Composite Flow Matching for Reinforcement Learning with Shifted-Dynamics Data

Lingkai Kong* **Haichuan Wang*** **Tonghan Wang*** **Guojun Xiong** **Milind Tambe**
School of Engineering and Applied Sciences
Harvard University

Abstract

Incorporating pre-collected offline data can significantly improve the sample efficiency of reinforcement learning (RL), but this benefit is often challenged by discrepancies between the transition dynamics of the offline data and the online environment. Existing methods typically address this issue by penalizing or filtering out offline transitions in high dynamics-gap regions. However, their estimation of the dynamics gap often relies on KL divergence or mutual information, which can be ill-defined when the source and target dynamics have different support. To overcome these limitations, we propose CompFlow, a method grounded in the theoretical connection between flow matching and optimal transport. Specifically, we model the online dynamics as a conditional flow built upon the output distribution of the offline flow, rather than learning it directly from a Gaussian prior. This composite structure offers two key advantages: (1) improved generalization for learning online dynamics, and (2) a principled estimation of the dynamics gap via the Wasserstein distance between offline and online transitions. Leveraging our principled estimation of the dynamics gap, we further introduce an optimistic active data collection strategy that prioritizes exploration in regions of high dynamics gap, and theoretically prove that it reduces the performance disparity with the optimal policy. Empirically, CompFlow outperforms strong baselines across various RL benchmarks with shifted dynamics.

1 Introduction

Reinforcement Learning (RL) has demonstrated remarkable performance in complex sequential decision-making tasks such as playing Go and Atari games [52, 54], supported by access to large amounts of online interactions with the environment. However, in many real-world domains such as robotics [26, 59], healthcare [67], and wildlife conservation [29, 30, 64, 65], access to such interactions is often prohibitively expensive, unsafe, or infeasible. The limited availability of interactions presents a major challenge for learning effective and reliable policies. To address this challenge and improve sample efficiency during online training, a promising strategy is to incorporate a pre-collected offline dataset generated by a previous policy [44, 58]. This approach enables the agent to learn from a broader set of experiences, which can help accelerate learning and improve performance [58].

A critical challenge in RL with offline data arises when the transition dynamics of the offline dataset differ from those of the online environment, where the agent is actively interacting and learning [49]. This issue, commonly referred to as *shifted dynamics*, can introduce severe mismatches that bias policy updates, destabilize the learning process, and ultimately degrade performance. For instance,In robotics, differences in physical parameters such as friction can result in transition dynamics that diverge from the source domain. In the conservation domain, historical data collected during one period may not accurately represent current conditions, as changes in environmental conditions and ecological patterns over time can lead to shifts in poaching behavior [64].

*Equal contribution. Corresponding author: lingkaikong@g.harvard.edu

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

To address these challenges, existing methods either penalize the rewards or value estimates of offline transitions with high dynamics gap [37, 46], or filter out such transitions entirely [62]. However, these approaches face key limitations. Most notably, the estimation of the dynamics gap typically relies on KL divergence or mutual information, both of which can be ill-defined when the offline and online transition dynamics have different supports [2, 48].

In this paper, we propose CompFlow, a new method for RL with shifted-dynamics data that leverages the theoretical connection between flow matching and optimal transport. We model the transition dynamics of the online environment using a composite flow architecture, where the online flow is defined on top of the output distribution of a learned offline flow rather than being initialized from a Gaussian prior. This design enables a principled estimation of the dynamics gap using the Wasserstein distance between the offline and online transition dynamics. On the theoretical side, we show that the composite flow formulation reduces generalization error compared standard flow matching by reusing structural knowledge embedded in the offline data, particularly when online interactions are limited.

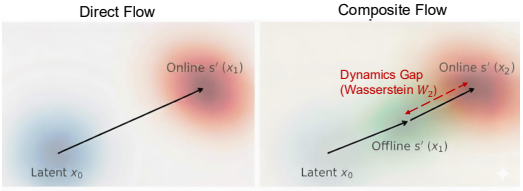

Figure 1: Comparison between direct and composite flow matching. Composite flow first transports from a Gaussian latent variable to the offline transition distribution, then adapts to the online distribution via optimal transport flow matching.

Building on the dynamics gap estimated by composite flow matching using the Wasserstein distance, we go beyond selectively merging offline transitions with low dynamics gap and further propose an active data collection strategy that targets regions in the online environment where the dynamics gap relative to the offline data is high. Such regions are often underrepresented in the replay buffer due to the dominance of low-gap samples. Our theoretical analysis shows that targeted exploration in these high-gap regions can further close the performance gap with respect to the optimal policy.

Our contributions are summarized as follows: (1) We introduce a composite flow model that estimates the dynamics gap by computing the Wasserstein distance between conditional transition distributions. We provide theoretical analysis showing that this approach achieves lower generalization error compared to learning the online dynamics from scratch. (2) Leveraging this principled estimation, we propose a new data collection strategy that encourages the policy to actively explore regions with high dynamics gap. We also provide a theoretical analysis of its performance benefits. (3) We empirically validate our method on various RL benchmarks with shifted dynamics and demonstrate that CompFlow outperforms or matches state-of-the-art baselines across these tasks.

2 Problem Statement and Background

2.1 Problem Definition

We consider two infinite-horizon MDPs: $\mathcal{M}_{\text{off}} := (\mathcal{S}, \mathcal{A}, p_{\text{off}}, r, \gamma)$ and $\mathcal{M}_{\text{on}} := (\mathcal{S}, \mathcal{A}, p_{\text{on}}, r, \gamma)$, sharing the same state/action spaces, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in (0, 1)$, but differing in transition dynamics:

$$p_{\text{off}}(s'|s, a) \neq p_{\text{on}}(s'|s, a) \quad \text{for some } (s, a).$$

We assume rewards are bounded, i.e., $|r(s, a)| \leq r_{\max}$ for all s, a. For any policy π, let $(s_t, a_t)_{t\geq 0}$ be the trajectory generated by $\mathcal{M}$ and π. We define the discounted state–action visitation (occupancy) measure as $\rho^{\pi}_{\mathcal{M}}(s, a) := (1 - \gamma)\,\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t\, \mathbf{1}\{s_t = s,\, a_t = a\}]$, and the discounted state visitation as $d^{\pi}_{\mathcal{M}}(s) := \sum_a \rho^{\pi}_{\mathcal{M}}(s, a)$. The expected return is $\eta_{\mathcal{M}}(\pi) := \mathbb{E}_{(s,a)\sim\rho^{\pi}_{\mathcal{M}}}[r(s, a)]$.

Definition 2.1 (Online Policy Learning with Shifted-Dynamics Offline Data)**.** Given an offline dataset $\mathcal{D}_{\text{off}} = \{(s_i, a_i, s'_i, r_i)\}_{i=1}^{N}$ from $\mathcal{M}_{\text{off}}$ and limited online access to $\mathcal{M}_{\text{on}}$, the objective is to learn a policy π maximizing $\eta_{\mathcal{M}_{\text{on}}}(\pi)$, ideally approaching $\eta_{\mathcal{M}_{\text{on}}}(\pi^\star)$, where $\pi^\star := \arg\max_\pi \eta_{\mathcal{M}_{\text{on}}}(\pi)$.

Lemma 2.2 (Return Bound between Two Environments [37])**.** *Let the empirical behavior policy in* $\mathcal{D}_{\text{off}}$ *be* $\pi_{\mathcal{D}_{\text{off}}}(a \mid s)$*. Define* $C_1 = \frac{2r_{\max}}{(1-\gamma)^2}$*. Then for any policy* π*,*

$$\begin{aligned}\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\mathcal{M}_{\text{off}}}(\pi) \geq\; & -2C_1\, \mathbb{E}_{(s,a)\sim\rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}},\, s'\sim p_{\text{off}}} \left[D_{\text{TV}}(\pi(\cdot|s') \parallel \pi_{\mathcal{D}_{\text{off}}}(\cdot|s'))\right] \\ & - C_1\, \mathbb{E}_{(s,a)\sim\rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}}} \left[D_{\text{TV}}(p_{\text{on}}(\cdot|s, a) \parallel p_{\text{off}}(\cdot|s, a))\right].\end{aligned}$$

This bound highlights two key sources of return gap between domains: (1) the mismatch between the learned policy and the behavior policy in the offline dataset, and (2) the shift in environment dynamics. The former can be mitigated through behavior cloning [37, 63], while the latter can be addressed by filtering out source transitions with large dynamics gaps [39, 62]. A central challenge lies in accurately estimating this gap. Existing methods often rely on KL divergence or mutual information [37, 46], which can be ill-defined when the two dynamics have different supports.

2.2 Flow Matching

In this paper, we will adopt flow matching [1, 34, 36] to model transition dynamics in reinforcement learning, owing to its ability to capture complex distributions. Flow Matching (FM) offers a simpler alternative to denoising diffusion models [21, 56], which are typically formulated using stochastic differential equations (SDEs). In contrast, FM is based on deterministic ordinary differential equations (ODEs), providing advantages such faster inference, and often improved sample quality.

The goal of FM is to learn a time-dependent velocity field $v_\theta(x,t) : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, parameterized by θ, which defines a flow map $\psi_\theta(x_0, t)$. This map is the solution to the ODE

$$\frac{d}{dt}\psi_\theta(x_0,t) = v_\theta(\psi_\theta(x_0,t),t), \quad \psi_\theta(x_0,0) = x_0,$$

and transports samples from a simple source distribution $p_0(x)$ (e.g., an isotropic Gaussian) at time $t = 0$ to a target distribution $p_1(x)$ at time $t = 1$. In practice, generating a sample from the target distribution involves drawing $x_0 \sim p_0(x)$ and integrating the learned ODE to obtain $x_1 = \psi_\theta(x_0, 1)$.

A commonly used training objective in flow matching is the *linear path matching loss*

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_0\sim p_0(x_0), x_1\sim p_1(x_1)}\left[\|v_\theta(x_t,t) - (x_1 - x_0)\|_2^2\right], \tag{1}$$

where $x_t = (1-t)x_0 + tx_1$ denotes the linear interpolation between x_0 and x_1. Using linear interpolation paths encourages the learned flow to follow nearly straight-line trajectories, which reduces discretization error and improves the computational efficiency of ODE solvers during sampling [36].

2.3 Optimal Transport Flow Matching and Wasserstein Distance

Optimal Transport Flow Matching (OT-FM) establishes a direct connection between flow-based modeling and Optimal Transport (OT), providing a principled framework for quantifying the discrepancy between two distributions. This connection is especially valuable in our setting, where a key challenge is to measure the distance between two conditional transition distributions.

Let $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a cost function. Optimal transport aims to find a coupling $q^\star \in \Pi(p_0, p_1)$—a joint distribution with marginals p_0 and p_1—that minimizes the expected transport cost:

$$\inf_{q\in\Pi(p_0,p_1)} \int c(x_0,x_1)\,\mathrm{d}q(x_0,x_1).$$

This minimum defines the Wasserstein distance $W_c(p_0, p_1)$ between the two distributions under the cost function c. When $c(x_0, x_1) = \|x_0 - x_1\|^2$, the resulting distance is known as the squared 2-Wasserstein distance.

In the training objective of Eq. 1, when sample pairs (x_0, x_1) are drawn from the optimal coupling $q^\star$, the flow model trained with the linear path matching loss learns a vector field that approximates the optimal transport plan. The transport cost of the learned flow is equal to the Wasserstein distance

$$\mathbb{E}_{x_0\sim p_0}\left[\|\psi_\theta(x_0,1) - x_0\|_2^2\right] = W_2^2(p_0,p_1).$$

For theoretical justification, see Theorem 4.2 in [47].

In practice, the optimal coupling $\pi^\star$ is approximated using mini-batches by solving a discrete OT problem between empirical samples from p_0 and p_1. The use of mini-batch OT has also been shown to implicitly regularize the transport plan [13, 14], as the stochasticity from independently sampled batches induces behavior similar to entropic regularization [9]. Further details of the OT-FM training procedure are provided in Appendix C.

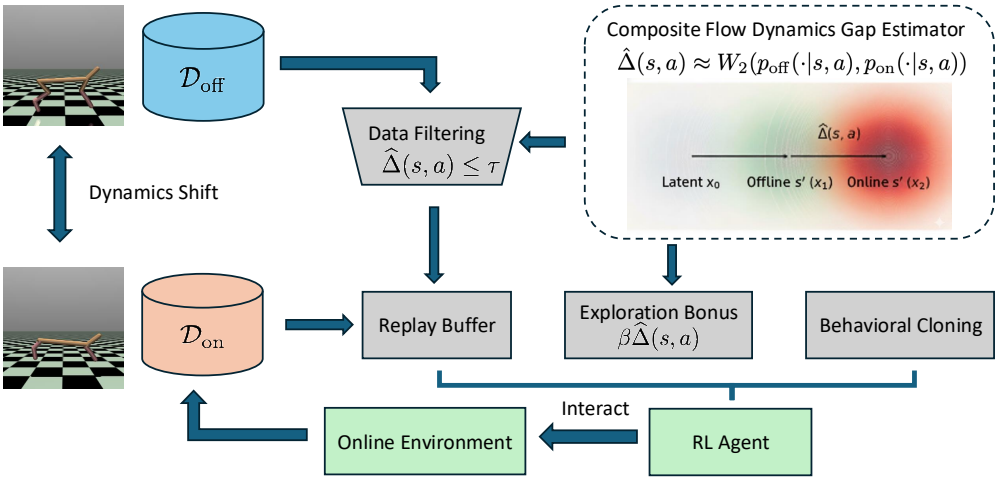

Figure 2: Overall Framework of CompFlow. To estimate the dynamics gap, we propose composite flow matching, which computes the Wasserstein distance between offline and online transition dynamics. Guided by the estimated dynamics gap, we augment policy training with offline transitions that exhibit low discrepancy from the online dynamics, and incorporate a behavior cloning objective to stabilize learning. To enhance data diversity and facilitate adaptation, we further encourage exploration in regions with high dynamics gap.

3 Proposed Method

In this section, we introduce our method, CompFlow. We begin by presenting a composite flow matching approach to estimate the dynamics gap via Wasserstein distance. Next, we propose a data collection strategy that actively explores high dynamics-gap regions and provide a theoretical analysis of its benefits. Finally, we describe the practical implementation details of our method.

3.1 Estimating Dynamics Gap via Composite Flow

To estimate the gap between dynamics, we first learn the transition models for both the offline dataset and the online environment. Flow matching provides a flexible framework for modeling complex transition dynamics; however, the limited number of samples available in the online environment presents a significant challenge. Training separate flow models for each environment can result in poor generalization in the online environment, as the model may overfit to the small amount of available data.

To mitigate this, we propose a **composite flow** formulation. Instead of learning the online transition model $p_{\text{on}}(s'|s,a)$ from scratch, we leverage structural knowledge from a well-trained offline model $p_{\text{off}}(s'|s,a)$. This enables to incorporate prior knowledge to improve the generalization.

Offline flow. We begin by learning a conditional flow model for the offline data. Let $x_0 \sim \mathcal{N}(0, \mathbf{I})$ be the initial latent variable. The offline flow map $\psi_\theta^{\text{off}}(x_0, t|s,a)$ is defined as the solution to the following ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_\theta^{\text{off}}(x_0,t|s,a) = v_\theta^{\text{off}}(\psi_\theta^{\text{off}}(x_0,t|s,a),t,s,a), \quad \psi_\theta^{\text{off}}(x_0,0|s,a) = x_0.$$

Solving this ODE from $t=0$ to 1 produces an intermediate representation: $x_1 = \psi_\theta^{\text{off}}(x_0,1|s,a)$.

Online flow. Instead of learning a online flow directly from a Gaussian prior, we initialize it from the intermediate representation x_1 produced by the offline flow. The target flow map $\psi_\phi^{\text{on}}(x,t|s,a)$ is defined as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_\phi^{\text{on}}(x,t|s,a) = v_\phi^{\text{on}}(\psi_\phi^{\text{on}}(x,t|s,a),t,s,a), \quad \psi_\phi^{\text{on}}(x,1|s,a) = x_1.$$

Solving this ODE from $t=1$ to 2 yields the final prediction for the online environment: $s' \doteq x_2 = \psi_\phi^{\text{on}}(x_1,2|s,a)$.

We now show that when the offline flow induces a distribution $\hat{p}_{\text{off}}(s'|s,a)$ that is closer to $p_{\text{on}}(s'|s,a)$ than the standard Gaussian distribution, the proposed composite flow method enjoys a smaller generalization error bound.

Theorem 3.1 (Conditions for Composite Flow Yielding Smaller Errors)**.** *Assume that composite flow and direct flow share the same hypothesis class $\mathcal{H}$ of (measurable) vector fields $v : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, and that there exists $B \in (0,\infty)$ such that $\sup_{v\in\mathcal{H}} \sup_{x\in\mathbb{R}^d,\, t\in[0,1]} \|v(x,t)\|_2 \le B$. Also, assume that $\max\{\mathrm{Tr}(\Sigma_{\text{on}}), \mathrm{Tr}(\widehat{\Sigma}_{\text{off}})\} \le C_{\text{TR}}$ for some C_{TR}. The composite flow enjoys a strictly tighter high-probability generalization bound than the direct flow if and only if*

$$W_2(p_G, p_{\text{on}}) > W_2(\hat{p}_{\text{off}}, p_{\text{on}}) \tag{2}$$

Here, $p_G := \mathcal{N}(0, I_d)$, Σ_{on} is the covariance of p_{on}, and $\widehat{\Sigma}_{\text{off}}$ is the covariance of $\hat{p}_{\text{off}}$.

Remark 3.2. In our setting, we assume that the offline data and the online environment share meaningful similarities. Given the abundance of offline data available to accurately learn the offline flow, this assumption is expected to hold well.

To compute the Wasserstein distance between the offline and online transition dynamics, we can use the OT-FM objective to train the online flow, initialized from the offline flow distribution. However, when the state-action pair (s,a) lies in a continuous space, it is not feasible to obtain a batch of samples corresponding to a fixed (s,a) from the online environment.

To address this, we follow prior works [18, 23] that handle samples with distinct conditioning variables. The key idea is to incorporate the conditioning variables into the cost function when solving for the optimal transport coupling. Specifically, we define the cost between two sample pairs $(s_{\text{off}}, a_{\text{off}}, s'_{\text{off}})$ and $(s_{\text{on}}, a_{\text{on}}, s'_{\text{on}})$ as:

$$c\big((s_{\text{off}}, a_{\text{off}}, s'_{\text{off}}, s_{\text{on}}, a_{\text{on}}, s'_{\text{on}}\big) = \|s'_{\text{off}} - s'_{\text{on}}\|_2^2 + \eta\left(\|s_{\text{off}} - s_{\text{on}}\|_2^2 + \|a_{\text{off}} - a_{\text{on}}\|_2^2\right),$$

where $\eta > 0$ is a weighting coefficient that controls the influence of the conditional terms.

Then the training objective of the online flow follows

$$\mathcal{L}_{\text{on}}(\phi) = \mathbb{E}_{((s_{\text{off}},a_{\text{off}},s'_{\text{off}} \doteq x_1),(s_{\text{on}},a_{\text{on}},s'_{\text{on}} \doteq x_2))\sim q^*, t\sim\mathcal{U}[1,2]}\left[\left\|v_\phi\left(x_t, t, s_{\text{on}}, a_{\text{on}}\right) - (s'_{\text{on}} - s'_{\text{off}})\right\|^2\right],$$

where q^* is the optimal coupling of empirical distribution $(s_{\text{off}}, a_{\text{off}}, s'_{\text{off}})$ generated by learned source flow and empirical transition $(s_{\text{on}}, a_{\text{on}}, s'_{\text{on}})$ from target domain replay buffer. x_t is the linear interpolation between x_1 and x_2.

The complete training algorithms of the offline flow and the online flow are in Appendix C.

Proposition 3.3 (Informal; shared-latent coupling is W_2-optimal)**.** Given $(s,a) \in \mathcal{S}\times\mathcal{A}$. As $\eta \to \infty$ and the training batch size $\to \infty$, the Wasserstein distance between the source and target transition distributions satisfies

$$W_2^2(p_{\text{off}}(\cdot|s,a), p_{\text{on}}(\cdot|s,a)) = \mathbb{E}_{x_0\sim\mathcal{N}(0,\mathbf{I})}\left[\left\|\psi_\theta^{\text{off}}(x_0,1|s,a) - \psi_\phi^{\text{on}}(\psi_\theta^{\text{off}}(x_0,1|s,a),2|s,a)\right\|_2^2\right].$$

Monte Carlo Estimator. This result yields a practical Monte Carlo estimator of the *dynamics gap* $\Delta(s,a)$ using shared latent variables:

$$\widehat{\Delta}(s,a) = \left(\frac{1}{M}\sum_{j=1}^{M}\left\|\psi_\theta^{\text{off}}(x_0^{(j)},1|s,a) - \psi_\phi^{\text{on}}(\psi_\theta^{\text{off}}(x_0^{(j)},1|s,a),2|s,a)\right\|_2^2\right)^{\frac{1}{2}}, \quad x_0^{(j)} \sim \mathcal{N}(0,\mathbf{I}). \tag{3}$$

3.2 Data Collection at High Dynamics Gap Region

As discussed in Section 2.1, using behavior cloning and augmenting the replay buffer with low dynamics gap offline data can alleviate performance drop from distribution shift. However, relying solely on such data may limit state-action coverage and hinder policy learning. To address this, we propose a new data collection strategy.

At each training iteration, we construct the replay buffer by selectively incorporating offline transitions with small estimated dynamics gap:

$$\mathcal{B} = \left\{(s,a) \in \mathcal{D}_{\text{off}} : \widehat{\Delta}(s,a) \le \tau\right\} \cup \mathcal{D}_{\text{on}}, \tag{4}$$

where τ is a predefined threshold. To improve data diversity and encourage better generalization, we actively explore regions with high dynamics gap—areas likely underrepresented in the buffer due to the dominance of low-gap samples. We adopt an optimistic exploration policy that selects actions by

$$a = \arg\max_{a \in \mathcal{A}} \left[Q(s,a) + \beta \, \widehat{\Delta}(s,a) \right], \tag{5}$$

where β is a hyperparameter that trades off return and exploration of underexplored dynamics. To ensure sufficient coverage during training, we can further incorporate stochasticity by adding small perturbations to the selected actions [17] or following a stochastic policy [19] consistent with widely used deep RL frameworks.

Theorem 3.4 (Large Dynamics Gap Exploration Reduces Performance Gap)**.** *Compared to behavior cloning policy π_{bc} on the offline dataset, training a policy $\hat{\pi}$ by replacing all offline samples with a dynamics gap exceeding κ (as estimated by the composite flow) with online environment samples can reduce the performance gap to the optimal online policy π^*_{on} with high probability by*

$$\frac{2L_r(1+\gamma)}{(1-\gamma)(1-\gamma L_p)} \big(\Delta_{W_2} - \kappa - \sqrt{(C_0 + C_1 \, W_2(\hat{p}_{\mathrm{off}}, p_{\mathrm{on}}))\, \Gamma_{N_{\mathrm{on}},\delta}} \big). \tag{6}$$

Here L_r and L_P are the Lipschitz constants for the reward and transition function, respectively. $\gamma L_p < 1$. $\Delta_{W_2} := \sup_{s,a} W_2\big(p_{\mathrm{off}}(\cdot|s,a),\ p_{\mathrm{on}}(\cdot|s,a)\big)$ is the largest dynamics gap. C_0 and C_1 are two constants. $\Gamma_{N_{\mathrm{on}},\delta} := \mathfrak{R}_{N_{\mathrm{on}}}(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{N_{\mathrm{on}}}}$, where $\mathcal{H}$ is the same as in Theorem 3.1 and N_{on} is the number of samples used to train the online flow.

From Theorem 3.4, exploration in regions with high dynamics gap can reduce the performance gap relative to the optimal policy. The parameter κ serves as a threshold: the bound holds when the policy is trained without offline samples whose dynamics gap exceeds κ. As β increases, more samples are collected from high-gap regions, which decreases κ. This increases performance gap reduction, resulting in a policy with higher expected return.

3.3 Practical Implementation

Our method can be instantiated using standard actor-critic algorithms with a critic $Q_\varsigma(s,a)$ and a policy $\pi_\varphi(a|s)$. To incorporate the dynamics gap, we apply rejection sampling to retain a fixed percentage of offline transitions with the lowest estimated gap in each iteration. The critic is trained by minimizing

$$\mathcal{L}_Q = \mathbb{E}_{(s,a,s')\sim D_{\mathrm{on}}}[(Q_\varsigma(s,a) - y)^2] + \mathbb{E}_{(s,a,s')\sim D_{\mathrm{off}}}[\mathbf{1}(\widehat{\Delta}(s,a) \le \widehat{\Delta}_{\xi\%})(Q_\varsigma(s,a) - y)^2], \tag{7}$$

where $\widehat{\Delta}_{\xi\%}$ is the ξ-quantile of the estimated dynamics gap in the offline minibatch. The target value is $y = r + \gamma Q_\varsigma(s',a') + \beta\widehat{\Delta}(s,a)$, with $a' \sim \pi_\varphi(\cdot|s')$.

The policy is updated using a combination of policy improvement and behavior cloning, with the following objective:

$$\mathcal{L}_\pi = \mathbb{E}_{s\sim D_{\mathrm{off}}\cup\mathcal{D}_{\mathrm{on}},\, a\sim\pi_\varphi(\cdot|s)}[Q_\varsigma(s,a)] - \omega\, \mathbb{E}_{(s,a)\sim\mathcal{D}_{\mathrm{off}},\, \tilde{a}\sim\pi_\varphi(\cdot|s)}[\|a-\tilde{a}\|^2], \tag{8}$$

where ω is a hyperparameter that balances policy improvement with imitation of source actions. The behavior cloning term encourages sampled actions from the policy to match those in the offline dataset as suggested in Lemma 2.2.

The pseudocode of CompFlow, instantiated with Soft Actor-Critic (SAC) [19], is presented in Appendix D.

4 Related Work

Online RL with offline dataset. Online RL often requires extensive environment interactions [53, 66], which can be costly or impractical in real-world settings. To improve sample efficiency, Offline-to-Online RL leverages pre-collected offline data to bootstrap online learning [43]. A typical two-phase approach trains an initial policy offline, then fine-tunes it online [32, 43, 44]. However, conservative strategies used to mitigate distributional shift, such as pessimistic value estimation [31], can result in suboptimal initial policies and limit effective exploration [40, 44]. To resolve this tension, recent methods propose ensemble-based pessimism [32], value calibration [44], optimistic

action selection [32], and policy expansion [68]. Others directly incorporate offline data into the replay buffer of off-policy algorithms [3], improving stability via ensemble distillation and layer normalization. However, these works assume identical dynamics between the source and target environments. In contrast, our work explicitly addresses the dynamics shift between the offline and online environments.

RL with dynamics shift. Our work is related to cross-domain RL, where the source and target domains share the same observation and action spaces but differ in transition dynamics. Prior work has addressed such discrepancies via system identification [5, 8, 10], domain randomization [41, 55, 60], imitation learning [20, 24], and meta-RL [42, 50], often assuming shared environment distributions [57] or requiring expert demonstrations. More recent work has relaxed these assumptions. One line of research focuses on *reward modification*, which adjusts the reward function to penalize source transitions that are unlikely under the target dynamics [12, 35], or down-weights value estimates in regions with high dynamics gap [46]. Another line of work explores *data filtering*, which selects only source transitions with low estimated dynamics gap [63], using metrics such as transition probability ratios [12], mutual information [62], value inconsistency [63], or representation-based KL divergence [37]. However, these metrics can become unstable or ill-defined when the transition dynamics have different supports, and value-based methods are prone to instability caused by bootstrapping bias.

In contrast, our approach leverages the theoretical connection between optimal transport and flow matching to estimate the dynamics gap in a principled manner. A closely related work by [39] also applies optimal transport to quantify the dynamics gap. However, their setting assumes both the source and target domains are offline, and their method estimates the gap based on the concatenated tuple (s, a, s') observed in offline data, rather than comparing conditional transition distributions. As a result, their metric does not accurately capture the gap in transition dynamics and cannot be used to compute exploration bonuses, which require gap estimation conditioned on a given state-action pair.

RL with diffusion and flow models. Diffusion models [21, 28, 56] and flow matching [11, 34] have emerged as powerful generative tools capable of modeling complex, high-dimensional distributions. Their application in RL is consequently expanding. Researchers have employed these generative models for various tasks, including planning and trajectory synthesis [22], representing expressive multimodal policies [7, 51], providing behavior regularization [6], or augmenting training datasets with synthesized experiences. While these works often focus on policy learning or modeling dynamics within a single environment, our approach targets the transfer learning setting. Specifically, we address scenarios characterized by a *dynamics gap* between the offline data and the online environment. We utilize Flow Matching, leveraging its connection to optimal transport, to estimate this gap.

5 Experiments

In this section[2], we first evaluate our approach across a range of environments in Gym-MuJoCo that exhibit different types of dynamics shifts. We then conduct ablation and hyperparameter studies to better understand the design choices and behavior of CompFlow. Finally, we assess the effectiveness of our method in a real-world inspired wildlife conservation task.

5.1 Gym-MuJoCo

5.1.1 Experimental Setup

Tasks and datasets. We evaluate our algorithm under three types of dynamics shifts, namely morphology, kinematic and friction changes, across three OpenAI Gym locomotion tasks: HalfCheetah, Hopper, and Walker2d [4]. Each experiment involves an offline environment and an online environment with modified transition dynamics following [38]. Morphology shifts alter the sizes of body parts, kinematic shifts impose constraints on joint angles, and friction shifts modify the static, dynamic, and rolling friction coefficients. For each task, we use three D4RL source datasets: medium, medium replay, and medium expert, which capture varying levels of data quality [16]. Additional environment details are in Appendix J.

Baselines. We compare CompFlow against the following baselines: BC-SAC extends SAC by incorporating both offline and online data, with a behavior cloning (BC) term for the offline data. H2O [46] penalizes Q-values for state-action pairs with large dynamics gaps. BC-VGDF [63] selects

[2]Our code is available at `https://github.com/Haichuan23/CompositeFlow`

Dataset	Task Name	SAC	BC-SAC	H2O	BC-VGDF	BC-PAR	Ours
MR	HalfCheetah (Morphology)	1457 ± 89	2495 ± 43	1430 ± 408	2765 ± 124	1790 ± 91	3119 ± 107
MR	HalfCheetah (Kinematic)	2255 ± 197	4868 ± 186	4257 ± 609	4392 ± 403	4179 ± 441	5189 ± 262
MR	HalfCheetah (Friction)	2069 ± 184	7799 ± 157	6397 ± 673	7829 ± 821	8056 ± 512	8241 ± 180
MR	Hopper (Morphology)	364 ± 82	346 ± 4	361 ± 18	348 ± 21	354 ± 25	355 ± 6
MR	Hopper (Kinematic)	737 ± 547	1024 ± 0	1025 ± 0	1024 ± 0	1024 ± 1	1024 ± 1
MR	Hopper (Friction)	234 ± 4	228 ± 2	229 ± 1	230 ± 3	232 ± 5	280 ± 27
MR	Walker2D (Morphology)	253 ± 60	598 ± 475	1014 ± 193	672 ± 576	458 ± 151	1094 ± 791
MR	Walker2D (Kinematic)	152 ± 22	2973 ± 185	1967 ± 851	1586 ± 923	948 ± 131	1568 ± 1315
MR	Walker2D (Friction)	301 ± 9	311 ± 5	296 ± 14	302 ± 12	321 ± 26	344 ± 20
M	HalfCheetah (Morphology)	1467 ± 89	1522 ± 72	1720 ± 273	1829 ± 345	1427 ± 196	2282 ± 287
M	HalfCheetah (Kinematic)	2316 ± 92	5451 ± 195	5019 ± 773	4972 ± 381	5243 ± 120	5593 ± 44
M	HalfCheetah (Friction)	2028 ± 238	7108 ± 1001	6968 ± 846	6802 ± 956	7800 ± 525	7871 ± 238
M	Hopper (Morphology)	396 ± 60	436 ± 45	410 ± 8	406 ± 52	418 ± 13	604 ± 173
M	Hopper (Kinematic)	724 ± 535	1022 ± 1	970 ± 98	934 ± 43	1020 ± 3	1023 ± 2
M	Hopper (Friction)	229 ± 5	232 ± 1	228 ± 4	229 ± 4	233 ± 5	300 ± 66
M	Walker2D (Morphology)	301 ± 177	457 ± 317	577 ± 201	584 ± 219	431 ± 177	886 ± 372
M	Walker2D (Kinematic)	258 ± 174	1966 ± 1155	1965 ± 568	1921 ± 928	806 ± 278	2039 ± 936
M	Walker2D (Friction)	301 ± 9	286 ± 54	298 ± 45	289 ± 47	308 ± 24	320 ± 31
ME	HalfCheetah (Morphology)	1392 ± 238	1195 ± 241	1147 ± 169	1072 ± 102	1207 ± 53	1485 ± 67
ME	HalfCheetah (Kinematic)	2323 ± 97	4211 ± 262	5143 ± 330	4603 ± 498	4399 ± 164	5750 ± 84
ME	HalfCheetah (Friction)	1950 ± 312	4185 ± 732	2140 ± 733	4078 ± 1032	4989 ± 500	5596 ± 1557
ME	Hopper (Morphology)	359 ± 75	349 ± 47	444 ± 15	357 ± 63	407 ± 28	462 ± 89
ME	Hopper (Kinematic)	724 ± 535	1024 ± 1	1031 ± 3	1022 ± 3	1027 ± 8	1022 ± 2
ME	Hopper (Friction)	229 ± 4	230 ± 3	232 ± 5	230 ± 6	232 ± 8	266 ± 70
ME	Walker2D (Morphology)	228 ± 51	429 ± 117	1103 ± 444	502 ± 301	380 ± 231	648 ± 180
ME	Walker2D (Kinematic)	386 ± 184	850 ± 953	1514 ± 782	1204 ± 734	755 ± 268	1511 ± 1206
ME	Walker2D (Friction)	266 ± 66	240 ± 114	258 ± 8	245 ± 51	242 ± 24	326 ± 26
	Average Return	**878**	**1920**	**1783**	**1868**	**1803**	**2193**

Table 1: Comparison of return under different dynamics shift scenarios and dataset types after 40K environment interactions. MR = Medium Replay, M = Medium, ME = Medium Expert. A cell is green if the method has the highest mean and improves over the second best by at least 2%. Cells within 2% of the top mean are marked in yellow.

offline transitions with value targets consistent with the online environment and adds a BC term. BC-PAR [37] applies a reward penalty based on representation mismatch between offline and online transitions, also including a BC term. Implementation details are provided in Appendix J.

5.1.2 Main Results

We evaluate the return of each algorithm in the target domain after 40K environment interactions and 400K gradient steps, reflecting the limited interaction setting. The performance of all methods is reported in Table 1. Our key findings are summarized as follows:

(1) CompFlow consistently outperforms recent off-dynamics reinforcement learning baselines across a wide range of dataset qualities and types of dynamics shifts. Specifically, it achieves the highest return on **19 out of 27** tasks and matches the top performance on an additional **5** tasks. On average, CompFlow attains a score of **2193**, compared to **1920** achieved by the second-best baseline, BC-SAC, representing a relative improvement of **14.2%**.

(2) Incorporating offline data can significantly enhance learning performance. On average, CompFlow achieves a **149.8%** improvement over SAC. In addition, CompFlow outperforms the base algorithm BC SAC, which directly incorporates all offline data, on **23 out of 27** tasks and matches its performance on the remaining **3**, providing strong empirical support for the effectiveness of our method.

(3) Additionally, we observe that recent baselines often perform similarly to BC-SAC across many tasks, consistent with the findings in [38], suggesting that they fail to effectively leverage the offline source domain data. This may stem from methods like H2O and BC-PAR relying on KL divergence or mutual information, which can be ill-defined under large dynamics gaps or different supports. Meanwhile, BC-VGDF adopts a data filtering strategy based on estimated value functions, which is often more challenging than learning transition dynamics due to bootstrapping bias and target non-stationarity. In contrast, our method, CompFlow, leverages flow matching to model complex

transition dynamics and exploits its theoretical connection to optimal transport to estimate the dynamics gap via Wasserstein distance—a more robust and principled metric.

5.1.3 Ablation and Hyperparameter Analysis

Composite Flow. We first evaluate the effectiveness of the proposed composite flow matching design. For comparison, we train a direct flow model on the target domain initialized from a Gaussian prior. We compute the mean squared error (MSE) on a held-out 10% validation set and report the average MSE across different epochs during RL training. As shown in Figure 3, the composite flow significantly reduces the MSE of the transition dynamics on the target domain. This improvement stems from its ability to reuse structural knowledge learned from the source domain, where abundant data is available. These empirical findings support the theoretical insight presented in Theorem 3.1 that our composite flow can improve the generalization ability.

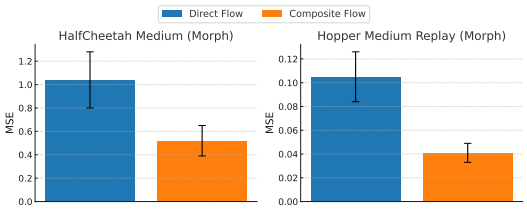

Figure 3: Comparison of MSE between direct flow and composite flow.

Impact of data selection ratio $\xi\%$. The data selection ratio ξ decides how many source domain data in a sampled batch can be shared for policy training. A larger ξ indicates that more source domain data will be admitted. To examine its influence, We sweep ξ across $\{70, 50, 30, 20\}$. The results is shown in Figure 4. Although the optimal ξ seems task dependent, moderate values of ξ (e.g., 30 or 50) generally yield good performance across tasks, striking a balance between leveraging useful source data and avoiding high dynamics mismatch. When ξ is too large (e.g., 70), performance often degrades, particularly in Walker Medium (Morph) and Walker Medium Replay (Morph), likely due to the inclusion of low-quality transitions with large dynamics gap. HalfCheetah consistently achieves higher returns compared to other domains, suggesting that knowledge transfer and policy learning in this environment are easier. Consequently, overly conservative filtering (e.g., $\xi = 20$) may exclude valuable source data, leading to slower learning. In this case, allowing more source data (e.g., $\xi = 70$) appears beneficial.

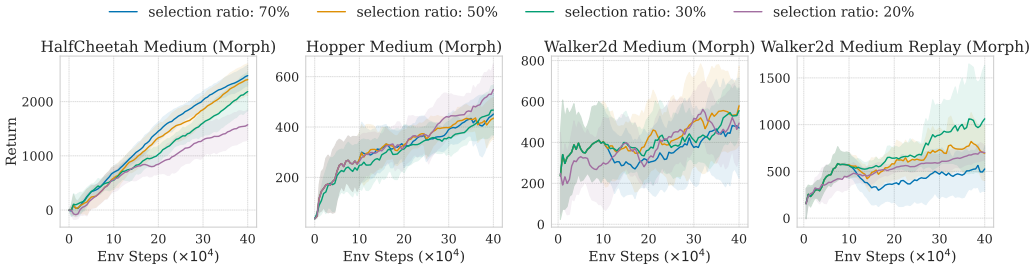

Figure 4: Comparison of return under different data selection ratios across tasks.

Impact of exploration strength β. β controls the strength of exploration toward regions with large dynamics gap. A large β indicates higher incentive to explore such regions. As shown in Figure 5, the effect of β on return is task-dependent, possibly due to differences in the underlying MDPs. In the Friction tasks, we observe that larger exploration bonuses lead to higher asymptotic returns. This suggests that incentivizing exploration helps the algorithm discover high-reward regions more effectively. In contrast, in the Morphology tasks, moderate values of β typically outperform both smaller and larger values. This indicates that excessive exploration may not be effective in some tasks. While the optimal β varies by task, one trend is consistent: across all five experiments, the setting with no exploration ($\beta = 0$) consistently ranks as the lowest or the second lowest performers. This empirical finding confirms the importance of the exploration bonus and provides evidence supporting our algorithmic design.

5.2 Patrol Policy Learning for Wildlife Conservation

We evaluate CompFlow in a wildlife conservation setting, where the objective is to learn a patrol policy that allocates limited ranger effort to reduce poaching and protect wildlife. We adopt the simulation environment introduced by Xu et al. [65]. The environment is represented as a spatial

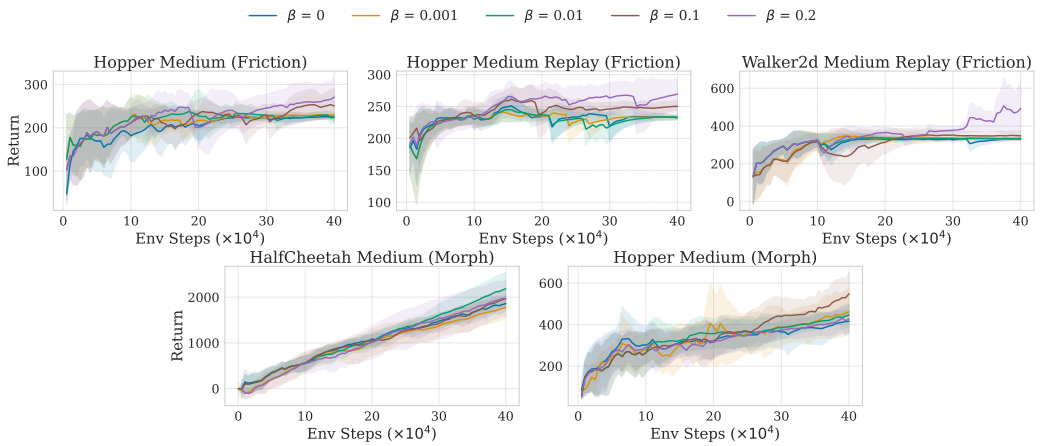

Figure 5: Comparison of return under different exploration strengths across tasks.

grid, partitioned into 1×1 km cells. At each time step, the agent selects a distribution of patrol effort across the cells, subject to a fixed budget. The state, which includes wildlife population density and poaching risk for each cell, evolves over time according to dynamics that depend on both poacher behavior and the deployed patrols. The per-step reward reflects the expected number of animals preserved through deterrence and spatial coverage achieved by the patrol strategy.

We assume access to an offline dataset collected under a previous policy in Murchison Falls National Park. The objective is to use this data to learn an improved patrol policy for Queen Elizabeth National Park, with only limited interaction with the new environment. Due to differences in ecological conditions and poacher behavior, the transition dynamics in the two parks are not the same. Figure 6 presents the results. CompFlow achieves the highest reward, exceeding the best baseline, BC-PAR, by 8.8%. Compared to training a policy from scratch using SAC, CompFlow improves the average reward by 20.8% under the same interaction budget. This improvement is especially important in conservation settings where ranger capacity is limited and the protected area is large. For instance, Murchison Falls National Park spans approximately 3,900 square kilometers, while Queen Elizabeth National Park covers about 1,980 square kilometers. The large size of these areas makes full coverage difficult, and learning effective patrol strategies with minimal exploration is crucial for protecting wildlife.

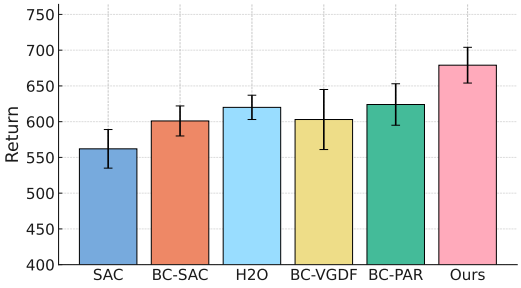

Figure 6: Reward on the wildlife conservation task.

6 Conclusion and Limitations

In this paper, we proposed CompFlow, a new method for estimating the dynamics gap in reinforcement learning with shifted-dynamics data. CompFlow leverages the theoretical connection between flow matching and optimal transport. To address data scarcity in the online environment, we adopt a composite flow structure that builds the online flow model on top of the output distribution from the offline flow. This composite formulation improves generalization and enables the use of Wasserstein distance between offline and online transitions as a robust measure of the dynamics gap. Using this pricinpled estimation, we further encourage the policy to explore regions with high dynamics gap and provide a theoretical analysis of the benefits. Empirically, we demonstrate that CompFlow consistently outperforms or matches state-of-the-art baselines across diverse RL tasks with varying types of dynamics shift.

Due to the space limit, we discuss the limitations of CompFlow in Appendix A.

Acknowledgement

We are thankful to the Uganda Wildlife Authority for granting us access to incident data from Murchison Falls and Queen Elizabeth National Park. We also thank the anonymous reviewers for their valuable feedback. This work was supported by ONR MURI N00014-24-1-2742.

References

[1] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=li7qeBbCR1t`.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

[3] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[5] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[6] Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=xCRr9DrolJ`.

[7] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[8] Ignasi Clavera, Anusha Nagabandi, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt: Meta-learning for model-based control. *arXiv preprint arXiv:1803.11347*, 3:3, 2018.

[9] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[10] Yuqing Du, Olivia Watkins, Trevor Darrell, Pieter Abbeel, and Deepak Pathak. Auto-tuned sim-to-real transfer. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1290–1296. IEEE, 2021.

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[12] Benjamin Eysenbach, Swapnil Asawa, Shreyas Chaudhari, Sergey Levine, and Ruslan Salakhutdinov. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. *arXiv preprint arXiv:2006.13916*, 2020.

[13] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch wasserstein: asymptotic and gradient properties. *arXiv preprint arXiv:1910.04091*, 2019.

[14] Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021.

[15] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

[16] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

[17] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[18] Adam P Generale, Andreas E Robertson, and Surya R Kalidindi. Conditional variable flow matching: Transforming conditional densities with amortized conditional optimal transport. *arXiv preprint arXiv:2411.08314*, 2024.

[19] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[20] Donald Hejna, Lerrel Pinto, and Pieter Abbeel. Hierarchically decoupled imitation for morphological transfer. In *International Conference on Machine Learning*, pages 4159–4171. PMLR, 2020.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pages 9902–9915. PMLR, 2022.

[23] Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Dynamic conditional optimal transport through simulation-free flows. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=tk0uaRynhH`.

[24] Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Domain adaptive imitation learning. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2020.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Representation Learning*, 2015.

[26] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[27] Lingkai Kong, Harshavardhan Kamarthi, Peng Chen, B Aditya Prakash, and Chao Zhang. Uncertainty quantification in deep learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5809–5810, 2023.

[28] Lingkai Kong, Yuanqi Du, Wenhao Mu, Kirill Neklyudov, Valentin De Bortoli, Dongxia Wu, Haorui Wang, Aaron Ferber, Yi-An Ma, Carla P Gomes, et al. Diffusion models as constrained samplers for optimization with unknown constraints. *arXiv preprint arXiv:2402.18012*, 2024.

[29] Lingkai Kong, Haichuan Wang, Charles A Emogor, Vincent Börsch-Supan, Lily Xu, and Milind Tambe. Generative ai against poaching: Latent composite flow matching for wildlife conservation. *arXiv preprint arXiv:2508.14342*, 2025.

[30] Lingkai Kong, Haichuan Wang, Yuqi Pan, Cheol Woo Kim, Mingxiao Song, Alayna Nguyen, Tonghan Wang, Haifeng Xu, and Milind Tambe. Robust optimization with diffusion models for green security. *arXiv preprint arXiv:2503.05730*, 2025.

[31] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33: 1179–1191, 2020.

[32] Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pages 1702–1712. PMLR, 2022.

[33] Yinghao Li, Lingkai Kong, Yuanqi Du, Yue Yu, Yuchen Zhuang, Wenhao Mu, and Chao Zhang. MUBen: Benchmarking the uncertainty of molecular representation models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=qYceFeHgm4.

[34] 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*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[35] Jinxin Liu, Hongyin Zhang, and Donglin Wang. Dara: Dynamics-aware reward augmentation in offline reinforcement learning. *arXiv preprint arXiv:2203.06662*, 2022.

[36] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[37] Jiafei Lyu, Chenjia Bai, Jingwen Yang, Zongqing Lu, and Xiu Li. Cross-domain policy adaptation by capturing representation mismatch. *arXiv preprint arXiv:2405.15369*, 2024.

[38] Jiafei Lyu, Kang Xu, Jiacheng Xu, Jing-Wen Yang, Zongzhang Zhang, Chenjia Bai, Zongqing Lu, Xiu Li, et al. Odrl: A benchmark for off-dynamics reinforcement learning. *Advances in Neural Information Processing Systems*, 37:59859–59911, 2024.

[39] Jiafei Lyu, Mengbei Yan, Zhongjian Qiao, Runze Liu, Xiaoteng Ma, Deheng Ye, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain offline policy adaptation with optimal transport and dataset constraint. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] Max Sobol Mark, Ali Ghadirzadeh, Xi Chen, and Chelsea Finn. Fine-tuning offline policies with optimistic action selection. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

[41] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. In *Conference on Robot Learning*, pages 1162–1176. PMLR, 2020.

[42] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

[43] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[44] Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Advances in Neural Information Processing Systems*, 36:62244–62269, 2023.

[45] Ariel Neufeld and Julian Sester. Bounding the difference between the values of robust and non-robust markov decision problems. *Journal of Applied Probability*, pages 1–14, 2023.

[46] Haoyi Niu, Yiwen Qiu, Ming Li, Guyue Zhou, Jianming Hu, Xianyuan Zhan, et al. When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. *Advances in Neural Information Processing Systems*, 35:36599–36612, 2022.

[47] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

[48] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[49] Chengrui Qu, Laixi Shi, Kishan Panaganti, Pengcheng You, and Adam Wierman. Hybrid transfer reinforcement learning: Provable sample efficiency from shifted-dynamics data. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL `https://openreview.net/forum?id=ndWOLqVRHC`.

[50] Roberta Raileanu, Max Goldstein, Arthur Szlam, and Rob Fergus. Fast adaptation to new environments via policy-dynamics value functions. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7920–7931, 2020.

[51] Allen Z. Ren, Justin Lidard, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=mEpqHvbD2h`.

[52] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[53] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[54] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[55] Reda Bahi Slaoui, William R Clements, Jakob N Foerster, and Sébastien Toth. Robust domain randomization for reinforcement learning. 2019.

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[57] Yuda Song, Aditi Mavalankar, Wen Sun, and Sicun Gao. Provably efficient model-based policy adaptation. *arXiv preprint arXiv:2006.08051*, 2020.

[58] Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

[59] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28694–28698, 2025.

[60] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[61] 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*, pages 5026–5033. IEEE, 2012.

[62] Xiaoyu Wen, Chenjia Bai, Kang Xu, Xudong Yu, Yang Zhang, Xuelong Li, and Zhen Wang. Contrastive representation for data filtering in cross-domain offline reinforcement learning. *arXiv preprint arXiv:2405.06192*, 2024.

[63] Kang Xu, Chenjia Bai, Xiaoteng Ma, Dong Wang, Bin Zhao, Zhen Wang, Xuelong Li, and Wei Li. Cross-domain policy adaptation via value-guided data filtering. *Advances in Neural Information Processing Systems*, 36:73395–73421, 2023.

[64] Lily Xu, Elizabeth Bondi, Fei Fang, Andrew Perrault, Kai Wang, and Milind Tambe. Dual-mandate patrols: Multi-armed bandits for green security. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14974–14982, 2021.

[65] Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. Robust reinforcement learning under minimax regret for green security. In *Uncertainty in Artificial Intelligence*, pages 257–267. PMLR, 2021.

[66] Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, Bo Liu, Jia Chen, Zhao Liu, Fuhao Qiu, Hongsheng Yu, et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:621–632, 2020.

[67] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[68] Haichao Zhang, We Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. *arXiv preprint arXiv:2302.00935*, 2023.

[69] Yuchen Zhuang, Yue Yu, Lingkai Kong, Xiang Chen, and Chao Zhang. Dygen: Learning from noisy labels via dynamics-enhanced generative modeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3674–3686, 2023.

Appendix for Composite Flow Matching for Reinforcement Learning with Shifted-Dynamics Data

A Limitations

(1) COMPFLOW is currently limited to settings where the offline data and the online environment share the same state and action spaces. Future work could explore estimating the dynamics gap in a shared latent embedding space to relax this assumption [69]. (2) Our evaluation is conducted entirely in simulated environments; applying COMPFLOW to real-world scenarios remains an important direction for future work. (3) We have not yet incorporated the uncertainty in estimating the dynamics gap. Future work may explore how to quantify this uncertainty and use it to improve both data filtering and exploration strategies [27, 33].

B Broader Impacts

Our work aims to make reinforcement learning more practical in real-world domains such as healthcare, robotics, and conservation, where online interaction is often costly, limited, or unsafe. By addressing the dynamics shift between offline data and the online environment, our method enables more reliable and sample-efficient policy learning. This capability supports safer deployment in high-stakes applications, including clinical decision support and adaptive anti-poaching strategies. Nonetheless, we emphasize that policies trained on historical data should be applied with caution, as misaligned dynamics or biased datasets may lead to unintended consequences if not carefully validated.

C Algorithms of Training Optimal Transport Flow Matching

The full training algorithm of Optimal Transport Flow Matching is given in Algorithm 1.

Algorithm 1 Training Optimal Transport Flow Matching

Require: Dataset $\mathcal{D}$, batch size k, cost function $c(\cdot,\cdot)$, learning rate lr, flow model v_θ

while not converged **do**
  Sample latent batch $\{x_0^{(i)}\}_{i=1}^k \sim \mathcal{N}(0, \mathbf{I})$
  Sample data batch $\{x_1^{(j)}\}_{j=1}^k \sim \mathcal{D}$
  Compute optimal transport plan:

$$A \doteq \arg\min_{A \in B_k} \sum_{i,j} A(i,j) \cdot c(x_0^{(i)}, x_1^{(j)}), \quad c(x_0^{(i)}, x_1^{(j)}) = ||x_0^{(i)} - x_1^{(j)}||_2^2$$

  ▷ B_k: set of $k \times k$ doubly-stochastic matrices
  Sample k index pairs $\{(i_\ell, j_\ell)\}_{\ell=1}^k \sim A(i,j)$
  Sample interpolation times $\{t^{(\ell)}\}_{\ell=1}^k \sim \mathcal{U}[0,1]$
  for $\ell = 1$ to k **do**
    Define matched pair:

$$x_0^{(\ell)} \doteq x_0^{(i_\ell)}, \quad x_1^{(\ell)} \doteq x_1^{(j_\ell)}$$

    Interpolate:

$$x_t^{(\ell)} \doteq t^{(\ell)} x_1^{(\ell)} + (1 - t^{(\ell)}) x_0^{(\ell)}$$

  end for
  Compute loss:

$$\mathcal{L} \doteq \frac{1}{k} \sum_{\ell=1}^{k} \left\| v_\theta(x_t^{(\ell)}, t^{(\ell)}) - (x_1^{(\ell)} - x_0^{(\ell)}) \right\|_2^2$$

  Update parameters:

$$\theta \leftarrow \theta - \mathsf{lr} \nabla_\theta \mathcal{L}$$

end while
return θ

D Algorithms of Training Offline and Online Flows

The full training algorithms of the offline flow and online flow are given in Algorithm 2 and Algorithm 3 respectively.

Algorithm 2 Training of Offline Flow Matching

Require: Offline dataset $\mathcal{D}_{\text{off}}$, batch size k, flow model v_θ, learning rate lr

while not converged **do**
  Sample a batch of transitions $\{(s^{(i)}, a^{(i)}, s'^{(i)})\}_{i=1}^k \sim \mathcal{D}_{\text{off}}$
  For each i, define target sample $x_1^{(i)} \doteq s'^{(i)}$
  Sample $\{x_0^{(i)}\}_{i=1}^k \sim \mathcal{N}(0, \mathbf{I})$
  Sample interpolation times $\{t^{(i)}\}_{i=1}^k \sim \mathcal{U}[0,1]$
  for $i = 1$ to k **do**
    Compute interpolated point: $x_t^{(i)} = t^{(i)} x_1^{(i)} + (1 - t^{(i)}) x_0^{(i)}$
  end for
  Compute training loss:

$$\mathcal{L} = \frac{1}{k} \sum_{i=1}^{k} \left\| v_\theta(x_t^{(i)}, t^{(i)}, s^{(i)}, a^{(i)}) - (x_1^{(i)} - x_0^{(i)}) \right\|_2^2$$

  Update $\theta \leftarrow \theta - \mathsf{lr} \nabla_\theta \mathcal{L}$
end while
return Trained parameters θ

Algorithm 3 Training of Online Flow via Optimal Transport

Require: Pre-trained offline flow model $\psi_\theta(x, t \mid s, a)$, offline dataset $\mathcal{D}_{\text{off}}$, target dataset $\mathcal{D}_{\text{on}}$, batch size k, cost function $c(\cdot)$, regularization weight η, learning rate lr, target flow model v_ϕ

while not converged **do**

  Sample latent vectors $\{x_0^{(i)}\}_{i=1}^k \sim \mathcal{N}(0, \mathbf{I})$

  Sample offline batch $\{s_{\text{off}}^{(i)}, a_{\text{off}}^{(i)}\}_{i=1}^k \sim \mathcal{D}_{\text{off}}$

  for $i = 1$ to k **do**

    Predict next state for the offline environment:

$$s_{\text{off}}'^{(i)} \doteq x_1^{(i)} \doteq \psi_\theta(x_0^{(i)}, 1 \mid s_{\text{off}}^{(i)}, a_{\text{off}}^{(i)})$$

  end for

  Sample online batch $\{s_{\text{on}}^{(j)}, a_{\text{on}}^{(j)}, s_{\text{on}}'^{(j)}\}_{j=1}^k \sim \mathcal{D}_{\text{on}}$

  Define $x_2^{(j)} \doteq s_{\text{on}}'^{(j)}$

  Compute optimal transport plan:

$$A \doteq \arg\min_{A \in B_k} \sum_{i,j} A(i,j) \cdot c(i,j)$$

  where the cost is:

$$c(i,j) \doteq \|x_1^{(i)} - x_2^{(j)}\|_2^2 + \eta \left(\|s_{\text{off}}^{(i)} - s_{\text{on}}^{(j)}\|_2^2 + \|a_{\text{off}}^{(i)} - a_{\text{on}}^{(j)}\|_2^2 \right)$$

  ▷ B_k: set of $k \times k$ doubly-stochastic matrices

  Sample k index pairs $\{(i_\ell, j_\ell)\}_{\ell=1}^k \sim A(i,j)$

  Sample interpolation times $\{t^{(\ell)}\}_{\ell=1}^k \sim \mathcal{U}[1,2]$ ▷ Time t for the online flow is from 1 to 2

  for $\ell = 1$ to k **do**

    Compute:

$$x_t^{(\ell)} \doteq (t^{(\ell)} - 1) x_2^{(i_\ell)} + (2 - t^{(\ell)}) x_1^{(j_\ell)}$$

  end for

  Compute training loss:

$$\mathcal{L} \doteq \frac{1}{k} \sum_{\ell=1}^k \left\| v_\phi(x_t^{(\ell)}, t^{(\ell)}, s_{\text{on}}^{(\ell)}, a_{\text{on}}^{(\ell)}) - (x_2^{(j_\ell)} - x_1^{(i_\ell)}) \right\|_2^2$$

  Update parameters:

$$\phi \leftarrow \phi - \mathsf{lr} \nabla_\phi \mathcal{L}$$

end while

return ϕ

E Algorithm of CompFlow built on Soft-Actor-Critic

The full training algorithm of CompFlow built on SAC is given in Algorithm 4.

Algorithm 4 COMPFLOW built on Soft Actor-Critic (SAC)

Input: Offline dataset $\mathcal{D}_{\text{off}}$, Online environment $\mathcal{M}_{\text{on}}$, max interaction steps $T_{\max}$, target model training frequency train_freq, source data selection ratio ξ, gap reward scale β, batch size B, behavior cloning weight ω, warmup steps warmup_steps, learning rate lr, target update rate ϖ

Initialization: Policy π_φ, Q-functions $\{Q_{\varsigma_i}\}_{i=1,2}$, target Q-functions $\{Q^{\text{tgt}}_{\varsigma_i}\}_{i=1,2} \leftarrow \{Q_{\varsigma_i}\}_{i=1,2}$, target replay buffer $\mathcal{D}_{\text{on}}$

Pretrain offline flow model $\psi_\theta(x, t \mid s, a)$ on $\mathcal{D}_{\text{off}}$ via Algorithm 2

for $t = 1$ to $T_{\max}$ **do**

  Sample transition $(s_{\text{on}}, a_{\text{on}}, r_{\text{on}}, s'_{\text{on}}) \sim \mathcal{M}_{\text{on}}$ using policy π_φ

  Update replay buffer: $\mathcal{D}_{\text{on}} \leftarrow \mathcal{D}_{\text{on}} \cup \{(s_{\text{on}}, a_{\text{on}}, r_{\text{on}}, s'_{\text{on}})\}$

  if $t \bmod$ train_freq $= 0$ and $t >$ warmup_steps **then**

    Train target flow $\psi_\phi(x, t \mid s, a)$ on $\mathcal{D}_{\text{on}}$ via Algorithm 3

  end if

  for $k = 1$ to K **do**

    Sample $b_{\text{off}} = \{(s^{(i)}_{\text{off}}, a^{(i)}_{\text{off}}, r^{(i)}_{\text{off}}, s'^{(i)}_{\text{off}})\}_{i=1}^{B} \sim D_{\text{off}}$

    Sample $b_{\text{on}} = \{(s^{(i)}_{\text{on}}, a^{(i)}_{\text{on}}, r^{(i)}_{\text{on}}, s'^{(i)}_{\text{on}})\}_{i=1}^{B} \sim \mathcal{D}_{\text{on}}$

    Estimate $\widehat{\Delta}(s^{(i)}_{\text{off}}, a^{(i)}_{\text{off}})$ for each $(s^{(i)}_{\text{off}}, a^{(i)}_{\text{off}}) \in b_{\text{off}}$ via Eq. 3

    Select top $\xi\%$ of b_{off} with lowest gap: $\tilde{b}_{\text{off}}$

    For each $(s, a, s', r) \in \tilde{b}_{\text{off}}$, update reward: $r \leftarrow r + \beta\widehat{\Delta}(s, a)$

    for $i = 1, 2$ **do**

      Compute Bellman target using target Q-networks:

$$y = r + \gamma \mathbb{E}_{a' \sim \pi_\varphi(\cdot|s')} \left[\min_{j=1,2} Q^{\text{tgt}}_{\varsigma_j}(s', a') - \alpha \log \pi_\varphi(a'|s') \right]$$

      Compute Q-function loss gradient:

$$\nabla_{\varsigma_i} \mathcal{L}_Q \leftarrow \frac{1}{B} \sum_{(s^{\text{on}}, a_{\text{on}}, s'_{\text{on}}, r_{\text{on}}) \in b_{\text{on}}} \nabla_{\varsigma_i} \left(Q_{\varsigma_i}(s_{\text{on}}, a_{\text{on}}) - y \right)^2 + \frac{1}{|\tilde{b}_{\text{off}}|} \sum_{(s_{\text{off}}, a_{\text{off}}, s'_{\text{off}}, r_{\text{off}}) \in \tilde{b}_{\text{off}}} \nabla_{\varsigma_i} \left(Q_{\varsigma_i}(s_{\text{off}}, a_{\text{off}}) - y \right)^2$$

      Update Q-network: $\varsigma_i \leftarrow \varsigma_i - \text{lr} \nabla_{\varsigma_i} \mathcal{L}_Q$

    end for

    # Soft update of target Q-networks

    for $i = 1, 2$ **do**

      $\varsigma^{\text{tgt}}_i \leftarrow \varpi \varsigma_i + (1 - \varpi) \varsigma^{\text{tgt}}_i$

    end for

    Compute BC weight:

$$\lambda = \omega \Bigg/ \left\{ \frac{1}{2B} \sum_{\tilde{s} \in \tilde{b}_{\text{src}} \cup b_{\text{tar}}} \left| \min \left\{ Q_{\varsigma_1}(\tilde{s}, \tilde{a}), Q_{\varsigma_2}(\tilde{s}, \tilde{a}) \right\} \right| \right\}$$

    Compute policy loss gradient:

$$\nabla_\varphi \mathcal{L}_\pi \leftarrow - \frac{\lambda}{|\tilde{b}_{\text{off}} \cup b_{\text{on}}|} \sum_{\tilde{s} \in \tilde{b}_{\text{off}} \cup b_{\text{on}}} \nabla_\varphi \left[\min_{i=1,2} Q_{\varsigma_i}(\tilde{s}, \tilde{a}) + \alpha H[\pi_\varphi(\cdot|\tilde{s})] \right] + \frac{1}{|b_{\text{off}}|} \sum_{(s,a) \in b_{\text{off}}} \nabla_\varphi \left\| a - \bar{a} \right\|^2$$

    where $\tilde{a} \sim \pi_\varphi(\cdot|\tilde{s}), \quad \bar{a} \sim \pi_\varphi(\cdot|s)$ *are independent samples.*

    Update policy: $\varphi \leftarrow \varphi + \text{lr} \nabla_\varphi \mathcal{L}_\pi$

  end for

end for

Output: Learned policy π_φ

F Proof of Lemma 2.2

Lemma 2.2 (Return Bound between Two Environments [37])**.** *Let the empirical behavior policy in $\mathcal{D}_{\text{off}}$ be $\pi_{\mathcal{D}_{\text{off}}}(a \mid s)$. Define $C_1 = \frac{2r_{\max}}{(1-\gamma)^2}$. Then for any policy π,*

$$\begin{aligned}\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\mathcal{M}_{\text{off}}}(\pi) \geq & -2C_1\, \mathbb{E}_{(s,a)\sim\rho_{\mathcal{M}}^{\pi_{\mathcal{D}_{\text{off}}}}, s'\sim p_{\text{off}}}\left[D_{\text{TV}}(\pi(\cdot|s') \parallel \pi_{\mathcal{D}_{\text{off}}}(\cdot|s'))\right] \\ & - C_1\, \mathbb{E}_{(s,a)\sim\rho_{\mathcal{M}}^{\pi_{\mathcal{D}_{\text{off}}}}}\left[D_{\text{TV}}(p_{\text{on}}(\cdot|s,a) \parallel p_{\text{off}}(\cdot|s,a))\right].\end{aligned}$$

Proof. Proof of Lemma 2.2 is already provided as an intermediate step for Theorem A.4 in [37], the offline performance bound case. We include their proof here with slight modifications for completeness.

We first cite the following two necessary lemmas also used in [37]. □

Lemma F.1 (Extended telescoping lemma)**.** *Denote $\mathcal{M}_1 = (\mathcal{S}, \mathcal{A}, P_1, r, \gamma)$ and $\mathcal{M}_2 = (\mathcal{S}, \mathcal{A}, P_2, r, \gamma)$ as two MDPs that* only differ in their transition dynamics*. Suppose we have two policies π_1, π_2, we can reach the following conclusion:*

$$\eta_{\mathcal{M}_1}(\pi_1) - \eta_{\mathcal{M}_2}(\pi_2) = \frac{1}{1-\gamma}\, \mathbb{E}_{\rho_{\mathcal{M}_1}^{\pi_1}(s,a)}\left[\mathbb{E}_{s'\sim P_1,\, a'\sim\pi_1}\left[Q_{\mathcal{M}_2}^{\pi_2}(s',a')\right] - \mathbb{E}_{s'\sim P_2,\, a'\sim\pi_2}\left[Q_{\mathcal{M}_2}^{\pi_2}(s',a')\right]\right].$$

Proof. *This is Lemma C.2 in [63].* □

Lemma F.2. *Denote $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ as the underlying MDP. Suppose we have two policies π_1, π_2, then the performance difference of these policies in the MDP gives:*

$$\eta_{\mathcal{M}}(\pi_1) - \eta_{\mathcal{M}}(\pi_2) = \frac{1}{1-\gamma}\, \mathbb{E}_{\rho_{\mathcal{M}}^{\pi_1}(s,a),\, s'\sim P}\left[\mathbb{E}_{a'\sim\pi_1}\left[Q_{\mathcal{M}}^{\pi_2}(s',a')\right] - \mathbb{E}_{a'\sim\pi_2}\left[Q_{\mathcal{M}}^{\pi_2}(s',a')\right]\right].$$

Proof. *This is Lemma B.3 in [37].* □

Proof. We have access to the offline data $\mathcal{M}_{\text{off}}$ and the empirical behavioral policy $\pi_{\mathcal{D}_{\text{off}}}$, so we bound the performance between $\eta_{\mathcal{M}_{\text{on}}}(\pi)$ and $\eta_{\mathcal{M}_{\text{off}}}(\pi_{\mathcal{D}_{\text{off}}})$. We have

$$\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\pi_{\mathcal{D}_{\text{off}}}}(\pi) = \underbrace{\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\mathcal{M}_{\text{off}}}(\pi_{\mathcal{D}_{\text{off}}})}_{(a)} + \underbrace{\eta_{\pi_{\mathcal{D}_{\text{off}}}}(\pi_{\mathcal{D}_{\text{off}}}) - \eta_{\pi_{\mathcal{D}_{\text{off}}}}(\pi)}_{(b)}. \tag{9}$$

To bound (a) term, we use Lemma F.1 in the second equality

$$\begin{aligned}&\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\mathcal{M}_{\text{off}}}(\mathcal{D}_{\text{off}}) \\ &= -\left(\eta_{\mathcal{M}_{\text{off}}}(\mathcal{D}_{\text{off}}) - \eta_{\mathcal{M}_{\text{on}}}(\pi)\right) \\ &= -\frac{1}{1-\gamma}\, \mathbb{E}_{(s,a)\sim\rho_{\mathcal{M}_{\text{off}}}^{\mathcal{D}_{\text{off}}}}\left[\mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}},\, a'\sim\pi_{\mathcal{D}_{\text{off}}}}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{off}},a')\right] - \mathbb{E}_{s'_{\text{on}}\sim p_{\mathcal{M}_{\text{on}}},\, a'\sim\pi}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{on}},a')\right]\right] \\ &= -\frac{1}{1-\gamma}\, \mathbb{E}_{(s,a)\sim\rho_{\mathcal{M}_{\text{off}}}^{\mathcal{D}_{\text{off}}}}\Bigg[\\ &\quad \underbrace{\left(\mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}},\, a'\sim\pi_{\mathcal{D}_{\text{off}}}}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{off}},a')\right] - \mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}},\, a'\sim\pi}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{off}},a')\right]\right)}_{\text{(c)}} \\ &\quad + \underbrace{\left(\mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}},\, a'\sim\pi}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{off}},a')\right] - \mathbb{E}_{s'_{\text{on}}\sim p_{\mathcal{M}_{\text{on}}},\, a'\sim\pi}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{on}},a')\right]\right)}_{\text{(d)}}\Bigg].\end{aligned}$$

To bound term (c), we use

$$\begin{aligned}&\mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}},\, a'\sim\pi_{\mathcal{D}_{\text{off}}}}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{off}},a')\right] - \mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}},\, a'\sim\pi}\left[Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{off}},a')\right] \\ &\leq \mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}}}\left[\sum_{a'\in\mathcal{A}} \left|\mathcal{D}_{\text{off}}(a' \mid s'_{\text{off}}) - \pi(a' \mid s'_{\text{off}})\right| \cdot \left|Q_{\mathcal{M}_{\text{on}}}^{\pi}(s'_{\text{off}},a')\right|\right] \\ &\leq \frac{2r_{\max}}{1-\gamma}\, \mathbb{E}_{s'_{\text{off}}\sim p_{\mathcal{M}_{\text{off}}}}\left[D_{\text{TV}}\left(\pi_{\mathcal{D}_{\text{off}}}(\cdot \mid s'_{\text{off}}) \,\|\, \pi(\cdot \mid s'_{\text{off}})\right)\right],\end{aligned}$$

where the last inequality comes from the fact that $|Q^{\pi}_{\mathcal{M}_{\text{on}}}(s'_{\text{off}}, a')| \leq \frac{r_{\max}}{1-\gamma}$ and the definition of TV distance.

$$
\begin{aligned}
(d) &= \mathbb{E}_{s' \sim p_{\mathcal{M}_{\text{off}}},\, a' \sim \pi}\left[Q^{\pi}_{\mathcal{M}_{\text{on}}}(s', a')\right] - \mathbb{E}_{s' \sim p_{\mathcal{M}_{\text{on}}},\, a' \sim \pi}\left[Q^{\pi}_{\mathcal{M}_{\text{on}}}(s', a')\right] \\
&= \int_{\mathcal{S}} \left(p_{\mathcal{M}_{\text{off}}}(s' \mid s, a) - p_{\mathcal{M}_{\text{on}}}(s' \mid s, a)\right) \left(\sum_{a'} \pi(a' \mid s') Q^{\pi}_{\mathcal{M}_{\text{on}}}(s', a')\right) ds' \\
&\leq \frac{r_{\max}}{1-\gamma} \int_{\mathcal{S}} \left|p_{\mathcal{M}_{\text{off}}}(s' \mid s, a) - p_{\mathcal{M}_{\text{on}}}(s' \mid s, a)\right| ds' \\
&= \frac{2r_{\max}}{1-\gamma} \left[D_{\text{TV}}\left(p_{\mathcal{M}_{\text{off}}}(\cdot \mid s, a) \,\|\, p_{\mathcal{M}_{\text{on}}}(\cdot \mid s, a)\right)\right],
\end{aligned}
$$

where the inequality comes from the fact that $Q^{\pi}_{\mathcal{M}_{\text{on}}}(s', a')$ weighted by probability is bounded by $\frac{r_{\max}}{1-\gamma}$, the upper bound for all Q-values.

Combine the bounds for (c) and (d), we obtain a bound for the (a) term:

$$
\begin{aligned}
\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\mathcal{M}_{\text{off}}}(\pi_{\mathcal{D}_{\text{off}}}) \geq &- \frac{2r_{\max}}{(1-\gamma)^2} \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}_{\text{off}}},\, s' \sim p_{\mathcal{M}_{\text{off}}}} \left[D_{\text{TV}}\left(\pi_{\mathcal{D}_{\text{off}}}(\cdot \mid s') \,\|\, \pi(\cdot \mid s')\right)\right] \\
&- \frac{2r_{\max}}{(1-\gamma)^2} \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}_{\text{off}}}} \left[D_{\text{TV}}\left(p_{\mathcal{M}_{\text{off}}}(\cdot \mid s, a) \,\|\, p_{\mathcal{M}_{\text{on}}}(\cdot \mid s, a)\right)\right].
\end{aligned}
$$

Now we try to bound term (b).

$$
\begin{aligned}
&\eta_{\mathcal{M}_{\text{off}}}(\pi_{\mathcal{D}_{\text{off}}}) - \eta_{\mathcal{M}_{\text{off}}}(\pi) \\
&= \frac{1}{1-\gamma} \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}_{\text{off}}},\, s' \sim p_{\mathcal{M}_{\text{off}}}(\cdot|s,a)} \left[\mathbb{E}_{a' \sim \pi_{\mathcal{D}_{\text{off}}}}\left[Q^{\pi}_{\mathcal{M}_{\text{off}}}(s', a')\right] - \mathbb{E}_{a' \sim \pi}\left[Q^{\pi}_{\mathcal{M}_{\text{off}}}(s', a')\right]\right] \\
&\geq -\frac{1}{1-\gamma} \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}_{\text{off}}},\, s' \sim p_{\mathcal{M}_{\text{off}}}(\cdot|s,a)} \left|\mathbb{E}_{a' \sim \pi_{\mathcal{D}_{\text{off}}}}\left[Q^{\pi}_{\mathcal{M}_{\text{off}}}(s', a')\right] - \mathbb{E}_{a' \sim \pi}\left[Q^{\pi}_{\mathcal{M}_{\text{off}}}(s', a')\right]\right| \\
&\geq -\frac{1}{1-\gamma} \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}_{\text{off}}},\, s' \sim p_{\mathcal{M}_{\text{off}}}(\cdot|s,a)} \left|\sum_{a' \in \mathcal{A}} \left(\pi_{\mathcal{D}_{\text{off}}}(a' \mid s') - \pi(a' \mid s')\right) Q^{\pi}_{\mathcal{M}_{\text{off}}}(s', a')\right| \\
&\geq -\frac{r_{\max}}{(1-\gamma)^2} \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}_{\text{off}}},\, s' \sim p_{\mathcal{M}_{\text{off}}}(\cdot|s,a)} \left|\sum_{a' \in \mathcal{A}} \left(\pi_{\mathcal{D}_{\text{off}}}(a' \mid s') - \pi(a' \mid s')\right)\right| \\
&= -\frac{2r_{\max}}{(1-\gamma)^2} \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}_{\text{off}}},\, s' \sim p_{\mathcal{M}_{\text{off}}}(\cdot|s,a)} \left[D_{\text{TV}}\left(\pi_{\mathcal{D}_{\text{off}}}(\cdot \mid s') \,\|\, \pi(\cdot \mid s')\right)\right],
\end{aligned}
$$

where the first equality is a direct application of Lemma F.2. Combine the bound for term (a) and (b), we conclude that

$$
\begin{aligned}
\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\mathcal{M}_{\text{off}}}(\pi) \geq &- 2C_1 \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}},\, s' \sim p_{\text{off}}} \left[D_{\text{TV}}(\pi(\cdot|s') \parallel \pi_{\mathcal{D}_{\text{off}}}(\cdot|s'))\right] \\
&- C_1 \mathbb{E}_{(s,a) \sim \rho^{\pi_{\mathcal{D}_{\text{off}}}}_{\mathcal{M}}} \left[D_{\text{TV}}(p_{\text{on}}(\cdot|s,a) \parallel p_{\text{off}}(\cdot|s,a))\right],
\end{aligned}
$$

where $C_1 = \frac{2r_{\max}}{(1-\gamma)^2}$. □

G Proof of Theorem 3.1

Definition G.1. For any distributions p_0, p_1 on R^d with finite second moments, define $D_2(p_0, p_1) := E||X_1 - X_0||_2^2, X_0 \sim p_0, X_1 \sim p_1$ independent.

Theorem 3.1 (Conditions for Composite Flow Yielding Smaller Errors)**.** *Assume that composite flow and direct flow share the same hypothesis class $\mathcal{H}$ of (measurable) vector fields $v : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, and that there exists $B \in (0, \infty)$ such that $\sup_{v \in \mathcal{H}} \sup_{x \in \mathbb{R}^d,\, t \in [0,1]} \|v(x,t)\|_2 \leq B$. Also, assume that $\max\{\text{Tr}(\Sigma_{\text{on}}), \text{Tr}(\widehat{\Sigma}_{\text{off}})\} \leq C_{\text{TR}}$ for some C_{TR}. The composite flow enjoys a strictly tighter high-probability generalization bound than the direct flow if and only if*

$$
W_2(p_G, p_{\text{on}}) > W_2(\hat{p}_{\text{off}}, p_{\text{on}}) \tag{2}
$$

Here, $p_G := \mathcal{N}(0, I_d)$, Σ_{on} is the covariance of p_{on}, and $\widehat{\Sigma}_{\text{off}}$ is the covariance of $\hat{p}_{\text{off}}$.

Proof. We consider *linear-path* flow matching (FM) trained with squared loss. A single training example is generated by: $t \sim \mathrm{Unif}[0,1]$; $X_0 \sim p_0$ (the *start* distribution), independently of t; $X_1 \sim p_{\mathrm{on}}$, independently of (t, X_0); The interpolation $X_t := (1-t)X_0 + tX_1$ and the label $Y := X_1 - X_0 \in \mathbb{R}^d$.

Let $\mathcal{H}$ be a hypothesis class of (measurable) vector fields $v : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ and define the population risk

$$R(v) := \mathbb{E}\Big[\, \|v(X_t,t) - Y\|_2^2 \Big], \tag{10}$$

where the expectation is over the sampling mechanism above. Let

$$v^\star(x,t) := \mathbb{E}[\, Y \mid X_t = x,\, t\,] \tag{11}$$

denote the Bayes (population) minimizer. Given n i.i.d. samples, let $\hat{v}$ be any empirical risk minimizer over $\mathcal{H}$ for the squared loss.

Assumptions. **A1** (*Uniform boundedness*) There exists $B \in (0,\infty)$ such that $\sup_{v\in\mathcal{H}} \sup_{x\in\mathbb{R}^d,\, t\in[0,1]} \|v(x,t)\|_2 \le B$. This ensures integrability and controls the Lipschitz constants used below.

FM as regression; Bayes predictor and excess-risk identity. By definition,

$$R(v) \;=\; \mathbb{E}\, \|v(X_t,t) - Y\|_2^2, \quad Y = X_1 - X_0.$$

For any measurable v, the *Bayes* (population) minimizer for the squared loss is the conditional mean

$$v^\star(x,t) \;=\; \mathbb{E}[\, Y \mid X_t = x,\, t\,].$$

A standard identity for squared loss (we derive it fully) is

$$R(v) - R(v^\star) \;=\; \mathbb{E}\, \|\, v(X_t,t) - v^\star(X_t,t)\, \|_2^2. \tag{12}$$

Derivation of (12). Expand and add/subtract $v^\star(X_t,t)$:

$$\begin{aligned} R(v) &= \mathbb{E}\, \|\, v(X_t,t) - v^\star(X_t,t) + v^\star(X_t,t) - Y\, \|_2^2 \\ &= \mathbb{E}\, \|v - v^\star\|_2^2 \;+\; 2\,\mathbb{E}\,\langle v - v^\star,\, v^\star - Y\rangle \;+\; \mathbb{E}\, \|v^\star - Y\|_2^2, \end{aligned}$$

where all functions are evaluated at (X_t,t) but notationally suppressed for readability. Condition on (X_t,t): by definition, $\mathbb{E}[Y \mid X_t,t] = v^\star(X_t,t)$, hence $\mathbb{E}[\, v^\star - Y \mid X_t,t\,] = 0$ and the cross term vanishes after taking expectations. Therefore

$$R(v) = \mathbb{E}\, \|v - v^\star\|_2^2 + R(v^\star),$$

which rearranges to (12).

In Lemma G.3, we prove that there exists an absolute constant $c > 0$ such that, for all $\delta \in (0,1)$, with probability at least $1-\delta$ (over the sample draw),

$$R(\hat{v}) \;\le\; R(v^\star) \;+\; c\left(B + \sqrt{\mathbb{E}\, \|Y\|_2^2}\right)\Gamma_{n,\delta}, \qquad \Gamma_{n,\delta} := \mathfrak{R}_n(\mathcal{H}) \;+\; \sqrt{\tfrac{\log(1/\delta)}{n}}, \tag{13}$$

where $\mathfrak{R}_n(\mathcal{H})$ is the (data-independent) Rademacher complexity of $\mathcal{H}$. In short: symmetrization + vector contraction + Lipschitzness of the squared loss on the B-ball yields (13).)

Subtract $R(v^\star)$ on both sides of (13) and apply (12) with $v = \hat{v}$:

$$\mathbb{E}\, \|\hat{v} - v^\star\|_2^2 \;\le\; c\left(B + \sqrt{\mathbb{E}\, \|Y\|_2^2}\right)\Gamma_{n,\delta}. \tag{14}$$

Identify $\mathbb{E}\, \|Y\|_2^2$ as $D_2(p_0, p_{\mathrm{on}})$ and linearize the square root. By independence of $X_0 \sim p_0$ and $X_1 \sim p_{\mathrm{on}}$,

$$\mathbb{E}\, \|Y\|_2^2 \;=\; \mathbb{E}\, \|X_1 - X_0\|_2^2 \;=:\; D_2(p_0, p_{\mathrm{on}}). \tag{15}$$

By applying Lemma G.4, we have

$$\sqrt{\mathbb{E}\, \|Y\|_2^2} \le \sqrt{\mathrm{Tr}(\Sigma_{\mathrm{on}}) + C_{\mathrm{TR}}} + W_2(p_0, p_{\mathrm{on}}). \tag{16}$$

Therefore, we obtain

$$\mathbb{E}\ \|\hat{v} - v^\star\|_2^2 \le c\left(B + \sqrt{\mathrm{Tr}(\Sigma_{\mathrm{on}}) + C_{\mathrm{TR}}} + W_2(p_0, p_{\mathrm{on}})\right)\Gamma_{n,\delta} \tag{17}$$

This gives the following master bound (18) with $C_0 := c(B + \sqrt{\mathrm{Tr}(\Sigma_{\mathrm{on}}) + C_{\mathrm{TR}}}$ and $C_1 := c$.

Master bound. Under **A1**, for any start p_0 (with fixed target p_{on}), with probability at least $1 - \delta$,

$$\mathbb{E}\ \|\hat{v} - v^\star\|_2^2 \ \le\ \underbrace{C_0\,\Gamma_{n,\delta}}_{p_0\text{-independent}} \ +\ \underbrace{C_1\,W_2(p_0, p_{\mathrm{on}})\,\Gamma_{n,\delta}}_{\text{depends on the start } p_0}, \qquad C_0 := c(B+\sqrt{\mathrm{Tr}(\Sigma_{\mathrm{on}}) + C_{\mathrm{TR}}}),\ \ C_1 := c. \tag{18}$$

Composite vs. direct. Consider two trainings sharing the same online p_{on} and class $\mathcal{H}$:

$$\textbf{(Composite step)} \quad p_0 \ =\ \hat{p}_{\mathrm{off}}(\cdot \mid s, a), \tag{19}$$

$$\textbf{(Direct step)} \quad p_0 \ =\ p_G \ =\ \mathcal{N}(0, I_d). \tag{20}$$

Applying (18) to (19) and (20) yields

$$\mathbb{E}\ \left\|\hat{v}_{\mathrm{comp}} - v^\star_{\mathrm{comp}}\right\|_2^2 \ \le\ C_0\,\Gamma_{n,\delta} + C_1\,W_2\big(\hat{p}_{\mathrm{off}}, p_{\mathrm{on}}\big)\,\Gamma_{n,\delta}, \tag{21}$$

$$\mathbb{E}\ \|\hat{v}_G - v^\star_G\|_2^2 \ \le\ C_0\,\Gamma_{n,\delta} + C_1\,W_2\big(p_G, p_{\mathrm{on}}\big)\,\Gamma_{n,\delta}. \tag{22}$$

When is the composite bound strictly tighter? Applying (18) to $p_0 = \hat{p}_{\mathrm{off}}$ (composite) and $p_0 = p_G$ (direct) gives (21) and (22). Since C_0 and $\Gamma_{n,\delta}$ are identical across the two trainings (they depend on $(B, c, \Sigma_{\mathrm{on}}, C_{\mathrm{TR}})$, n, δ, and $\mathcal{H}$, but not on p_0), the *only* difference in the right-hand sides is the factor $W_2(\cdot, p_{\mathrm{on}})$.

Therefore the composite bound (21) is *strictly smaller* than the direct bound (22) if and only if

$$W_2\big(\hat{p}_{\mathrm{off}}, p_{\mathrm{on}}\big) \ <\ W_2\big(p_G, p_{\mathrm{on}}\big). \tag{23}$$

□

Remark G.2 (Optional D_2 strengthening). One can analogously replace $W_2(p_0, p_{\mathrm{on}})$ by $D_2(p_0, p_{\mathrm{on}})$ when bounding 14, using the standard inequality trick $\sqrt{x} \le \frac{1}{2}(x+1)$ for all $x \ge 0$. We use W_2 metric because it's more commonly adopted in the literature, but D_2 leads to a tighter bound since we no longer needs the constant C_{TR}. We do not expand the D_2 case here, as the argument mirrors the W_2 case.

Lemma G.3 (Generalization gap for squared loss under uniform boundedness)**.** *Assume **A1**. Let $\ell_v(z) := \|v(x,t) - y\|_2^2$ for $z = (x, t, y)$ generated as in the theorem. Then there exists a positive constant $c > 0$ such that, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over an i.i.d. sample of size n,*

$$R(\hat{v}) \ \le\ R(v^\star) \ +\ c\left(B + \sqrt{\mathbb{E}\ \|Y\|_2^2}\right)\Gamma_{n,\delta}, \qquad \Gamma_{n,\delta} := \mathfrak{R}_n(\mathcal{H}) + \sqrt{\tfrac{\log(1/\delta)}{n}}.$$

Proof. *We sketch the standard steps and make constants explicit where needed:*

(i) Symmetrization. *Let $\hat{R}(v)$ denote the empirical risk. For ERM $\hat{v}$,*

$$R(\hat{v}) - R(v^\star) \ \le\ 2\ \sup_{v \in \mathcal{H}}\big(R(v) - \hat{R}(v)\big)$$

up to negligible terms; this is standard (e.g., by a one-sided symmetrization plus a chaining step). Thus it suffices to bound $\sup_{v\in\mathcal{H}}(R(v) - \hat{R}(v))$.

(ii) Lipschitz envelope via contraction. *For fixed (x, t, y) with $\|v(x,t)\| \le B$ and any $u \in \mathbb{R}^d$ with $\|u\| \le B$,*

$$\Big|\ \|u - y\|^2 - \|v(x,t) - y\|^2\ \Big| = \big|\ \langle u - v(x,t),\ (u + v(x,t) - 2y)\rangle\ \big| \le 2\,(B + \|y\|)\ \|u - v(x,t)\|\,.$$

Thus, as a function of u, the map $u \mapsto \|u - y\|^2$ is $(2(B + \|y\|))$-Lipschitz on the B-ball. Applying the vector contraction inequality to the class $\{v(\cdot, \cdot) \in \mathcal{H}\}$ gives

$$\mathbb{E} \sup_{v \in \mathcal{H}} \big(R(v) - \hat{R}(v)\big) \lesssim \Big(\mathbb{E}\,[\,B + \|Y\|\,]\Big)\,\mathfrak{R}_n(\mathcal{H}).$$

By Cauchy–Schwarz and Jensen, $\mathbb{E}\,\|Y\| \le \sqrt{\mathbb{E}\,\|Y\|^2}$.

(iii) High-probability upgrade. *A standard bounded differences (or Bernstein-type) argument for Lipschitz, sub-quadratic losses upgrades the expected sup-gap to a high-probability bound, adding the usual $\sqrt{\log(1/\delta)/n}$ term. Collecting constants and using $\mathbb{E}\,\|Y\| \le \sqrt{\mathbb{E}\,\|Y\|^2}$ yields the claimed form with some absolute $c > 0$:*

$$\sup_{v \in \mathcal{H}} \big(R(v) - \hat{R}(v)\big) \le c\,\Big(B + \sqrt{\mathbb{E}\,\|Y\|^2}\Big)\,\Gamma_{n,\delta}.$$

Since $\hat{v}$ is an ERM, $\hat{R}(\hat{v}) \le \hat{R}(v^\star)$, and therefore $R(\hat{v}) - R(v^\star) \le 2 \sup_v (R(v) - \hat{R}(v))$ in the final step, absorbing the factor 2 into c. □

Lemma G.4. *Let $\mu.$ and $\Sigma.$ denote mean and covariance, respectively. We show that if $\mathrm{Tr}(\Sigma_0) \le C_{\mathrm{TR}}$ for some $C_{\mathrm{TR}} > 0$, we have:*

$$\sqrt{D_2(p_0, p_1)} \le \sqrt{\mathrm{Tr}(\Sigma_1) + C_{\mathrm{TR}}} + W_2(p_0, p_1)$$

Proof. Let $X_0 \sim p_0$ and $X_1 \sim p_1$ be independent with means μ_0, μ_1 and covariances Σ_0, Σ_1. We compute

$$D_2(p_0, p_1) = \mathbb{E}\,\|X_1 - X_0\|_2^2 = \mathbb{E}\,\|X_1\|_2^2 + \mathbb{E}\,\|X_0\|_2^2 - 2\,\mathbb{E}\,\langle X_1, X_0\rangle.$$

By independence, $\mathbb{E}\,\langle X_1, X_0\rangle = \langle \mathbb{E}X_1, \mathbb{E}X_0\rangle = \langle \mu_1, \mu_0\rangle$. Moreover, for any random vector Z with mean μ and covariance Σ, $\mathbb{E}\,\|Z\|_2^2 = \|\mu\|_2^2 + \mathrm{Tr}(\Sigma)$. Thus

$$\begin{aligned} D_2(p_0, p_1) &= \big(\,\|\mu_1\|_2^2 + \mathrm{Tr}(\Sigma_1)\big) + \big(\,\|\mu_0\|_2^2 + \mathrm{Tr}(\Sigma_0)\big) - 2\,\langle \mu_1, \mu_0\rangle \\ &= \|\mu_1 - \mu_0\|_2^2 + \mathrm{Tr}(\Sigma_1 + \Sigma_0), \end{aligned}$$

Let $\Gamma(p_0, p_1)$ be all couplings of p_0, p_1. For any $\pi \in \Gamma(p_0, p_1)$ with $(X_0, X_1) \sim \pi$, we have

$$\mathbb{E}_\pi \|X_1 - X_0\|_2^2 \ge \big\|\mathbb{E}_\pi[X_1 - X_0]\big\|_2^2 = \|\mu_1 - \mu_0\|_2^2,$$

by Jensen's inequality since $z \mapsto \|z\|_2^2$ is convex. Taking the infimum over all couplings gives

$$W_2^2(p_0, p_1) = \inf_{\pi \in \Gamma(p_0, p_1)} \mathbb{E}_\pi \|X_1 - X_0\|_2^2 \ge \|\mu_1 - \mu_0\|_2^2.$$

Therefore, we have $\|\mu_1 - \mu_0\|_2^2 \le W_2^2(p_0, p_1)$. Hence,

$$\begin{aligned} \sqrt{D_2(p_0, p_1)} &= \sqrt{\|\mu_1 - \mu_0\|_2^2 + \mathrm{Tr}(\Sigma_1 + \Sigma_0)} \\ &\le \sqrt{\mathrm{Tr}(\Sigma_0 + \Sigma_1)} + \sqrt{\|\mu_1 - \mu_0\|_2^2} \\ &\le \sqrt{\mathrm{Tr}(\Sigma_1) + C_{\mathrm{TR}}} + W_2(p_0, p_1) \end{aligned}$$

This completes the proof. □

The assumption that $\mathrm{Tr}(\Sigma_0) \le C_{\mathrm{TR}}$ makes sense because we assume bounded state space for all settings considered in our paper.

H Proof of Theorem 3.4

Theorem 3.4 (Large Dynamics Gap Exploration Reduces Performance Gap)**.** *Compared to behavior cloning policy π_{bc} on the offline dataset, training a policy $\hat{\pi}$ by replacing all offline samples with a dynamics gap exceeding κ (as estimated by the composite flow) with online environment samples can reduce the performance gap to the optimal online policy π^*_{on} with high probability by*

$$\frac{2L_r(1+\gamma)}{(1-\gamma)(1-\gamma L_p)}\big(\Delta_{W_2} - \kappa - \sqrt{(C_0 + C_1\, W_2(\hat{p}_{\mathrm{off}}, p_{\mathrm{on}}))\,\Gamma_{N_{\mathrm{on}},\delta}}\big). \tag{6}$$

Here L_r and L_P are the Lipschitz constants for the reward and transition function, respectively. $\gamma L_p < 1$. $\Delta_{W_2} := \sup_{s,a} W_2\big(p_{\mathrm{off}}(\cdot|s,a),\ p_{\mathrm{on}}(\cdot|s,a)\big)$ is the largest dynamics gap. C_0 and C_1 are two constants. $\Gamma_{N_{\mathrm{on}},\delta} := \mathfrak{R}_{N_{\mathrm{on}}}(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{N_{\mathrm{on}}}}$, where $\mathcal{H}$ is the same as in Theorem 3.1 and N_{on} is the number of samples used to train the online flow.

Proof. Let's first list the assumptions:

R1: Lipschitz reward. For every $a \in \mathcal{A}$, $s \mapsto r(s,a)$ is L_r-Lipschitz.

R2: Lipschitz dynamics. For $m \in \{\mathrm{off}, \mathrm{on}\}$, $W_1\big(p_m(\cdot \mid s,a), p_m(\cdot \mid s',a)\big) \le L_p\, d(s,s')$ for all s, s', a.

R3: Contraction. $\gamma L_p < 1$.

We proceed in four steps: (A) a policy-dependent model-gap bound; (B) a uniform high-probability W_2-gap estimation bound for the rectified flow (α); (C) learning-gap bounds for π_{bc} and $\hat{\pi}$ (β); (D) assembly via the standard three-term decomposition.

Step A: Policy-dependent model-gap bound. For any fixed policy π, we cite Theorem 3.1 of [45] for the following bound. Under **R1**–**R3**, we have

$$\begin{aligned}
\|V^\pi_{\mathrm{on}} - V^\pi_{\mathrm{off}}\|_\infty &\le 2L_r\Delta_{W_1}(1+\gamma)\sum_{i=0}^{\infty}\gamma^i\sum_{j=0}^{i}(L_P)^j \\
&= 2L_r\Delta_{W_1}(1+\gamma)\sum_{j=0}^{\infty}(L_P)^j\sum_{i=j}^{\infty}\gamma^i \\
&= 2L_r\Delta_{W_1}(1+\gamma)\sum_{j=0}^{\infty}(L_P)^j\cdot\frac{\gamma^j}{1-\gamma} \\
&= \frac{2L_r\Delta_{W_1}(1+\gamma)}{1-\gamma}\sum_{j=0}^{\infty}(\gamma L_P)^j \\
&= \frac{2L_r\Delta_{W_1}(1+\gamma)}{(1-\gamma)(1-\gamma L_P)}
\end{aligned} \tag{24}$$

Averaging over the initial state-distribution μ gives us a bound on what we call the model gap:

$$\begin{aligned}
\big|\eta_{\mathcal{M}_{\mathrm{on}}}(\pi) - \eta_{\mathcal{M}_{\mathrm{off}}}(\pi)\big| &= \Big|\int_{\mathcal{S}}\big(V^\pi_{\mathrm{on}}(s) - V^\pi_{\mathrm{off}}(s)\big)\,\mu(ds)\Big| \\
&\le \int_{\mathcal{S}}\big|V^\pi_{\mathrm{on}}(s) - V^\pi_{\mathrm{off}}(s)\big|\,\mu(ds). \\
&\le \|V^\pi_{\mathrm{on}} - V^\pi_{\mathrm{off}}\|_\infty \\
&\le \frac{2L_r\Delta_{W_1}(1+\gamma)}{(1-\gamma)(1-\gamma L_p)}.
\end{aligned}$$

Moreover, the bounded state space assumption in Theorem 3.1 implies bounded second moment for both p_{off} and p_{on}, so we have $W_1(p_{\text{off}}, p_{\text{on}}) \le W_2(p_{\text{off}}, p_{\text{on}})\ \forall p_{\text{off}}, p_{\text{on}}$. Hence, we obtain that

$$|\eta_{\mathcal{M}_{\text{on}}}(\pi) - \eta_{\mathcal{M}_{\text{off}}}(\pi)| \ \le\ \frac{2L_r(1+\gamma)}{(1-\gamma)(1-\gamma L_p)}\Delta_{W_1} \tag{25}$$

$$\le\ \frac{2L_r(1+\gamma)}{(1-\gamma)(1-\gamma L_p)}\Delta_{W_2}. \tag{26}$$

Step B: FM-based uniform W_2 estimation bound (construction of α). We need a high-probability *uniform* control of the error of the rectified-flow W_2 estimator to ensure that thresholding by κ actually caps the *on-policy* mismatch for $\hat{\pi}$.

B.1. Flow-to-map error. Let T be the Brenier OT map from $\hat{p}_{\text{off}}(\cdot \mid s,a)$ to $p_{\text{on}}(\cdot \mid s,a)$ (guaranteed by standard assumptions), and let $\widehat{T}$ be the terminal map obtained by integrating the learned FM velocity field $\hat{v}$ from $t=0$ to $t=1$. Under the linear path and Lipschitz regularity of $v^\star$ and $\hat{v}$ (Theorem 3.1 assumptions), a stability argument (Gronwall) yields

$$\mathbb{E}_{X_0\sim\hat{p}_{\text{off}}}\big\|\widehat{T}(X_0) - T(X_0)\big\|_2^2 \ \le\ K_{\text{stab}} \int_0^1 \mathbb{E}\big\|\hat{v}(X_t,t) - v^\star(X_t,t)\big\|_2^2\, dt, \tag{27}$$

for some finite K_{stab} depending on Lipschitz constants of the flow. By the "master bound" (Eq. (7) in Theorem 3.1), uniformly over $t\in[0,1]$, with prob. $\ge 1-\delta$,

$$\mathbb{E}\big\|\hat{v} - v^\star\big\|_2^2 \ \le\ C_0\,\Gamma_{N_{\text{on}},\delta} + C_1\, W_2(\hat{p}_{\text{off}}, p_{\text{on}})\,\Gamma_{N_{\text{on}},\delta}. \tag{28}$$

So we have

$$\underbrace{\mathbb{E}\|\widehat{T}(X_0) - T(X_0)\|_2^2}_{\text{terminal map MSE}} \overset{\text{Gronwall}}{\le} K_{\text{stab}} \int_0^1 \underbrace{\mathbb{E}\|\hat{v} - v^\star\|_2^2}_{\text{FM excess risk at } t}\, dt \tag{29}$$

$$\le K_{\text{stab}}\,(C_0 + C_1\, W_2(\hat{p}_{\text{off}}, p_{\text{on}}))\,\Gamma_{N_{\text{on}},\delta} \tag{30}$$

B.2. From map MSE to W_2 error. By the coupling construction $(\hat{s}, s) = (\widehat{T}(X_0), T(X_0))$ with $X_0 \sim \hat{p}_{\text{off}}$ and the triangle inequality for W_2,

$$\sup_{(s,a)} \Big|\, \widehat{\Delta}_{W_2}(s,a) - \Delta_{W_2}(s,a) \,\Big| \ \le\ \sup_{(s,a)}\ W_2\big(\hat{p}_{\text{on}}, p_{\text{on}}\big) \tag{31}$$

Define the coupling $\gamma_{(s,a)} := (T,\widehat{T})_{\#}\hat{p}_{\text{off}}(\cdot \mid s,a)$, i.e., if $S := T(X_0)$ and $\widehat{S} := \widehat{T}(X_0)$ then $(S,\widehat{S}) \sim \gamma_{(s,a)}$ and $\gamma_{(s,a)} \in \Pi\big(p_{\text{on}}(\cdot \mid s,a), \hat{p}_{\text{on}}(\cdot \mid s,a)\big)$. By the definition of the 2-Wasserstein distance,

$$W_2^2\big(p_{\text{on}}(\cdot \mid s,a), \hat{p}_{\text{on}}(\cdot \mid s,a)\big) = \inf_{\gamma\in\Pi(p_{\text{on}},\hat{p}_{\text{on}})} \int \|x-y\|_2^2\, \mathrm{d}\gamma(x,y) \tag{32}$$

$$\le\ \int \|x-y\|_2^2\, \mathrm{d}\gamma_{(s,a)}(x,y) = \mathbb{E}\,\|\widehat{T}(X_0) - T(X_0)\|_2^2. \tag{33}$$

Taking square roots gives, for each (s,a),

$$W_2\big(p_{\text{on}}(\cdot \mid s,a), \hat{p}_{\text{on}}(\cdot \mid s,a)\big) \ \le\ \sqrt{\mathbb{E}\,\|\widehat{T}(X_0) - T(X_0)\|_2^2}.$$

Finally, taking the supremum over (s,a) preserves the inequality:

$$\sup_{(s,a)} W_2\big(\hat{p}_{\text{on}}(\cdot \mid s,a), p_{\text{on}}(\cdot \mid s,a)\big) \ \le\ \sup_{(s,a)} \sqrt{\mathbb{E}\,\|\widehat{T}(X_0) - T(X_0)\|_2^2}. \tag{34}$$

Combining (34) and (31) gives

$$\sup_{(s,a)} \Big|\, \widehat{\Delta}_{W_2}(s,a) - \Delta_{W_2}(s,a) \,\Big| \ \le\ \alpha_{N_{\text{on}},\delta} := \sqrt{K_{\text{stab}}\,(C_0 + C_1\, W_2(\hat{p}_{\text{off}}, p_{\text{on}}))\,\Gamma_{N_{\text{on}},\delta}},$$

with probability at least $1-\delta$.

B.3. Capping the on-policy gap for $\hat{\pi}$. By construction of the replacement rule, for any (s,a) encountered during fine-tuning of $\hat{\pi}$ we have either $\widehat{\Delta}_{W_2}(s,a) \le \kappa$ (kept offline sample) or we replaced it by an *online* sample. In either case, on those (s,a) we ensure

$$\Delta_{W_2}(s,a) \;\le\; \widehat{\Delta}_{W_2}(s,a) + \alpha_{N_{\mathrm{on}},\delta} \;\le\; \kappa + \alpha_{N_{\mathrm{on}},\delta}.$$

Therefore the *on-policy* gap of $\hat{\pi}$ obeys

$$\Delta_{W_2}^{(\hat{\pi})} \;\le\; \kappa + \alpha_{N_{\mathrm{on}},\delta}. \tag{35}$$

Step C: Learning-gap bounds for π_{bc} and $\widehat{\pi}$ (construction of β). Recall the three-term decomposition for any policy π:

$$\eta_{\mathrm{on}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{on}}(\pi) = \underbrace{\big(\eta_{\mathrm{on}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}})\big)}_{\text{model(a)}} + \underbrace{\big(\eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\pi)\big)}_{\text{learning(b)}} + \underbrace{\big(\eta_{\mathrm{off}}(\pi) - \eta_{\mathrm{on}}(\pi)\big)}_{\text{model(c)}}. \tag{36}$$

We now bound the *learning* term (b) by ERM generalization.

Policies are learned by ERM on a surrogate imitation loss $\mathcal{L}(\pi)$ over a policy class Π with Rademacher complexity $\mathfrak{R}_N(\Pi)$. There exists a calibration constant C_{val} (depends on concentrability/Lipschitzness; fixed for the class) such that, for any π,

$$\eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\pi) \;\le\; C_{\mathrm{val}}\Big(\mathcal{L}(\pi) - \inf_{\pi'\in\Pi} \mathcal{L}(\pi')\Big). \tag{37}$$

Moreover, for datasets of sizes $N_{\mathrm{off}}, N_{\mathrm{mod}}$, with probability at least $1-\delta$,

$$\mathcal{L}(\hat{\pi}) - \inf_{\pi'\in\Pi} \mathcal{L}(\pi') \;\le\; C_\Pi\Big(\mathfrak{R}_N(\Pi) + \sqrt{\tfrac{\log(1/\delta)}{N}}\Big), \qquad N \in \{N_{\mathrm{off}}, N_{\mathrm{mod}}\}, \tag{38}$$

for an absolute constant C_Π.

Let π_{bc} be the behavior cloning policy trained on the *offline* dataset of size N_{off}, and let $\hat{\pi}$ be ERM on the modified dataset (size N_{mod}): for any (s,a) encountered, if the FM-estimated gap satisfies $\widehat{\Delta}_{W_2}(s,a) > \kappa$, replace the *offline* (s,a,s') sample by an *online* (s,a,s'_{on}) sample (collecting N_{on} such online transitions), leaving all others unchanged. Denote N_{mod} the resulting (modified) dataset size.

By calibration (37) and the generalization inequality (38), with probability at least $1-\delta$ we have

$$\eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\pi_{\mathrm{bc}}) \;\le\; C_{\mathrm{val}}\, C_\Pi\Big(\mathfrak{R}_{N_{\mathrm{off}}}(\Pi) + \sqrt{\tfrac{\log(1/\delta)}{N_{\mathrm{off}}}}\Big),$$

and similarly

$$\eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\hat{\pi}) \;\le\; C_{\mathrm{val}}\, C_\Pi\Big(\mathfrak{R}_{N_{\mathrm{mod}}}(\Pi) + \sqrt{\tfrac{\log(1/\delta)}{N_{\mathrm{mod}}}}\Big).$$

By a union bound (probability $1-2\delta$) and adding these two upper bounds we obtain a common envelope

$$\max\Big\{\eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\pi_{\mathrm{bc}}),\; \eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\hat{\pi})\Big\} \;\le\; \beta, \tag{39}$$

with

$$\beta := C_{\mathrm{val}} C_\Pi\left(\mathfrak{R}_{N_{\mathrm{off}}}(\Pi) + \mathfrak{R}_{N_{\mathrm{mod}}}(\Pi) + \sqrt{\tfrac{\log(2/\delta)}{N_{\mathrm{off}}}} + \sqrt{\tfrac{\log(2/\delta)}{N_{\mathrm{mod}}}}\right). \tag{40}$$

Step D: Assemble bounds and compare. Incorporate (26) and (39) to (36) with $\pi = \pi_{\mathrm{bc}}$, we have with high probability,

$$\eta_{\mathrm{on}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{on}}(\pi_{\mathrm{bc}}) \le \big(\eta_{\mathrm{on}}(\pi^\star_{\mathrm{on}}) - \eta_{\mathrm{off}}(\pi^\star_{\mathrm{on}})\big) + \beta + \frac{2L_r(1+\gamma)}{(1-\gamma)(1-\gamma L_p)}\Delta_{W_2}$$

Incorporate (26) and (39) to (36) with $\pi = \hat{\pi}$, we have with high probability

$$\eta_{\text{on}}(\pi^\star_{\text{on}}) - \eta_{\text{on}}(\widehat{\pi}) \;\le\; \big(\eta_{\text{on}}(\pi^\star_{\text{on}}) - \eta_{\text{off}}(\pi^\star_{\text{on}})\big) + \beta + \frac{2L_r(1+\gamma)}{(1-\gamma)(1-\gamma L_p)}(\kappa + \alpha_{N_{\text{on}},\delta})$$

The difference between these two upper bounds is:

$$\frac{2L_r(1+\gamma)}{(1-\gamma)(1-\gamma L_p)}\big(\Delta_{W_2} - \kappa - \alpha_{N_{\text{on}},\delta}\big).$$

Absorbing K_{stab} into C_0 and C_1 gets the result. □

I Proof of Proposition 3.3

We provide the formal version of Proposition 3.3 as below.

Proposition I.1 (Shared-Latent Coupling Approximates W_2-Optimal Transport)**.** We first introduce some notations for simplicity:

Notation. Fix a state–action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, and omit (s, a) in the following for notational simplicity. Let the latent distribution be $p_0(x_0) = \mathcal{N}(0, \mathbf{I})$. Define the offline transition distribution as $p_1(x_1) := p_{\text{off}}(s'|s, a)$, and the online transition distribution as $p_2(x_2) := p_{\text{on}}(s'|s, a)$.

We denote the pretrained source flow $\psi_\theta(x, t|s, a)$ by $\psi_\theta(x, t)$, and similarly write the target flow $\psi_\phi(x, t|s, a)$ as $\psi_\phi(x, t)$. Let $\psi^k_\phi(x, t)$ denote the target flow model trained on a batch of size k from the target domain.

Assumptions.

(A1) The distributions p_1 and p_2 have bounded support: there exists $C > 0$ such that $\|x\| \le C$ for all $x \in \text{supp}(p_1) \cup \text{supp}(p_2)$.

(A2) The distribution p_1 admits a density, and the optimal transport map between p_1 and p_2 under the quadratic cost is continuous.

(A3) At each training iteration, we compute the optimal transport plan for the current minibatch.

Result. Let $W_2^2(p_1, p_2)$ denote the squared 2-Wasserstein distance between p_1 and p_2, defined as

$$W_2^2(p_1, p_2) := \min_{q \in \Pi(p_1, p_2)} \mathbb{E}_{(x_1, x_2) \sim q} \left[\|x_2 - x_1\|_2^2\right],$$

where $\Pi(p_1, p_2)$ denotes the set of all couplings (i.e., joint distributions) with marginals p_1 and p_2.

Then, in the limit as $\eta \to \infty$ (in the cost function of Algorithm 3) and $k \to \infty$, we have

$$\lim_{k\to\infty} \mathbb{E}_{x_0 \sim \mathcal{N}(0,\mathbf{I})} \left[\left\|\psi_\theta(x_0, 1) - \psi^k_\phi\left(\psi_\theta(x_0, 1), 2\right)\right\|^2\right] = W_2^2(p_1, p_2),$$

where ψ_θ denotes the source flow and ψ^k_ϕ is the target flow trained on a batch of size k.

Proof. We begin by reparameterizing the expectation using the definition of the pushforward distribution induced by the source flow, $p_1 = \psi_\theta(x, 1)\#p_0$. Specifically,

$$\lim_{k\to\infty} \mathbb{E}_{x_0 \sim p_0} \left[\left\|\psi_\theta(x_0, 1) - \psi^k_\phi\left(\psi_\theta(x_0, 1), 2\right)\right\|^2\right] = \lim_{k\to\infty} \mathbb{E}_{x_1 \sim p_1} \left[\left\|x_1 - \psi^k_\phi(x_1, 2)\right\|^2\right].$$

According to Proposition 3.1 in [18], as $\eta \to \infty$, mass transport in the conditional variables (s, a) vanishes. That is, the transport is concentrated exclusively in the next-state space, and optimal transport is performed independently within each fixed (s, a) pair when $\eta \to \infty$.

Therefore, for any fixed (s, a), we can invoke Theorem 4.2 from [47], which guarantees that the expected transport cost under the batch-trained flow ψ^k_ϕ converges to the squared 2-Wasserstein distance between p_1 and p_2 as the batch size $k \to \infty$. Hence,

$$\lim_{k\to\infty} \mathbb{E}_{x_1 \sim p_1} \left[\left\|x_1 - \psi^k_\phi(x_1, 2)\right\|^2\right] = W_2^2(p_1, p_2),$$

which concludes the proof. □

J Experimental Details of Gym-MuJoCo

In this section, we describe the detailed experimental setup as well as the hyperparameter setup used in this work.

J.1 Environment Setting

J.1.1 Offline Dataset

We use the MuJoCo datasets from D4RL [16] as our offline data. These datasets are collected from continuous control environments in Gym [4], simulated using the MuJoCo physics engine [61]. We focus on three benchmark tasks: *HalfCheetah*, *Hopper*, and *Walker2d*, and evaluate across three dataset types: *medium*, *medium-replay*, and *medium-expert*.

- The *medium* datasets consist of trajectories generated by an SAC policy trained for 1M steps and then early stopped.
- The *medium-replay* datasets capture the replay buffer of a policy trained to the performance level of the medium agent.
- The *medium-expert* datasets are formed by mixing equal proportions of medium and expert data (50-50).

J.1.2 Kinematic Shift Tasks

We use Kinematic Shift Tasks from the benchmark [38]. We select most shift level 'hard' to make the tasks more challenging

• **HalfCheetah Kinematic Shift:** The rotation range of the foot joint is modified to be:

```
<joint axis="0 1 0" damping="3" name="bfoot" pos="0 0 0" range="-.08 .157" stiffness
    ="120" type="hinge"/>
<joint axis="0 1 0" damping="1.5" name="ffoot" pos="0 0 0" range="-.1 .1" stiffness=
    "60" type="hinge"/>
```

• **Hopper Kinematic Shift:** the rotation range of the foot joint is modified from $[-45, 45]$ to $[-9, 9]$:

```
<joint axis="0 -1 0" name="foot_joint" pos="0 0 0.1" range="-9 9" type="hinge"/>
```

• **Walker2D Kinematic Shift:** the rotation range of the foot joint is modified from $[-45, 45]$ to $[-9, 9]$:

```
<joint axis="0 -1 0" name="foot_joint" pos="0 0 0.1" range="-9 9" type="hinge"/>
<joint axis="0 -1 0" name="foot_left_joint" pos="0 0 0.1" range="-9 9" type="hinge"
    />
```

J.1.3 Morphology Shift Tasks

We use Morphology Shift Tasks from the benchmark [38]. We select most shift level 'hard' to make the tasks more challenging

• **HalfCheetah Morphology Shift:** the front thigh size and the back thigh size are modified to be:

```
<geom fromto="0 0 0.02 0 -0.02" name="bthigh" size="0.046" type="capsule"/>
<body name="fshin" pos="0.02 0 -0.02">
  <geom fromto="0 0 0 -.13 0 -.15" name="bshin" rgba="0.9 0.6 0.6 1" size="0.046"
    type="capsule"/>
</body>
<body name="bfoot" pos="-.13 0 -.15">
  <geom fromto="0 0 0 -.04 0 -0.05" name="fthigh" size="0.046" type="capsule"/>
</body>
<body name="fshin" pos="0 -.04 0 -0.05">
  <geom fromto="0 0 0 .11 0 -.13" name="fshin" rgba="0.9 0.6 0.6 1" size="0.046"
    type="capsule"/>
</body>
<body name="ffoot" pos=".11 0 -.13"/>
```

• **Hopper Morphology Shift:** the foot size is revised to be 0.4 times of that within the source domain:

```
<geom friction="2.0" fromto="-0.052 0 0.1 0.104 0 0.1" name="foot_geom" size="0.024"
     type="capsule"/>
```

• **Walker2D Morphology Shift:** the leg size of the robot is revised to be 0.2 times of that in the source domain.

```
<geom friction="0.9" fromto="0 0 1.05 0 0 0.2" name="thigh_geom" size="0.05" type="
    capsule"/>
<joint axis="0 -1 0" name="leg_joint" pos="0 0 0.2" range="-150 0" type="hinge"/>
<geom friction="0.9" fromto="0 0 0.2 0 0 0.1" name="leg_geom" size="0.04" type="
    capsule"/>
<geom friction="0.9" fromto="0 0 1.05 0 0 0.2" name="thigh_left_geom" rgba=".7 .3 .6
     1" size="0.05" type="capsule"/>
<joint axis="0 -1 0" name="leg_left_joint" pos="0 0 0.2" range="-150 0" type="hinge"
    />
<geom friction="0.9" fromto="0 0 0.2 0 0 0.1" name="leg_left_geom" rgba=".7 .3 .6 1"
     size="0.04" type="capsule"/>
```

J.1.4 Friction Shift Tasks

Following [38], the friction shift is implemented by altering the friction attribute in the geom elements. The frictional components are adjusted to 5.0 times the frictional components in the offline environment. The following is an example for the Hopper robot.

Listing 1: Geometry Definitions for Walker2D

```
# torso
<geom friction="4.5" fromto="0 0 1.45 0 0 1.05" name="torso_geom" size="0.05" type="
    capsule"/>
# thigh
<geom friction="4.5" fromto="0 0 1.05 0 0 0.6" name="thigh_geom" size="0.05" type="
    capsule"/>
# leg
<geom friction="4.5" fromto="0 0 0.6 0 0 0.1" name="leg_geom" size="0.04" type="
    capsule"/>
# foot
<geom friction="10.0" fromto="-0.13 0 0.1 0.26 0 0.1" name="foot_geom" size="0.06"
    type="capsule"/>
```

J.2 Implementation Details

BC-SAC: This baseline leverages both offline and online transitions for policy learning. Since learning from offline data requires conservatism while online data does not, we incorporate a behavior cloning term into the actor update of the SAC algorithm. Specifically, the critic is updated using standard Bellman loss on the combined offline and online datasets, and the actor is optimized as:

$$\mathcal{L}_{\text{actor}} = \lambda \cdot \mathbb{E}_{s \sim \mathcal{D}_{\text{off}} \cup \mathcal{D}_{\text{on}},\, a \sim \pi_\varphi(\cdot|s)} \left[\min_{i=1,2} Q_{\varsigma_i}(s,a) - \alpha \log \pi_\varphi(\cdot|s) \right] + \mathbb{E}_{(s,a) \sim \mathcal{D}_{\text{off}},\, \hat{a} \sim \pi_\varphi(\cdot|s)} \left[(a - \hat{a})^2 \right], \tag{41}$$

where $\lambda = \frac{\omega}{\frac{1}{N} \sum_{(s_j, a_j)} \min_{i=1,2} Q_{\varsigma_i}(s_j, a_j)}$ and $\omega \in \mathbb{R}^+$ is a normalization coefficient. We train BC-SAC for 400K gradient steps, collecting online data every 10 steps. We use the hyperparameters recommended in [38].

H2O: H2O [46] trains domain classifiers to estimate dynamics gaps and uses them as importance sampling weights during critic training. It also incorporates a CQL loss to encourage conservatism. Since the original H2O is designed for the Online-Offline setting, we adapt the objective to the Offline-Online setting. The critic loss is:

$$\begin{aligned} \mathcal{L}_{\text{critic}} = {} & \mathbb{E}_{\mathcal{D}_{\text{on}}} \left[(Q_{\varsigma_i}(s,a) - y)^2 \right] + \mathbb{E}_{\mathcal{D}_{\text{off}}} \left[\omega(s,a,s') (Q_{\varsigma_i}(s,a) - y)^2 \right] \\ & + \beta_{\text{CQL}} \left(\mathbb{E}_{s \sim \mathcal{D}_{\text{off}},\, \hat{a} \sim \pi_\varphi(\cdot|s)} \left[\omega(s,a,s') Q_{\varsigma_i}(s,\hat{a}) \right] - \mathbb{E}_{\mathcal{D}_{\text{off}}} \left[\omega(s,a,s') Q_{\varsigma_i}(s,a) \right] \right), \quad i \in \{1,2\}, \end{aligned} \tag{42}$$

where $\omega(s, a, s')$ is the dynamics-based importance weight, and β_{CQL} is the penalty coefficient. We set $\beta_{\text{CQL}} = 10.0$, which performs better than the default 0.01. We reproduce H2O using the official codebase,[3] and adopt the suggested hyperparameters. H2O is trained for 40K environment steps, with 10 gradient updates per step.

BC-VGDF: BC-VGDF [63] filters offline transitions whose estimated values align closely with those from the online environment. It trains an ensemble of dynamics models to predict next states from raw state-action pairs under the online dynamics. Each predicted next state is evaluated by the policy to obtain a value ensemble $\{Q(s'_i, a'_i)\}_{i=1}^{M}$, forming a Gaussian distribution. A fixed percentage ($\xi\%$) of offline transitions with the highest likelihood under this distribution are retained. The critic loss is:

$$\begin{aligned}\mathcal{L}_{\text{critic}} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_{\text{on}}}\left[\left(Q_{\varsigma_i}(s,a) - y\right)^2\right] \\ + \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_{\text{off}}}\left[\mathbf{1}\left(\Lambda(s,a,s') > \Lambda_{\xi\%}\right)\left(Q_{\varsigma_i}(s,a) - y\right)^2\right], \quad i \in \{1,2\},\end{aligned} \tag{43}$$

where $\Lambda(s, a, s')$ denotes the fictitious value proximity (FVP), and $\Lambda_{\xi\%}$ is the ξ-quantile threshold. VGDF also trains an exploration policy. Actor training includes a behavior cloning term, as in BC-SAC. We follow the official implementation,[4] use the recommended hyperparameters, and train for 40K environment steps with 10 gradient updates per step.

BC-PAR: BC-PAR [37] addresses dynamics mismatch through representation mismatch, measured as the deviation between the encoded source state-action pair and its next state. It employs a state encoder f_ψ and a state-action encoder g_ξ, both trained on the target domain. The encoder loss is:

$$\mathcal{L}(\psi, \xi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}_{\text{on}}}\left[\left(g_\xi(f_\psi(s), a) - \text{SG}(f_\psi(s'))\right)^2\right], \tag{44}$$

where SG is the stop-gradient operator. Rewards of the offline data are adjusted as:

$$\hat{r}_{\text{PAR}} = r_{\text{off}} - \beta \cdot \left\|g_\xi(f_\psi(s_{\text{off}}), a_{\text{off}}) - f_\psi(s'_{\text{off}})\right\|^2, \tag{45}$$

where β controls the penalty strength. The actor (π_φ) and critic (Q_{ς_i}) are jointly trained using both offline and online data. Actor training includes a behavior cloning term, similar to BC-SAC. We implement BC-PAR using the official codebase,[5] adopt the suggested hyperparameters, and train for 40K environment steps with 10 gradient updates per step.

CompFlow. When training the target flow, we use a quadratic cost function and employ the Python Optimal Transport (POT) library [15] to compute the optimal transport plan for each minibatch using the `exact` solver. Additional hyperparameters are provided in Table 2 and Table 3. Since the exploration bonus term is closely tied to properties of the environment—such as the state space, action space, and reward structure—it is expected that the optimal exploration strength β varies across tasks. We perform a sweep over $\beta \in \{0.01, 0.1, 0.2\}$, and select the offline data selection ratio $\xi\%$ from $\{30\%, 50\%\}$.

K Experimental Details of Wildlife Conservation

We use the green security simulator in [65]. The model is a Markov decision process with state, action, transitions, and a terminal return. We summarize the parts needed to reproduce our experiments.

State and action. At time t, the state is

$$s_t = \big(a_{t-1},\, w_{t-1},\, t\big), \qquad s_0 = (0,\, w_0,\, 0),$$

where $w_t = (w_t^1, \ldots, w_t^N)$ is wildlife across N cells and $a_t = (a_t^1, \ldots, a_t^N)$ is patrol effort. The defender chooses $a_t \in [0, 1]^N$ under the budget $\sum_{i=1}^N a_t^i \le B$.

Adversary behavior. At each step, the poacher places a snare in cell i with probability

$$p_t^i \;=\; \text{logistic}\Big(z^i \;+\; \beta\, a_{t-1}^i \;+\; \eta \sum_{j\in\mathcal{N}(i)} a_{t-1}^j\Big),$$

where z^i is the baseline attractiveness of cell i. The parameters $\beta < 0$ and $\eta > 0$ capture deterrence from prior patrol and displacement from neighboring patrols. The realized attack is

$$k_t^i \sim \text{Bernoulli}\big(p_t^i\big).$$

[3] `https://github.com/t6-thu/H2O`

[4] `https://github.com/Kavka1/VGDF`

[5] `https://github.com/dmksjfl/PAR`

Hyperparameter	Value
Actor network architecture	(256, 256)
Critic network architecture	(256, 256)
Batch size	128
Learning rate	3×10^{-4}
Optimizer	Adam [25]
Discount factor (γ)	0.99
Replay buffer size	10^6
Warmup steps	10^5
Activation	ReLU
Target update rate	5×10^{-3}
SAC temperature coefficient (α)	0.2
Maximum log standard deviation	2
Minimum log standard deviation	-20
Normalization coefficient (ω)	5

Table 2: Hyperparameters for RL training.

Hyperparameter	Offline Flow	Online Flow
Number of hidden layers	6	6
Hidden dimension	256	256
Activation	ReLU	ReLU
Batch size	1024	1024
ODE solver method	Euler	Euler
ODE solver steps	10	10
Training frequency	—	5000
Optimizer	Adam	Adam

Table 3: Hyperparameter setup for the offline and online flows

Wildlife transition. After attacks, wildlife in each cell evolves by natural growth and poaching losses:

$$w_t^i \;=\; \max\Big\{0,\; \big(w_{t-1}^i\big)^{\phi} \;-\; \alpha\, k_{t-1}^i \,\big(1 - a_t^i\big)\Big\},$$

where $\phi > 1$ is the growth rate and $\alpha > 0$ is the loss per uncovered attack. This defines the transition kernel T_z over states given actions:

$$s_{t+1} \sim T_z\big(s_t,\, a_t\big).$$

Return. The episode return is the total wildlife at the horizon,

$$R(s_T) \;=\; \sum_{i=1}^{N} w_T^i,$$

and the expected return of a policy π under environment parameters z is

$$r(\pi, z) \;=\; \mathbb{E}[R(s_T)]\,, \qquad s_{t+1} \sim T_z\big(s_t,\, \pi(s_t)\big),\; s_0 = (0, w_0, 0).$$

We assume access to an offline dataset of 100,000 transitions collected in Murchison Falls National Park using a well-trained SAC policy. For the online environment in Queen Elizabeth National Park, we are allowed a budget of 40,000 interactions.

L Discussion on Computational Cost

The practical cost of data filtering is relatively small. For example, with a batch size of 256 for training the policy network and a Monte Carlo sample size of 30, the entire filtering process takes just 0.03 seconds on an A100 GPU. We will highlight in the paper that this efficiency is due to the simplicity of our flow training objective, which follows a linear path. As a result, solving the corresponding ODE at inference time is very easy—A basic Euler method with just 10 time steps is sufficient.

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: We propose a new method for reinforcement learning with shifted-dynamics data based on flow matching.

   Guidelines:

   - The answer NA 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 NA 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: We discuss the limitations of our method in Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We provide all proofs for the theorems, propositions, and lemmas in the Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide all the experimental details in Appendix J.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We have provided the code in the paper.

   Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all the experimental details in Appendix J.

   Guidelines:

   - The answer NA 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: We report the standard deviation across different random seeds for all tasks.

   Guidelines:

   - The answer NA 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: All the computing infrastructure information has been provided Appendix J.

   Guidelines:

   - The answer NA 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: We conform with the NeurIPS code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have discussed both potential positive societal impacts and negative societal impacts in Appendix B.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA 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 have cited all code, data, and models we used in the paper.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We did not use LLMs.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.