Mitigating Compounding Errors in Online Reinforcement Learning via Optimal Transport Regularized Flow Matching

Boxiang Tao
China University of Petroleum (East China)
Qingdao, China
z24070102@s.upc.edu.cn

Lei Guo
China University of Petroleum (East China)
Qingdao, China
guolei@upc.edu.cn

Bin Wang
China University of Petroleum (East China)
Qingdao, China
wangbin@upc.edu.cn

Zexin Wang
Huazhong University of Science and Technology
Wuhan, China
u202315704@hust.edu.cn

Hongbo Dou
China University of Petroleum (East China)
Qingdao, China
z23070160@s.upc.edu.cn

Abstract

Model-based reinforcement learning (MBRL) promises high sample efficiency but is often crippled by compounding model errors: small prediction inaccuracies accumulate rapidly over multi-step rollouts, leading to biased policy optimization. We propose FTA, a trajectory-level generative framework using conditional flow matching (CFM) instead of step-wise dynamics. To avoid curved, energy-inefficient paths, we introduce an optimal transport regularizer grounded in the Benamou–Brenier formula. This regularizer minimizes kinetic energy while enforcing exact state displacement, yielding straight, low-energy trajectories with provable linear error propagation. Since a static generator fails in online RL due to (i) shifting data distributions and (ii) mismatch between training (historical) and desired (high-return) conditionals, we add periodic retraining from the replay buffer and a value-guided ODE sampler that biases generation toward high returns via the current Q-function. Theoretically, we show that the regularizer drives the flow toward the Wasserstein-2 optimal transport map and analyze the conservative equilibrium behavior of return-weighted value estimation. On MuJoCo benchmarks, FTA consistently outperforms strong MBRL baselines and generative policies in sample efficiency and final performance, and generalizes across tasks. By mitigating compounding errors at the trajectory level and adapting to online shifts, FTA offers a robust, data-efficient alternative to traditional dynamics models.

1 Introduction

Model-based reinforcement learning (MBRL) [1, 2, 3, 4] offers a promising path to improve sample efficiency by learning an environment dynamics model and using it to generate fictitious experiences. However, its practical success is often limited by a fundamental issue: *compounding model errors* [5]. Even small prediction inaccuracies at each step accumulate rapidly over multi-step rollouts [6, 7, 8], causing the agent to exploit erroneous regions of the model and leading to severely biased policy optimization.

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

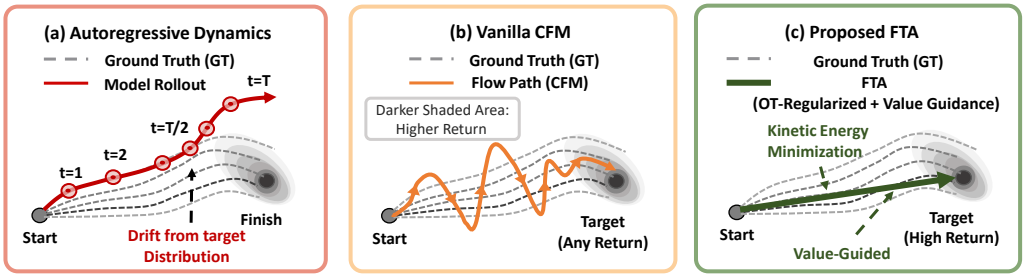


Figure 1: Comparison of trajectory generation paradigms. (a) Autoregressive dynamics accumulate prediction error, causing progressive drift from the target distribution. (b) Vanilla CFM matches the data distribution but follows a highly non-physical, oscillatory path. (c) The proposed FTA produces a direct, well-aligned trajectory via optimal transport regularization and value guidance, achieving stable and efficient evolution toward high-return regions.

Existing MBRL methods attempt to mitigate this problem through uncertainty-aware rollouts [9, 10], multi-step direct prediction [11, 12, 13]. Despite their merits, they share a common underlying assumption: the model predicts next states step-by-step [14, 15]. This local autoregressive design inevitably amplifies errors over long horizons, and the problem remains open [16, 17].

We take a fundamentally different perspective. Instead of learning a step-wise transition function, we propose to directly model the distribution over entire trajectories conditioned on the task context. This trajectory-level generative formulation [18, 19, 20] naturally circumvents the sequential bootstrapping of errors. Specifically, we adopt a conditional flow matching (CFM) model [21, 22], which learns a time-dependent vector field to transport a simple Gaussian prior to the target trajectory distribution. Compared to diffusion models, CFM requires no iterative denoising, making it significantly more efficient [23] for online training and sampling.

However, as illustrated in Figure 1, naive CFM produces arbitrarily curved paths that waste kinetic energy [24, 25, 26] and yield oscillatory trajectories—undesirable for stable policy learning. To address this, we introduce an optimal transport regularizer grounded in the Benamou–Brenier formula [27]. This regularizer minimizes the kinetic energy of the learned flow while enforcing exact state displacement, encouraging straight, low-energy trajectories that provably achieve linear error propagation. This is a sharp contrast to step-wise models, whose errors grow exponentially with horizon. Even with an optimal transport regularizer, a static generative model fails in online RL for two reasons. First, the data distribution shifts continuously as the policy improves, rendering a fixed model obsolete. Second, the conditional distribution used for training (historical data) differs from the desired sampling distribution (high-return trajectories). We tackle these challenges with two lightweight yet effective mechanisms: (i) periodic retraining from the environment replay buffer, and (ii) a value-guided ODE sampler that biases generation toward high-return regions using the current Q-function.

Theoretically, we show that the optimal transport regularizer drives the learned flow toward the Wasserstein-2 optimal transport map [28], consistent with recent OT-flow RL frameworks [29, 30], and analyze the conservative equilibrium behavior of return-weighted value estimation. Empirically, on MuJoCo benchmarks [31], the proposed online RL algorithm, **F**low-based **T**rajectory Tr**A**nsport (**FTA**), consistently outperforms strong MBRL baselines and generative policies in both sample efficiency and final performance, while also demonstrating favorable cross-task generalization. By mitigating compounding errors at the trajectory level and adapting to online distribution shifts, FTA offers a robust, data-efficient alternative to traditional dynamics models for online MBRL.

2 Related Works

2.1 Model-Based RL and the Compounding Error Problem

Model-based reinforcement learning (MBRL) learns a dynamics model to generate synthetic experience [32], but suffers from compounding errors that bias policy optimization [5, 14]. Existing methods mitigate this via several strategies. MBPO [6] and PETS [33] use short or uncertainty-truncated

rollouts to limit error accumulation. ME-TRPO [34] and Steering [35] incorporate ensemble robustness or information-directed exploration. MACURA [36] and BMPO [7] adapt rollout length based on model confidence. Other approaches reduce autoregressive dependence: ADMPO [13] directly predicts future states skipping intermediate steps; MPPVE [12] evaluates multi-step plans from real states; Infoprop [8] uses information-theoretic rollout for consistency. Despite these advances, all remain step-wise in nature and the autoregressive assumption is never fully eliminated. In contrast, our FTA outputs an entire trajectory in one shot, bypassing this bottleneck.

2.2 Generative Models for Reinforcement Learning

Generative models, particularly diffusion models and flow matching, have recently gained traction in RL for data synthesis, planning, and policy representation. Among these approaches, Diffuser [37] represents a key approach leveraging diffusion for planning [38, 18]. Other methods, such as Dacer [19] and Diffusion-QL [39], utilize the denoising process to construct expressive policies that replace conventional Gaussian representations. To address efficiency, FQL [40] employs flow matching but avoids iterative backpropagation by distilling a one-step policy to maximize value. Research on data synthesis [41, 42, 43] remains a closely related frontier. For instance, GTA [20] uses conditional diffusion with classifier-free guidance for offline trajectory augmentation, while MTDiff-S [44] integrates Transformers and prompt learning to serve as both a planner and a high-fidelity synthesizer. To streamline external signaling, EFM [45] proposes energy-weighted flow matching, incorporating guidance directly without auxiliary discriminators. Despite these advances, most existing methods are confined to offline RL, leaving the potential of generative models in online RL largely unexplored.

3 Preliminaries

3.1 Deep Reinforcement Learning

Reinforcement learning (RL) [46] is typically formalized as a Markov Decision Process (MDP), defined by a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$. Here, $\mathcal{S} \subset \mathbb{R}^{d_s}$ and $\mathcal{A} \subset \mathbb{R}^{d_a}$ represent the state space and action space, respectively. The transition probability $P(s_{t+1} \mid s_t, a_t)$ defines the distribution over the next state s_{t+1} given the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. The reward function $r = \mathcal{R}(s, a)$ provides an immediate reward r_t at each time step t, and $\gamma \in [0, 1)$ is the discount factor. In this paper, we focus on online RL, where an agent learns by directly interacting with the environment. Starting from an initial state distribution $s_0 \sim \rho$, the agent learns a policy $\pi(a_t \mid s_t)$ to maximize the expected discounted cumulative return: $\mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)\right]$. A trajectory τ generated by a policy π is a sequence of state-action-reward tuples: $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{T-1}, a_{T-1}, r_{T-1})$, where T is the horizon. We denote the conditional distribution of trajectories under policy π given context c as $p_\tau(\tau \mid c)$.

3.2 Conditional Flow Matching

Given a data sample $\tau_1 \sim p_{\mathcal{T}}(\tau \mid c)$ conditioned on context c and a noise sample $\tau_0 \sim p_0(\tau)$ drawn from a simple prior (e.g., a standard Gaussian), CFM [21] defines an interpolation path $\tau_h = (1 - h) \cdot \tau_0 + h \cdot \tau_1$, where $h \in [0, 1]$. The model trains a time-dependent conditional vector field $v_\psi(\tau_h, h, c)$ to approximate the ground-truth velocity. The learning objective is formulated as a regression loss:

$$\mathcal{L}_{\text{CFM}}(\psi) = \mathbb{E}_{h,\tau_0,\tau_1,c}\Big[\|v_\psi(\tau_h, h, c) - (\tau_1 - \tau_0)\|^2\Big], \tag{1}$$

which encourages the learned vector field to transport samples from $p_0(\tau)$ to the target distribution $p_{\mathcal{T}}(\tau \mid c)$ in a coherent and condition-aware manner. At inference time, the generative process is equivalent to solving an Ordinary Differential Equation (ODE) [47]:

$$\frac{d\tau_h}{dh} = v_\psi(\tau_h, h, c), \tag{2}$$

integrating forward from $h = 0$ with $\tau_0 \sim p_0(\tau)$ to $h = 1$ to obtain synthesized samples.

4 Methodology

In this section, we introduce the implementation of the flow-based trajectory transport (FTA), a generative framework for online MBRL that mitigates compounding errors by modeling entire trajectories instead of step-wise dynamics. The overall architecture follows a Dyna-style [32] interaction loop. The loop consists of three phases: (i) training the CFM model on real environment transitions, (ii) generating synthetic trajectories and storing them in a separate model buffer, and (iii) optimizing the policy using a mixture of real and generated data. A CFM model is periodically trained on real environment transitions from $\mathcal{B}_{\text{env}}$; it then generates synthetic trajectories that are stored in a separate model buffer $\mathcal{B}_{\text{model}}$; the policy is optimized using a mixture of real and generated data.

4.1 Trajectory-Level Generative Model

Let a trajectory sampled with a policy π_θ in the RL environment be denoted as $\tau^\pi = (s_0, a_0, r_0, \cdots, s_{T-1}, a_{T-1}, r_{T-1})$, where $s_t \in \mathcal{S}$, $a_t \in \mathcal{A}$, $r_t \in \mathbb{R}$, and T is the trajectory horizon. The CFM model $\mathcal{M}$ learns a parameterized continuous vector field v_ψ that captures the trajectory distribution $p_{\mathcal{T}}(\tau)$ under a given policy π_θ, allowing trajectories to be sampled from it. To enhance the model's ability to handle trajectories of varying horizons, it is a common scenario in online RL where episodes may terminate early, we adopt a random length strategy during training. At each training iteration, the horizon T is uniformly sampled from the integer interval $[1, \mathscr{T}]$, where $\mathscr{T}$ is a predefined maximum length. Then, the initial trajectory data $\tau_0 \in \mathbb{R}^{(d_s+d_a+1)\times T}$ is sampled from the standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, and we take τ_0 as the input to the model:

$$\tau_0 := \begin{pmatrix} s_0, s_1, s_2, \cdots, s_{T-1} \\ a_0, a_1, a_2, \cdots, a_{T-1} \\ r_0, r_1, r_2, \cdots, r_{T-1} \end{pmatrix}. \tag{3}$$

We sample real trajectory data τ_1 of length T from the environment replay buffer $\mathcal{B}_{\text{env}}$, and construct a conditional probability path τ_h between τ_0 and τ_1 via linear interpolation to characterize the continuous evolution of the flow from $h = 0$ to $h = 1$. The model aims to learn a vector field v_ψ that predicts the instantaneous velocity of trajectory samples at arbitrary flow time h. Accordingly, the training objective follows the standard CFM loss, formally defined as:

$$\mathcal{L}_{\text{CFM}}(\psi) = \mathbb{E}_{h\sim U(0,1),\tau_0\sim\mathcal{N}(0,\mathbf{I}),\tau_1\sim\mathcal{B}_{\text{env}}}\Big[\, \|v_\psi(\tau_h, h) - \widehat{v}(\tau_h, h \mid \tau_1)\|^2 \,\Big], \tag{4}$$

where $\tau_h = (1-h)\tau_0 + h\tau_1$. Here, the context c is extracted as the first step of τ_1.

Implementation. We build our trajectory-level generative model upon Scalable Interpolant Transformers (SiT) [48], a unified framework that generalizes diffusion and flow-based models via flexible stochastic interpolants. Compared to standard diffusion models, SiT adopts velocity prediction, and continuous-time training. Recently, some works take velocity prediction as the learning objective and validate its effectiveness in offline RL [49].

4.2 Low-Energy Optimal Transport Regularization

While vanilla CFM can fit the endpoint τ_1, the lack of constraints on intermediate paths often results in high-energy, curved trajectories, leading to policy oscillations and degraded sample efficiency in online RL. To achieve straight, low-energy transport paths, we incorporate principles from Optimal Transport theory [28]. According to the Benamou-Brenier formula [27], the flow that minimizes total kinetic energy corresponds to a Wasserstein-2 geodesic, effectively the straightest path. Based on this, we propose the following OT regularization term:

$$\begin{aligned} \mathcal{L}_{\mathcal{E}}(\psi) = \mathbb{E}_{h\sim U(0,1),\tau_0\sim\mathcal{N}(0,\mathbf{I}),\tau_1\sim\mathcal{B}_{\text{env}}}\Big[\|v_\psi(\tau_h, h)\|^2\Big] + \\ \lambda_{\mathcal{E}}\mathbb{E}_{\tau_0\sim\mathcal{N}(0,\mathbf{I}),\tau_1\sim\mathcal{B}_{\text{env}}}\left[\left\|\int_0^1 v_\psi^{(s)}(\tau_h, h)dh - (s_1 - s_0)\right\|^2\right], \end{aligned} \tag{5}$$

where $v_\psi^{(s)}$ denotes the state component of the learned vector field, and $\lambda_{\mathcal{E}} > 0$ is a trade-off parameter balancing the two regularization objectives. In Equation 5, the first term suppresses path curvature by

penalizing the instantaneous kinetic energy of the flow, while the second enforces a state displacement constraint that ensures the net state displacement of generated trajectories aligns with the ground-truth transition $\Delta s = s_1 - s_0$. We provide the theoretical analysis of this in Section 4.3.

In dynamic online RL environments, the data distribution shifts continuously as the policy evolves, making robust trajectory generation and uncertainty estimation critical. To implement the OT regularizer under epistemic uncertainty, we extend the base formulation to a multi-head architecture consisting of K independent prediction heads $\{\psi_k\}_{k=1}^K$ that branch out from a shared Transformer backbone. For each individual head, the prediction uncertainty at a given flow state τ_h is quantified by its squared deviation from the ensemble mean field:

$$\begin{aligned} \mathcal{U}_k(\tau_h, h) &= \|v_{\psi_k}(\tau_h, h) - \bar{v}(\tau_h, h)\|^2, \\ \text{where } \bar{v}(\tau_h, h) &= \frac{1}{K}\sum_{j=1}^{K} v_{\psi_j}(\tau_h, h). \end{aligned} \tag{6}$$

Heads with larger prediction variances are more prone to generating unreliable, energy-inefficient curved paths. To suppress this behavior adaptive to the model's epistemic confidence, we dynamically scale the regularization strength for each head using normalized uncertainty weights:

$$\lambda_{\mathcal{E}}^{(k)} = \frac{\mathcal{U}_k(\tau_h)}{\sum_{j=1}^{K} \mathcal{U}_j(\tau_h)}. \tag{7}$$

By substituting the head-specific adaptive weight $\lambda_{\mathcal{E}}^{(k)}$ into the boundary constraint of Equation 5, we denote the individual regularizer for the k-th head as $\mathcal{L}_{\mathcal{E}}(\psi_k; \lambda_{\mathcal{E}}^{(k)})$. This adaptive scheme applies stronger geometric regularization to highly uncertain heads, steering them back toward straight trajectories while preserving necessary prediction diversity for exploration.

The final joint training objective for our trajectory generative model $\mathcal{M}$ is optimized by averaging the combined losses over all K heads:

$$\mathcal{L}_{\mathcal{M}}(\psi) = \frac{1}{K}\sum_{k=1}^{K}\left[\mathcal{L}_{\mathrm{CFM}}(\psi_k) + \mathcal{L}_{\mathcal{E}}(\psi_k; \lambda_{\mathcal{E}}^{(k)})\right]. \tag{8}$$

4.3 Theoretical Analysis

We establish the core theoretical properties of the FTA framework. We begin with a Lipschitz continuity assumption on the MDP, then prove the transitivity of state-wise error propagation, the kinetic energy lower bound and its connection to Wasserstein-2 geodesics, and finally the linear error propagation theorem for OT-regularized flow matching. Please see detailed proof in the appendix C.

Assumption 4.1 (Lipschitz Continuity of the MDP)**.** *Assume the environment transition function $P(s_{t+1} \mid s_t, a_t)$ and the policy $\pi(a_t \mid s_t)$ satisfy: there exist constants $L_s, L_a > 0$ such that for any $s, s' \in \mathcal{S}$ and $a, a' \in \mathcal{A}$,*

$$\|P(\cdot \mid s, a) - P(\cdot \mid s', a')\|_{\mathrm{TV}} \le L_s\|s - s'\| + L_a\|a - a'\|, \tag{9}$$

and the policy π is L_π-Lipschitz: $\|\pi(\cdot \mid s) - \pi(\cdot \mid s')\|_{\mathrm{TV}} \le L_\pi\|s - s'\|$.

This assumption is standard in continuous control and ensures that the system responds smoothly to changes in states and actions.

Proposition 4.2 (Transitivity of State Error)**.** *Let $\tau = (s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1})$ be a real trajectory and $\hat{\tau}$ a generated one. Suppose $\|\hat{s}_0 - s_0\| \le \epsilon_0$ and for each step t*

$$\|\hat{s}_{t+1} - s_{t+1}\| \le \kappa\|\hat{s}_t - s_t\| + \delta, \tag{10}$$

with constants $\kappa \ge 0, \delta \ge 0$. The for all t,

$$\|\hat{s}_t - s_t\| \le \kappa^t\epsilon_0 + \frac{\kappa^t - 1}{\kappa - 1}\delta \quad (\kappa \ne 1). \tag{11}$$

Lemma 4.3 (Kinetic Energy Lower Bound and Wasserstein-2 Geodesics)**.** *Consider a probability path $p_h(\tau)$ evolving in time $h \in [0,1]$ with velocity field $v_h(\tau)$ that satisfies the continuity equation $\partial_h p_h + \nabla \cdot (p_h v_h) = 0$ and boundary conditions $p_0 = p_0, p_1 = p_{target}$. The the kinetic energy*

$$\mathcal{K} = \int_0^1 \mathbb{E}_{\tau_h \sim p_h} \left[\|v_h(\tau_h)\|^2 \right] dh \tag{12}$$

satisfies $\mathcal{K} \geq W_2^2(p_0, p_{\text{target}})$, where W_2 is the Wasserstein-2 distance. Equality holds if and only if the path is a Wasserstein-2 geodesic, i.e., the velocity field is a gradient field of a convex function.

Theorem 4.4 (Linear Error Propagation of OT-CFM Flows)**.** *Under Assumption 4.1, for each considered horizon T, let τ and $\hat{\tau}$ denote the true and OT-CFM generated trajectories on a specified coupled probability space with a common initial context. Assume a compact convex domain with uniformly positive true conditional densities, $W_2(p_{\text{CFM}}, p_{\text{true}}) \leq \delta$, and Lipschitz implicit kernels $\hat{\mu}, \hat{\pi}$ with conditional TV errors bounded by $C_{\text{cond}}\delta$. Suppose the state error $e_t = \hat{s}_t - s_t$ satisfies the following recursion for $0 \leq t < T-1$, with the bias and matrix-product bounds holding almost surely:*

$$\begin{aligned} e_{t+1} &= D_t e_t + b_t + \eta_t, \quad e_0 = 0, \\ \|b_t\| &\leq C_b \delta, \quad \mathbb{E}\|\eta_t\|^2 \leq \epsilon^2, \\ \|D_{t-1} \cdots D_{k+1}\|_{\text{op}} &\leq 1, \quad 0 \leq k < t < T, \end{aligned} \tag{13}$$

where an empty product is the identity, $C_b \geq 0$ is finite, D_t is the error propagation matrix and ϵ bounds the effective one-step state prediction error induced by CFM velocity approximation. For the fixed policy, context, and family of couplings, assume these bounds hold on each corresponding probability space with the same constants C_b, δ, ϵ, independent of t and T. Then, for some constant $C > 0$ independent of t and T,

$$\mathbb{E}\left[\|\hat{s}_t - s_t\|^2\right] \leq C t^2 \max\{\delta^2, \epsilon^2\}, \quad 0 \leq t < T. \tag{14}$$

Thus, the root-mean-square state error grows at most linearly with t under the stated conditions. The product bound and conditional approximation bounds are explicit assumptions; the OT penalty alone is not asserted to establish them.

Corollary 4.5. *In contrast, conventional step-wise dynamics models satisfy $\mathbb{E}\left[\|\hat{s}_t - s_t\|^2\right] \leq C' e^{\lambda t} \epsilon^2$, for some $\lambda > 0$ (exponential growth). Therefore, OT-CFM reduces error propagation from exponential to linear, fundamentally mitigating the compounding error problem in online MBRL.*

4.4 Value-Guided Sampling

The OT-regularized CFM model described above is trained on historical data collected by the current policy. Consequently, its conditional distribution $p_\tau(\tau \mid c)$ approximates the *average* return of past trajectories. However, during policy optimization we desire to sample trajectories with *high* returns. This mismatch between the training distribution (historical) and the desired sampling distribution (high-return) cannot be corrected by the OT regularizer alone.

To bridge this gap, we introduce value-guided sampling: a lightweight procedure that biases the ODE integration toward trajectories with larger expected returns, without retraining the model. For a partially generated trajectory τ_h at interpolation time h, we define its discounted return as $J(\tau_h) = \sum_{t=0}^{T-1} \gamma^t r_t$, where r_t is the reward at step t extracted from τ_h. The return $J(\tau_h)$ directly measures trajectory quality and provides a natural guidance signal for modulating the learned flow. Then, we define the value-guided vector field $\mathcal{G}$ as

Definition 4.6 (Value-Guided Vector Field)**.** Let $v_\psi(\tau_h, h)$ be the original learned vector field. We modulate its magnitude using a monotonic weighting function that slows the flow in high-return regions:

$$\mathcal{G}(\tau_h, h) = \Upsilon\big(J(\tau_h)\big) v_\psi(\tau_h, h) \tag{15}$$

with $\Upsilon(\cdot) > 0$. In practice, we use a fixed horizon T and an exponential weighting that normalizes returns within a mini-batch:

$$\Upsilon(J) = \text{clip}\left(\exp\left(-\lambda_\Upsilon \frac{J - \mu_J}{\sigma_J}\right), \nu_{\min}, \nu_{\max}\right), \tag{16}$$

where μ_J and σ_J denote the batch mean and standard deviation of returns, and $\lambda_\Upsilon > 0$ controls the guidance strength. Higher-return trajectories receive smaller velocity scaling factors, locally slowing the ODE evolution in high-return regions. This modulation is intended to favor sampling in these regions, while clipping ensures numerical stability.

Theorem 4.7 (Q-Estimation Equilibrium and Bias Bounds for Value-Guided Sampling)**.** *Fix the policy, initial state-action pair (s_0, a_0), normalization statistics, and a compactly supported return distribution p. Write $q = Q_\phi(s_0, a_0)$ and $\mu = \mathbb{E}_p[J]$. Let $\Upsilon(J)$ be a measurable, non-increasing weight with $0 < \nu_{\min} \le \Upsilon(J) \le \nu_{\max} < \infty$, and let $\bar{\Upsilon} = \mathbb{E}_p[\Upsilon(J)] > 0$. Consider the scalar expected-update model*

$$\mathbb{E}[\Delta q] = a\,\mathbb{E}_p[\Upsilon(J)(J - q)], \qquad a > 0. \tag{17}$$

There is a unique equilibrium target Q^* for which this expected update vanishes, given by

$$Q^* = \frac{\mathbb{E}_p[\Upsilon(J)J]}{\bar{\Upsilon}} = \mu + \frac{\mathrm{Cov}_p(\Upsilon(J), J)}{\bar{\Upsilon}}. \tag{18}$$

Furthermore,

$$\mathrm{sign}\big(\mathbb{E}[\Delta q]\big) = \mathrm{sign}(Q^* - q). \tag{19}$$

Since $\Upsilon(J)$ is non-increasing, $\mathrm{Cov}_p(\Upsilon(J), J) \le 0$, yielding $Q^* \le \mu$. Its absolute deviation from the base mean is exactly

$$|Q^* - \mu| = \frac{|\,\mathrm{Cov}_p(\Upsilon(J), J)|}{\bar{\Upsilon}}. \tag{20}$$

For the normalized exponential weight $\Upsilon_\lambda(J) = \exp(-\lambda(J - \mu_J)/\sigma_J)$ with fixed $\sigma_J > 0$, as $\lambda \downarrow 0$,

$$Q^* - \mu = -\frac{\lambda}{\sigma_J}\,\mathrm{Var}_p(J) + \mathcal{O}(\lambda^2). \tag{21}$$

The same expansion applies to its clipped version when $0 < \nu_{\min} < 1 < \nu_{\max}$, since clipping is then inactive for sufficiently small λ. In particular, $Q^* \to \mu$ as $\lambda \downarrow 0$. Exact equality $Q^* = \mu$ holds whenever $\mathrm{Cov}_p(\Upsilon(J), J) = 0$, including constant weights.

By scaling magnitude instead of direction, value guidance preserves the local direction of the learned vector field while adjusting its evolution according to trajectory returns. The resulting effects on trajectory quality and physical consistency are evaluated empirically.

5 Experiments

We evaluate FTA on eight continuous control benchmarks from MuJoCo-v4 [31] using the Gymnasium interface [50]. Our primary goal is to assess whether trajectory-level generative modeling with optimal transport (OT) regularization mitigates compounding errors more effectively than existing MBRL methods that rely on step-wise dynamics. We therefore compare against representative baselines that explicitly address error accumulation: MBPO [6] (short branched rollouts), ADMPO [13] (any-step prediction), MPPVE [12] (multi-step planning) and MACURA [36] (uncertainty-aware rollout truncation). We also include the model-free SAC [51] to highlight the benefit of model-based synthesis, and two generative-model-based baselines, i.e., FlowRL [30] and SDAC [52], to examine whether our OT-regularized trajectory-level generative modeling brings further gains over existing generative approaches. Recent world models such as Dreamer [53] learn latent dynamics and are highly effective in vision-based tasks, but their focus is on compact state representation and long-horizon imagination rather than on directly combating compounding errors in low-dimensional continuous control.

5.1 Experimental Results

As shown in Figure 2, the empirical results across the MuJoCo suite demonstrate that FTA achieves better sample efficiency and asymptotic performance. By leveraging trajectory-level conditional flow matching with OT regularization, our framework effectively straights the generative paths and minimizes kinetic energy, which reduces compounding errors from exponential to linear growth. This theoretical advantage translates into superior stability and faster convergence, particularly in high-dimensional tasks like Walker2d, HalfCheetah, Ant, and Humanoid, where FTA consistently surpasses advanced ensemble and model-based baselines. The remaining Swimmer and Reacher

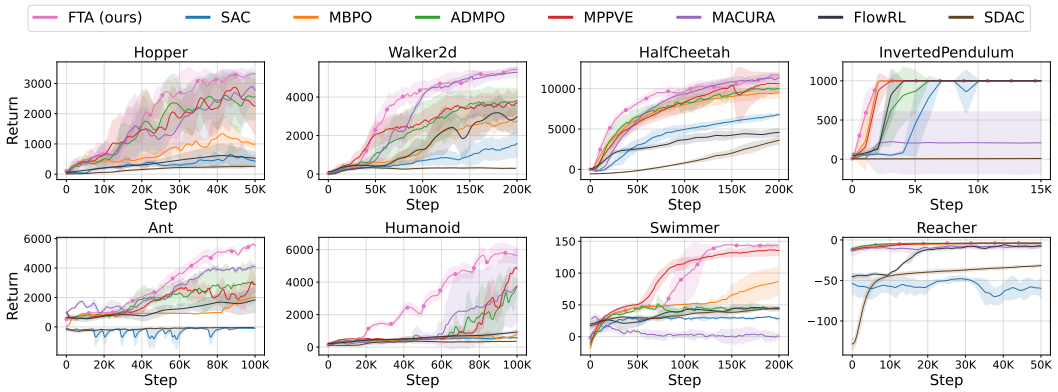


Figure 2: Performance comparison of FTA (ours) against state-of-the-art baselines on MuJoCo benchmarks. Training curves depict mean returns over total interaction steps, with shaded regions representing the standard deviation across random seeds.

tasks, together with the generative baselines FlowRL and SDAC, exhibit the same overall trend, with FTA remaining competitive or superior in most cases and generally outperforming existing generative approaches. Ultimately, these findings validate that integrating value guidance with structured generative priors enables the synthesis of high-quality experiences that exceed the performance of recorded historical data.

5.2 Ablation Study

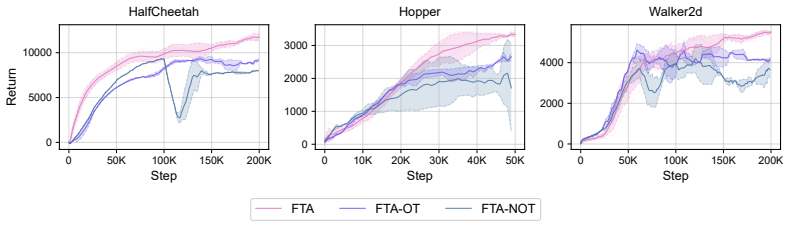
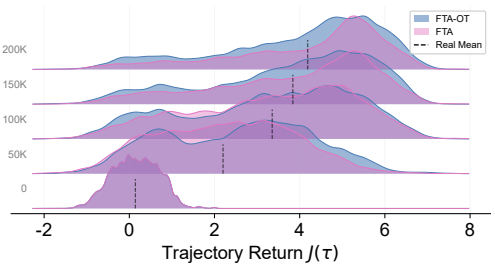


Figure 3: Ablation study of the FTA framework components across MuJoCo benchmarks. The results compare the full framework (FTA), the variant with OT regularization but without value guidance (FTA-OT), and the vanilla conditional flow matching baseline (FTA-NOT).

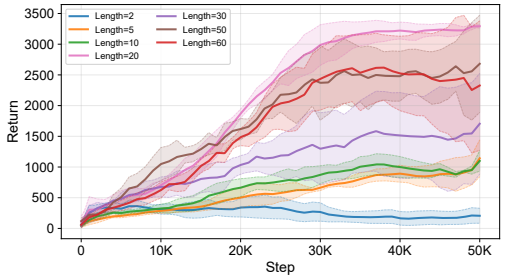
The ablation study illustrated in Figure 3 provides a comprehensive comparison between FTA, FTA-OT, and FTA-NOT, highlighting the functional impact of OT and value guidance. Across three MuJoCo benchmarks, FTA-OT consistently demonstrates superior stability and performance compared to the vanilla FTA-NOT. This trend is most evident in the HalfCheetah environment, where FTA-NOT suffers a significant performance collapse around 110,000 steps, whereas FTA-OT maintains a steady and robust learning curve. These empirical results directly validate our methodological framework: while vanilla flow matching generates oscillatory, high-energy trajectories that lead to high variance and compounding prediction errors, the OT regularizer enforces kinetic energy minimization via the Benamou–Brenier formula. By constraining the generative flow to follow straight-line Wasserstein-2 geodesics, FTA-OT effectively achieves the linear error propagation predicted in Theorem 4.4, providing a reliable foundation for policy optimization. Furthermore, while FTA yields the highest final returns by utilizing value-guided sampling to focus on high-reward regions, the persistent performance gap between FTA-OT and FTA-NOT confirms that trajectory-level OT regularization is the primary mechanism for mitigating compounding errors in our framework.

The ridge plot in Figure 4a provides a granular look at how value guidance modulates the generative process. Early in training (0–50K steps), the generated return distributions for both variants largely overlap with the real environment mean, as the model primarily focuses on capturing the basic transition dynamics. However, as training progresses to 200K steps, a clear divergence emerges: while the FTA-OT distribution remains centered near the historical average, the FTA distribution

significantly shifts toward the higher reward spectrum ($J(\tau) > 4$). This rightward shift confirms the effectiveness of the value-guided vector field defined in Equation 15, which biases sampling toward trajectories with higher predicted returns. Crucially, the fact that the pink distribution consistently surpasses the Real Mean baseline demonstrates that our framework does not merely replicate historical successes but actively hallucinates improved trajectories. This informed optimism facilitates more efficient policy optimization by providing the agent with high-quality synthetic transitions that are otherwise scarce in the replay buffer.



(a) Distributional evolution of trajectory returns $J(\tau)$ over training steps. We compare generated returns from FTA and FTA-OT against the environment average (Real Mean). The value-guided sampling progressively shifts the generative distribution toward high-return regions, enabling the model to synthesize experiences that exceed the performance of recorded historical data.



(b) Impact of trajectory length T_{gen} on performance. We evaluate FTA under varying generation horizons from $T_{\text{gen}} = 2$ to $T_{\text{gen}} = 60$. Performance scales significantly as the length increases, with $T_{\text{gen}} = 20$ and $T_{\text{gen}} = 50$ achieving the highest returns. However, the marginal gains diminish beyond $T_{\text{gen}} = 20$, and excessively long horizons (e.g., $T_{\text{gen}} = 60$) exhibit higher variance and slightly lower stability.

Figure 4: Evaluation of value-guided distribution shift and trajectory horizon sensitivity.

The sensitivity analysis regarding the trajectory length T_{gen} reveals a critical trade-off between local precision and global optimization. As illustrated in Figure 4b, very short horizons ($T_{\text{gen}} = 2, 5$) lead to poor performance, as the model lacks the necessary look-ahead to capture the long-term consequences of current actions, effectively falling into myopic policy traps. As the length increases to $T_{\text{gen}} = 20$, we observe a substantial performance boost, which validates our trajectory-level generative approach: a longer horizon allows the conditional flow matching to synthesize more coherent and purposeful transitions. Interestingly, further increasing the length to $T_{\text{gen}} = 60$ does not yield proportional improvements and instead introduces greater variance (shaded regions). This suggests that while a larger T_{gen} provides a richer context for value guidance, it also increases the complexity of the generative task, making it more difficult for the model to maintain the linear error propagation property over extremely long sequences. Consequently, $T_{\text{gen}} = 20$ represents an empirical sweet spot for balancing computational efficiency with robust performance in MuJoCo environments.

5.3 Results Analysis

The compounding error analysis empirically validates the theoretical assertions made in Theorem 4.4 regarding error propagation. As illustrated in Figure 5, vanilla CFM suffers from a steep error growth with a slope of 4.21, indicating that small perturbations in the vector field lead to an exponential divergence from the ground-truth trajectory over the horizon. In contrast, FTA maintains a much lower growth rate with a slope of 2.44. This significant reduction confirms that the OT regularization successfully straightens the flow, thereby constraining the cumulative drift to a linear scale relative to the physical time $t \cdot h$. By effectively narrowing the

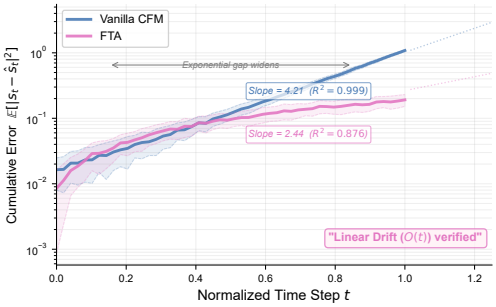


Figure 5: Compounding error analysis.

exponential gap, FTA ensures high-fidelity trajectory synthesis even across extended horizons, which is a critical factor for the superior performance observed in our main MuJoCo benchmarks.

Figure 6 provides a granular qualitative and quantitative assessment of the geometric properties inherent in the generated flows, specifically focusing on phase-space trajectories and their corresponding kinetic energy profiles. In the absence of optimal transport constraints, vanilla CFM tends to generate paths characterized by significant oscillations and erratic zigzag patterns, suggesting physically inefficient mappings. In stark contrast, FTA yields near-linear geodesics between s_0 and s_1. This visual evidence directly validates our OT-based kinetic energy minimization. Quantitatively, FTA consistently achieves a substantial reduction in average kinetic energy, with savings ranging from 30% to 45% across benchmarks. These results demonstrate that simplifying generative dynamics into **straight-line** motions provides essential structural priors for stable and efficient trajectory synthesis.

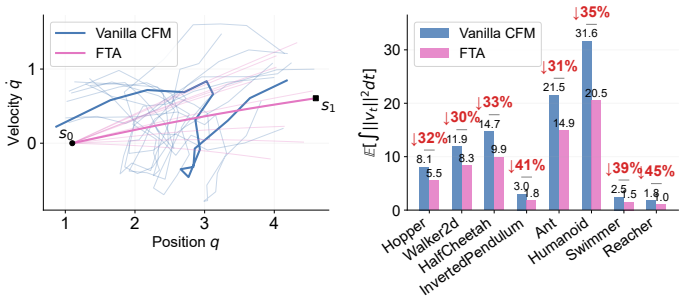


Figure 6: (Left) Phase-Space Paths: Comparison of generated trajectories between vanilla CFM and FTA in the phase space $(q, \dot{q})$. (Right) Kinetic Energy Comparison: Average kinetic energy $\mathbb{E}[\int_0^1 \|v_t\|^2 dt]$ across Eight MuJoCo environments.

To test the robustness of the learned generative priors, we evaluate the zero-shot transfer performance from HalfCheetah to Walker2d. Although these agents possess different skeletal structures, they share fundamental physical consistencies in locomotion dynamics. As shown in Figure 7, FTA maintains superior sample efficiency and achieves a final return of approximately 5,000, outperforming baselines trained from scratch on the target task. This confirms that the OT regularization in FTA distills task-agnostic structural regularities, such as energy-efficient and straight-line motion, rather than merely overfitting to specific environment transitions. Such generalization capability is crucial for developing versatile agents capable of navigating novel state-action spaces with minimal performance loss.

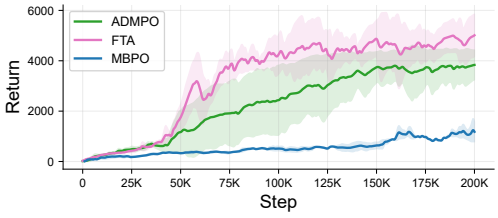


Figure 7: Cross-task generalization from HalfCheetah to Walker2d.

6 Conclusion and Discussion

We proposed FTA, a trajectory-level generative framework for online MBRL that replaces step-wise dynamics with conditional flow matching and an optimal transport regularizer. By minimizing kinetic energy as prescribed by the Benamou-Brenier formula, the framework produces straight-line trajectory geodesics; both theoretical analysis and empirical results confirm that this reduces error propagation from exponential to linear, thereby significantly enhancing the stability of policy optimization. Our experimental results further demonstrate that the integration of a value-guided ODE sampling mechanism allows FTA to synthesize high-quality, high-return experiences, outperforming traditional dynamics models in sample efficiency and final performance across various MuJoCo benchmarks. Despite these strengths, several limitations remain, such as the increased computational overhead of ODE solving compared to simple MLP predictions and the dependence on Q-function accuracy, which can introduce guidance bias early in training. Additionally, while a 20-step horizon proved effective, extending this trajectory-level approach to extremely long sequences or high-dimensional pixel-based environments represents a key direction for future research.

Acknowledgments and Disclosure of Funding

Funding. The authors received no external funding or third-party support for this work.

Competing Interests. The authors declare that they have no competing interests related to this work.

References

[1] Zirui Wang, Yue Deng, Junfeng Long, and Yin Zhang. Parallelizing model-based reinforcement learning over the sequence length. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[2] Abdullah Akgül, Manuel Haussmann, and Melih Kandemir. Deterministic uncertainty propagation for improved model-based offline reinforcement learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[3] Mao Hong, Zhengling Qi, and Yanxun Xu. Model-based reinforcement learning for confounded pomdps. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 18668–18710. PMLR / OpenReview.net, 2024.

[4] Renhao Zhang, Haotian Fu, Yilin Miao, and George Konidaris. Model-based reinforcement learning for parameterized action spaces. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 58935–58954. PMLR / OpenReview.net, 2024.

[5] Kavosh Asadi, Dipendra Misra, Seungchan Kim, and Michael L. Littman. Combating the compounding-error problem with a multi-step model. *CoRR*, abs/1905.13320, 2019.

[6] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12498–12509, 2019.

[7] Hang Lai, Jian Shen, Weinan Zhang, and Yong Yu. Bidirectional model-based policy optimization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5618–5627. PMLR, 2020.

[8] Bernd Frauenknecht, Devdutt Subhasish, Friedrich Solowjow, and Sebastian Trimpe. On rollouts in model-based reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[9] Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. Coplanner: Plan to roll out conservatively but to explore optimistically for model-based RL. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[10] Wenjun Huang, Yunduan Cui, Huiyun Li, and Xinyu Wu. Practical probabilistic model-based reinforcement learning by integrating dropout uncertainty and trajectory sampling. *IEEE Trans. Neural Networks Learn. Syst.*, 36(7):12812–12826, 2025.

[11] Daniel Palenicek, Michael Lutter, Joao Carvalho, and Jan Peters. Diminishing return of value expansion methods in model-based reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[12] Haoxin Lin, Yihao Sun, Jiaji Zhang, and Yang Yu. Model-based reinforcement learning with multi-step plan value estimation. In Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu, editors, *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 1481–1488. IOS Press, 2023.

[13] Haoxin Lin, Yu-Yan Xu, Yihao Sun, Zhilong Zhang, Yi-Chen Li, Chengxing Jia, Junyin Ye, Jiaji Zhang, and Yang Yu. Any-step dynamics model improves future predictions for online and offline reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[14] Nathan O. Lambert, Kristofer S. J. Pister, and Roberto Calandra. Investigating compounding prediction errors in learned dynamics models. *CoRR*, abs/2203.09637, 2022.

[15] Anne Somalwar, Bruce D. Lee, George J. Pappas, and Nikolai Matni. Learning with imperfect models: When multi-step prediction mitigates compounding error. In *64th IEEE Conference on Decision and Control, CDC 2025, Rio de Janeiro, Brazil, December 9-12, 2025*, pages 82–89. IEEE, 2025.

[16] Nan Jiang. A note on loss functions and error compounding in model-based reinforcement learning. *CoRR*, abs/2404.09946, 2024.

[17] Brett Barkley and David Fridovich-Keil. Stealing that free lunch: Exposing the limits of dyna-style reinforcement learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[18] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pages 9902–9915. PMLR, 2022.

[19] Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, and Shengbo Li. Diffusion actor-critic with entropy regulator. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[20] Jaewoo Lee, Sujin Yun, Taeyoung Yun, and Jinkyoo Park. GTA: generative trajectory augmentation with guidance for offline reinforcement learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[21] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[22] Sean Ye and Matthew C. Gombolay. Efficient trajectory forecasting and generation with conditional flow matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pages 2816–2823. IEEE, 2024.

[23] Srishti Gupta and Yashasvee Taiwade. Efficiency vs. fidelity: A comparative analysis of diffusion probabilistic models and flow matching on low-resource hardware. *CoRR*, abs/2511.19379, 2025.

[24] Kacper Kapusniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael M. Bronstein, Avishek Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[25] Ziyun Li, Ben Dai, Huancheng Hu, Henrik Boström, and Soon Hoe Lim. Enfopath: Energy-informed analysis of generative trajectories in flow matching. *CoRR*, abs/2511.19087, 2025.

[26] Ziyun Li, Huancheng Hu, Soon Hoe Lim, Xuyu Li, Fei Gao, Enmao Diao, Zezhen Ding, Michalis Vazirgiannis, and Henrik Boström. A kinetic-energy perspective of flow matching. *CoRR*, abs/2602.07928, 2026.

[27] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[28] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[29] Ting Gao, Stavros Orfanoudakis, Nan Lin, Elvin Isufi, Winnie Daamen, and Serge P. Hoogendoorn. Flow matching policy with entropy regularization. *CoRR*, abs/2603.17685, 2026.

[30] Lei Lyu, Yunfei Li, Yu Luo, Fuchun Sun, Tao Kong, Jiafeng Xu, and Xiao Ma. Flow-based policy for online reinforcement learning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

[31] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 5026–5033. IEEE, 2012.

[32] Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Bruce W. Porter and Raymond J. Mooney, editors, *Machine Learning, Proceedings of the Seventh International Conference on Machine Learning, Austin, Texas, USA, June 21-23, 1990*, pages 216–224. Morgan Kaufmann, 1990.

[33] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4759–4770, 2018.

[34] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[35] Souradip Chakraborty, Amrit S. Bedi, Alec Koppel, Mengdi Wang, Furong Huang, and Dinesh Manocha. STEERING : Stein information directed exploration for model-based reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 3949–3978. PMLR, 2023.

[36] Bernd Frauenknecht, Artur Eisele, Devdutt Subhasish, Friedrich Solowjow, and Sebastian Trimpe. Trust the model where it trusts itself - model-based actor-critic with uncertainty-aware rollout adaption. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[37] Chang Chen, Fei Deng, Kenji Kawaguchi, Caglar Gulcehre, and Sungjin Ahn. Simple hierarchical planning with diffusion. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[38] João Carvalho, An T. Le, Mark Baierl, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *IROS*, pages 1916–1923, 2023.

[39] Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[40] Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[41] Cong Lu, Philip J. Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[42] Shichao Fan, Quantao Yang, Yajie Liu, Kun Wu, Zhengping Che, Qingjie Liu, and Min Wan. Diffusion trajectory-guided policy for long-horizon robot manipulation. *IEEE Robotics Autom. Lett.*, 10(12):12788–12795, 2025.

[43] Qianlan Yang and Yu-Xiong Wang. Rtdiff: Reverse trajectory synthesis via diffusion for offline reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[44] Haoran He, Chenjia Bai, Kang Xu, Zhuoran Yang, Weinan Zhang, Dong Wang, Bin Zhao, and Xuelong Li. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[45] Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[46] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018.

[47] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[48] Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXVII*, Lecture Notes in Computer Science, pages 23–40. Springer, 2024.

[49] Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[50] Mark Towers, Ariel Kwiatkowski, Jordan K. Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *CoRR*, abs/2407.17032, 2024.

[51] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018.

[52] Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Soft diffusion actor-critic: Efficient online reinforcement learning for diffusion policy. *CoRR*, abs/2502.00361, 2025.

[53] Haoyu Ma, Jialong Wu, Ningya Feng, Chenjun Xiao, Dong Li, Jianye Hao, Jianmin Wang, and Mingsheng Long. Harmonydream: Task harmonization inside world models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

A Symbols

The symbols in the paper are listed in Table 1.

Table 1: Symbols and Explanations

Category	Symbol	Explanation
Environment	$\mathcal{S}$	State space
	s	State
	d_s	Dimension of state
	$\mathcal{A}$	Action space
	a	Action
	d_a	Dimension of action
	$\mathcal{R}$	Reward function
	r	Immediate Reward
	P	Transition probability function
	γ	Discount factor
	t	Environmental time step
	π	Policy of agent
Deep Q Learning	$\mathcal{H}$	Entropy of policy π
	Q^π	Action value function under policy π
	V^π	State-Action value function under policy π
Conditional Flow Matching Model	v	Time-dependent vector field
	$\widehat{v}$	Analytically constructed target vector field
	h	Interpolation time in $[0, 1]$
	c	Conditional context
Trajectory-level Generative Model	τ^π	A trajectory sampled with a policy π
	$\mathcal{T}$	The support set of trajectory
	τ	A trajectory sampled from the support set $\mathcal{T}$
	$\mathscr{T}$	Maximum integer horizon for trajectory generation
	T	Randomly sampled integer horizon from $[1, \mathscr{T}]$ during training
	$p_0(\tau)$	Initialized trajectory distribution
	τ_0	A initialized trajectory sampled from $p_0(\tau)$
	$p_{\mathcal{T}}(\tau)$	Target trajectory distribution
	τ_1	A target trajectory sampled from $p_{\mathcal{T}}(\tau)$
	$\hat{\tau}_1$	A trajectory sampled from the model
	τ_h	A trajectory at interpolation time h
	$p_h(\tau)$	Trajectory distribution at interpolation time h
	$\mathcal{U}$	The uncertainty of the model $\mathcal{M}$
	Υ	A bounded scaler function
	$\nu_{\min}, \nu_{\max}$	The bound of scaler function Υ
Learning Parameters	θ	Parameter of Policy
	ϕ	Parameter of Q-network
	ψ	Parameter of learnable vector field v
	α	Temperature parameter of policy's entropy
Replay Buffer	$\mathcal{B}_{\text{env}}$	Replay buffer of environment
	$\mathcal{B}_{\text{model}}$	Replay buffer of model

B Pseudocode

We present the pseudocode of the FTA framework. Algorithm 1 outlines the main online reinforcement learning loop, which follows a Dyna-style architecture. The agent interacts with the environment, stores real transitions in a replay buffer, and periodically retrains a CFM model on this real data. Synthetic trajectories are then generated by a value-guided ODE sampler and stored in a separate model buffer. The policy and Q-networks are updated using a mixture of real and synthetic samples. The CFM training subroutine is detailed in Algorithm 2. It employs a random length strategy, linear interpolation paths, and a multi-head architecture. The loss combines a standard conditional flow matching term with an OT regularizer that penalises kinetic energy and enforces exact state displacement. Per-head uncertainty weights the regularizer adaptively, steering uncertain heads more strongly toward straight, low-energy flows. Algorithm 3 describes the value-guided trajectory generation procedure. An ODE solver integrates the learned vector field from a Gaussian prior, while a non-increasing weighting function $\Upsilon(J)$ modulates the field based on the partially accumulated discounted return J. This concentrates the sampling distribution over high-return regions without altering the underlying transition dynamics, thus providing optimistic yet physically consistent synthetic experiences for policy improvement.

Algorithm 1 Flow-based Trajectory Transport (FTA)

Input: maximum trajectory length $\mathcal{T}$, number of CFM heads K, regularization weight β, balancing coefficient $\lambda_{\mathcal{E}}$, value guidance parameters $\lambda_{\Upsilon}, \nu_{\min}, \nu_{\max}$, ODE integration steps N_{ode}, retrain frequency F_{model}, batch sizes $B_{\text{real}}, B_{\text{model}}$, policy update ratio r_{model}
Initialize environment replay buffer $\mathcal{B}_{\text{env}} \leftarrow \emptyset$, model replay buffer $\mathcal{B}_{\text{model}} \leftarrow \emptyset$
Initialize CFM model $\mathcal{M}$ with K output heads $\{\psi_k\}_{k=1}^{K}$
Initialize policy π_θ and Q-networks Q_ϕ (e.g., SAC)
Initialize environment, obtain initial state s_0
for each interaction step $n = 1, 2, \ldots$ **do**
    Sample action $a \sim \pi_\theta(\cdot|s)$; execute a, observe r, s', done flag d
    Store (s, a, r, s', d) in $\mathcal{B}_{\text{env}}$
    $s \leftarrow s'$; if d then reset environment and obtain new s_0
    if $n \bmod F_{\text{model}} = 0$ **then**
        Train CFM($\mathcal{B}_{\text{env}}$) ▷ Algorithm 2
        $\mathcal{B}_{\text{model}} \leftarrow$ Generate Trajectories($\mathcal{M}, \pi_\theta, Q_\phi, N_{\text{gen}}$) ▷ Algorithm 3
    end if
    if update condition met (e.g., every environment step) **then**
        Sample mini-batches from $\mathcal{B}_{\text{env}}$ and $\mathcal{B}_{\text{model}}$ with ratio r_{model}
        Update policy π_θ and Q-networks Q_ϕ using standard SAC updates
    end if
end for

C Theoretical Results and Detailed Proofs

We provide the complete proofs for all theoretical claims in Sections 4.3 and 4.4 of the main text. We prove Proposition 4.2, Lemma 4.3, Theorem 4.4, Corollary 4.5, and Theorem 4.7 in order, following the notation and problem setup established in the paper.

C.1 Proof of Proposition 4.2

Proposition 4.2 (Transitivity of State Error). Let $\tau = (s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1})$ be a real trajectory and $\hat{\tau}$ a generated one. Suppose $\|\hat{s}_0 - s_0\| \le \epsilon_0$ and for each step t

$$\|\hat{s}_{t+1} - s_{t+1}\| \le \kappa \|\hat{s}_t - s_t\| + \delta,$$

with constants $\kappa \ge 0, \delta \ge 0$. The for all t,

$$\|\hat{s}_t - s_t\| \le \begin{cases} \kappa^t \epsilon_0 + \frac{\kappa^t - 1}{\kappa - 1}\delta, & \kappa \ne 1, \\ \epsilon_0 + t\delta, & \kappa = 1. \end{cases}$$

Algorithm 2 Train CFM($\mathcal{B}_{\text{env}}$)

for each training iteration **do**
  Sample a mini-batch of real trajectories $\{\tau_1^{(i)}\}$ from $\mathcal{B}_{\text{env}}$
  for each $\tau_1^{(i)}$ in mini-batch **do**
    Randomly sample horizon $T \sim \text{Uniform}(1, \mathscr{T})$
    Truncate $\tau_1^{(i)}$ to the first T steps (pad if necessary)
    Sample noise trajectory $\tau_0^{(i)} \sim \mathcal{N}(0, \mathbf{I})$ of shape $(d_s + d_a + 1) \times T$
    Sample interpolation time $h \sim \mathcal{U}(0, 1)$
    $\tau_h^{(i)} \leftarrow (1-h)\,\tau_0^{(i)} + h\,\tau_1^{(i)}$
    Extract context $c^{(i)}$ as the first step of $\tau_1^{(i)}$
  end for
  for each head $k = 1, \ldots, K$ **do**
    Compute vector field $\mathbf{v}_{\psi_k}(\tau_h^{(i)}, h, c^{(i)})$ for all i
    $\mathcal{L}_{\text{CFM}}^{(k)} \leftarrow \frac{1}{\text{batch}} \sum_i \|\mathbf{v}_{\psi_k}^{(i)} - (\tau_1^{(i)} - \tau_0^{(i)})\|^2$
    Compute epistemic uncertainty $\mathcal{U}_k^{(i)} = \|\mathbf{v}_{\psi_k}^{(i)} - \bar{\mathbf{v}}^{(i)}\|^2$, where $\bar{\mathbf{v}}^{(i)} = \frac{1}{K}\sum_{j=1}^{K} \mathbf{v}_{\psi_j}^{(i)}$
    Normalize head weight $\lambda_{\mathcal{E}}^{(k,i)} \leftarrow \frac{\mathcal{U}_k^{(i)}}{\sum_{j=1}^{K} \mathcal{U}_j^{(i)}}$
    $\mathcal{L}_{\mathcal{E}}^{(k,i)} \leftarrow \frac{1}{\text{batch}} \sum_i \left[\|\mathbf{v}_{\psi_k}^{(i)}\|^2 + \lambda_{\mathcal{E}}^{(k,i)} \left\| \int_0^1 \mathbf{v}_{\psi_k}^{(s)}(\tau_h, h, c)\, dh - (s_1^{(i)} - s_0^{(i)}) \right\|^2 \right]$
    $\mathcal{L}^{(k)} \leftarrow \mathcal{L}_{\text{CFM}}^{(k)} + \frac{1}{\text{batch}} \sum_i \mathcal{L}_{\mathcal{E}}^{(k,i)}$
  end for
  $\mathcal{L}_{\mathcal{M}} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}^{(k)}$
  Update model parameters $\{\psi_k\}$ by gradient descent on $\mathcal{L}_{\mathcal{M}}$
end for

Proof. We proceed by induction on t.

- For $t = 0$, the bound $\|\hat{s}_0 - s_0\| \le \epsilon_0$ holds by assumption, which satisfies the formula.
- Assume the bound holds for some $t \ge 0$, and consider $t+1$. By the recurrence relation: $\|\hat{s}_{t+1} - s_{t+1}\| \le \kappa \|\hat{s}_t - s_t\| + \delta$. Substitute the induction hypothesis:

  Case 1: $\kappa = 1$. Then

$$\|\hat{s}_{t+1} - s_{t+1}\| \le 1 \cdot (\epsilon_0 + t\delta) + \delta = \epsilon_0 + (t+1)\delta. \tag{22}$$

  Case 2: $\kappa \neq 1$. Then

$$\begin{aligned} \|\hat{s}_{t+1} - s_{t+1}\| &\le \kappa \left(\kappa^t \epsilon_0 + \frac{\kappa^t - 1}{\kappa - 1} \delta \right) + \delta \\ &= \kappa^{t+1} \epsilon_0 + \frac{\kappa^{t+1} - \kappa}{\kappa - 1} \delta + \delta \\ &= \kappa^{t+1} \epsilon_0 + \frac{\kappa^{t+1} - \kappa + \kappa - 1}{\kappa - 1} \delta \\ &= \kappa^{t+1} \epsilon_0 + \frac{\kappa^{t+1} - 1}{\kappa - 1} \delta. \end{aligned} \tag{23}$$

This completes the induction. □

The proposition characterizes the worst-case error propagation under a linear recurrence: exponential growth when $\kappa > 1$, and linear growth when $\kappa = 1$.

C.2 Proof of Lemma 4.3

Lemma 4.3 (Kinetic Energy Lower Bound and Wasserstein-2 Geodesics). Consider a probability path $p_h(\tau)$ evolving in time $h \in [0, 1]$ with velocity field $v_h(\tau)$ that satisfies the continuity equation

$\partial_h p_h + \nabla \cdot (p_h v_h) = 0$ and boundary conditions $p_0 = p_0, p_1 = p_{\text{target}}$. The the kinetic energy

$$\mathcal{K} = \int_0^1 \mathbb{E}_{\tau_h \sim p_h} \left[\|v_h(\tau_h)\|^2 \right] dh$$

satisfies $\mathcal{K} \geq W_2^2(p_0, p_{\text{target}})$, where W_2 is the Wasserstein-2 distance. Equality holds if and only if the path is a Wasserstein-2 geodesic, i.e., the velocity field is a gradient field of a convex function.

Proof. By the Benamou–Brenier formula, the squared 2-Wasserstein distance can be expressed as the infimum over all probability paths and velocity fields that satisfy the continuity equation and boundary conditions:

$$\begin{aligned} \mathcal{W}_2^2(p_0, p_{\text{target}}) = \inf_{(p_h, v_h)} \Bigg\{ \int_0^1 \int_{\mathbb{R}^d} \|v_h(\tau)\|^2 p_h(\tau) d\tau dh \Bigg| \partial_h p_h + \\ + \nabla \cdot (p_h v_h) = 0, p_0 = p_0, p_1 = p_{\text{target}} \Bigg\}, \end{aligned} \tag{24}$$

where d is the dimension of the trajectory τ.

For any admissible pair (p_h, v_h) satisfying the boundary conditions, the definition of the infimum directly implies

$$\mathcal{K} = \int_0^1 \mathbb{E}_{\tau_h \sim p_h} \left[\|v_h(\tau_h)\|^2 \right] dh \geq W_2^2(p_0, p_{\text{target}}). \tag{25}$$

This establishes the lower bound. Regarding the equality condition, optimal transport theory states that the unique minimizer of the above variational problem is given by the McCann interpolation, for which the optimal velocity field is a time-independent gradient field $v_h(\tau) = \nabla \phi(\tau)$ with ϕ convex, and the corresponding path is a constant-speed Wasserstein-2 geodesic. Conversely, if a pair (p_h, v_h) achieves the infimum $\mathcal{K} = W_2^2(p_0, p_{\text{target}})$, it must be a minimizer and thus possesses the gradient-field structure; hence it is a geodesic. □

Algorithm 3 Generate Trajectories($\mathcal{M}, \pi_\theta, Q_\phi, N_{\text{gen}}$)

$\mathcal{B}'_{\text{model}} \leftarrow \emptyset$
for $n = 1, \ldots, N_{\text{gen}}$ **do**
  Sample initial state s_0 from environment or replay buffer; set context $c = s_0$
  Sample prior $\tau_0 \sim \mathcal{N}(0, \mathbf{I})$ of shape $(d_s + d_a + 1) \times T_{\text{gen}}$
  Initialize $\tau \leftarrow \tau_0$, $h \leftarrow 0$, $\Delta h \leftarrow 1/N_{\text{ode}}$
  for $t = 1, \ldots, N_{\text{ode}}$ **do** ▷ ODE integration loop
    $h \leftarrow h + \Delta h$
    Extract partial state, action, reward from τ; compute discounted return $J(\tau)$
    Compute $\Upsilon(J) = \text{clip}\left(\exp\left(-\lambda_\Upsilon \frac{J - \mu_J}{\sigma_J}\right), \nu_{\min}, \nu_{\max}\right)$ ▷ μ_J, σ_J are batch statistics
    Compute mean vector field $\bar{\mathbf{v}} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \mathbf{v}_{\psi_k}(\tau, h, c)$
    Modulate vector field: $\mathbf{g} \leftarrow \Upsilon(J) \cdot \bar{\mathbf{v}}$
    Update trajectory: $\tau \leftarrow \tau + \Delta h \cdot \mathbf{g}$
  end for
  Store generated trajectory τ into $\mathcal{B}'_{\text{model}}$
end forreturn $\mathcal{B}'_{\text{model}}$

C.3 Proof of Theorem 4.4

Theorem 4.4 (Linear Error Propagation of OT-CFM Flows). Under Assumption 4.1, for each considered horizon T, let τ and $\hat{\tau}$ denote the true and OT-CFM generated trajectories on a specified coupled probability space with a common initial context. Assume a compact convex domain with uniformly positive true conditional densities, $W_2(p_{\text{CFM}}, p_{\text{true}}) \leq \delta$, and Lipschitz implicit kernels $\hat{\mu}, \hat{\pi}$ with conditional TV errors bounded by $C_{\text{cond}}\delta$. Suppose the state error $e_t = \hat{s}_t - s_t$ satisfies

the following recursion for $0 \le t < T-1$, with the bias and matrix-product bounds holding almost surely:

$$\begin{aligned} e_{t+1} &= D_t e_t + b_t + \eta_t, \quad e_0 = 0, \\ \|b_t\| &\le C_b\delta, \quad \mathbb{E}\|\eta_t\|^2 \le \epsilon^2, \\ \|D_{t-1}\cdots D_{k+1}\|_{\mathrm{op}} &\le 1, \quad 0 \le k < t < T, \end{aligned}$$

where an empty product is the identity, $C_b \ge 0$ is finite, D_t is the error propagation matrix and ϵ bounds the effective one-step state prediction error induced by CFM velocity approximation. For the fixed policy, context, and family of couplings, assume these bounds hold on each corresponding probability space with the same constants C_b, δ, ϵ, independent of t and T. Then, for some constant $C > 0$ independent of t and T,

$$\mathbb{E}\|\hat{s}_t - s_t\|^2 \le Ct^2 \max\{\delta^2, \epsilon^2\}, \quad 0 \le t < T.$$

Thus, the root-mean-square state error grows at most linearly with t under the stated conditions. The product bound and conditional approximation bounds are explicit assumptions; the OT penalty alone is not asserted to establish them.

Proof. We use the error recursion and matrix-product bound assumed in the theorem. Define $\Phi_{t,k+1} = D_{t-1}\cdots D_{k+1}$ for $0 \le k < t$, with $\Phi_{t,t} = I$. Since $e_0 = 0$, repeated substitution gives

$$e_t = \sum_{k=0}^{t-1} \Phi_{t,k+1}(b_k + \eta_k). \tag{26}$$

The assumed product and bias bounds imply, almost surely,

$$\|e_t\| \le tC_b\delta + \sum_{k=0}^{t-1} \|\eta_k\|. \tag{27}$$

For $t \ge 1$, applying $(a+b)^2 \le 2a^2 + 2b^2$ and Cauchy-Schwarz to the finite sum yields

$$\begin{aligned} \mathbb{E}\|e_t\|^2 &\le 2t^2 C_b^2\delta^2 + 2t\sum_{k=0}^{t-1} \mathbb{E}\|\eta_k\|^2 \\ &\le 2t^2(C_b^2\delta^2 + \epsilon^2) \\ &\le 2(C_b^2+1)t^2 \max\{\delta^2, \epsilon^2\}. \end{aligned} \tag{28}$$

The case $t = 0$ follows from the common initial state. Thus $C = 2(C_b^2+1)$ is valid. Independence and zero mean of η_k are not needed. The square-root bound is linear in t, using the uniform assumptions stated in the theorem.

□

C.4 Remark on Corollary 4.5

Corollary 4.5 states that conventional step-wise dynamics models typically satisfy $\mathbb{E}[\|\hat{s}_t - s_t\|^2] \le C'e^{\lambda t}\epsilon^2$ for some $\lambda > 0$. This follows directly from the autoregressive nature: each single-step prediction error gets amplified through the recurrence, corresponding to the case $\kappa > 1$ in Proposition 4.2. In contrast, OT-CFM constrains the flow to be a geodesic, which brings the effective propagation constant down to 1, thereby converting the error growth from exponential to linear and fundamentally mitigating the compounding error problem.

C.5 Proof of Theorem 4.7

Theorem 4.7 (Q-Estimation Equilibrium and Bias Bounds for Value-Guided Sampling). Fix the policy, initial state-action pair (s_0, a_0), normalization statistics, and a compactly supported return distribution p. Write $q = Q_\phi(s_0, a_0)$ and $\mu = \mathbb{E}_p[J]$. Let $\Upsilon(J)$ be a measurable, non-increasing weight with $0 < \nu_{\min} \le \Upsilon(J) \le \nu_{\max} < \infty$, and let $\bar{\Upsilon} = \mathbb{E}_p[\Upsilon(J)] > 0$. Consider the scalar expected-update model

$$\mathbb{E}[\Delta q] = a\,\mathbb{E}_p[\Upsilon(J)(J-q)], \qquad a > 0. \tag{29}$$

There is a unique equilibrium target Q^* for which this expected update vanishes, given by

$$Q^* = \frac{\mathbb{E}_p[\Upsilon(J)J]}{\bar{\Upsilon}} = \mu + \frac{\mathrm{Cov}_p(\Upsilon(J), J)}{\bar{\Upsilon}}. \tag{30}$$

Furthermore,

$$\mathrm{sign}\big(\mathbb{E}[\Delta q]\big) = \mathrm{sign}(Q^* - q). \tag{31}$$

Since $\Upsilon(J)$ is non-increasing, $\mathrm{Cov}_p(\Upsilon(J), J) \le 0$, yielding $Q^* \le \mu$. Its absolute deviation from the base mean is exactly

$$|Q^* - \mu| = \frac{|\,\mathrm{Cov}_p(\Upsilon(J), J)|}{\bar{\Upsilon}}. \tag{32}$$

For the normalized exponential weight $\Upsilon_\lambda(J) = \exp(-\lambda(J - \mu_J)/\sigma_J)$ with fixed $\sigma_J > 0$, as $\lambda \downarrow 0$,

$$Q^* - \mu = -\frac{\lambda}{\sigma_J}\,\mathrm{Var}_p(J) + \mathcal{O}(\lambda^2). \tag{33}$$

The same expansion applies to its clipped version when $0 < \nu_{\min} < 1 < \nu_{\max}$, since clipping is then inactive for sufficiently small λ. In particular, $Q^* \to \mu$ as $\lambda \downarrow 0$. Exact equality $Q^* = \mu$ holds whenever $\mathrm{Cov}_p(\Upsilon(J), J) = 0$, including constant weights.

Proof. All expectations below are taken under the fixed return distribution p.

Well-defined expectations. Compact support provides a finite $M \ge 1$ such that $|J| \le M$ almost surely. Consequently,

$$\mathbb{E}_p|\Upsilon(J)J| \le \nu_{\max} M < \infty, \qquad 0 < \nu_{\min} \le \bar{\Upsilon} \le \nu_{\max} < \infty. \tag{34}$$

Thus all expectations and covariance terms in the statement are finite, and division by $\bar{\Upsilon}$ is valid.

Unique equilibrium and update direction. With the return distribution and weights fixed, define

$$g(q) := \mathbb{E}[\Delta q] = a\big(\mathbb{E}_p[\Upsilon(J)J] - q\bar{\Upsilon}\big). \tag{35}$$

This is an affine function of the scalar q with strictly negative slope $-a\bar{\Upsilon}$. Hence it has exactly one zero, namely

$$Q^* = \frac{\mathbb{E}_p[\Upsilon(J)J]}{\bar{\Upsilon}}. \tag{36}$$

The covariance identity

$$\mathrm{Cov}_p(\Upsilon(J), J) = \mathbb{E}_p[\Upsilon(J)J] - \bar{\Upsilon}\mu \tag{37}$$

then gives

$$Q^* = \mu + \frac{\mathrm{Cov}_p(\Upsilon(J), J)}{\bar{\Upsilon}}, \qquad g(q) = a\bar{\Upsilon}(Q^* - q). \tag{38}$$

Since $a\bar{\Upsilon} > 0$,

$$\mathrm{sign}\, g(q) = \mathrm{sign}(Q^* - q), \tag{39}$$

including $q = Q^*$ under the convention $\mathrm{sign}(0) = 0$.

Let J' be an independent copy of J. Since Υ is non-increasing,

$$\begin{aligned} 2\,\mathrm{Cov}_p(\Upsilon(J), J) &= \mathbb{E}[(J - J')(\Upsilon(J) - \Upsilon(J'))] \\ &\le 0. \end{aligned} \tag{40}$$

Therefore $Q^* \le \mu$. Taking absolute values yields

$$|Q^* - \mu| = \frac{|\,\mathrm{Cov}_p(\Upsilon(J), J)|}{\bar{\Upsilon}}. \tag{41}$$

Small-strength expansion. Write Q_λ^* for the equilibrium associated with $\Upsilon_\lambda(J) = \exp(-\lambda(J - \mu_J)/\sigma_J)$. The statistics $\mu_J \in \mathbb{R}$ and $\sigma_J > 0$ are fixed. The common factor $\exp(\lambda\mu_J/\sigma_J)$ cancels, so

$$Q_\lambda^* = \frac{\mathbb{E}_p[Je^{-\lambda J/\sigma_J}]}{\mathbb{E}_p[e^{-\lambda J/\sigma_J}]}. \tag{42}$$

Fix a finite $\lambda_0 > 0$. Taylor's theorem gives

$$e^{-\lambda J/\sigma_J} = 1 - \frac{\lambda J}{\sigma_J} + R_\lambda(J), \qquad |R_\lambda(J)| \le K\lambda^2, \quad 0 \le \lambda \le \lambda_0, \tag{43}$$

where

$$K = \frac{M^2}{2\sigma_J^2} \exp\left(\frac{\lambda_0 M}{\sigma_J}\right) < \infty \tag{44}$$

is independent of J and λ in this range. In particular, $\mathbb{E}_p|R_\lambda(J)| \le K\lambda^2$ and $\mathbb{E}_p|JR_\lambda(J)| \le MK\lambda^2$. Taking expectations therefore gives

$$\begin{aligned} D_\lambda &:= \mathbb{E}_p[e^{-\lambda J/\sigma_J}] = 1 - \frac{\lambda\mu}{\sigma_J} + \mathcal{O}(\lambda^2), \\ N_\lambda &:= \mathbb{E}_p[Je^{-\lambda J/\sigma_J}] = \mu - \frac{\lambda}{\sigma_J}\mathbb{E}_p[J^2] + \mathcal{O}(\lambda^2). \end{aligned} \tag{45}$$

It follows that

$$N_\lambda - \mu D_\lambda = -\frac{\lambda}{\sigma_J}\left(\mathbb{E}_p[J^2] - \mu^2\right) + \mathcal{O}(\lambda^2). \tag{46}$$

Since $D_\lambda = 1 + \mathcal{O}(\lambda)$ and $D_\lambda \to 1$, its reciprocal is $1 + \mathcal{O}(\lambda)$ for sufficiently small λ. Thus

$$Q_\lambda^* - \mu = \frac{N_\lambda - \mu D_\lambda}{D_\lambda} = -\frac{\lambda}{\sigma_J}\operatorname{Var}_p(J) + \mathcal{O}(\lambda^2). \tag{47}$$

The remainder constants may depend on the fixed distribution, M, σ_J, and the chosen neighborhood of zero; they are independent of λ in that neighborhood.

Clipping and the zero-strength limit. Set $Z = (J - \mu_J)/\sigma_J$. There is a finite $L \ge 0$ with $|Z| \le L$ almost surely. Under $0 < \nu_{\min} < 1 < \nu_{\max}$, let

$$m = \min\{-\log \nu_{\min}, \log \nu_{\max}\} > 0. \tag{48}$$

If $L = 0$, the weights equal one and clipping is inactive for every λ. If $L > 0$, then for $0 \le \lambda < m/L$,

$$\nu_{\min} < e^{-\lambda L} \le e^{-\lambda Z} \le e^{\lambda L} < \nu_{\max} \quad \text{almost surely.} \tag{49}$$

Hence clipping does not alter the weights throughout this neighborhood, and the same expansion applies to the clipped equilibrium. In either case $Q_\lambda^* \to \mu$ as $\lambda \downarrow 0$. Finally, the exact covariance formula shows that $Q^* = \mu$ if and only if $\operatorname{Cov}_p(\Upsilon(J), J) = 0$; constant weights satisfy this condition. □

D Baselines

MBPO [6] A MBRL algorithm that strategically uses short model-generated rollouts to improve sample efficiency while maintaining asymptotic performance comparable to model-free methods. The core idea is to decouple the task horizon from the model horizon by generating short rollouts from states sampled from a replay buffer, rather than from the initial state distribution. This reduces compounding model errors and prevents model exploitation.

Theorem D.1 (Theorem 4.1 Monotonic Improvement Guarantee)**.** *The authors derive a bound relating the true return $\eta[\pi]$ and the model return $\hat{\eta}[\pi]$,*

$$\eta[\pi] \ge \hat{\eta}[\pi] - C(\epsilon_m, \epsilon_\pi)$$

where

$$C(\epsilon_m, \epsilon_\pi) = \frac{2\gamma r_{\max}(\epsilon_m + 2\epsilon_\pi)}{(1-\gamma)^2} + \frac{4r_{\max}\epsilon_\pi}{(1-\gamma)}$$

This ensures improvement in the real environment if the policy improves under the model by more than C. Then,

Theorem D.2 (Theorem 4.2 Branched Rollout Bound)**.** *For a k-step branched rollout starting from a real state*

$$\eta[\pi] \geq \eta^{\text{branch}}[\pi] - 2r_{\max}\left[\frac{\gamma^{k+1}\epsilon_\pi}{(1-\gamma)^2} + \frac{\gamma^k + 2}{(1-\gamma)}\epsilon_\pi + \frac{k}{1-\gamma}(\epsilon_m + 2\epsilon_\pi)\right]$$

This allows dynamic adjustment between model-based and model-free updates.

ADMPO The paper [13] introduces the *Any-step Dynamics Model (ADM)*, a model designed to reduce the compounding error in MBRL. Unlike standard single step models that predict the next state using only the current state-action pair, ADM can take a variable length history as input to directly predict a future state. This reduces the reliance on bootstrapped predictions, which are a primary source of accumulating error during long model rollouts. The objective of ADM is

$$J_T(\theta) = \frac{1}{m}\sum_{k=1}^{m}\mathbb{E}_{(s_t,a_{t:t+k-1},r_{t+k},s_{t+k})\sim\mathcal{D}_{\text{em}}}\left[\log \hat{T}_\theta(s_{t+k}, r_{t+k}|s_t, a_{t:t+k-1})\right]. \quad \text{Equation (2)}$$

The authors develop two algorithms: ADMPO-ON for online MBRL (improving sample efficiency) and ADMPO-OFF for offline MBRL (enabling better uncertainty quantification). In this work, we adopt its online RL variant as a baseline method, i.e., ADMPO-ON.

MACURA This paper [36] introduces a Dyna-style MBRL algorithm that adapts model-based rollout lengths based on local model accuracy rather than training time. Instead of asking *When to trust the model?*, MACURA asks *Where to trust the model?*. It defines a subset of the state space $\mathcal{E} \subseteq \mathcal{S}$ where the learned dynamics model $\tilde{p}$ is sufficiently accurate. Rollouts are allowed to continue only while they remain within $\mathcal{E}$.

The set $\mathcal{E}$ is constructed using a model uncertainty measure. For a probabilistic ensemble (PE) model, the epistemic uncertainty is quantified using the geometric Jensen–Shannon (GJS) divergence:

$$u_{\text{GJS}}(s,a) = \frac{2}{E(E-1)}\sum_{e=1}^{E}\sum_{f=1}^{e-1} D_{\text{GJS}}\left(\mathcal{N}_e \| \mathcal{N}_f\right), \quad \text{Equation (15)}$$

where $\mathcal{N}_i := \mathcal{N}(\mu_{\theta_i}(s,a), \Sigma_{\theta_i}(s,a))$. Then,

$$\mathcal{E} := \{s \in \mathcal{S} \mid u_{\text{GJS}}(s,a) < \kappa, a \sim \pi(\cdot \mid s)\}. \quad \text{Equation (20)}$$

Here κ is an adaptive uncertainty threshold.

MPPVE This paper [12] proposes a model-based planning policy learning with multi-step plan value estimation, a algorithm designed to improve sample efficiency by reducing the negative impact of model error in long-horizon rollouts. Instead of estimating the value of single-step actions in model generated states. MPPVE evaluates multi-step action plans directly from real states. It introduces a k-step plan value function $Q(s_t, \tau_t^k)$, where $\tau_t^k = (a_t, a_{t+1}, \cdots, a_{t+k-1})$, and updates the policy using gradients computed only from the initial real state s_t, avoiding reliance on error-prone fake states. Building upon these, a deep Q-network $Q_\psi(s_t, \tau_t^k)$ is trained to minimize the TD error

$$\hat{y}_{\psi^-} = \sum_{m=0}^{k-1}\gamma^m r_{t+m} + \gamma^k Q_{\psi^-}(s_{t+k}, \tau_{t+k}^k), \quad \text{Equation (16)}$$

where τ_{t+k}^k is sampled from the current planning policy.

FlowRL This paper [30] proposes an online reinforcement learning framework with flow-based generative policies constrained by Wasserstein-2 distance. Instead of using Gaussian or diffusion policies with expensive iterative sampling, FlowRL parameterizes the policy as a velocity field $v_\theta(t, s, a^t)$ and maximizes Q-values under a W-2 distance constraint to an optimal behavior policy. To avoid direct sampling from this optimal policy, it introduces an implicit advantage-based weighting function f to transform the constraint into a weighted conditional flow matching loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a\sim\mathcal{D},a'\sim\pi_\theta}\left[Q^{\pi_\theta}(s,a') - \lambda\left(f(Q^{\pi_{\beta^*}} - Q^{\pi_\theta})\|v_\theta - (a - a^0)\|^2 - \epsilon\right)\right],$$

where $f \propto \max(Q^{\pi_{\beta^*}} - Q^{\pi_\theta}, 0)$ adaptively guides the policy velocity field toward high-performing historical actions while enabling efficient one-step inference.

SDAC This paper [52] proposes an efficient online reinforcement learning algorithm that trains diffusion policies using a Reverse Sampling Score Matching (RSSM) mechanism. Instead of sampling from the inaccessible optimal policy or backpropagating through the expensive and unstable reverse diffusion process, SDAC treats diffusion models as noise-perturbed energy-based models and uses only the soft Q-function as the energy function. Building upon these, the score network s_θ is trained to minimize a Q-reweighted score matching loss:

$$\mathcal{L}^\pi(\theta; Q, \lambda) := \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{s, a_t, \tilde{a}_0} \left[\exp\left(\frac{\bar{Q}(s, \tilde{a}_0)}{\lambda} \right) \| s_\theta(a_t, s, t) - \nabla_{a_t} \log \tilde{q}_{0|t}(\tilde{a}_0 | a_t) \|^2 \right],$$

where $\bar{Q}$ is the normalized Q-function, and $\tilde{a}_0$ is generated by applying the reverse sampling trick to $a_t \sim \tilde{p}_t$, bypassing target policy sampling and recursive gradient backpropagation while maintaining lightweight computation.

E Environmental Setup

We conduct experiments on MuJoCo [31] tasks provided by the Gymnasium platform [50]. This section provides detailed descriptions of their setup and features.

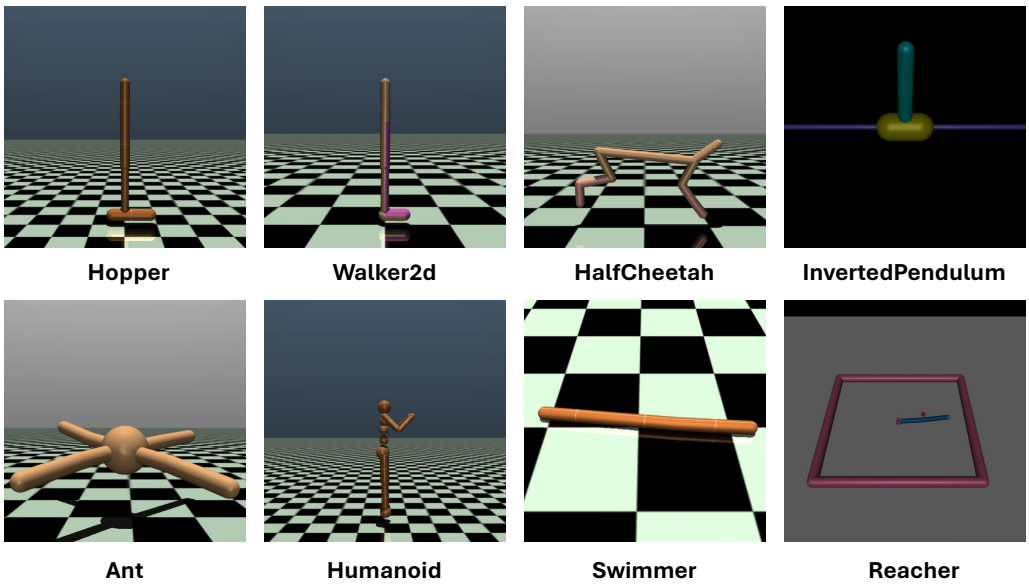


Figure 8: The eight continuous control benchmarks are standard MuJoCo tasks.

E.1 Hopper

This environment defines a two-dimensional single-leg locomotion task. The agent controls a hopper composed of four rigid body segments—torso, thigh, leg, and foot—connected by three joints. By applying torques at these joints, the agent must generate stable hopping motions that propel the body forward. The task is characterized by continuous dynamics and a relatively high-dimensional state and control space, requiring coordinated control for balance and forward movement.

State space

The hopper's state using joint positions and velocities. By default, it consists of 11 continuous values, including 5 position variables and 6 velocity variables, with the global x-position excluded. Optionally, the x-coordinate of the robot can be included, expanding the observation to 12 dimensions.

Action space

The action space is continuous with dimension 3. Each action corresponds to continuous torque values applied to the three hinge joints of the hopper.

Reward function

The reward function can be expressed as the following formulation

$$r_t = r_{\text{healthy}} + r_{\text{forward}} - r_{\text{ctrl}}.$$

Specifically,

1. Healthy reward

$$r_{\text{healthy}} = \begin{cases} \text{healthy_reward}, & \text{if the hopper is healthy at timestep } t, \\ 0, & \text{otherwise}, \end{cases}$$

2. Forward reward

$$r_{\text{forward}} = w_f \frac{x_{t+1} - x_t}{\Delta t},$$

where x_t denotes the torso's position along the forward (x) direction at timestep t. Δt is the time interval between actions, and w_f is the forward reward weight.

3. Control cost

$$r_{\text{ctrl}} = w_c \|a_t\|_2^2$$

where a_t is the continuous action (joint torques) applied at timestep t, and w_c is the control cost weight.

E.2 HalfCheetah

HalfCheetah is a two-dimensional locomotion task involving a planar, articulated robot with multiple linked body segments and joints. The agent controls the robot by applying continuous torques to its joints in order to generate coordinated running motions. The objective is to maximize forward velocity while maintaining stability, without explicit constraints on balance or contact failure. The task features continuous dynamics and a relatively high-dimensional state and action space, emphasizing efficient and smooth locomotion.

State space

The state space consists of joint positions and joint velocities of the robot with dimension 17. By default, it includes all internal configuration variables except the global x-position of the torso. The resulting state is a continuous vector capturing the kinematic and dynamic state of the system.

Action space

The action space is continuous with dimension 6. Each action corresponds to continuous torque commands applied to the six actuated joints of the HalfCheetah.

Reward function Except for the healthy reward, the rest is identical to Hopper.

E.3 Walker2d

Walker2d is a two-dimensional bipedal locomotion task in which the agent controls a simplified humanoid robot composed of a torso and two legs connected by multiple joints. By applying continuous torques at the joints, the agent must learn to walk forward while maintaining an upright posture and balance. The task requires coordinated control to prevent falling, combining stability maintenance with forward progression under continuous-time dynamics.

State space

The state space represents the walker's full kinematic state, including joint positions and velocities with dimension 17. By default, the global x-position is excluded, while orientation, joint angles, and their velocities are retained. The state is a continuous, high-dimensional state vector describing balance and locomotion status.

Action space

The action space is continuous with dimension 6. Each action specifies continuous torques applied to the six joints controlling the two legs of the walker.

Reward function Same as Hopper.

E.4 InvertedPendulum

The Inverted Pendulum is a classic control task in which the agent must balance a rigid pendulum upright on a cart. The system consists of a single pendulum connected to a movable cart along a one-dimensional track. The agent applies continuous forces to the cart to prevent the pendulum from falling over. The task emphasizes stability and precise control under continuous dynamics.

State space

The state space is continuous with dimension 4 and typically consists of cart position and velocity, pendulum angle and angular velocity. These variables form a low-dimensional vector that fully describes the system's state. By default, the global x-position of the cart may be excluded or included depending on the environment configuration.

Action space

The action space is continuous with dimension 1. Each action corresponds to a continuous force applied to the cart along the horizontal axis.

Reward function

The reward at each timestep can be expressed as

$$r_t = r_{\text{alive}} - r_{\text{ctrl}}$$

where

1. Alive reward

$$r_{\text{alive}} = \begin{cases} 1, & \text{if the pendulum is upright at timestep } t, \\ 0, & \text{otherwise.} \end{cases}$$

2. Control cost

$$r_{\text{ctrl}} = w_c a_t^2,$$

where a_t is the continuous force applied at timestep t and w_c is a small weight penalizing large actions.

E.5 Ant

Ant is a three-dimensional quadrupedal locomotion task in which the agent controls a four-legged robot composed of a central torso and four legs, each with multiple joints. The agent must apply continuous torques to learn coordinated trotting or galloping gaits to move forward efficiently. Compared to planar tasks, Ant involves complex 3D dynamics, requiring the agent to manage balance and body orientation while navigating the environment.

State space

The state space includes the global position and orientation of the torso, joint angles, and their velocities. It is typically high-dimensional with a dimension of 111 (or 27 if contact forces are excluded), capturing both the kinematic and dynamic states of the quadruped.

Action space

The action space is continuous with dimension 8. Each action corresponds to continuous torque commands applied to the eight joints controlling the four legs.

Reward function The reward primarily encourages forward velocity, with penalties on control effort and, in some variants, contact forces to promote smooth and energy-efficient locomotion.

E.6 Humanoid

Humanoid is a three-dimensional bipedal locomotion task that presents one of the most challenging continuous control benchmarks. The agent controls a humanoid figure with a torso, two arms, and two legs connected by numerous joints. The objective is to achieve stable forward walking or running while maintaining an upright posture. The high degree of freedom and the complex dynamics make balance maintenance and coordination extremely difficult.

State space

The state space is of very high dimension (e.g., 376 dimensions in standard MuJoCo settings). It includes the positions and velocities of all body parts, joint angles, and external forces, requiring the agent to process a rich set of proprioceptive information.

Action space

The action space is continuous with dimension 17. Each action specifies continuous torques applied to the joints in the legs, arms, and torso.

Reward function The reward is typically composed of a forward progress term, a healthy reward for maintaining an upright position without falling, and a penalty on the magnitude of the control torques to encourage natural and efficient movements.

E.7 Swimmer

Swimmer is a three-dimensional locomotion task simulating a snake-like or fish-like robot moving in a viscous fluid environment. The agent controls a multi-link body through continuous joint torques to propel itself forward. The task emphasizes the ability to generate efficient undulating motions in a fluid medium, where the dynamics are heavily influenced by drag forces.

State space

The state space is relatively low-dimensional with dimension 8. It consists of the joint angles, joint velocities, and the position and velocity of the front tip of the swimmer.

Action space

The action space is continuous with dimension 2. Each action corresponds to the torque applied to the two actuated joints connecting the body segments.

Reward function The reward is proportional to the forward velocity of the swimmer, typically with a small penalty on control effort to encourage smooth motion.

E.8 Reacher

Reacher is a two-dimensional manipulation task involving a two-link robotic arm. Unlike the locomotion tasks, the objective here is to control the arm's joints to bring its end-effector (fingertip) to a randomly generated target position in the plane. The task requires precise control and continuous adjustment to reach and stay at the target.

State space

The state space has a dimension of 10. It includes the cosine and sine of the two joint angles, the angular velocities, and the coordinates and velocities of the target position.

Action space

The action space is continuous with dimension 2. Each action corresponds to the continuous torque commands applied to the two joints of the arm.

Reward function The reward is typically dense and based on the negative distance between the end-effector and the target position, often supplemented by a small penalty on the action magnitude to encourage smooth control.

F Implementation Details

We adopt Soft Actor-Critic (SAC) [51] as our base RL algorithm because of its stability. In practice, for each update we sample a mini-batch from $\mathcal{B}_{\text{env}}$ and another from $\mathcal{B}_{\text{model}}$ with concentrating $\mathcal{D}$. The critic network is trained by minimizing the soft Bellman residual:

$$J_Q(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\phi(s,a) - \left(r + \gamma\mathbb{E}_{a'\sim\pi_\theta}\left[\min_{i=1,2} Q_{\text{target},i}(s',a') - \alpha\log\pi_\theta(a'|s')\right]\right)\right)^2\right]. \tag{50}$$

where a clipped double-Q mechanism is employed to mitigate the overestimation bias prevalent in actor-critic methods. Simultaneously, the actor network is optimized by minimizing the objective:

$$J_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\alpha\log\pi_\theta(a|s) - \min_{i=1,2} Q_{\phi_i}(s,a)\right] \tag{51}$$

utilizing the reparameterization trick to allow for stable backpropagation through the policy's stochastic action selection. To maintain a consistent exploration pressure throughout training, the temperature parameter α is dynamically adjusted by minimizing:

$$J(\alpha) = \mathbb{E}_{a\sim\pi_\theta}\left[-\alpha\log\pi_\theta(a|s) - \alpha\bar{\mathcal{H}}\right], \tag{52}$$

where $\bar{\mathcal{H}}$ denotes the target entropy, typically set to the negative dimension of the action space.

G Experimental Details

Our experiments are conducted on a machine equipped with an Intel Xeon CPU, an NVIDIA GeForce RTX 3090 GPU, and Ubuntu 20.04 operating system. The models are implemented using Pytorch. Below, we list the detailed parameters and settings used during model training.

We first introduce the hyperparameters used for training the trajectory-level generative model. As described in the previous section, the generative model is implemented with a Transformer as the backbone network. Specifically, we employ an 8-layer Transformer, where each hidden layer has a dimensionality of 256, and the number of attention heads is set to 8.

Subsequently, task-specific training parameters are configured according to task difficulty. Taking the Hopper-v4 task as an example, the model is trained once every 250 environment steps. For each training phase, the batch size is set to 128, while the number of training epochs is dynamically determined based on the size of the environment data buffer. The number of epochs is computed as

$$e = \min(\log(|\mathcal{B}_{\text{env}}+1|) * \mathcal{B}_k * \mathscr{T}, e_{\max}),$$

where $e_{\max}$ denotes the maximum number of training epochs, and

$$\mathcal{B}_k = 1 + \frac{\mathcal{B}_{\text{env}}}{|\mathcal{B}_{\text{env}}|_{\max}}$$

represents the sampling ratio from the environment replay buffer. For the Hopper task, we set $e_{\max} = 500$. During training, a different trajectory generation length $T \in [1, \mathscr{T}]$ is randomly selected for each iteration, where $\mathscr{T}$ denotes the predefined maximum generation length. In Hopper, we set $\mathscr{T} = 20$.

After the completion of each training phase, trajectory data can be generated either immediately or at fixed intervals measured in environment steps. In the Hopper task, we synchronize the sampling interval with the training interval, generating trajectories every 250 environment steps. Since trajectory sampling must wait for model parameter updates, we adopt an environment-step-based schedule to determine the appropriate generation length

$$T = f(N) = \min(\max(x + \frac{t-a}{b-a}\cdot(y-x), x), y),$$

where N denotes the number of environment steps, and a, b, x, y are parameters defined in the rollout schedule. For Hopper-v4, we set $a = 0, b = 30k, x = 3$ and $y = 20$, corresponding to the start and

end environment steps (a, b), and the minimum and maximum generation lengths (x, y), respectively. During sampling, the batch size is set to 3000. Trajectories are initialized according to the selected length by sampling from a Gaussian distribution, with the first time step replaced by a real transition from the environment. The initialized trajectories are then used as inputs to the ODE solver, with the integration interval fixed to $[0, 1]$ and the number of integration steps set to 100. For value-guided sampling, the guidance coefficient is set to $\lambda_\Upsilon = 1.0$. We adapt Adam as the optimizer for training the model and the learning rate of it is 2×10^{-4}.

All other tasks use the same parameter settings as Hopper-v4. Detailed configurations for each task are summarized in the corresponding Table 2. Then, we also provide the parameters and settings of SAC [51] for policy optimization in Table 3.

Table 2: Parameters and settings in different tasks

Hyperparameter	**Walker2d**	**HalfCheetah**	**InvertedPendulum**	**Ant**	**Humanoid**	**Swimmer**	**Reacher**
Model Training Interval	250	250	250	250	250	250	250
Train Batch Size	256	256	256	512	512	256	256
Maximum Training Epochs	600	600	300	600	600	600	600
Maximum Generated Length	20	20	10	20	20	20	20
Model Sampling Interval	250	250	250	250	250	250	250
Total Training time steps	200k	200k	15k	100k	100k	200k	50k
Rollout Schedule	0, 1e5, 2, 20	1e4, 1e5, 2, 20	2e3, 1e4, 3, 10	1e4, 1e5, 3, 20	1e4, 1e5, 3, 20	1e4, 1e5, 3, 20	1e4, 3e4, 3, 20
Sampling Batch Size	8000	8000	5000	6000	6000	8000	8000
Guidance Coefficient	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 3: Parameters and settings in different tasks

Hyperparameter	**Value**
The Number of Critics	2
The Number of FC layers in Actor Network	256,256
The Number of FC layers in Critic Network	256,256
Target Network Smoothing Coefficient	5×10^{-3}
Discount Factor	0.99
Learning Rate of Actor	1×10^{-4}
Learning Rate of Critic	3×10^{-4}
Optimizers of the Actor and Critics	Adam
Batch Size	256

H Computational Complexity Analysis

We analyze the computational cost of FTA from two perspectives: (i) training the conditional flow matching (CFM) model with optimal transport (OT) regularization, and (ii) generating synthetic trajectories via value-guided ODE sampling. Let $d_x = d_s + d_a + 1$ denote the per-step dimension of the flattened trajectory representation, T the sampled trajectory length (maximum $\mathscr{T}$), B the batch size, and K the number of ensemble heads in the CFM architecture. The backbone is a Scalable Interpolant Transformer [48], which scales quadratically with sequence length.

H.1 Training Complexity

Each training iteration processes a mini-batch of B real trajectory segments of length $T \sim \mathcal{U}(1, \mathscr{T})$. The computational graph consists of:

- **Forward pass.** A transformer forward pass on a sequence of length T and dimension d_x requires $\mathcal{O}(BT^2 d_x + BTd_x^2)$ operations, dominated by the self-attention mechanism. For K heads, the output projection multiplies the hidden state into K distinct vector fields; this increases the final layer cost by a factor of K but remains a lower-order term compared to the shared backbone.
- **Loss computation.** The CFM loss (Eq. 4) evaluates the squared error between predicted and target velocities. The OT regularizer (Eq. 5) adds penalty terms involving the kinetic energy and the state displacement constraint; the integral $\int_0^1 v_\psi^{(s)} dh$ is approximated via numerical quadrature along the interpolation path, contributing $\mathcal{O}(BTd_x)$. The per-head uncertainty weights (Equation 6) require computing the mean prediction and variance across heads, with negligible overhead $\mathcal{O}(BTd_xK)$.
- **Backward pass.** The gradient computation scales proportionally to the forward pass, yielding an overall training iteration complexity of

$$\mathcal{O}\big(BT^2 d_x + BTd_x^2 + BTd_xK\big). \tag{53}$$

Since T is randomly sampled per batch, the expected cost is governed by the average length $\mathbb{E}[T] = (\mathscr{T} + 1)/2$. Compared to standard step-wise dynamics models (e.g., an MLP that predicts the next state in $\mathcal{O}(Bd_x^2)$), the CFM backbone is more expensive due to the transformer's quadratic dependence on T. However, the overhead is substantially amortized by the fact that a single forward pass synthesizes an entire trajectory, avoiding the sequential error accumulation that forces step-wise models to use short rollouts or expensive uncertainty estimation. In practice, $\mathscr{T} \leq 20$ or 50 keeps the training cost manageable on a single GPU.

H.2 Sampling Complexity

Generating a synthetic trajectory of length T_{gen} (a fixed horizon during deployment) with value-guided ODE sampling requires N_{ode} integration steps. At each integration step h, the following operations are performed:

- **Vector field evaluation:** Forward pass through the transformer backbone with all K heads, averaged to obtain $\bar{v}(\tau_h, h, c)$. Complexity per step is $\mathcal{O}(T_{\text{gen}}^2 d_x + T_{\text{gen}} d_x^2 + T_{\text{gen}} d_x K)$.
- **Value guidance:** Compute the discounted return $J(\tau_h)$ from the partial trajectory and apply the modulation function $\Upsilon(J)$ (Eq. 13). This involves a scan over T_{gen} steps, costing $\mathcal{O}(T_{\text{gen}})$.
- **Euler update:** $\tau_{h+\Delta h} = \tau_h + \Delta h \cdot \Upsilon(J)\, \bar{v}$, at $\mathcal{O}(T_{\text{gen}} d_x)$.

The dominant term is the N_{ode} repeated transformer forward passes, giving a total sampling complexity per trajectory of

$$\mathcal{O}\big(N_{\text{ode}}(T_{\text{gen}}^2 d_x + T_{\text{gen}} d_x^2)\big). \tag{54}$$

In our experiments, $T_{\text{gen}} \in \{20, 50\}$ and $N_{\text{ode}} \in [10, 50]$, making generation computationally more demanding than a single forward pass of an MLP dynamics model but still efficient enough for online use: generating a batch of 5000 trajectories (typical for a model buffer refill) takes $\sim$ seconds on a modern GPU. Importantly, the ODE solver is invoked only periodically (every F_{model} environment steps), so the amortized cost per real interaction remains low.

H.3 Comparison with Step-wise Models

Classical MBRL approaches (e.g., MBPO, MPPVE) employ autoregressive dynamics models that predict one step at a time. For a rollout of length H, they perform H sequential forward passes, each costing $\mathcal{O}(d_x^2)$ (for a standard MLP) or $\mathcal{O}(L_{\text{enc}} d_x^2)$ (for an ensemble). In contrast, FTA generates the entire trajectory in one shot but requires N_{ode} forward passes of a more expensive transformer. Thanks

to the OT-regularized geodesic structure, FTA achieves linear error propagation while step-wise models suffer from exponential growth; the extra computation is therefore a worthwhile investment in sample efficiency and policy stability.

Algorithm Name	Environment Steps to Reach 11,000 Score	Total Training Time
MBPO	240k	8h
FTA (ours)	150k	12h
ADMPO	210k	10h

Table 4: Performance comparison between FTA and baseline algorithms. We report the number of environment steps required to reach a target score of 11,000 and the corresponding total wall-clock training time.

As shown in Table 4, on a single NVIDIA 3090 GPU, FTA requires approximately $1.2 - 1.5\times$ the total training time of the baselines. However, due to its substantially improved sample efficiency, the number of training steps needed to reach 90% of final performance is reduced by 40–60% compared to MBPO and ADMPO in HalfCheetah.

I Additional Experiments Results

I.1 The Main Results on MuJoCo

Table 5: Performance on MuJoCo environments over random seeds. The best result in each environment is highlighted in bold.

Algorithm	Hopper	Walker2d	HalfCheetah	InvertedPendulum	Ant	Humanoid	Swimmer	Reacher
				Foundation RL				
SAC	409.00 ± 135.40	1303.40 ± 1006.53	6821.20 ± 262.77	1000.00 ± 0.00	45.83 ± 91.34	545.36 ± 76.62	35.87 ± 2.16	-7.42 ± 20.88
MBPO	1033.00 ± 166.15	2885.63 ± 552.12	9505.36 ± 590.69	1000.00 ± 0.00	1838.78 ± 216.37	948.91 ± 418.82	96.98 ± 28.67	-4.63 ± 0.39
				Advanced MBRL				
ADMPO	2595.11 ± 809.51	3878.06 ± 595.43	9925.70 ± 472.78	1000.00 ± 0.00	2773.20 ± 1064.73	3975.57 ± 1021.71	44.88 ± 3.07	-4.02 ± 0.37
MACURA	2480.47 ± 774.48	5276.01 ± 403.38	11411.76 ± 362.65	212.72 ± 440.51	4022.54 ± 213.93	5602.83 ± 415.17	-1.64 ± 2.69	-13.47 ± 12.50
MPPVE	2210.96 ± 1091.91	3987.06 ± 859.49	10686.28 ± 1638.38	1000.00 ± 0.00	2897.78 ± 505.06	4624.32 ± 209.12	136.25 ± 6.83	-3.79 ± 1.00
				Generative Policies				
FlowRL	515.37 ± 198.21	3319.65 ± 1397.87	4606.33 ± 482.45	1000.00 ± 0.00	1916.93 ± 1345.83	4472.11 ± 1927.24	48.36 ± 21.04	-4.51 ± 1.73
SDAC	235.95 ± 21.46	376.01 ± 403.38	751.30 ± 703.09	30.00 ± 0.00	391.28 ± 102.07	417.00 ± 105.48	46.31 ± 2.49	-30.88 ± 2.46
FTA (ours)	**3333.58 ± 74.10**	**5511.38 ± 32.41**	**11762.30 ± 487.35**	**1000.00 ± 0.00**	**5795.53 ± 89.32**	**5605.18 ± 444.73**	**144.94 ± 7.04**	**-3.40 ± 0.23**

As reported in Table 5, FTA demonstrates superior quantitative performance across almost all MuJoCo benchmarks, consistently outperforming both traditional RL and generative baselines by notable margins. The most striking numerical advantage is observed in the Ant environment, where FTA achieves a return of 5795.53, surpassing the second-best method (MACURA at 4022.54) by approximately 44%, and outperforming the generative baselines FlowRL (1916.93) and SDAC (391.28) by a factor of 3 to 14. A similarly substantial gap is evident in Hopper, with FTA reaching 3333.58 compared to the next highest ADMPO (2595.11). Furthermore, on complex, high-dimensional locomotion tasks like Walker2d and HalfCheetah, FTA not only secures the highest mean returns (5511.38 and 11762.30, respectively) but also exhibits significantly lower variance (e.g., $\pm$32.41 on Walker2d) than advanced MBRL baselines like MACURA ($\pm$403.38) and MPPVE ($\pm$859.49), indicating more stable learning. While FTA ties for the maximum score of 1000.00 on the simpler InvertedPendulum task, it strictly dominates on Swimmer and Reacher as well. Overall, the numerical differences highlight that FTA achieves robust asymptotic performance, drastically narrowing the performance gap between generative policies and MBRL methods.

I.2 The Exploration Ability of FTA

The empirical discrepancy between generated data and real data in FTA serves as a direct validation of the value-guided mechanism's ability to drive robust exploration. As shown in Figure 9, while MBPO exhibits a collapse toward local real-data clusters, FTA leverages its trajectory-level alignment to produce synthetic states that are more diverse and further from the current buffer.

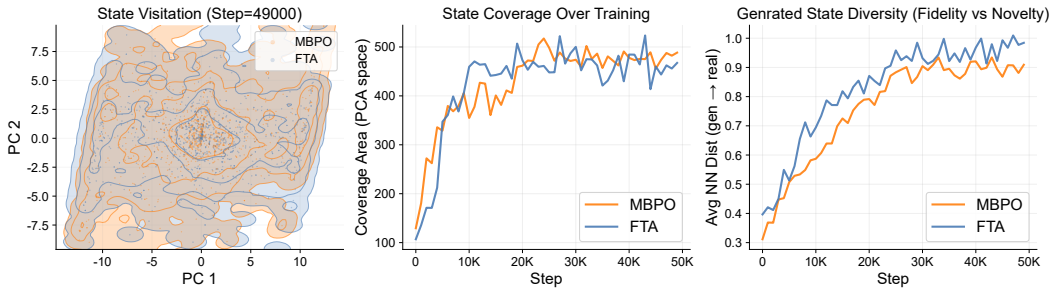


Figure 9: Comparative analysis of state-space exploration and synthetic data fidelity. (a) State Visitation (projected via PCA) at 49,000 steps illustrates that FTA achieves a significantly broader reach across the state manifold, whereas the MBPO baseline is largely confined to narrow clusters. (b) State Coverage Over Training quantifies this advantage, showing that FTA consistently maintains a higher cumulative coverage area throughout the learning process. (c) Generated State Diversity measures the average nearest-neighbor distance between generated states and the real-data buffer; the higher distance in FTA indicates the generation of novel states that systematically extend beyond the existing data distribution to facilitate global discovery.

This behavior can be partly understood through the equilibrium properties of the weighted TD objective. Theoretically, when the weighting function $\Upsilon(J)$ is non-increasing, the equilibrium satisfies $Q^* \leq \mu$, yielding a conservative (downward-biased) estimate. In model-based settings, such conservatism may help mitigate the over-optimistic value estimates caused by model inaccuracies, which can lead to premature exploitation. This provides a theoretical perspective on conservative value estimation, while its effect on exploration is evaluated empirically.

Crucially, the higher diversity in FTA's generated data is accompanied by stable learning performance in our experiments. Theorem 4.7 shows that the expected update in the weighted-update model satisfies $\text{sign}(\mathbb{E}[\Delta q]) = \text{sign}(Q^* - q)$, directing the estimate toward the conservative equilibrium Q^*. This may mitigate overestimation bias, although a reduction in absolute estimation bias is not guaranteed. Together, these findings suggest that FTA can achieve a wider exploration footprint while maintaining stable policy optimization.

I.3 State Transition Accuracy of Generated Data

The results in Figure 10 highlight the robustness of FTA framework in maintaining long-horizon fidelity. In vanilla CFM, errors in individual state transitions accumulate exponentially as the rollout length increases, leading to model drift where the terminal state of a generated sequence diverges significantly from reality. This is evident in the $H = 15$ plot, where Vanilla CFM's similarity plateaus at a lower level with noticeably wider confidence intervals. FTA overcomes this by shifting the learning objective from local, step-by-step transitions to global trajectory-level alignment. By incorporating OT principles—specifically leveraging the Benamou-Brenier formulation to minimize kinetic energy along the flow, FTA ensures that the entire generated path remains structurally consistent with real-world dynamics. This global constraint effectively pins the terminal state to the correct region of the manifold, even for longer horizons. From a theoretical standpoint, this high fidelity is the necessary counterpart to the value-guided mechanism discussed previously. While the VG scheme introduces a controlled, conservative bias ($Q^* \leq \mu$) to prevent over-exploitation, the high cosine similarity shown here ensures that this pessimism is applied to accurate representations of the environment. Without the high-fidelity trajectory alignment provided by FTA, the agent might be conservative about the wrong states; with it, the agent can reliably explore the vicinity of the true expected return using synthetic data that truthfully reflects the underlying MDP.

I.4 Visualization of Sampling Trajectory

We provide a qualitative visualization analysis of the trajectories generated by FTA in order to gain intuitive insights into the effects of different sampling strategies on trajectory quality. Specifically, we save trajectory samples generated under two settings: value-guided sampling and non–value-guided sampling. To facilitate visualization, we apply dimensionality reduction to the high-dimensional

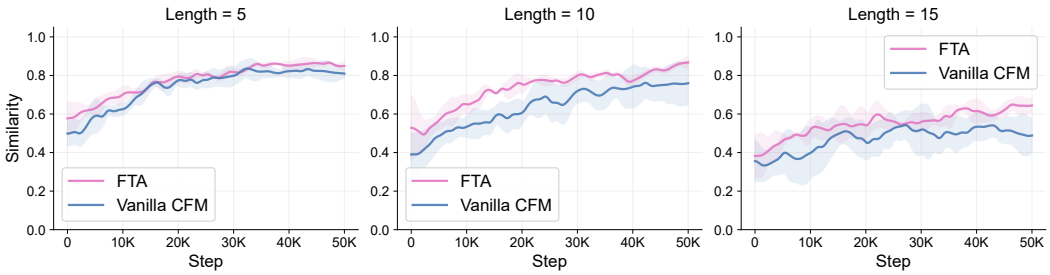


Figure 10: Fidelity of terminal state predictions across varying trajectory lengths. We compare the cosine similarity between the terminal states of generated trajectories and environment transitions for rollout lengths $H \in \{5, 10, 15\}$.

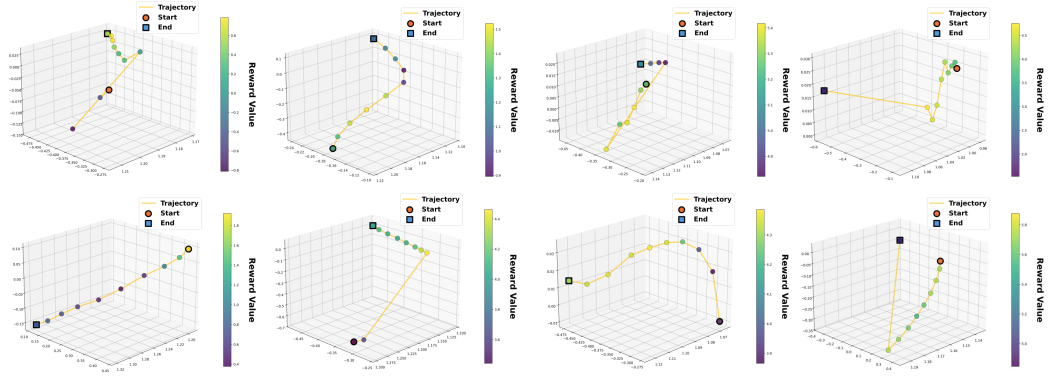


Figure 11: Visualization of trajectory sampled by FTA with value-guided sampling in HalfCheetah

state space and project the resulting trajectories into a three-dimensional space, as illustrated in Figures 11 and 12. In the visualization, the starting point and ending point of each trajectory are explicitly marked. Each circular marker corresponds to a single environment state at a specific time step. To further characterize how rewards evolve along the trajectory, we color-code the state points according to the magnitude of the reward obtained at that state, enabling a direct comparison of reward progression in the state space. For clarity and to avoid visual clutter caused by excessively long rollouts, we generate trajectories with relatively short horizons during sampling. From the results, we observe that trajectories generated via value-guided sampling exhibit a clear and consistent trend of gradually increasing rewards, in contrast to those generated without value guidance. Moreover, the state transitions of value-guided trajectories form smoother and more coherent paths in the low-dimensional space, indicating more stable dynamics. Notably, these trajectories rarely suffer from transition collapse, where the generated states abruptly deviate from plausible regions of the state space or converge to degenerate dynamics. These observations suggest that incorporating value guidance effectively steers the trajectory generation process toward high-reward regions, while the control energy minimization constraint introduced during training plays a critical role in suppressing abrupt or unstable state transitions. Together, these components significantly improve the stability and realism of the generated trajectories. This qualitative evidence further substantiates the necessity and effectiveness of enforcing control energy regularization in the FTA framework.

I.5 Visualization of Testing Video

We utilize the built-in episode video recording functionality provided by the MuJoCo environment to capture several evaluation rollouts, thereby obtaining dynamic visualizations of the agent's ground-truth behavioral trajectories during real environment interactions. These recorded videos offer an intuitive view of the agent's action execution, state evolution, and overall motion patterns over continuous time steps, serving as qualitative evidence for assessing the realism and stability of the

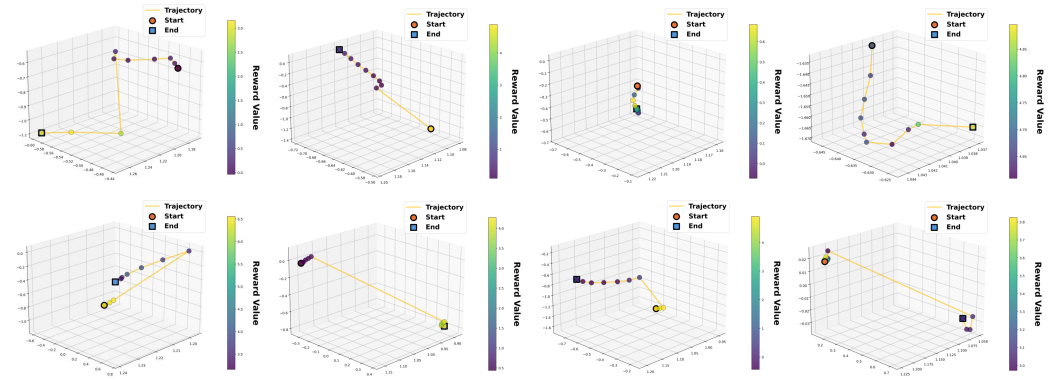


Figure 12: Visualization of trajectory sampled by FTA without value-guided sampling in HalfCheetah

learned behavior. We show the video of two tasks: Hopper-v4 and Walker2d-v4 in Figures 13, 14, respectively.

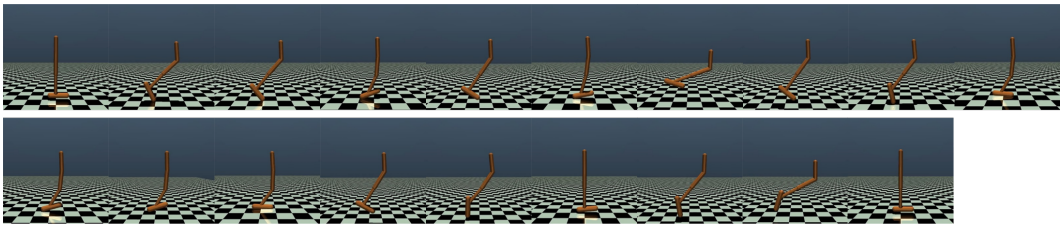


Figure 13: Test video of the Hopper task at 20,000 environment steps.

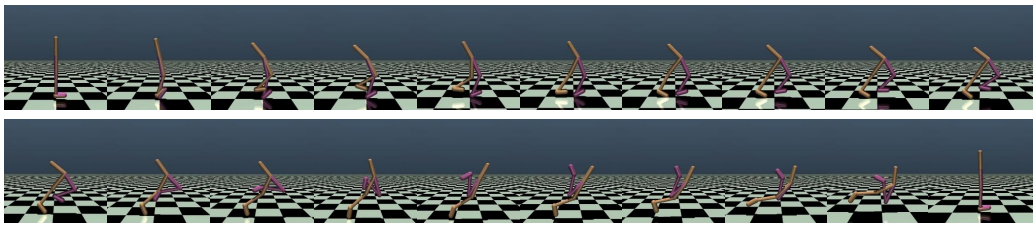


Figure 14: Test video of the Walker2d task at 150,000 environment steps.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the proposed FTA framework, its components (CFM, OT regularizer, value-guided sampling), theoretical contributions (linear error propagation, value-estimation equilibrium analysis), and experimental results on MuJoCo, all of which are supported by the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 explicitly discusses limitations, including ODE solving overhead, dependence on Q-function accuracy, potential guidance bias early in training, and challenges with very long sequences or pixel-based environments.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Assumptions are stated (Assumption 4.1) and all theorems/lemmas are numbered, cross-referenced, and proved either in the main text with proof sketches or in the appendix. Relied-upon theorems (e.g., Benamou–Brenier) are properly cited.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper provides architecture details (SiT backbone), training objectives, hyperparameters (see Appendix / parameter tables), and baselines; all are described to an extent that allows reproducibility, though code is not yet released for anonymity.

Guidelines:

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

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

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

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

   Answer: [No]

   Justification: The implementation and model checkpoints are not publicly available at the time of camera-ready submission. The paper provides algorithmic details, pseudocode, and experimental configurations to support reproducibility.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 and Appendices E–G describe the benchmark environments, implementation details, network architectures, hyperparameters, and training configurations. Additional experimental results are provided in Appendix I.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All learning curves report mean returns and standard deviation over multiple random seeds (stated in Figure 2 caption), providing a measure of statistical variability.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the detail of computer resources in the appendix G.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics; it uses standard RL benchmarks, involves no human subjects, and respects anonymity.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The work is foundational research on model-based RL and does not present immediate societal impacts. The methods could potentially improve sample efficiency in real-world robotics, but no direct harmful application is foreseen.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper releases neither pre-trained models that could be misused nor scraped datasets; it only reports results on standard MuJoCo tasks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We cite the original papers for all baselines (MBPO, ADMPO, etc.) and the MuJoCo and Gymnasium libraries. All are standard and used in accordance with their licenses. The experiments use Gymnasium (MIT License) and MuJoCo (Apache License

2.0). The corresponding original works are properly cited, and the software is used in accordance with its licensing terms.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: This work introduces a new learning framework but does not release any new datasets or pretrained models as part of the submission.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects research.

Guidelines:

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

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

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

Answer: [N/A]

Justification: No human subjects were involved, so IRB approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs were not used as an important, original, or non-standard component of the proposed method; they were only used for minor writing edits, which does not require declaration per the NeurIPS policy.

Guidelines:

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