_T, m_T)$. In the convention of Eq. equation 29, $a_t = \sqrt{2\nu}\,u_t$ and $\lambda = 2\nu$. The model-specific costs and initial laws are given below; horizons and time grids are listed in Table 6.

Transport–congestion. Here $b = 0$, $\nu = 0.5$, and $X_0 \sim \mathcal{N}(-0.75, 0.20^2)$. The target is $x_\star = 0.75$, with costs

$$f(x, m) = \frac{0.25}{2}(x - x_\star)^2 + 6\int_{\mathbb{R}} \exp\left[-\frac{(x-y)^2}{2(0.30)^2}\right] m(\mathrm{d}y),$$
$$g(x, m) = \frac{1.5}{2}(x - x_\star)^2.$$

The unnormalized Gaussian kernel penalizes proximity to the population, while the quadratic terms attract agents to the target.

Non-potential LQ. Here $b(x) = Ax$, $\nu = 0.08$, and $X_0 \sim \mathcal{N}(\bar{x}_0, \Sigma_0)$. Writing $\bar{m}_t = \int x\, m_t(\mathrm{d}x)$, the costs are

$$f_t(x, m_t) = \tfrac{1}{2} x^\top Q x + x^\top [C\bar{m}_t - q(t)], \quad q(t) = q_0 + q_1 \sin(\pi t/T),$$
$$g(x, m_T) = \tfrac{1}{2} x^\top Q_T x - q_T^\top x, \qquad C = \begin{pmatrix} 0.35 & 0.25 \\ -0.25 & 0.35 \end{pmatrix}.$$

The nonsymmetric matrix C gives the non-potential mean coupling. The initial mean and covariance are approximately

$$\bar{x}_0 = \begin{pmatrix} 0.3641 \\ -0.4238 \end{pmatrix}, \qquad \Sigma_0 = \begin{pmatrix} 0.16064 & 0.00257 \\ 0.00257 & 0.17036 \end{pmatrix}.$$

The dense matrices A, Q, Q_T, vectors q_0, q_1, q_T, and initial law are fixed across methods, with $Q, Q_T \succ 0$. Their full numerical coefficients will be supplied with the code.

Two-exit routing. Write the state as $z = (x, y)$. Here $b = 0$, $\nu = 0.12$, and $Z_0 \sim \mathcal{N}((0, -1.55)^\top, \mathrm{diag}(0.28^2, 0.24^2))$. The running state cost is $f(z, m) = \frac{0.35}{2} y^2$. Define the smooth right-exit fraction and terminal costs by

$$p(m_T) = \int_{\mathbb{R}^2} \frac{1}{1 + \exp(-x/0.16)}\, m_T(\mathrm{d}x\, \mathrm{d}y),$$
$$c_R(x, p) = \tfrac{12}{2}(x - 1.45)^2 + 2.8p + 0.45,$$
$$c_L(x, p) = \tfrac{12}{2}(x + 1.45)^2 + 2.8(1 - p),$$
$$g(z, m_T) = \tfrac{5}{2} y^2 - \tau \log\left[\frac{e^{-c_R(x, p(m_T))/\tau} + e^{-c_L(x, p(m_T))/\tau}}{2}\right], \qquad \tau = 0.12.$$

The terminal cost combines attraction to the exits at $(\pm 1.45, 0)$ with a crowding penalty at each exit and a fixed offset for the right exit.

G.3 Stochastic Optimal Control

Control error. Let $(t_i, x_i)_{i=1}^{n_{\mathrm{eval}}}$ denote the preterminal evaluation states along optimal-policy trajectories, and let $u^\star$ be the analytic control in the same coordinates as the learned control. The reported error is

$$\mathcal{E}_u = \frac{1}{n_{\mathrm{eval}} d_u} \sum_{i=1}^{n_{\mathrm{eval}}} \|u_\theta(t_i, x_i) - u^\star(t_i, x_i)\|^2. \tag{74}$$

Thus *Control* $\mathcal{L}_2$ is a mean squared error per control coordinate.

Local-sampling ablation. Table 7 varies branch count $M \in \{4, 8, 16\}$ and maximum segment length $H \in \{4, 8, 16\}$ on OU Linear. The actual segment has $(\tau - s)/\Delta t \le H$ steps. Each cell uses four paired runs at 60,000 updates and reports mean and sample standard deviation; the $(M, H) = (8, 8)$ cell reuses the corresponding comparison runs.

G.4 Bayesian Smoothing

Posterior reconstruction. All methods use the same observation and reference within a task. Ad-hoc performs no training; ELS ad-hoc and ELS variational in the figures denote Ad-hoc and Variational in the table. PI-VCM generates posterior approximations using Eq. equation 28. The reported metrics use the direct, unweighted deployment samples.

Mean and spread. Let $\widehat{\mu}_{i,j}, \widehat{s}_{i,j}$ be the estimated posterior mean and marginal standard deviation, with reference values $\mu^{\mathrm{ref}}_{i,j}, s^{\mathrm{ref}}_{i,j}$. Over the time-coordinate evaluation set $\mathcal{E}$,

$$\mathrm{RMSE}_{\mathrm{mean}} = \left[\frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} (\widehat{\mu}_{i,j} - \mu^{\mathrm{ref}}_{i,j})^2\right]^{1/2},$$
$$\mathrm{RMSE}_{\mathrm{std}} = \left[\frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} (\widehat{s}_{i,j} - s^{\mathrm{ref}}_{i,j})^2\right]^{1/2}. \tag{75}$$

Reaction–diffusion reports these mean and spread errors. Its Std. RMSE is an error in marginal standard deviations.

Marginals and mode probabilities. For scalar distributions with finite first moments, $W_1(P,Q) = \int_{\mathbb{R}} |F_P(z) - F_Q(z)|\,\mathrm{d}z$. Double well and Lorenz–63 average marginal distances as

$$\mathcal{E}_{W_1} = \frac{1}{Ld}\sum_{i=1}^{L}\sum_{j=1}^{d} W_1(\widehat{\pi}_{t_i,j}, \pi^{\mathrm{ref}}_{t_i,j}), \tag{76}$$

excluding the deterministic initial state. Double well also reports

$$\mathrm{MAE}_{\mathrm{mode}} = \frac{1}{L}\sum_{i=1}^{L} |\widehat{\pi}_{t_i}(X_1 > 0) - \pi^{\mathrm{ref}}_{t_i}(X_1 > 0)|. \tag{77}$$

These marginal diagnostics do not characterize the full joint path law.

Lorenz–63 marginal W_1 uses the normalized coordinates in Appendix G.2. Its unobserved-component RMSE uses physical posterior means in X_2, X_3, with the square root taken after averaging squared errors over both time and coordinates. The two columns therefore have different units. The Lorenz–63 table and figures use the common 10,000-update endpoint for trainable methods; double well uses 6,000 updates.

G.5 Mean-Field Games

Best-response gap. For a candidate population m, hold it fixed when computing the response and define

$$\mathrm{Gap}(u; m) = J^m(0, \mu_0; u) - \inf_v J^m(0, \mu_0; v). \tag{78}$$

This is nonnegative for exact costs and an exact optimum. The two-exit and transport–congestion results approximate it using candidate-specific numerical best responses. A signed estimated gap can be affected by discretization, response approximation, and Monte Carlo error. For two-exit routing, the learned control and its numerical response use 2,048 paths, 144 Euler steps, and common initial states and Brownian increments within each comparison.

Discrete LQ advantage. For a frozen population and transition kernel P_n^m, write

$$Q_n^{m,\star}(x,a) = \ell_n^m(x,a) + \mathbb{E}_{P_n^m(\cdot|x,a)} V_{n+1}^{m,\star}(X_{n+1}), \qquad V_n^{m,\star}(x) = \inf_a Q_n^{m,\star}(x,a),$$

where the one-step cost ℓ_n^m includes its time-step factor. Telescoping the continuation values gives

$$\mathbb{E}^{u,m}\sum_{n=0}^{N-1}[Q_n^{m,\star}(X_n, u_n(X_n)) - V_n^{m,\star}(X_n)] = J_\Delta^m(u) - \mathbb{E}_{\mu_0} V_0^{m,\star}(X_0). \tag{79}$$

For the LQ task, the physical action increment is $\Delta t\, a$ and its cost is $\Delta t\|a\|^2/2$. If P_{n+1} is the Hessian of the exact discrete value, the one-step advantage is

$$\frac{\Delta t}{2}(a - a_n^\star(x))^\top (I + \Delta t P_{n+1})(a - a_n^\star(x)).$$

The reported score sums this quantity along learned-policy paths and then averages over paths. Non-potential LQ uses DPI-MGF, the modified DPI method described in Section 5.3.

Population interpretation. A small optimality error concerns a response to the frozen candidate population. An equilibrium also requires that the induced population agree with that candidate, as distinguished in Appendix F. Supplemental particle plots describe distributional shape and do not replace this second requirement.

Lyon density forecasts. The Lyon experiment uses sequences 1B, 2D, and 2E, with a three-second observation window and a two-second forecast horizon. PI-VCM, DEDA-FP, and MFDGM are compared using the relative error of pedestrian intensity fields on a common spatial grid:

$$\mathcal{E}_\rho = \frac{\|\widehat{\rho} - \rho^{\mathrm{obs}}\|_{2,\mathrm{core}}}{\|\rho^{\mathrm{obs}}\|_{2,\mathrm{core}}}. \tag{80}$$

Each observed pedestrian contributes one unit of kernel mass; all forecast paths from one initial pedestrian together contribute one unit. The fields retain their population counts. Relative error means division by the reference-field norm, rather than separately normalizing both fields to unit mass. Evaluation uses a 0.25-m grid; figures use a 0.10-m display grid and 0.30-m Gaussian bandwidth.

The comparison uses three paired repetitions. In the Average column, each repetition first averages its errors over the three sequences; the table then reports the mean and sample standard deviation of those averages. Each sequence has equal weight. These are exploratory results on previously inspected recordings. Forecasts start from pedestrians visible at the observation endpoint; future observations can include arrivals and departures, so density error can reflect both position and population-count differences.

Local-sampling ablation study. To assess sensitivity to local sampling, we compare PI-VCM variants on OU Linear with branch counts $M \in \{4, 8, 16\}$ and maximum segment lengths $H \in \{4, 8, 16\}$ steps. All variants are evaluated with the same control-error metric.

Table 7: Local-sampling ablation for PI-VCM on OU Linear.

	MAX. ROLLOUT STEPS H		
BRANCHES M	4	8	16
4	$0.00000288 \pm 0.00000010$	$0.00000254 \pm 0.00000028$	$0.00000256 \pm 0.00000028$
8	$0.00000284 \pm 0.00000012$	$\mathbf{0.00000248 \pm 0.00000016}$	$0.00000257 \pm 0.00000039$
16	$0.00000287 \pm 0.00000016$	$0.00000253 \pm 0.00000023$	$0.00000251 \pm 0.00000034$

H SUPPLEMENTARY EXPERIMENTAL FIGURES

H.1 BAYESIAN SMOOTHING

Reaction–diffusion. Figure 3 complements the aggregate scores in Table 3 by resolving posterior-mean errors across spatial coordinates and physical time. It shows where reconstruction errors occur within the observation interval.

Joint posterior structure. Figures 4 and 5 complement one-dimensional marginals with joint particle views: separated modes in double well and curved dependence between the unobserved Lorenz–63 components. All panels within each figure share a reference-density background and coordinate range. ELS ad-hoc (or ELS) and ELS variational in the figures correspond to Ad-hoc and Variational in the table. Double well uses the 6,000-update endpoint and Lorenz–63 uses 10,000 updates. The particle and marginal plots pool 10,000 saved paths per run over four runs; the reference contains 16,384 paths.

Marginals and time-resolved errors. Figure 6 compares the terminal Lorenz–63 marginal, while Figure 7 shows how distributional and reconstruction errors vary over time. The latter distinguishes an isolated endpoint discrepancy from an error that persists through the observation interval.

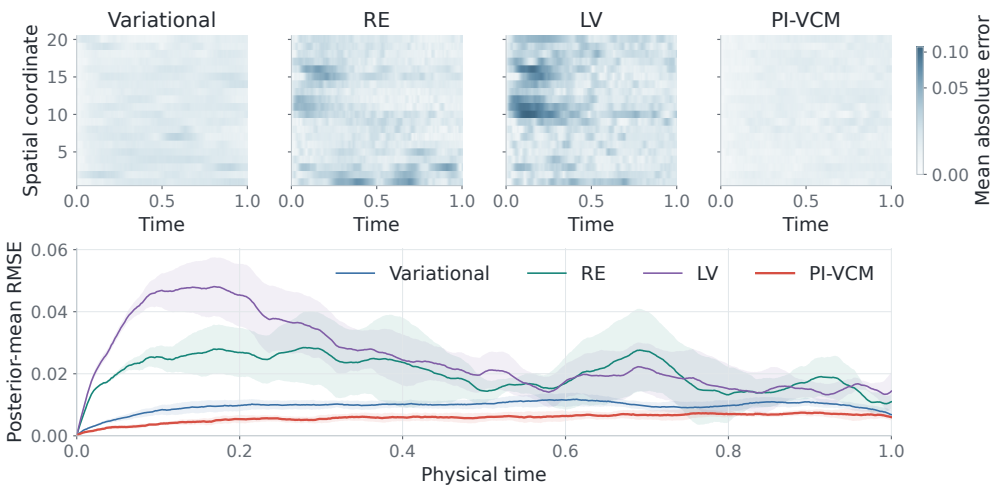

Figure 3: Reaction–diffusion posterior-mean errors for four trained methods. Top: absolute errors averaged over five runs. Bottom: time-resolved RMSE, with the run mean and one sample standard deviation.

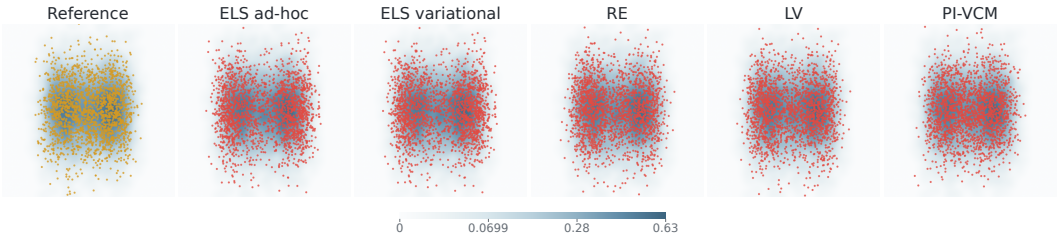

Figure 4: Double-well joint posterior in (X_1, X_2) at $t = 2.5$. Displayed particles overlay the shared reference density; 1,920 particles are shown per panel.

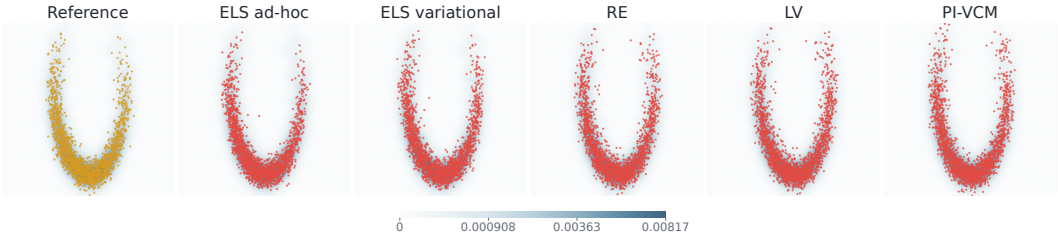

Figure 5: Lorenz–63 joint posterior in physical (X_2, X_3) at $t = 1$. Displayed particles overlay the shared reference density; 1,920 particles are shown per panel.

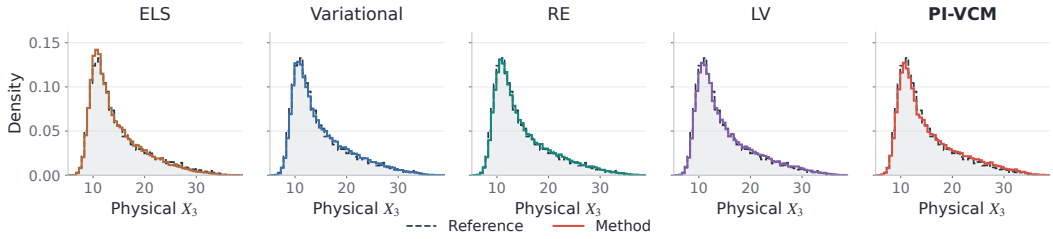

Figure 6: Lorenz–63 terminal posterior of physical X_3. Dashed curves show the reference; panels share bins and axis limits. Histograms are normalized by the full path count and bin width.

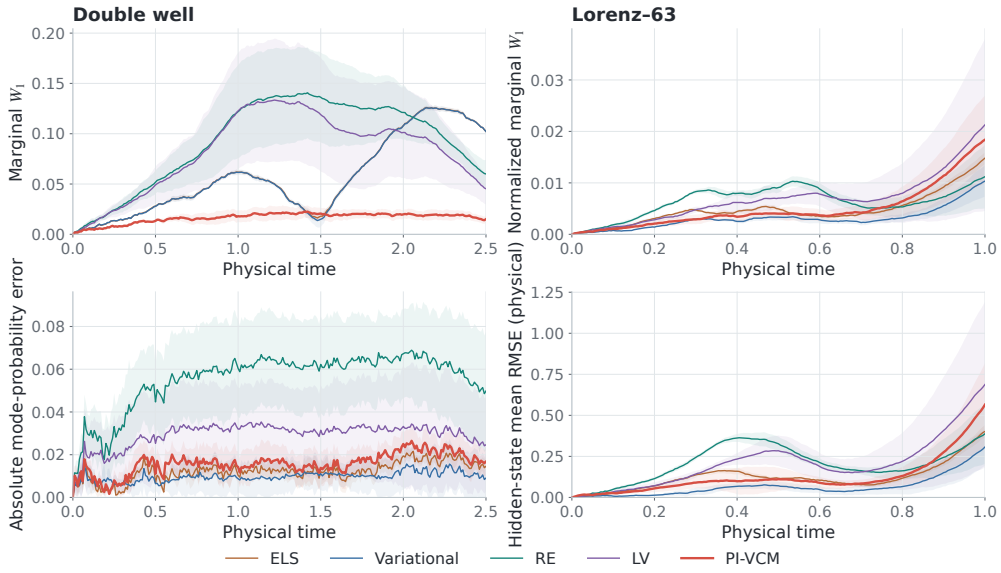

Figure 7: Bayesian smoothing errors over physical time. Lines and shading show the run mean and one sample standard deviation. Lorenz–63 marginal distances use normalized coordinates; hidden-state mean RMSE uses physical coordinates. Its table score takes the square root after averaging squared errors, rather than averaging this RMSE curve over time.

H.2 Mean-Field Games

Two-exit routing. Figure 8 extends the terminal comparison in Figure 2 across time. Figure 9 resolves the terminal modes and their spread. Both use an independent equilibrium reference; these distributional views complement the fixed-population best-response scores and do not by themselves establish equilibrium.

Transport–congestion. Figure 10 extends the terminal comparison in Figure 2 to five times, showing how the population moves and spreads under congestion. This fixed-run comparison uses a shared numerical MFG reference and displays five learned methods.

Non-potential LQ. Figure 11 shows the terminal distributions induced by four displayed controls in both state coordinates. This fixed-run qualitative comparison complements the repeated-run optimality scores in Table 4; it does not include every method in that table.

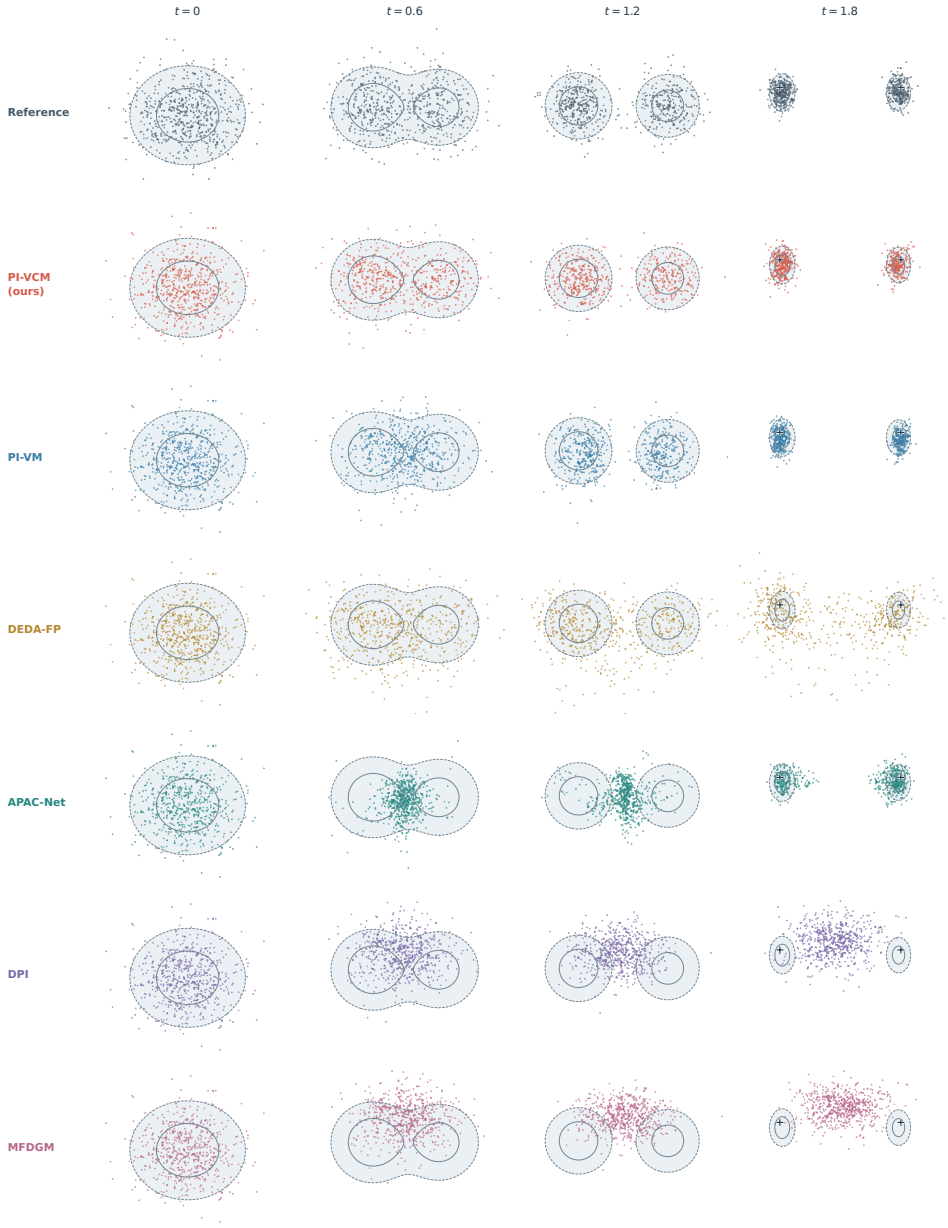

Figure 8: Two-exit population evolution for all six methods. Columns run from initial to terminal time. Shared reference contours enclose 50% and 90% probability; terminal crosses mark the exits.

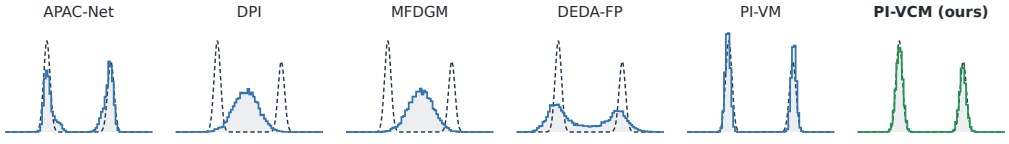

Figure 9: Two-exit terminal position distributions. Solid curves show mean densities across three runs, and dashed curves show the independent equilibrium reference. PI-VCM is highlighted in green. All panels share bins and scales.

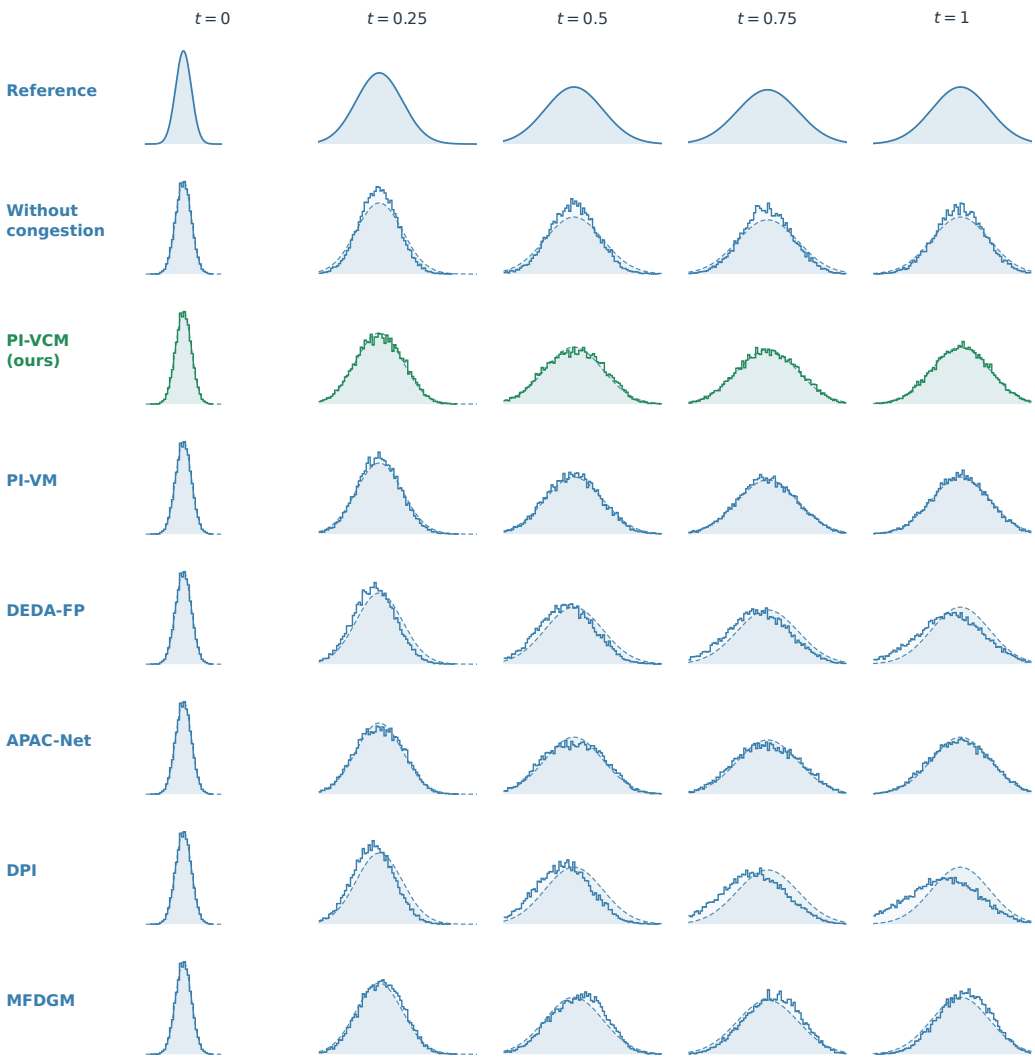

Figure 10: Transport–congestion population densities at $t = 0, 0.25, 0.5, 0.75, 1$. The first two rows show the MFG reference and the population without congestion, respectively. The remaining rows compare six learned methods with the reference. PI-VCM is highlighted in green.

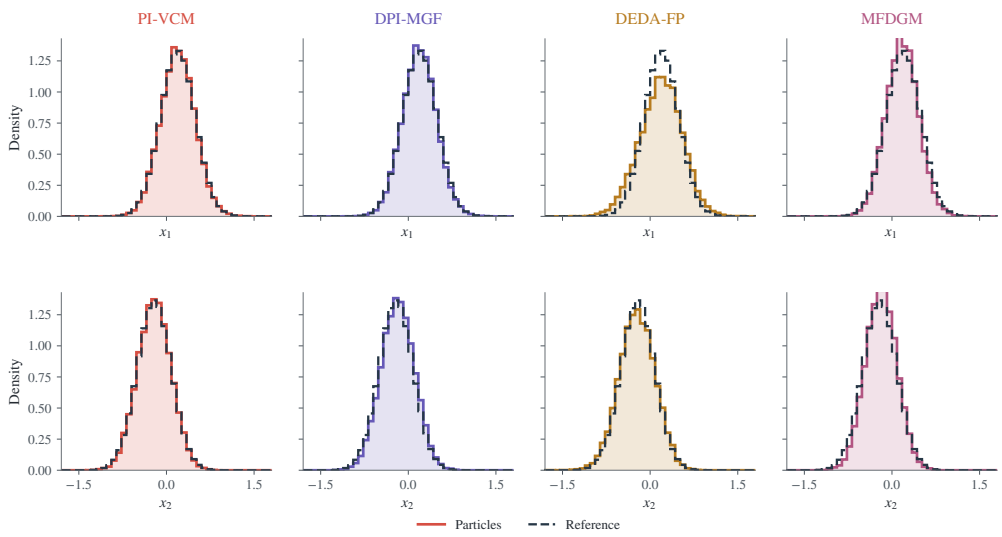

Figure 11: Non-potential LQ terminal marginals at $t = 1$: x_1 (top) and x_2 (bottom). Colored histograms use 16,384 policy-deployment samples per method; dashed curves show the exact 16-step discrete LQ equilibrium reference. Panels share bins and scales.

H.3 Lyon Density Forecasts

Figures 12 and 13 show where the methods place future pedestrian intensity across sequences and over time. They use the same three-second observation window and two-second forecast horizon as Table 4. The maps retain pedestrian counts and share coordinates and a linear density scale within each figure. At the observation endpoint, all methods start from the same observed population.

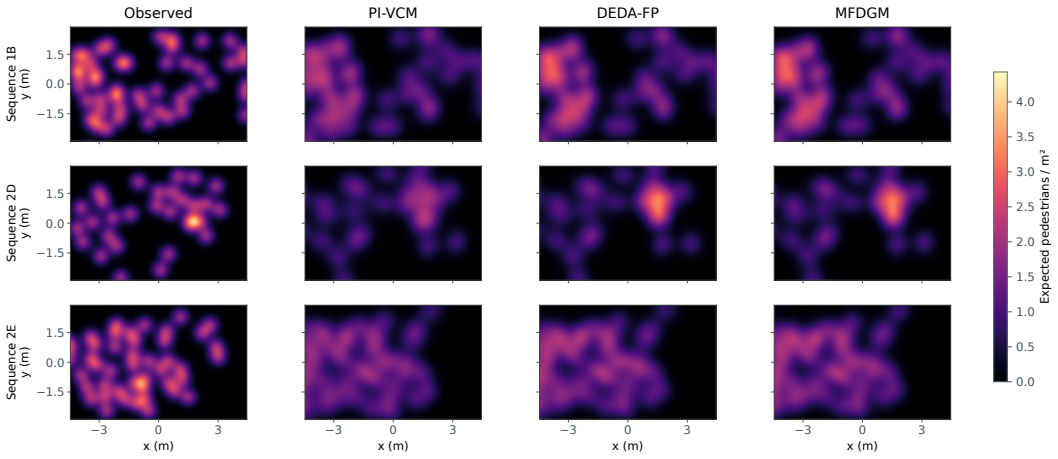

Figure 12: Lyon pedestrian densities on sequences 1B, 2D, and 2E at +2 seconds after three seconds of observation. All panels share a density scale in expected pedestrians per square meter.

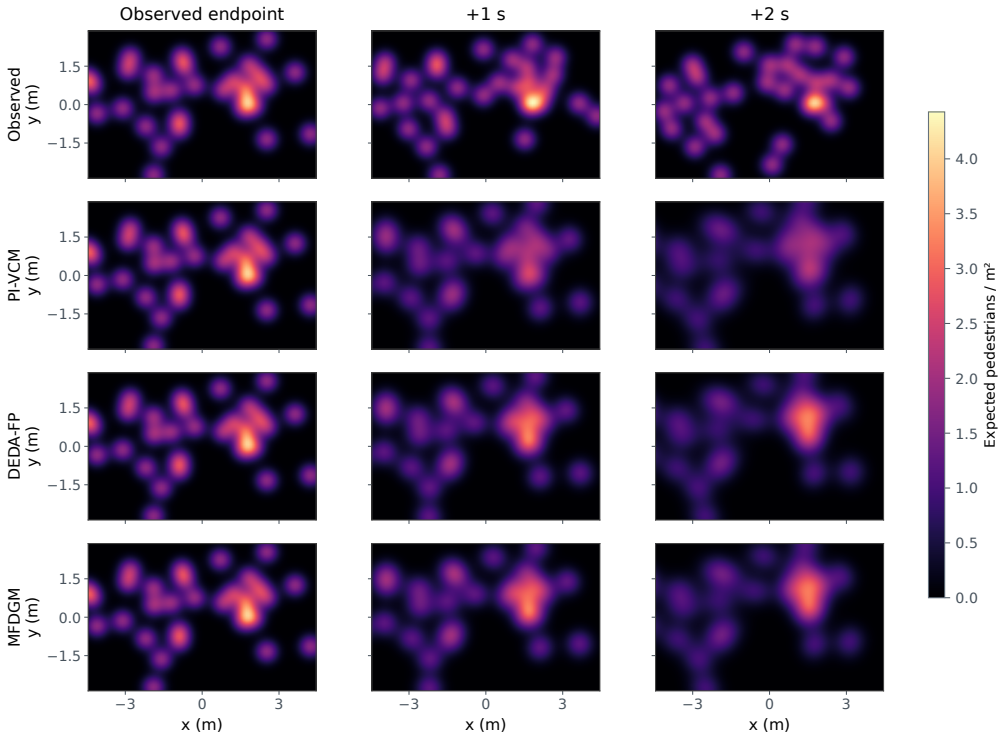

Figure 13: Lyon sequence 2D at the observation endpoint, +1 second, and +2 seconds. The first column repeats the common observed initial population; all panels share one density scale.