Fisher Decorator: Refining Flow Policy via A Local Transport Map

Xiaoyuan Cheng[1*] **Haoyu Wang**[2*] **Wenxuan Yuan**[3] **Ziyan Wang**[4]
Zonghao Chen[1] **Li Zeng**[5] **Zhuo Sun**[6,7†]
[1]University College London, [2]Shanghai Jiao Tong University, [3]Imperial College London,
[4]King's College London, [5]Shenzhen Research Institute of Big Data,
[6]School of Statistics and Data Science, Shanghai University of Finance and Economics,
[7]Institute of Big Data Research, Shanghai University of Finance and Economics

Abstract

Recent advances in flow-based offline reinforcement learning (RL) have achieved strong performance by parameterizing policies via flow matching. However, they still face critical trade-offs among expressiveness, optimality, and efficiency. In particular, existing flow policies interpret the L_2 regularization as an upper bound of the 2-Wasserstein distance (W_2), which can be problematic in offline settings. This issue stems from a fundamental geometric mismatch: the behavioral policy manifold is inherently anisotropic, whereas the L_2 (or upper bound of W_2) regularization is isotropic and density-insensitive, leading to systematically misaligned optimization directions. To address this, we revisit offline RL from a geometric perspective and show that policy refinement can be formulated as a local transport map—an initial flow policy augmented by a residual displacement. By analyzing the induced density transformation, we derive a local quadratic approximation of the KL-constrained objective governed by the Fisher information matrix, enabling a tractable anisotropic optimization formulation. By leveraging the score function embedded in the flow velocity, we obtain a corresponding quadratic constraint for efficient optimization. Our results reveal that the optimality gap in prior methods arises from their isotropic approximation. In contrast, our framework achieves a controllable approximation error within a provable neighborhood of the optimal solution. Extensive experiments demonstrate state-of-the-art performance across diverse offline RL benchmarks.

1 Introduction

Offline reinforcement learning (RL) algorithms hold tremendous promise for transforming large datasets into powerful decision-making systems [1, 2]. A central challenge in offline RL is formulated as *KL-constrained policy optimization*, which balances reward maximization with the constraint to remain near the behavioral distribution, thereby mitigating distributional shift [3]. As datasets grow larger and span increasingly diverse domains, their behavioral policy distributions become more complex and often highly multimodal [4, 5]. Effectively modeling such distributions requires expressive policy parameterizations. Consequently, offline RL introduces a key challenge for policy learning: balancing policy expressiveness with reliable policy optimization.

Improving policy expressiveness is therefore crucial for solving complex decision-making problems. Recent works have explored modern generative modeling techniques, such as diffusion models

*Equal contribution. Contact Xiaoyuan Cheng with ucesxc4@ucl.ac.uk, and Haoyu Wang with create_arc@sjtu.edu.cn. Project page: https://github.com/ARC0127/Fisher-Decorator.

†Correspondence Author. Correspondence to Zhuo Sun: sunzhuo@mail.shufe.edu.cn.

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

[6, 7, 8] and flow-based models [9], to address this challenge. However, directly incorporating these models into policy extraction remains non-trivial. Existing approaches typically derive a generative policy from a learned value function using techniques such as weighted behavioral cloning [10, 11, 12] or reparameterized policy gradients [13, 14]. In practice, weighted behavioral cloning often suffers from a limited number of effective training samples and insufficient policy expressiveness, which can lead to suboptimal performance, particularly on complex tasks [15]. Meanwhile, reparameterized policy gradient methods require backpropagation through time [16], making them computationally expensive and prone to unstable optimization.

To improve training efficiency and stability, recent works have explored one-step distillation techniques that compress iterative generative processes into single-step generators [17, 18]. Building on this idea, recent state-of-the-art methods such as flow Q-learning [19] and its variants [20, 21] distill iterative flow models into one-step generators, enabling more tractable policy optimization while remaining faithful to the behavioral distribution. However, the distillation objective corresponds to minimizing an upper bound of the 2-Wasserstein distance [22]. This is because existing methods typically resort to an L_2 relaxation as a computationally tractable surrogate for the W_2 objective [19]. However, such a simplification inherently reduces the optimization to an isotropic, density-insensitive penalty. In other words, it treats all parts of the behavior policy's support equally, regardless of how much probability mass the behavior policy assigns to them. As a result, it fails to distinguish between deviations in high-density regions (i.e., frequently visited state–action pairs under the behavior policy) and low-density regions. In contrast, KL constraints explicitly emphasize alignment in high-density regions, and this mismatch can lead to suboptimal updates and distributional shift. Consequently, the resulting one-step policy may suffer from mode collapse and face challenges such as distributional shift, especially when the Q function is inaccurate. Another recent work [23] introduces a residual refinement component with an L_2 penalty to mitigate these effects. However, this regularization still corresponds to the same W_2 upper bound, and thus inherits the same isotropic, density-insensitive behavior, leading to suboptimal policy updates. This exposes a fundamental mismatch: offline RL relies on KL-constrained updates that are inherently anisotropic and density-aware, whereas existing flow-based methods rely on W_2 regularization whose practical instantiations reduce to isotropic regularization (see more comprehensive literature review in Appendix A).

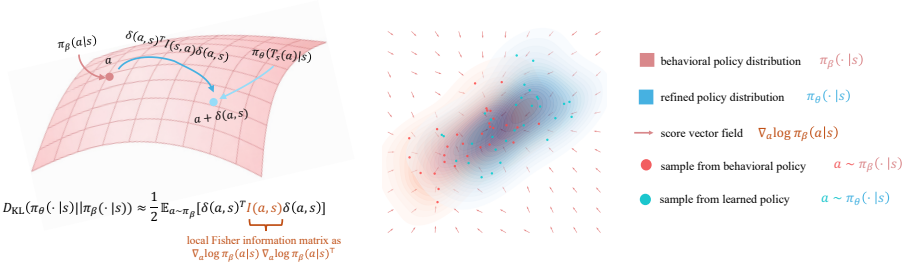

Figure 1: Geometric interpretation of offline policy optimization. (Left) Each action a within the behavioral policy support is refined via a local transport map that induces a constrained displacement. The KL divergence between the behavioral policy $\pi_\beta(\cdot|s)$ and the refined pushforward measure $\pi_\theta(\cdot|s) = (T_s)_\# \pi_\beta(\cdot|s)$ is captured by the expectation of the local quadratic form, which is structurally governed by the anisotropic Fisher information matrix. (Right) Visualization of the policy evolution: samples from the behavioral policy distribution (red) are transported to the refined policy distribution (blue). Their trajectories are anchored by the Fisher information matrix, ensuring the refinement respects the underlying manifold.

To address this fundamental mismatch, we revisit flow-based policy learning in offline RL from a geometric perspective (see Figure 1). We model policy refinement as a local transport map that augments the flow policy with a residual displacement. By analyzing the pushforward of the probability density through this transport map, we show that the KL constraint admits a local quadratic approximation governed by the Fisher information matrix. This reveals that policy updates inherently evolve on a statistical manifold with anisotropic structure [24], whose metric can be efficiently estimated from the flow's velocity field. As a result, we obtain geometry-aware, anisotropic updates without requiring backpropagation through time or relying on biased one-step distillation. Based on this insight, we propose a simple residual policy, termed **Fisher Decorator (FiDec)**. Our contributions are summarized as follows:

(1) *Geometric Characterization of the KL–Wasserstein Gap:* We identify a fundamental geometric mismatch between KL-constrained policy optimization and its W_2 surrogate. By analyzing the pushforward induced by a local transport map, we show that the KL constraint admits a local quadratic form governed by the local Fisher information metric, which is inherently anisotropic and density-aware. In contrast, the W_2 upper bound reduces to an isotropic, density-insensitive L_2 penalty. This mismatch explains the optimality gap of prior methods and reveals why they induce distributional shift and mode collapse in multimodal settings.

(2) *Tractable Anisotropic Policy Optimization:* Building on this characterization, we reformulate KL-constrained policy optimization as a quadratic objective defined on the transport map. We show that the local Fisher information metric is naturally encoded in the score function of the velocity field and can be efficiently estimated. This yields a tractable anisotropic, metric-aware update rule that better preserves distribution support and avoids mode averaging.

(3) *The Fisher Decorator (FiDec) Framework:* We propose FiDec, a simple residual policy that parametrizes refinement as a local transport map. By constraining the residual under the local Fisher information metric, FiDec performs anisotropic, trust-region updates that are aligned with the underlying geometry. This mitigates distributional shift and mode collapse, while remaining efficient without ODE solving or one-step distillation. Empirically, FiDec achieves state-of-the-art (SOTA) performance across a wide range of benchmarks.

2 Preliminary

Notation for Offline RL. We consider the standard Markov decision process (MDP) $\mathcal{M}$ [25], defined by the tuple $(\mathcal{S}, \mathcal{A}, r, \rho, p)$, where $\mathcal{S} \subset \mathbb{R}^n$ is the state space, $\mathcal{A} \subset \mathbb{R}^d$ is the action space, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\rho \in \Delta(\mathcal{S})$ is the initial state distribution, and $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition dynamics. In offline RL, the goal is to learn the parameters θ of a policy $\pi_\theta : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected discounted return $\mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta}[Q_\phi(s,a)] = \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta}[\sum_{t=0}^{N} \gamma^t r(s_t, a_t)]$ using a fixed dataset $\mathcal{D} = \{\tau^{(n)}\}_{n=1}^{M}$ without additional environment interaction. Here, each trajectory τ is given by $(s_0, a_0, \ldots, s_N, a_N)$, and $\gamma \in (0,1)$ denotes the discount factor.

Problem Formulation. Let π_β denote the behavioral policy extracted from the dataset $\mathcal{D}$. Following [1] and [3], offline RL can be formulated as a KL-constrained optimization problem:

$$\max_{\pi_\theta} \quad \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\theta(\cdot|s)}[Q_\phi(s,a)], \qquad \text{s.t.} \quad \mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}\big(\pi_\theta(\cdot|s) \,\|\, \pi_\beta(\cdot|s)\big)\right] \le \epsilon, \tag{1}$$

where $\epsilon > 0$ is a predefined threshold that controls the trade-off between policy improvement and stability. This constraint ensures the learned policy π_θ remains within a principled proximity to π_β, thereby mitigating distributional shift by anchoring the optimization to the data support. In this work, we aim to learn an expressive policy π_θ that satisfies this objective. Specifically, we leverage the flow matching framework for policy parameterization, which facilitates the flexible modeling of complex, multimodal action distributions [26]. To provide the necessary background, we next briefly review the formulation of flow matching and its associated velocity fields.

Flow Matching and Policy. Flow models [27, 28, 29] are rooted in ordinary differential equations (ODEs), which enable simpler training and fast inference. Given a data distribution $p(x) \in \Delta(\mathbb{R}^d)$, flow matching aims to fit the parameter β of a time-dependent velocity field $v_\beta : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ that its corresponding flow [30] $\psi_\beta : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, defined by the unique solution to the ODE

$$\tfrac{d}{dt}\psi_\beta(t,x) = v_\beta(t, \psi_\beta(t,x)), \tag{2}$$

transforms a unit Gaussian at $t = 0$ to the target distribution p at $t = 1$. We consider the linear Gaussian interpolation path [31], the objective is to minimize

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t\sim\text{Unit}[0,1], x^0\sim\mathcal{N}(0,I_d), x^1\sim p(x)}[\|v_\beta(t,x^t) - (x^1 - x^0)\|_2^2], \tag{3}$$

where $\mathcal{N}(0, I_d)$ is the d-dimensional unit Gaussian, $\text{Unif}[0,1]$ denotes the uniform distribution over the unit interval, and $x^t = (1-t)x^0 + tx^1$ is the linear interpolation.

Flow Policies. We use flow matching to learn the behavioral policy, and the objective is minimizing

$$\mathbb{E}_{t\sim\text{Unit}[0,1], s, a=x^1\sim\mathcal{D}, x^0\sim\mathcal{N}(0,I_d)}[\|v_\beta(t,s,x^t) - (x^1 - x^0)\|_2^2], \tag{4}$$

where $v_\beta : [0,1] \times \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}^d$ is a state-time-dependent vector field that generates a state-dependent flow $\psi_\beta(t, s, x)$, which serves as the behavioral policy. For simplicity, we denote $\mu_\beta(s, z)$ as the conditional flow policy at $t = 1$, where z is sampled from the standard Gaussian $\mathcal{N}(0, I_d)$. Here, we ask the central question:

How can we design a policy refinement mechanism that respects the anisotropic structure induced by the KL constraint, while remaining tractable for optimization?

To address this, the following section identifies the theoretical discrepancy between existing flow-based methods and the foundational optimization objective. We then examine the inherent link between the pushforward of transport maps and KL-constrained optimization.

3 Methods

This section first identifies the theoretical gap of previous methods, then introduces a local transport map to bridge KL-constrained optimization and information geometry, and finally derives a practical algorithm (FiDec) for efficient policy refinement.

3.1 Revisit Theoretical Gap in Optimization

In Flow Q-learning (FQL) [19], its variants, as well as DeFlow [23], the refined policy is typically parameterized as a one-step flow $\mu_\theta(s, z)$. Under this formulation, the constraint term is equivalent to the upper bound of 2-Wasserstein distance:

$$\begin{aligned} \mathbb{E}_{s\sim\mathcal{D}, z\sim\mathcal{N}(0,I_d)}\left[\|\mu_\theta(s,z) - \mu_\beta(s,z)\|_2^2\right] &\geq \mathbb{E}_{s\sim\mathcal{D}}\Big[\inf_{\xi\in\Xi(\pi_\theta(\cdot|s),\pi_\beta(\cdot|s))} \mathbb{E}_{x,y\sim\xi}[\|x-y\|_2^2]\Big] \\ &= \mathbb{E}_{s\sim\mathcal{D}}\left[W_2\big(\pi_\theta(\cdot|s), \pi_\beta(\cdot|s)\big)^2\right], \end{aligned} \tag{5}$$

where $\Xi(\pi_\theta(\cdot|s), \pi_\beta(\cdot|s))$ denotes the set of coupling distributions of π_θ and π_β, and W_2 denotes the 2-Wasserstein distance with the Euclidean metric in the action space. However, the W_2 regularization in (5) is not equivalent to the KL constraint specified in (1).

Why the L_2 (or upper bound of W_2) in (5) is problematic? First, the nature of D_{KL} is anisotropic [32], which emphasizes the alignment of probability densities through its logarithmic ratio, acting as a strict support-preserving constraint. Specifically, the KL divergence becomes infinite when $\pi_\theta(\cdot|s)$ assigns positive probability mass outside the support of $\pi_\beta(\cdot|s)$, effectively preventing the learned policy from drifting into regions outside the behavioral support. This property is crucial in offline RL, where inaccurate Q-values often exert an 'extrapolative' pressure on the policy [15, 20].

In contrast, replacing KL with L_2 (or upper bound of W_2) introduces fundamental limitations due to its isotropic nature, particularly in multimodal settings. Specifically, when instantiated via its commonly used upper bound, the W_2 objective reduces to an isotropic L_2 penalty in the ambient space, treating all directions and regions uniformly regardless of the underlying distribution. This leads to several undesirable consequences: (1) *Relaxation of Support Constraints:* Unlike the 'zero-forcing' nature of KL, the isotropic L_2 penalty lacks a mechanism to strongly penalize deviations into low-density or unsupported regions. As a result, the optimized policy may assign non-negligible mass to regions outside the behavioral support, leading to distributional shift and OOD issues. (2) *Mass Averaging via Displacement Interpolation:* From the perspective of optimal transport [33, 22], W_2 optimization follows displacement geodesics. For a multimodal π_β, the isotropic penalty induces a 'centripetal' effect toward the Wasserstein barycenter, encouraging the policy to collapse toward the geometric mean of distinct modes and resulting in unrealistic mode-averaging behavior. (3) *Misaligned Updates under Intrinsic Geometry:* The isotropic L_2 regularization ignores the anisotropic statistical manifold induced by the behavioral distribution. Consequently, policy updates are misaligned with the underlying geometry, leading to suboptimal optimization.

To break this bottleneck, we revisit the policy refinement process and develop an optimization framework that is mathematically faithful to the original KL constraint while remaining computationally tractable within the flow-matching framework.

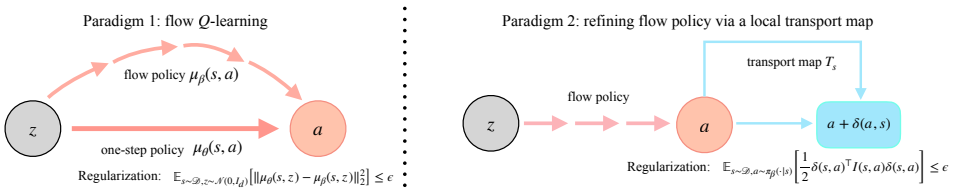

Figure 2: Comparison of flow policy refinement paradigms. (Left) Paradigm 1: flow Q-learning. The refined policy μ_θ is parameterized as an independent one-step mapping from noise z to action a. (Right) Paradigm 2: local transport map (Ours). Instead of re-learning the flow, we keep the original flow as a base and apply a local transport map T_s. The refined action is formulated as $T_s(a) = a + \delta(s,a)$, ensuring the refinement is constrained by local Fisher information matrix.

3.2 Policy Refinement via A Local Transport Map

Rather than learning a one-step policy (see difference in Figure 2), we parameterize the refined policy π_θ via a differentiable local *transport map* $T_s : \mathcal{A} \to \mathcal{A}$ for each $s \in \mathcal{S}$ (see Definition B.3 in Appendix B). This map is defined as a local displacement over the flow-generated action:

$$T_s(a) \coloneqq a + \delta(s,a) \quad \text{where } a = \mu_\beta(s,z). \tag{6}$$

Here, $\delta(s,\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ denotes a learned residual term that induces a local displacement field on the behavioral policy. By formulating the policy update as the *pushforward measure*[3] $\pi_\theta(\cdot|s) = (T_s)_\# \pi_\beta(\cdot|s)$, we can directly link the refinement process to the geometric structure of the behavioral policy distribution. This parameterization ensures that the optimization remains anchored to this support, inherently mitigating the risk listed in Section 3.1.

Applying the pushforward properties [34], the relationship between the refined policy distribution π_θ and the behavioral policy distribution π_β is given by

$$\pi_\theta(T_s(a)|s) = \pi_\beta(a|s) \cdot \left|\det \nabla_{a'} T_s^{-1}(a')\right|_{a'=T_s(a)}. \tag{7}$$

See detailed derivation in Appendix C.1. Here, $\det$ is the determinant of matrix and $\nabla_{a'} T_s^{-1}(a')$ denotes the Jacobian of the inverse map with respect to its input. Based on (6) and the inverse function theorem [35], the Jacobian of T_s is

$$\nabla_{a'} T_s^{-1}(a') = (I_d + \nabla_a \delta(s,a))^{-1}$$

Since δ represents a small localized displacement field, we obtain

$$\left|\det \nabla_{a'} T_s^{-1}(a')\right| = \left|\det(I_d + \nabla_a \delta(s,a))\right|^{-1} = 1 - \nabla \cdot \delta(s,a) + \mathcal{O}(\|\nabla_a \delta(s,a)\|^2), \tag{8}$$

where $\nabla \cdot \delta = \mathrm{tr}(\nabla_a \delta)$ denotes the divergence of the displacement field. The detailed derivation is in Appendix C. We then reveal how the local transport map changes the behavioral policy distribution.

Theorem 3.1 (Connecting the Transport Map to KL Divergence)**.** *Let $\pi_\theta = T_\# \pi_\beta$ be the pushforward of the behavioral policy under the residual map $T(s,a) = a + \delta(s,a)$. Under a small perturbation δ, the KL divergence admits the following second-order approximation:*

$$D_{KL}\big(\pi_\theta(\cdot|s) \,\|\, \pi_\beta(\cdot|s)\big) \approx \frac{1}{2} \mathbb{E}_{a\sim\pi_\beta(\cdot|s)} \left[\delta(s,a)^\top \, I(s,a) \, \delta(s,a)\right], \tag{9}$$

where $I(s,a) = \nabla_a \log \pi_\beta(a|s) \nabla_a \log \pi_\beta(a|s)^\top$ is the local Fisher information matrix (see Definition B.5) obtained at the behavioral policy distribution.

Formal statements and proof are given in Appendix C.3. Notably, this establishes a natural bridge between the transport map and KL divergence without altering the original optimization objective. *The final approximation reveals that the local displacement is penalized by an anisotropic quadratic form induced by the local Fisher information matrix $I(s,a)$.*

[3]Formally, the pushforward measure $\pi_\theta = (T)_\# \pi_\beta$ is defined such that for any measurable set $B \subset \mathcal{A}$, $\pi_\theta(B|s) = \pi_\beta(T_s^{-1}(B|s))$, meaning samples from π_θ are obtained by simply transforming samples from $\pi_\beta(\cdot|s)$ via $a' = T_s(a)$. More information can be found in Appendix B.

Having established this second-order approximation of KL divergence, ensuring consistency with (1) reduces to estimating the local score function, $\nabla_a \log \pi_\beta(a|s)$. Remarkably, in the context of flow matching, this score function is intrinsically linked to the learned velocity field v_β. For a Gaussian interpolation path, the score function can be derived using Lemma C.4 in the Appendix as follows:

$$\nabla_a \log \pi_\beta(a|s) = \lim_{t\to 1} \frac{t v_\beta(t,s,a) - a}{1-t}. \tag{10}$$

However, the expression in (10) exhibits a singularity in this limit (if directly calculate it via the velocity of flow), as both the numerator and denominator vanish simultaneously, leading to numerical instability. To circumvent this, we introduce a small perturbation ε to the time variable, yielding a numerically stable approximation. The following theorem establishes the convergence rate and error bound between this perturbed approximation and the ground-truth score function.

Theorem 3.2 (Pointwise Convergence of the Perturbed Score Function)**.** *Let $\pi_\beta(a|s)$ denote the behavioral policy distribution defined on a compact set of $\mathbb{R}^d$, and assume $\pi_\beta(\cdot|s) \in C^3(\mathbb{R}^d, \mathbb{R})$ over the compact subset of $\mathbb{R}^d$. Consider the linear Gaussian interpolation path, let $\mathbf{s}_t(a|s)$ be the associated score function at time t. For a small perturbation parameter $\varepsilon > 0$, define the perturbed time $t_\varepsilon = 1 - \varepsilon$. As $\varepsilon \to 0^+$, the approximation error between the perturbed score $\mathbf{s}_{1-\varepsilon}(a|s)$ and the ground truth score $\mathbf{s}_1(a|s)$ satisfies:*

$$\|\mathbf{s}_{1-\varepsilon}(a|s) - \mathbf{s}_1(a|s)\| = \varepsilon \left\|\nabla_a \left(a^\top \nabla_a \log \pi_\beta(a|s)\right)\right\| + \mathcal{O}(\varepsilon^2).$$

Consequently, the boundary perturbation scheme achieves a first-order convergence rate of $\mathcal{O}(\varepsilon)$.

Detailed proof is given in Appendix C.5. Based on the error analysis provided in the theorem above, we conclude that the Fisher information matrix $I(s,a)$ can be estimated via local perturbation with a controllable error bound. Thus, in flow policy, the optimization in (1) can be surrogate as

$$\max_T \quad \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\beta(\cdot|s)}[Q_\phi(s, T_s(a))], \quad \text{s.t.} \quad \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\beta(\cdot|s)}\left[\frac{1}{2}\delta(s,a)^\top I(s,a)\delta(s,a)\right] \le \epsilon. \tag{11}$$

Here, the original intractable problem becomes a trust-region optimization [36] as

$$I(s,a) \approx \frac{\left(t_\varepsilon v_\beta(t_\varepsilon,s,a) - a\right)\left(t_\varepsilon v_\beta(t_\varepsilon,s,a) - a\right)^\top}{(1-t_\varepsilon)^2} \tag{12}$$

is computed from the time perturbed score function. The new formulation (11) breaks previous bottlenecks (see the high-level comparison in Table 1): (1) *Preservation of Support.* Unlike global W_2 optimization, the residual formulation enforces a local coupling where $\pi_\theta(a + \delta(s,a)|s) \approx \pi_\beta(a|s)$ due to the near-unit Jacobian $|\det \nabla_a T| \approx 1$ (since δ is small). This ensures that the probability mass is merely "nudged" within the existing manifold rather than being reallocated across space. By maintaining $\pi_\theta(a') \approx 0$ whenever $\pi_\beta(a|s) = 0$, the policy is effectively prevented from shifting into the "voids" between modes, thereby reinstating the support-preserving nature of the KL divergence. (2) *Robust to Mode Averaging.* By restricting the update to a small displacement field δ, the optimization is transformed from a global search for the Wasserstein barycenter into a localized geometric refinement. Since the transport map T is anchored to near identity, it preserves the multimodal structure of π_β and counteracts the "centripetal" force that typically leads to mass averaging. (3) *Aligned Updates under Intrinsic Geometry.* Unlike the isotropic L_2 penalty induced by W_2, our formulation employs the local Fisher information metric to provide direction-aware regularization, better aligning policy updates with the structure of the behavioral distribution.

Table 1: Comparison of metric-aware behavioral regularizers in offline RL. Our method aligns with the KL-constrained formulation, preserving support and preventing mode averaging.

Method	Metric	Preserve Support	Robust to Mode Averaging	Anisotropic
FQL [19]	W_2^2	✗	✗	✗
DeFlow [23]	W_2^2	✗	✗	✗
FiDec (ours)	Fisher	✓	✓	✓

Furthermore, we analyze the value gap induced by the isotropic metric (i.e., L_2) from a trust-region optimization perspective. Under a first-order Taylor expansion, we show that this gap scales with the mismatch between the anisotropic metric and the L_2 metric as $\mathcal{O}\big(\nabla_a Q_\phi(s,a)^\top I(s,a)^\dagger \nabla_a Q_\phi(s,a) - \nabla_a Q_\phi(s,a)^\top \nabla_a Q_\phi(s,a)\big)$ (see derivation in Appendix C.5). In general data distributions where the induced geometry is anisotropic, this gap remains non-vanishing.

3.3 Practical Algorithm

The Fisher Decorator (FiDec) algorithm (shown in Algorithm 1 in Appendix D) is conceptually simple and can be decomposed into four components: (1) training the critic Q_ϕ, (2) learning the behavioral flow policy via the velocity field v_β, and (3) optimizing the transport map T, (4) updating the Lagrangian dual variables. Steps (1) and (2) largely follow the standard flow policy setting. In contrast, we emphasize the learning of the transport map T, and dual updates for the related Lagrangian formulation.

Learning Transport Map. We parameterize the transport map as a composition of a behavioral flow policy and a residual term. Specifically, the flow policy μ_β generates a base action, while a neural network δ_θ learns a correction term to further refine the action. The residual term δ_θ is constrained by a quadratic form induced by the local Fisher information matrix.

Following this formulation, we introduce the Lagrangian of (11) as (see details in Appendix C.4):

$$\min_{\lambda\geq 0}\max_{\delta_\theta}\ \mathbb{E}_{s\sim\mathcal{D},\,a\sim\pi_\beta}\Big[Q_\phi\big(s,T_s(\mu_\beta(s,z))\big)-\lambda\big(\tfrac{1}{2}\delta_\theta(s,a)^\top I(s,a)\delta_\theta(s,a)-\epsilon\big)\Big], \tag{13}$$

where $T_s(\mu_\beta(s,z))=\mu_\beta(s,z)+\delta_\theta(s,\mu_\beta(s,z))$ denotes the learned transport map, $I(s,a)$ is the approximated Fisher information matrix shown in (12), λ is the dual variable, and ϵ controls the strength of the constraint. Learning the transport map is equivalent to solving the inner problem.

Dual Updates. We update the dual variable λ via projected gradient ascent to enforce the constraint

$$\lambda\leftarrow \mathrm{ReLU}\big(\lambda+\eta\,\mathbb{E}_{s\sim\mathcal{D},\,a\sim\pi_\beta}\big[\tfrac{1}{2}\delta_\theta(s,a)^\top I(s,a)\delta_\theta(s,a)-\epsilon\big]\big)\,, \tag{14}$$

where η is the learning rate. This update increases λ when the constraint is violated and decreases it otherwise, thereby adaptively balancing policy improvement and constraint satisfaction.

We optimize the objective (13) via an alternating primal-dual procedure. In each iteration, we update the transport map parameter δ_θ by maximizing the Lagrangian with fixed λ, and update λ via projected gradient ascent. In practice, expectations are approximated using minibatch samples, and gradients through the behavioral policy μ_β are stopped when optimizing δ_θ for stability.

4 Numerical Experiments

In this section, we benchmark our proposed algorithm, FiDec, against state-of-the-art offline and offline-to-online RL methods across a diverse suite of challenging tasks. We answer the questions if our method can break the existing bottleneck shown in Table 1. Furthermore, we conduct extensive ablation studies and sensitivity analyses to elucidate the impact of key hyperparameter choices.

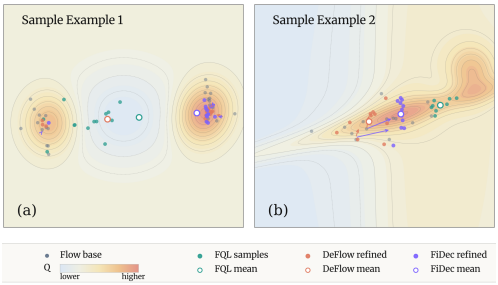

Figure 3: Isotropic vs. anisotropic policy refinement. (a) In multimodal settings, isotropic W_2-based methods exhibit mode averaging or interpolation: FQL collapses toward intermediate regions, while DeFlow spreads mass averagely across modes with its mean located in low-value areas. In contrast, FiDec preserves multimodality while shifting mass toward more favorable regions. (b) Along the distribution support, isotropic regularization either leads to OOD drift (FQL) or fails to reach high-value regions (DeFlow). FiDec instead follows the support and refines the policy toward higher-value regions.

Benchmarks. Followed previous work [37, 38, 19], we assess our method on multiple challenging benchmarks, including OGBench [5] and D4RL [39].

Baselines for Offline Setting. To ensure a comprehensive evaluation, we compare our method against eleven recent offline RL algorithms spanning a broad range of architectural designs and policy extraction paradigms. These include: (1) Gaussian policies: BC, IQL [40], and ReBRAC [37]; (2) diffusion policies: IDQL [6], SRPO [7], and CAC [14]; and (3) flow policies: FAWAC [41], IFQL [9], FQL [19], and DeFlow [23]. **Baselines for Offline-to-Online Setting.** For offline-to-online RL experiments, we consider prior offline RL methods (IQL, ReBRAC, IFQL, FQL and DeFlow) that support

fine-tuning and achieve strong performance. Meanwhile, we consider two other strong algorithms designed for online fine-tuning: Cal-QL [42] and RLPD [43].

Result Analysis. Our result analysis is structured around the core contributions of this work.

1. Can FiDec achieve strong performance in the offline setting across diverse benchmark tasks? The answer is **YES**. As shown in Table 2, FiDec consistently outperforms prior methods, including those based on Gaussian, diffusion, and flow policies. In particular, FiDec demonstrates clear advantages over recent strong baselines such as FQL and DeFlow, especially on challenging OGBench tasks (e.g., Humanoid, Puzzle, Antsoccer), which are inherently multimodal. This implicitly suggests that FiDec is able to effectively capture and preserve complex multimodal behavior distributions, while avoiding mode collapse and maintaining generalization across diverse task settings.

Table 2: Offline RL results. Results are averaged over 8 seeds. Our method (FiDec) achieves strong performance across 73 diverse tasks in 4 benchmarks: the top three methods are highlighted in **(1st)**, (2nd) and (3rd). We follow the default evaluation setting from [19] and [23]. More detailed results for each specific task are reported in Table 8.

	Gaussian Policies			Diffusion Policies			Flow Policies					
Task Category	BC	IQL	ReBRAC	IDQL	SRPO	CAC	FAWAC	FBRAC	IFQL	FQL	DeFlow	**FiDec**
OGBench antmaze-large-singletask (5)	11±1	53±3	81±5	21±5	11±4	33±4	6±1	60±6	28±5	79±3	81±11	**87±1**
OGBench antmaze-giant-singletask (5)	0±0	4±1	**26±8**	0±0	0±0	0±0	0±0	4±4	3±2	9±6	5±7	13±4
OGBench humanoidmaze-medium-singletask (5)	2±1	33±2	22±8	1±0	1±1	53±8	19±1	38±5	60±14	58±5	57±29	**72±2**
OGBench humanoidmaze-large-singletask (5)	1±0	2±1	2±1	1±0	0±0	0±0	0±0	2±0	**11±2**	4±2	5±5	8±3
OGBench antsoccer-arena-singletask (5)	1±0	8±2	0±0	12±4	1±0	2±4	12±0	16±1	33±6	60±2	62±20	**64±2**
OGBench cube-single-singletask (5)	5±1	83±3	91±2	95±2	80±5	85±9	81±4	79±7	79±2	**96±1**	90±6	94±1
OGBench cube-double-singletask (5)	2±1	7±1	12±1	15±6	2±1	6±2	5±2	15±3	14±3	29±1	38±22	**43±4**
OGBench scene-singletask (5)	5±1	28±1	41±3	46±3	20±1	40±7	30±3	45±5	30±3	**56±2**	51±40	**56±1**
OGBench puzzle-3x3-singletask (5)	2±0	9±1	21±1	10±2	18±1	19±0	6±2	14±4	19±1	30±1	24±38	**43±1**
OGBench puzzle-4x4-singletask (5)	0±0	7±1	14±1	**29±3**	10±3	15±3	1±0	13±1	25±5	17±2	4±2	15±1
D4RL antmaze (6)	17	57	78	79	74	30±3	44±3	64±7	65±7	84±3	81±7	**87±1**
D4RL adroit (12)	48	53	**59**	52±1	51±1	43±2	48±1	50±2	52±1	52±1	34±41	46±2
Visual manipulation (5)	–	42±4	60±2	–	–	–	–	22±2	50±5	**65±2**	59±38	**64±1**

2. Can our new formulation in (13) go beyond isotropic regularization to enable more effective policy refinement? **YES**. As shown in Figure 3(a), both FQL and DeFlow exhibit limitations under isotropic regularization. FQL suffers from mode averaging, while DeFlow produces a distribution that interpolates between modes, with its mean located in low-value regions. This behavior arises because the W_2 distance induces an isotropic L_2 penalty, which treats all directions uniformly and ignores the underlying score field of the distribution, resulting in ambiguous update directions.

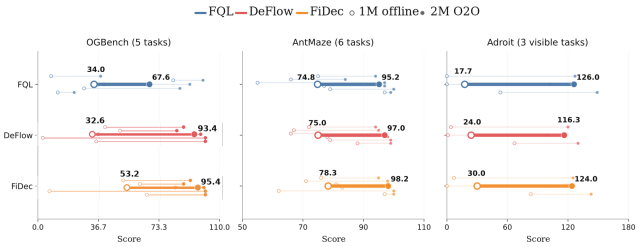

Figure 4: Offline-to-online fine-tuning performance. We evaluate 14 tasks. All methods continue training from the offline objective, with fine-tuning starting after 1M gradient steps (offline prior: ○, fine-tuning: ●). Light curves denote individual tasks, and dark curves show the average.

In contrast, our formulation is anisotropic and metric-aware: the update directions are guided by the local score field through the local Fisher information matrix. This preserves multimodality while biasing updates toward favorable regions: weaker-score areas (e.g., the left mode) allow larger displacements, whereas stronger-score regions are more constrained. A similar phenomenon is observed in Figure 3(b). The behavioral policy is supported on a thin manifold. Under isotropic regularization, DeFlow produces a more concentrated distribution but fails to reach higher-value regions, while FQL deviates from the data support and exhibits out-of-distribution drift. In contrast, FiDec follows the geometric structure and moves toward higher-value regions, demonstrating both optimality and effectiveness in policy refinement. This generally better performance in the offline setting (see Table 2) also verifies this point; see Figure 7 for more example details. We provide more examples in Appendix E.3.

3. Can our new formulation in (13) be seamlessly adopted in the offline-to-online setting? The answer is **YES**. FiDec can be directly fine-tuned without any modification and consistently outperforms prior flow policies (Table 9 in Appendix E.3, Figure 4). We attribute this advantage to two key factors. First, FiDec provides a stronger offline prior. By parameterizing the policy as a local transport map, it retains the expressive structure of the underlying flow policy while enabling flexible local refinement. As a result, FiDec achieves a higher-quality initialization compared to prior methods, where FQL often underperforms due to its limited expressiveness under one-step distillation. Second, FiDec benefits from anisotropic metric-aware updates during fine-tuning. Guided by the local Fisher information matrix, the policy can efficiently move probability mass with lower cost in low-score regions (due to weaker score magnitude) toward more favorable regions. This leads to faster improvement and a higher concentration of samples in high-Q regions, resulting in more effective online refinement.

Table 3: Isotropic vs. anisotropic metrics under identical hyperparameters and policy parameterization under offline setting.

Env	Anisotropic (Fisher)	Isotropic (L_2)
antsoccer-arena-navigate-singletask-task4	**0.49±0.04**	0.43±0.08
cube-double-play-singletask-task2	**0.58±0.02**	0.50±0.03
humanoidmaze-medium-navigate-singletask-task1	**0.76±0.07**	0.35±0.07
puzzle-4x4-play-singletask-task4	**0.05±0.01**	0.04±0.01
scene-play-singletask-task2	**0.70±0.07**	0.51±0.06
Avg. Performance	**0.51±0.25**	0.36±0.17

4. Is the anisotropic metric always better than the isotropic one under the same policy parameterization? **YES**. Under the same residual displacement parameterization and training setup, we observe that the anisotropic metric induced by the Fisher information matrix consistently outperforms its isotropic counterpart (L_2), as shown in Table 3. This suggests that properly accounting for the local geometric properties of the behavioral policy distribution leads to more effective policy updates and improved performance, which is consistent with our theoretical analysis of the optimization gap in Appendix C.5 from a trust-region perspective.

5. How efficient is FiDec compared to existing approaches? FiDec is computationally efficient at both the algorithmic and implementation levels. It avoids expensive procedures such as ODE solving and backpropagation through time, and performs policy refinement via a simple one-step residual update. The additional cost from the Fisher information matrix is minimal, as it is computed from the flow velocity using an outer product and applied through a lightweight quadratic form in JAX. From a framework perspective, FiDec operates in a one-step optimization scheme without distillation or iterative generation, and the primal-dual updates introduce negligible overhead. As shown in Table 4, FiDec achieves competitive training speed, indicating that the added anisotropic geometric structure does not compromise efficiency.

Table 4: Training efficiency comparison of flow policies.

Method	FQL	DeFlow	FiDec
Time (ms/step)	2.13 ± 0.14	2.61 ± 0.18	2.72 ± 0.20

6. Can the hyperparameter "perturbed time t_ε" be determined from first principles, rather than through heuristic tuning? **YES**. We provide a principled characterization of the optimal perturbation by analyzing the trade-off between approximation bias and numerical error (see Appendix C.3). This yields an optimal scaling $\varepsilon^* \sim \mathcal{O}(\delta_{\text{FP32}}^{1/4})$, which depends on both machine precision δ and the spatial deformation $\left\|\nabla_a\left(a^\top \nabla_a \log \pi_\beta(a|s)\right)\right\|^2$ of the behavioral distribution. In practice, this analysis suggests that $\varepsilon \sim \mathcal{O}(10^{-1.5})$ under standard FP32 precision, which aligns with empirical performance (Figure 5), where the best results are achieved within the range of $t_\varepsilon \in [0.70, 0.80]$. Detailed per-task results are reported in Table 10, Appendix E.

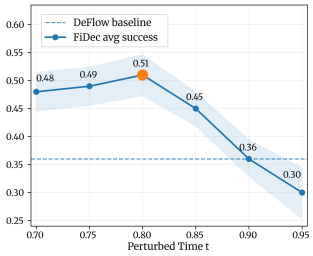

Figure 5: Ablations on perturbed time t_ε.

5 Conclusion

In this paper, we identify a fundamental geometric mismatch in flow-based offline RL: KL-constrained optimization induces an anisotropic geometry, while commonly used W_2 surrogates reduce to isotropic regularization, leading to suboptimal updates. To address this, we formulate policy refinement as a local transport map and derive a tractable quadratic approximation of the KL objective. This yields an efficient anisotropic optimization framework, instantiated as FiDec, that enables stable and expressive policy improvement without expensive iterative procedures. Empirically, FiDec achieves

state-of-the-art performance across diverse benchmarks. Our findings underscore the importance of geometry-aware policy optimization in offline RL.

Despite its effectiveness, our method has several limitations. First, FiDec relies on an approximate estimation of the Fisher information matrix derived from the flow velocity. While efficient, this approximation may be inaccurate when the learned flow model is imperfect or when the underlying distribution exhibits highly complex structure. Second, although we provide a principled analysis of the perturbation scale, its optimal value still depends on unknown properties of the data distribution and may require mild tuning in practice. An interesting direction for future work is to further exploit the analytical formula of the optimal policy refinement. As shown in Equation (59) in Appendix C.4, the optimal displacement admits a closed-form solution resembling a pointwise natural gradient update in the action space. This suggests that, given an accurate estimate of the local Fisher information matrix, policy refinement could potentially be performed without iterative optimization.

Acknowledgements

Zhuo Sun is supported by Fundamental Research Funds for the Central Universities 2025110590 of Shanghai University of Finance and Economics. This work is supported by the Shanghai Engineering Research Center of Finance Intelligence (Grant No,19DZ2254600).

References

[1] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[2] Yuwei Fu, Di Wu, and Benoit Boulet. A closer look at offline rl agents. *Advances in Neural Information Processing Systems*, 35:8591–8604, 2022.

[3] Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE transactions on neural networks and learning systems*, 35(8):10237–10257, 2023.

[4] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[5] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv:2410.20092*, 2024.

[6] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

[7] Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. *arXiv preprint arXiv:2310.07297*, 2023.

[8] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[9] Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. *arXiv preprint arXiv:2503.04975*, 2025.

[10] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, pages 22825–22855. PMLR, 2023.

[11] Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:67195–67212, 2023.

[12] Shutong Ding, Ke Hu, Zhenhao Zhang, Kan Ren, Weinan Zhang, Jingyi Yu, Jingya Wang, and Ye Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization. *Advances in Neural Information Processing Systems*, 37:53945–53968, 2024.

[13] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[14] Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. *arXiv preprint arXiv:2309.16984*, 2023.

[15] Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is value learning really the main bottleneck in offline rl? *Advances in Neural Information Processing Systems*, 37:79029–79056, 2024.

[16] Nicolas Espinosa-Dice, Kiante Brantley, and Wen Sun. Expressive value learning for scalable offline reinforcement learning. *arXiv preprint arXiv:2510.08218*, 2025.

[17] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[18] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[19] Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning, 2025.

[20] Songyuan Zhang, Oswin So, HM Ahmad, Eric Yang Yu, Matthew Cleaveland, Mitchell Black, and Chuchu Fan. Reform: Reflected flows for on-support offline rl via noise manipulation. *arXiv preprint arXiv:2602.05051*, 2026.

[21] Thanh Nguyen and Chang D. Yoo. One-step flow q-learning: Addressing the diffusion policy bottleneck in offline reinforcement learning, 2026.

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

[23] Zhancun Mu. Deflow: Decoupling manifold modeling and value maximization for offline policy extraction. *arXiv preprint arXiv:2601.10471*, 2026.

[24] Zikai Shen, Zonghao Chen, Dimitri Meunier, Ingo Steinwart, Arthur Gretton, and Zhu Li. Nonparametric instrumental variable regression with observed covariates. *arXiv preprint arXiv:2511.19404*, 2025.

[25] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[26] Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2(3), 2023.

[27] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[28] 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.

[29] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[30] Serge Lang. *Differential and Riemannian manifolds*. Springer Science & Business Media, 2012.

[31] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[32] Shun-ichi Amari. *Information geometry and its applications*. Springer, 2016.

[33] Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. *Statistical optimal transport*. Springer, 2025.

[34] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[35] Terence Tao. *Analysis*, volume 1. Springer, 2006.

[36] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

[37] Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

[38] Rafael Rafailov, Kyle Hatch, Anikait Singh, Laura Smith, Aviral Kumar, Ilya Kostrikov, Philippe Hansen-Estruch, Victor Kolev, Philip Ball, Jiajun Wu, et al. D5rl: Diverse datasets for data-driven deep reinforcement learning. *arXiv preprint arXiv:2408.08441*, 2024.

[39] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[40] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[41] Nair Ashvin, Dalal Murtaza, Gupta Abhishek, and L Sergey. Accelerating online reinforcement learning with offline datasets. *CoRR, vol. abs/2006.09359*, 2020.

[42] 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.

[43] 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.

[44] Harshit Sikchi, Qinqing Zheng, Amy Zhang, and Scott Niekum. Dual rl: Unification and new methods for reinforcement and imitation learning. *arXiv preprint arXiv:2302.08560*, 2023.

[45] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[46] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. Corl: Research-oriented deep offline reinforcement learning library. *Advances in Neural Information Processing Systems*, 36:30997–31020, 2023.

[47] 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.

[48] Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. Offline rl with no ood actions: In-sample learning via implicit value regularization. *arXiv preprint arXiv:2303.15810*, 2023.

[49] Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent rl without entropy. *arXiv preprint arXiv:2301.02328*, 2023.

[50] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in neural information processing systems*, 33:14129–14142, 2020.

[51] Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration by random network distillation. In *International conference on machine learning*, pages 26228–26244. PMLR, 2023.

[52] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

[53] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

[54] 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.

[55] 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.

[56] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

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

[58] 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.

[59] 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.

[60] Siddarth Venkatraman, Shivesh Khaitan, Ravi Tej Akella, John Dolan, Jeff Schneider, and Glen Berseth. Reasoning with latent diffusion in offline reinforcement learning. *arXiv preprint arXiv:2309.06599*, 2023.

[61] Xiaoyuan Cheng, Xiaohang Tang, and Yiming Yang. Safe and stable control via lyapunov-guided diffusion models. *arXiv preprint arXiv:2509.25375*, 2025.

[62] Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. *arXiv preprint arXiv:2502.00361*, 2025.

[63] Xiaoyi Dong, Jian Cheng, and Xi Sheryl Zhang. Maximum entropy reinforcement learning with diffusion policy. *arXiv preprint arXiv:2502.11612*, 2025.

[64] Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Rémi Munos, Alessandro Lazaric, and Ahmed Touati. Temporal difference flows. *arXiv preprint arXiv:2503.09817*, 2025.

[65] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

[66] Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[68] Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

[69] Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement-residual rl for precise assembly. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 01–08. IEEE, 2025.

[70] Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Yuqi Xie, Fengyuan Hu, Jimmy Wu, Zhengyi Luo, Linxi Fan, et al. Self-improving vision-language-action models with data generation via residual rl. *arXiv preprint arXiv:2511.00091*, 2025.

[71] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. *arXiv preprint arXiv:2412.13630*, 2024.

[72] Alfréd Rényi. *Probability theory*. Courier Corporation, 2007.

[73] Alice Guionnet and Bogusław Zegarlinksi. Lectures on logarithmic sobolev inequalities. In *Séminaire de probabilités XXXVI*, pages 1–134. Springer, 2004.

[74] Herbert Federer. *Geometric measure theory*. Springer, 2014.

[75] John Neuberger. *Sobolev gradients and differential equations*. Springer Science & Business Media, 2009.

[76] Haim Brezis and Haim Brézis. *Functional analysis, Sobolev spaces and partial differential equations*, volume 2. Springer, 2011.

[77] Alan Macdonald. *Vector and geometric calculus*, volume 12. CreateSpace Independent Publishing Platform Scotts Valley, CA, USA, 2012.

[78] Zonghao Chen, Atsushi Nitanda, Arthur Gretton, and Taiji Suzuki. Towards a unified analysis of neural networks in nonparametric instrumental variable regression: Optimization and generalization. *arXiv preprint arXiv:2511.14710*, 2025.

Notation

Notation	Meaning
a	action
s	state
$\mathbf{s}$	score function
t	time step
r	reward function
v	velocity field (or flow vector)
$\mathcal{A}$	action space
$\mathcal{S}$	state space
$\mathcal{B}$	Borel σ-algebra
$\mathcal{D}$	offline dataset
I	local Fisher information matrix
D_{KL}	Kullback–Leibler divergence
L_2	Lebesgue space equipped with $2-$norm
W_2	2-Wasserstein distance
$\mathcal{L}$	Lagrangian
λ	Lagrange multiplier (dual variable)
$\mathcal{N}$	Gaussian distribution
Q	action-value function
T	local transport map
Ξ	set of couplings (transport plans)
ξ	a coupling (joint distribution)
ψ	flow induced by a velocity field
δ	local displacement
μ_β	policy generated from the original flow
μ_θ	policy generated from the refined flow
π_β	behavior policy
π_θ	learned policy
$\#$	pushforward operator
$\nabla\cdot$	divergence operator
$\det(\cdot)$	determinant of a matrix
$\mathrm{tr}(\cdot)$	trace of a matrix
$\square^\dagger$	pseudo-inverse of a matrix

Appendix Overview

This appendix provides a comprehensive and self-contained presentation of the theoretical foundations, proofs, implementation details, and additional experimental results of our method.

Appendix A provides provide a more comprehensive literature review covering offline RL, generative policies and residual policies.

In Appendix B, we introduce the key technical definitions, including probability measures, Wasserstein distance, transport maps, and the local Fisher information matrix, which form the basis of our geometric formulation. Appendix C presents detailed derivations and proofs, including the change-of-variable formulation under transport maps, the second-order expansion of the KL divergence, and the connection to the local Fisher information matrix. We further analyze the trust-region optimization perspective and characterize the optimality gap induced by isotropic approximations.

Appendix D provides the full implementation details and algorithmic description of our method. Finally, Appendix E includes additional experimental results, benchmark descriptions, and hyperparameter settings.

Overall, the appendix is intended to complement the main paper by providing rigorous theoretical justification and complete experimental transparency.

A More Comprehensive Literature Review

We provide a more comprehensive literature review covering offline RL, generative policies and residual policies.

Offline RL. Offline RL aims to learn a policy solely from previously collected data. Over the years, many offline RL methods and techniques have been proposed, most of which can be viewed through a common principle: maximizing expected return while controlling the discrepancy between the state-action distribution induced by the learned policy and that of the dataset [1, 44]. Prior work has instantiated this principle in various forms, including behavioral regularization [41, 45, 46], conservative value estimation [47], in-sample maximization [40, 48, 49], out-of-distribution detection [50, 51], dual RL [52], and generative modeling [53, 52]. After offline training, the learned policy can be further improved through additional online interaction, a setting commonly known as offline-to-online RL, for which several methods have been developed [54, 55, 42, 43, 19, 20]. Our method, FiDec, is applicable to both offline and offline-to-online RL, and introduces anisotropic constraints to guide policy improvement.

RL with Generative Models. Inspired by the recent success of iterative generative modeling methods, including denoising diffusion [56, 57, 58] and flow matching [27, 59], a growing body of work has explored their integration into reinforcement learning. Prior studies have used iterative generative models for planning, hierarchical learning, and policy parameterization [60, 61, 62, 63], as well as for world modeling and data augmentation [64, 65, 66]. Our approach falls under flow-based policy parameterization: we represent the policy with an expressive flow network capable of capturing arbitrarily complex behavior-policy distributions.

Residual Policy. Residual learning has a long and influential history in machine learning, ranging from ResNets [67] in computer vision to residual policy learning in robotics [68]. More recent studies [69, 70, 71] extend this idea by training a refinement module that adjusts actions sampled from a frozen policy through online interaction. Mu [23] further investigate the use of residual policies within flow policies for offline reinforcement learning. In contrast, our work examines this paradigm from a different perspective: we study how the intrinsic properties of residual policies and flow matching can be leveraged to improve policy learning through an anisotropic geometric view.

Key differences between FiDec and TRPO, natural policy gradient, and mirror descent. FiDec differs from these methods primarily in where the trust-region geometry is defined and how the policy update is parameterized.

- *TRPO, natural policy gradient, and Fisher-based trust regions.* These methods constrain a parameter update $\Delta\theta$ using a parameter-space Fisher matrix:

  $$D_{\mathrm{KL}}(\pi_{\theta+\Delta\theta}\|\pi_\theta) \approx \frac{1}{2}\Delta\theta^\top F_\theta \Delta\theta.$$

  FiDec instead defines the trust region in action space. For the displacement $\delta(s,a) = T_s(a) - a$ that transports π_β to π_θ, the induced KL divergence is locally approximated as

  $$D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s)\|\pi_\beta(\cdot \mid s)\big) \approx \frac{1}{2}\mathbb{E}_{a\sim\pi_\beta(\cdot|s)}\left[\delta(s,a)^\top I(s,a)\delta(s,a)\right].$$

  Classical Fisher-based methods therefore define the trust region in policy-parameter space, whereas FiDec defines it in action space around a fixed behavior policy. This construction is designed for implicit flow policies and can be estimated from offline data without requiring an explicit policy likelihood.

- *Mirror descent.* Mirror descent directly performs a KL-proximal update over policies or distributions. FiDec instead restricts the updated policy to the pushforward of π_β under a local transport map and optimizes its displacement field. The key novelty is a tractable, action-space realization of KL-constrained refinement rather than direct optimization of the policy parameterization.

B Technical Definitions

Definition B.1 (Probability Measure [72])**.** Let $(\mathcal{X}, \mathcal{B})$ be a measurable space. A measure $\mu : \mathcal{B} \to [0, 1]$ is called a *probability measure* if:

- (null empty set) $\mu(\emptyset) = 0$,
- (countable additivity) μ is countably additive, i.e., for any sequence of disjoint sets $\{B_i\}_{i=1}^{\infty} \subseteq \mathcal{B}$,
$$\mu\left(\bigcup_{i=1}^{\infty} B_i\right) = \sum_{i=1}^{\infty} \mu(B_i),$$
- (normalization) $\mu(\mathcal{X}) = 1$.

Definition B.2 (2-Wasserstein Distance [73])**.** Let μ, ν be probability measures on $\mathcal{X}$ with finite second moments. The 2-Wasserstein distance between μ and ν is defined as
$$W_2(\mu, \nu) = \left(\inf_{\gamma \in \Xi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} \|x - y\|^2 \, d\gamma(x, y)\right)^{1/2},$$
where $\Xi(\mu, \nu)$ denotes the set of all couplings of μ and ν.

Definition B.3 (Transport Map [74])**.** Let $(\mathcal{X}, \mathcal{B})$ be a measurable space, and let μ and ν be probability measures on $\mathcal{X}$. A measurable map $T : \mathcal{X} \to \mathcal{X}$ is called a *transport map* from μ to ν if it pushes μ forward to ν, i.e.,
$$\nu = T_{\#}\mu,$$
meaning that for every measurable set $B \subseteq \mathcal{X}$,
$$\nu(B) = \mu\big(T^{-1}(B)\big).$$
Equivalently, for any measurable function f,
$$\int f(y) \, d\nu(y) = \int f(T(x)) \, d\mu(x).$$

Remark B.4. In our setting, for each state s, the mapping $T_s : \mathcal{A} \to \mathcal{A}$ defines a transport map between action distributions. Specifically, let $\pi_\beta(\cdot|s)$ denote the well-defined behavioral policy distribution and let $a' = T_s(a)$ with $a \sim \pi_\beta(\cdot|s)$. Then the refined policy $\pi_\theta(\cdot|s)$ is given by the pushforward
$$\pi_\theta(\cdot|s) = (T_s)_{\#}\, \pi_\beta(\cdot|s).$$
Thus, for each fixed state s, the map T_s transports the behavioral policy distribution $\pi_\beta(\cdot|s)$ to the refined distribution $\pi_\theta(\cdot|s)$.

Definition B.5 (Local Fisher Information Matrix [32])**.** Let $p(x)$ be a smooth density on $\mathbb{R}^d$ with score function $\mathbf{s}(x) \coloneqq \nabla_x \log p(x)$. The local Fisher information matrix is defined as
$$I(x) \coloneqq \mathbf{s}(x)\mathbf{s}(x)^\top.$$

Remark B.6 (Relation to information geometry). The local Fisher information matrix $I(x)$ can be viewed as the local contribution (integrand) of the classical Fisher information matrix.
$$I(p) \coloneqq \mathbb{E}_{x\sim p(x)}\mathbb{E}[\mathbf{s}(x)\mathbf{s}(x)^\top].$$
In information geometry, $I(p)$ defines a local Riemannian metric on a statistical manifold of probability distributions. However, the local object $I(x)$ itself does not directly define a Riemannian metric on the sample space, since it is local Fisher information matrix. Nevertheless, it encodes the local direction of maximal sensitivity of the log-density via the score function, and can be interpreted as the infinitesimal contribution to the local Fisher information metric under expectation.

Motivated by the above, our work focuses on the cost induced by local displacements in the action space. In particular, given a displacement field $\delta(s, a)$, the associated infinitesimal cost is governed by the local Fisher information matrix through
$$(\delta(s, a)^\top \mathbf{s}(a|s))^2 = \delta(s, a)^\top I(s, a)\delta(s, a),$$
where $I(s, a) = \mathbf{s}(a|s)\mathbf{s}(a|s)^\top$ is the local Fisher information matrix. This reveals that the underlying anisotropic geometric structure induced by the distribution is inherently anisotropic: *only directions aligned with the score function* $\mathbf{s}(a|s)$ *contribute to the cost, while directions orthogonal to* $\mathbf{s}(a|s)$ *incur no first-order variation. Consequently, the geometry is highly directional and depends on the local behavior of the log-density.*

C Technical Proofs

C.1 Refined Policy Distribution under Transport Map

Lemma C.1 (Change of Variables under Transport Maps)**.** *For all state $s \in \mathcal{S}$, let $T_s : \mathcal{A} \to \mathcal{A}$ be an invertible and differentiable transport map for a fixed state s, and let $\pi_\beta(\cdot|s)$ be a probability distribution on $\mathcal{A}$. Define the pushforward distribution*

$$\pi_\theta(\cdot|s) = (T_s)_{\#}\, \pi_\beta(\cdot|s).$$

Then the density of π_θ is given by the change-of-variables formula:

$$\pi_\theta(a'|s) = \pi_\beta\big(T_s^{-1}(a')|s\big) \cdot \left|\det \nabla_{a'} T_s^{-1}(a')\right|,$$

where $a' = T_s(a)$ and $\nabla_{a'} T_s^{-1}(a')$ denotes the Jacobian of the inverse map with respect to its input.

Proof. Fix an arbitrary state $s \in \mathcal{S}$ and let T_s be an invertible and differentiable map. By definition of transport map in Definition B.3, for any measurable set $B \subseteq \mathcal{A}$,

$$\pi_\theta(B|s) = \pi_\beta\big(T_s^{-1}(B)|s\big).$$

Assume that $\pi_\beta(\cdot|s)$ admits a density (still denoted by π_β). Then

$$\pi_\theta(B|s) = \int_{T_s^{-1}(B)} \pi_\beta(a|s)\, da.$$

Applying the change-of-variables formula with $a' = T(s,a)$ yields

$$\int_{T_s^{-1}(B)} \pi_\beta(a|s)\, da = \int_B \pi_\beta\big(T_s^{-1}(a')|s\big)\, \left|\det \nabla_{a'} T_s^{-1}(a')\right|\, da'.$$

Since this holds for all measurable sets B, we conclude that the density of $\pi_\theta(\cdot|s)$ is

$$\pi_\theta(a'|s) = \pi_\beta\big(T_s^{-1}(a')|s\big) \cdot \left|\det \nabla_{a'} T_s^{-1}(a')\right|.$$

□

Lemma C.2. *Let*

$$T_s(a) = a + \delta(s,a), \quad \delta(s,\cdot) \in C^1(\mathbb{R}^d, \mathbb{R}^d) \tag{15}$$

Assume that T is invertible and $\|\nabla_a \delta(s,a)\|$ is small. Then, for all $a' = T_s(a)$, we have

$$\left|\det \nabla_{a'} T_s^{-1}(a')\right| = 1 - \nabla_a \cdot \delta(s,a) + \mathcal{O}(\|\nabla_a \delta(s,a)\|^2), \tag{16}$$

where $a = T_s^{-1}(a')$.

Proof. The Jacobian of T is given by

$$\nabla_a T_s(a) = I + \nabla_a \delta(s,a). \tag{17}$$

Using the determinant expansion

$$\det(I + A) = 1 + \operatorname{tr}(A) + \underbrace{\frac{1}{2}\left[(\operatorname{tr} A)^2 - \operatorname{tr}(A^2)\right] + \mathcal{O}(\|A\|^3)}_{\text{high-order terms}}, \tag{18}$$

we obtain

$$\det \nabla_a T_s(a) = 1 + \nabla \cdot \delta(s,a) + \mathcal{O}(\|\nabla_a \delta(s,a)\|^2). \tag{19}$$

By the inverse function theorem,

$$\det \nabla_{a'} T_s^{-1}(a') = \frac{1}{\det \nabla_a T_s(a)}. \tag{20}$$

Applying the Taylor expansion $(1+\epsilon)^{-1} = 1 - \epsilon + \mathcal{O}(\epsilon^2)$ yields

$$|\det \nabla_{a'} T_s^{-1}(a')| = 1 - \nabla \cdot \delta(s,a) + \mathcal{O}(\|\nabla_a \delta(s,a)\|^2), \tag{21}$$

where $a = T_s^{-1}(a')$. □

Theorem C.3 (Second-order expansion of KL divergence under small transport)**.** *Let $\pi_\beta(\cdot|s)$ be a probability density on $\mathbb{R}^d$ with $\log \pi_\beta \in C^2(\mathbb{R}^d, \mathbb{R})$, and define a transport map*

$$T_s(a) = a + \delta(s, a), \tag{22}$$

where $\delta(s, \cdot) \in C^1(\mathbb{R}^d, \mathbb{R}^d)$ and $\|\nabla_a \delta(s, a)\|$ is small so that T is invertible.

Let $\pi_\theta(\cdot|s)$ be the pushforward of $\pi_\beta(\cdot|s)$ under T, i.e.,

$$\pi_\theta(a'|s) = \pi_\beta(a|s) \left|\det \nabla_{a'} T_s^{-1}(a')\right|, \quad a' = T(s, a). \tag{23}$$

Assume that π_β decays sufficiently fast at infinity so that integration by parts is valid.

Then, the Kullback–Leibler divergence admits the expansion

$$D_{\mathrm{KL}}\big(\pi_\theta(\cdot|s) \,\|\, \pi_\beta(\cdot|s)\big) \approx \frac{1}{2} \int_{\mathbb{R}^d} \pi_\beta(a|s)\, \delta(s, a)^\top \nabla_a \log \pi_\beta(a|s) \nabla_a \log \pi_\beta(a|s)^\top\, \delta(s, a)\, da. \tag{24}$$

Proof. We start from the definition of the Kullback–Leibler divergence:

$$D_{\mathrm{KL}}(\pi_\theta \| \pi_\beta) = \int_{\mathbb{R}^d} \pi_\theta(a'|s) \log \frac{\pi_\theta(a'|s)}{\pi_\beta(a'|s)}\, da'. \tag{25}$$

Let $a' = T(s, a) = a + \delta(s, a)$. Using the pushforward relation in Lemma C.1

$$\pi_\theta(a'|s) = \pi_\beta(a|s) \left|\det \nabla_a T_s(a)\right|^{-1}, \tag{26}$$

we obtain

$$D_{\mathrm{KL}} = \int_{\mathbb{R}^d} \pi_\beta(a|s) \log \frac{\pi_\theta(T_s(a)|s)}{\pi_\beta(T_s(a)|s)}\, da. \tag{27}$$

Substituting the density transformation yields

$$\log \frac{\pi_\theta(T_s(a)|s)}{\pi_\beta(T_s(a)|s)} = \log \pi_\beta(a|s) - \log \pi_\beta(T_s(a)|s) - \log \det \nabla_a T_s(a). \tag{28}$$

We now expand each term up to second order.

Step 1: Expansion of $\log \pi_\beta(T_s(a))$.

Using Taylor expansion,

$$\log \pi_\beta(a + \delta) = \log \pi_\beta(a|s) + \nabla_a \log \pi_\beta(a|s)^\top \delta + \frac{1}{2} \delta^\top \nabla_a^2 \log \pi_\beta(a|s)\, \delta + \mathcal{O}(\|\delta\|^3). \tag{29}$$

Step 2: Second-order expansion of the log-determinant.

Using the exact Taylor expansion for the log-determinant of the Jacobian $\nabla_a T_s(a) = I + \nabla_a \delta(s, a)$, we have

$$\begin{aligned} \log \det \nabla_a T_s(a) &= \mathrm{tr}(\nabla_a \delta(s, a)) - \frac{1}{2} \mathrm{tr}\left((\nabla_a \delta(s, a))^2\right) + \mathcal{O}(\|\nabla_a \delta\|^3) \\ &= \nabla \cdot \delta(s, a) - \frac{1}{2} \mathrm{tr}\left((\nabla_a \delta(s, a))^2\right) + \mathcal{O}(\|\nabla_a \delta\|^3). \end{aligned} \tag{30}$$

Unlike the first-order approximation, we must retain the quadratic trace term as it contributes to the second-order expansion of the KL divergence.

Step 3: Combine expansions.

Substituting the expansions of both the log-density and the log-determinant into the log-ratio, we obtain

$$\begin{aligned} \log \frac{\pi_\theta(T_s(a)|s)}{\pi_\beta(T_s(a)|s)} = &- \nabla_a \log \pi_\beta(a|s)^\top \delta - \frac{1}{2} \delta^\top \nabla_a^2 \log \pi_\beta(a|s)\, \delta \\ &- \nabla \cdot \delta + \frac{1}{2} \mathrm{tr}\left((\nabla_a \delta)^2\right) + \mathcal{O}(\|\delta\|^3). \end{aligned} \tag{31}$$

Step 4: Substitute into KL.

Integrating over the base distribution $\pi_\beta(a|s)$, the KL divergence becomes:

$$D_{\mathrm{KL}}\big(\pi_\theta(\cdot|s)\,\|\,\pi_\beta(\cdot|s)\big) = \int \pi_\beta(a|s)\Big[-\nabla_a \log \pi_\beta(a|s)^\top \delta - \nabla\cdot\delta \\ -\frac{1}{2}\delta^\top \nabla_a^2 \log \pi_\beta(a|s)\,\delta + \frac{1}{2}\,\mathrm{tr}\left((\nabla_a\delta)^2\right)\Big] da + \mathcal{O}(\|\delta\|^3). \tag{32}$$

Step 5: Cancellation of first-order terms.

By applying integration by parts and assuming π_β decays sufficiently fast at the boundary [75, 76], the expected divergence of the vector field equates to:

$$\int \pi_\beta\,\nabla\cdot\delta\,da = -\int \nabla_a\pi_\beta\cdot\delta\,da = -\int \pi_\beta \nabla_a \log\pi_\beta\cdot\delta\,da. \tag{33}$$

Consequently, the first-order terms exactly cancel out:

$$\int \pi_\beta\Big(-\nabla_a\log\pi_\beta\cdot\delta - \nabla\cdot\delta\Big)da = 0. \tag{34}$$

Step 6: Final result and reduction to Fisher form.

We are left with the second-order terms comprising the Hessian of the log-density and the trace of the squared Jacobian:

$$D_{\mathrm{KL}} = \frac{1}{2}\int \pi_\beta\Big[-\delta^\top\nabla_a^2\log\pi_\beta\,\delta + \mathrm{tr}\left((\nabla_a\delta)^2\right)\Big]da + \mathcal{O}(\|\delta\|^3). \tag{35}$$

Using the logarithmic derivative identity, we decompose the negative Hessian into two components:

$$-\nabla_a^2\log\pi_\beta = -\underbrace{\frac{\nabla_a^2\pi_\beta}{\pi_\beta}}_{\text{local curvature}} + \underbrace{\nabla_a\log\pi_\beta\nabla_a\log\pi_\beta^\top}_{\text{Fisher information matrix}}. \tag{36}$$

Substituting this into (35) yields:

$$D_{\mathrm{KL}} = \frac{1}{2}\int \pi_\beta\Big(\delta^\top\nabla_a\log\pi_\beta\nabla_a\log\pi_\beta^\top\delta\Big)da - \frac{1}{2}\int\delta^\top(\nabla_a^2\pi_\beta)\delta\,da + \frac{1}{2}\int\pi_\beta\,\mathrm{tr}\left((\nabla_a\delta)^2\right)da. \tag{37}$$

To resolve the curvature term $\int\delta^\top(\nabla_a^2\pi_\beta)\delta\,da$, we apply integration by parts. Since $\nabla_a\pi_\beta = \pi_\beta\nabla_a\log\pi_\beta$, the boundary-free integral transfers the derivative onto the displacement field:

$$\int\delta^\top(\nabla_a^2\pi_\beta)\delta\,da = -\int\pi_\beta\Big[(\nabla\cdot\delta)(\delta^\top\nabla_a\log\pi_\beta) + \delta^\top(\nabla_a\delta)\nabla_a\log\pi_\beta\Big]da. \tag{38}$$

Inserting (38) back into (37), we can group all terms involving the Jacobian $\nabla_a\delta$ into a single residual operator $\mathcal{E}$:

$$D_{\mathrm{KL}} = \frac{1}{2}\int\pi_\beta\Big(\delta^\top\nabla_a\log\pi_\beta\nabla_a\log\pi_\beta^\top\delta\Big)da + \mathcal{E}(\delta,\nabla_a\delta), \tag{39}$$

where

$$\mathcal{E} = \frac{1}{2}\int\pi_\beta\Big[\mathrm{tr}\left((\nabla_a\delta)^2\right) + (\nabla\cdot\delta)(\delta^\top\nabla_a\log\pi_\beta) + \delta^\top(\nabla_a\delta)\nabla_a\log\pi_\beta\Big]da. \tag{40}$$

In the context of residual policy learning, we naturally regularize the transport map $\delta(s,a)$ to be a **slowly varying, locally near-rigid displacement field** to avoid catastrophic distributional shift. This

physical constraint implies that the local deformation is vanishingly small compared to the translation itself, i.e., $\|\nabla_a \delta\| = o(\|\delta\|)$.

Under this setting, the residual $\mathcal{E}$, which scales as $\mathcal{O}(\|\nabla_a \delta\|^2)$ and $\mathcal{O}(\|\delta\|\|\nabla_a \delta\|)$, is dominated by the translational energy $\mathcal{O}(\|\delta\|^2)$. Omitting these higher-order deformational artifacts, we recover the elegant Fisher information form:

$$D_{\mathrm{KL}}\big(\pi_\theta(\cdot|s) \,\|\, \pi_\beta(\cdot|s)\big) \approx \frac{1}{2} \int \pi_\beta(a|s)\, \delta(s,a)^\top \big(\nabla_a \log \pi_\beta(a|s) \nabla_a \log \pi_\beta(a|s)^\top\big) \delta(s,a)\, da. \tag{41}$$

This result fundamentally demonstrates that constraining the Jacobian of the displacement field naturally reduces the KL divergence to a local quadratic form governed by the local Fisher information metric, measuring the translational discrepancy. □

Miscellaneous Complements: Formal Analysis of the Curvature Term

We provide a rigorous analysis of the term $\mathcal{J} = \int_{\mathcal{A}} \delta^\top (\nabla_a^2 \pi_\beta) \delta \, da$ and its physical implications for the transport map in refining flow policies.

1. Global Cancellation of Density Curvature We first establish that the total curvature of any valid probability density integrates to zero over its support. Let $\mathcal{A} \subseteq \mathbb{R}^d$ be the action space. By the definition of the Laplacian operator, we have $\nabla_a^2 \pi_\beta = \nabla \cdot (\nabla_a \pi_\beta)$. According to the Divergence Theorem (or Stokes' Theorem) [77], the integral over $\mathcal{A}$ can be converted into a boundary integral:

$$\int_{\mathcal{A}} \nabla_a^2 \pi_\beta \, da = \int_{\mathcal{A}} \nabla \cdot (\nabla_a \pi_\beta) \, da = \oint_{\partial \mathcal{A}} (\nabla_a \pi_\beta \cdot \vec{n}) \, dS, \tag{42}$$

where $\vec{n}$ denotes the unit outward normal vector on the boundary $\partial \mathcal{A}$ of the compact set $\mathcal{A}$. For behavioral policy distributions, we assume π_β decays sufficiently fast at infinity (or reaches zero at the boundary of the compact support), implying:

- $\pi_\beta(a|s) \to 0$ as $a \to \partial \mathcal{A}$,
- $\nabla_a \pi_\beta(a|s) \to 0$ (the density becomes flat at the boundary).

Under these regularity conditions, the boundary flux vanishes, leading to the global cancellation of curvature:

$$\int_{\mathcal{A}} \nabla_a^2 \pi_\beta \, da = 0. \tag{43}$$

2. Physical Insight: Avoiding Distributional Shift In our residual policy setting, we refine the behavior policy π_β via a displacement field $\delta(s,a)$. The curvature term $\frac{\nabla_a^2 \pi_\beta}{\pi_\beta}$ represents local **volumetric deformation**—the degree to which the action space is locally compressed or stretched.

Equation (43) shows that the expectation of this curvature over the entire action space is zero. When designing the residual term $\delta(s,a)$, our objective is to ensure that $\nabla_a \delta(s,a)$ does not change dramatically. A high gradient in δ would imply a significant deformation of the behavioral policy structure, potentially leading to a catastrophic **distributional shift**.

By ensuring $\|\nabla_a \delta(s,a)\|$ small relatively to $\|\delta(s,a)\|$, we allow to transport its high-probability mass while preserving the internal density structure (i.e., avoiding "collapsing" or "tearing" the distribution). Under such smoothness assumptions, δ can be treated as locally constant relative to the curvature, yielding:

$$\mathcal{J} = \int_{\mathcal{A}} \delta^\top (\nabla_a^2 \pi_\beta) \delta \, da \approx 0. \tag{44}$$

This supports the use of the Fisher information form as the leading-order metric for policy divergence: it captures the translational energy of the update, while the deformation-related curvature effects are asymptotically negligible under the near-rigid transport.

C.2 Score Function Estimation

Lemma C.4. *Let $a^1 \sim \pi_\beta(a|s)$ and $z \sim \mathcal{N}(0, I_d)$ be independent random variables. Define the convex combination:*

$$a^t = ta^1 + (1-t)z, \quad t \in (0,1).$$

Then, the score function of the marginalized distribution $\pi_\beta(a^t|s)$ is given by:

$$\nabla_a \log \pi_\beta(a^t|s) = \frac{t\,\mathbb{E}[a^1|a^t] - a^t}{(1-t)^2}.$$

Furthermore, given the conditional velocity field $v(t, s, a^t)$ defined in the flow matching framework, the score function can be equivalently expressed as:

$$\nabla_a \log \pi_\beta(a^t|s) = \frac{tv(t, s, a^t) - a^t}{1-t}. \tag{45}$$

Proof. Based on the linear Gaussian interpolation $a^t = ta^1 + (1-t)z$, the conditional distribution of a^t given a^1 is:

$$a^t|a^1 \sim \mathcal{N}(ta^1, (1-t)^2 I_d).$$

The log-likelihood gradient (score) of this conditional Gaussian density is:

$$\nabla_{a^t} \log p(a^t|a^1, s) = \nabla_{a^t} \log \mathcal{N}\left(a^t; ta^1, (1-t)^2 I_d\right) = -\frac{a^t - ta^1}{(1-t)^2}. \tag{46}$$

By the property of the marginal score function (or via the law of total expectation applied to the score):

$$\begin{aligned}
\nabla_a \log \pi_\beta(a^t|s) &= \mathbb{E}_{p(a^1|a^t,s)}\left[\nabla_a \log p(a^t|a^1, s)\right] \\
&= \mathbb{E}\left[-\frac{a^t - ta^1}{(1-t)^2}\middle| a^t\right] \\
&= \frac{t\mathbb{E}[a^1|a^t] - a^t}{(1-t)^2}.
\end{aligned}$$

In the context of Flow Matching with a Gaussian path, the optimal velocity field $v(t, s, a^t)$ satisfies the relationship:

$$v(t, s, a^t) = \mathbb{E}\left[\frac{d}{dt}a^t \middle| a^t\right] = \mathbb{E}[a^1 - z|a^t]. \tag{47}$$

Substituting $z = \frac{a^t - ta^1}{1-t}$ into the above, we obtain the identity:

$$\mathbb{E}[a^1|a^t] = a^t + (1-t)v(t, s, a^t). \tag{48}$$

Substituting this into the score expression:

$$\begin{aligned}
\nabla_a \log \pi_\beta(a^t|s) &= \frac{t\left(a^t + (1-t)v(t, s, a^t)\right) - a^t}{(1-t)^2} \\
&= \frac{(t-1)a^t + t(1-t)v(t, s, a^t)}{(1-t)^2} \\
&= \frac{tv(t, s, a^t) - a^t}{1-t}.
\end{aligned}$$

This completes the proof. □

Our objective is to recover the score function of the behavioral distribution as $t \to 1$. However, the expression in (45) exhibits a singularity in this limit, as both the numerator and denominator vanish simultaneously, leading to numerical instability. To circumvent this, we introduce a small perturbation ε to the time variable, yielding a numerically stable approximation. The following theorem establishes the convergence rate and error bound between this perturbed approximation and the ground-truth score function.

Theorem C.5 (Pointwise Convergence of the Perturbed Score Function)**.** *Let $\pi_\beta(a|s)$ denote the behavioral policy distribution defined on a compact set of $\mathbb{R}^d$, and assume $\pi_\beta(\cdot|s) \in C^3(\mathbb{R}^d, \mathbb{R})$ over the compact subset of $\mathbb{R}^d$. Consider the linear Gaussian interpolation path, let $\mathbf{s}_t(a|s)$ be the associated score function at time t. For a small perturbation parameter $\varepsilon > 0$, define the perturbed time $t_\varepsilon = 1 - \varepsilon$. As $\varepsilon \to 0^+$, the approximation error between the perturbed score $\mathbf{s}_{1-\varepsilon}(a|s)$ and the ground truth score $\mathbf{s}_1(a|s)$ satisfies:*

$$\|\mathbf{s}_{1-\varepsilon}(a|s) - \mathbf{s}_1(a|s)\| = \varepsilon \left\|\nabla_a\big(a^\top \nabla_a \log \pi_\beta(a|s)\big)\right\| + \mathcal{O}(\varepsilon^2).$$

Consequently, the boundary perturbation scheme achieves a first-order convergence rate of $\mathcal{O}(\varepsilon)$.

Proof. **1. Integral Representation with Drift**
In the linear Gaussian interpolation path $a_t = ta_0 + (1-t)\eta$, where $a_0 \sim \pi_\beta(\cdot|s)$ and $\eta \sim \mathcal{N}(0, I_d)$, the marginal density $p_t(a|s)$ at time $t = 1 - \varepsilon$ is given by:

$$p_{1-\varepsilon}(a|s) = \int \pi_\beta(a_0|s)\mathcal{N}\big(a; (1-\varepsilon)a_0, \varepsilon^2 I_d\big)da_0.$$

To evaluate this convolution, we reparameterize the integral by setting $y = (1-\varepsilon)a_0$. The change of measure yields $da_0 = (1-\varepsilon)^{-d}dy$, giving:

$$p_{1-\varepsilon}(a|s) = \int \underbrace{(1-\varepsilon)^{-d}\pi_\beta\left(\frac{y}{1-\varepsilon}\middle| s\right)}_{:=f_\varepsilon(y)} \mathcal{N}(a; y, \varepsilon^2 I_d)dy.$$

2. Taylor Expansion of the Base Density
We first expand the scaled density function $f_\varepsilon(y)$ for small ε. Using $(1-\varepsilon)^{-d} = 1 + d\varepsilon + \mathcal{O}(\varepsilon^2)$ and the Taylor expansion of π_β, we have:

$$\begin{aligned} f_\varepsilon(a) &= (1 + d\varepsilon)\left[\pi_\beta(a|s) + \nabla_a \pi_\beta(a|s)^\top(\varepsilon a)\right] + \mathcal{O}(\varepsilon^2) \\ &= \pi_\beta(a|s) + \varepsilon\big(d\pi_\beta(a|s) + a^\top \nabla_a \pi_\beta(a|s)\big) + \mathcal{O}(\varepsilon^2). \end{aligned}$$

Recognizing the term in the parenthesis as the divergence of the vector field $a\pi_\beta(a|s)$, we obtain:

$$f_\varepsilon(a) = \pi_\beta(a|s) + \varepsilon \nabla_a \cdot \big(a\pi_\beta(a|s)\big) + \mathcal{O}(\varepsilon^2).$$

3. Applying the Gaussian Mollifier
The integral $p_{1-\varepsilon}(a|s) = \int f_\varepsilon(y)\mathcal{N}(a; y, \varepsilon^2 I_d)dy$ is the standard heat equation evolution of f_ε for "heat-time" $\tau = \varepsilon^2$. Thus,

$$p_{1-\varepsilon}(a|s) = f_\varepsilon(a) + \frac{\varepsilon^2}{2}\Delta_a f_\varepsilon(a) + \mathcal{O}(\varepsilon^4).$$

Substituting our expansion for $f_\varepsilon(a)$, the ε^2 term from the heat kernel is absorbed into the higher-order terms, leaving the drift-induced first-order error as the dominant term:

$$p_{1-\varepsilon}(a|s) = \pi_\beta(a|s) + \varepsilon \nabla_a \cdot \big(a\pi_\beta(a|s)\big) + \mathcal{O}(\varepsilon^2). \tag{49}$$

4. Expansion of the Score Function
Taking the gradient of (49) with respect to a, we get:

$$\nabla_a p_{1-\varepsilon} = \nabla_a \pi_\beta + \varepsilon \nabla_a\big(\nabla_a \cdot (a\pi_\beta)\big) + \mathcal{O}(\varepsilon^2).$$

The score function is $\mathbf{s}_{1-\varepsilon} = \frac{\nabla_a p_{1-\varepsilon}}{p_{1-\varepsilon}}$. Using the first-order quotient linearization $\frac{u+\varepsilon\delta u}{v+\varepsilon\delta v} \approx \frac{u}{v} + \varepsilon\left(\frac{v\delta u - u\delta v}{v^2}\right)$, where $u = \nabla_a \pi_\beta$ and $v = \pi_\beta$, we have:

$$\mathbf{s}_{1-\varepsilon} = \frac{\nabla_a \pi_\beta}{\pi_\beta} + \varepsilon\left[\frac{\pi_\beta \nabla_a\big(\nabla_a \cdot (a\pi_\beta)\big) - \nabla_a \pi_\beta\big(\nabla_a \cdot (a\pi_\beta)\big)}{\pi_\beta^2}\right] + \mathcal{O}(\varepsilon^2).$$

The term in the bracket is exactly the gradient of the scalar field $\frac{\nabla_a \cdot (a\pi_\beta)}{\pi_\beta}$ via the quotient rule. Therefore:

$$\mathbf{s}_{1-\varepsilon}(a|s) - \mathbf{s}_1(a|s) = \varepsilon \nabla_a \left(\frac{\nabla_a \cdot (a\pi_\beta)}{\pi_\beta} \right) + \mathcal{O}(\varepsilon^2).$$

5. Final Simplification
We can simplify the scalar field inside the gradient:

$$\frac{\nabla_a \cdot (a\pi_\beta)}{\pi_\beta} = \frac{d\pi_\beta + a^\top \nabla_a \pi_\beta}{\pi_\beta} = d + a^\top \nabla_a \log \pi_\beta(a|s).$$

Since the gradient of the constant d is zero, the error term reduces to:

$$\mathbf{s}_{1-\varepsilon}(a|s) - \mathbf{s}_1(a|s) = \varepsilon \nabla_a \big(a^\top \nabla_a \log \pi_\beta(a|s) \big) + \mathcal{O}(\varepsilon^2).$$

Taking the norm completes the proof. □

C.3 Analysis of Optimal Choice of Time Perturbation

According to the approximation of the score function, the local quadratic form (or Fisher information equivalent) can be computed as an outer product:

$$I(s,a) \approx \frac{\big((1-\varepsilon)v_\beta(1-\varepsilon,s,a) - a\big)\big((1-\varepsilon)v_\beta(1-\varepsilon,s,a) - a\big)^\top}{\varepsilon^2}. \tag{50}$$

We now turn to analyze the optimal choice of the time perturbation ε. The total computation error, denoted by $\mathcal{E}_{\text{total}}(\varepsilon) = \mathcal{E}_{\text{trunc}} + \mathcal{E}_{\text{num}}$, consists of two primary components: the approximation error stemming from the time perturbation (truncation bias) and the numerical rounding error that arises as ε becomes vanishingly small.

Based on the first-order pointwise convergence established in Theorem C.5, the norm of the score error scales as $\mathcal{O}(\varepsilon)$. Consequently, the **truncation error** for the squared objective $I(s,a)$ satisfies:

$$\mathcal{E}_{\text{trunc}} \approx C_1 \varepsilon^2, \quad \text{where} \quad C_1 = \left\| \nabla_a \big(a^\top \nabla_a \log \pi_\beta(a|s) \big) \right\|^2. \tag{51}$$

The constant C_1 captures the magnitude of the "tidal force" or spatial deformation of the score field under radial shrinkage. A distribution with highly varying local curvature implies that the score field deforms sharply towards the origin, necessitating a smaller ε to minimize this truncation bias.

Conversely, the **numerical rounding error** is a consequence of finite floating-point precision (machine epsilon δ). In the expression (50), calculating the squared norm involves dividing by ε^2. As $\varepsilon \to 0$, the denominator amplifies the infinitesimal floating-point variance in the numerator. This numerical noise variance can be modeled as:

$$\mathcal{E}_{\text{num}} \approx \frac{C_2 \delta}{\varepsilon^2}, \tag{52}$$

where C_2 is a constant related to the magnitude of the velocity field v and the action a. To identify the optimal perturbation ε^*, we minimize the total mean squared error:

$$\min_\varepsilon \mathcal{E}_{\text{total}}(\varepsilon) = C_1 \varepsilon^2 + \frac{C_2 \delta}{\varepsilon^2}. \tag{53}$$

Taking the derivative with respect to ε and setting it to zero yields:

$$2C_1 \varepsilon - \frac{2C_2 \delta}{\varepsilon^3} = 0 \implies \varepsilon^* = \left(\frac{C_2 \delta}{C_1} \right)^{1/4}. \tag{54}$$

This derivation reveals a profound connection to classical numerical analysis: the optimal time perturbation ε^* scales with the machine precision following the classical first-order finite-difference scaling law, $\mathcal{O}(\delta^{1/4})$. Furthermore, ε^* is inversely proportional to the fourth root of the score field's radial deformation magnitude (C_1).

Consequently, while highly complex distributions theoretically require a finer perturbation, the practical lower bound of ε is strictly dictated by the hardware precision δ_{FP32}. In a standard FP32 environment ($\delta_{\text{FP32}} \sim 10^{-6}$), the optimal perturbation scales near $(10^{-6})^{1/4} \approx 10^{-1.5}$. Therefore, choosing $\varepsilon \sim \mathcal{O}(10^{-2})$ to $\mathcal{O}(10^{-1})$ typically achieves the most robust balance between approximation accuracy and numerical stability. Empirical results from the ablation studies are completely consistent with this theoretical scaling law, as shown in Figure 5 and Table 10.

C.4 Trust-Region Problem

The detailed algorithm is presented in Appendix D, and the following paragraphs describe the architectural components in detail.

Lagrangian Formulation. The constrained optimization problem in (11) can be reformulated by introducing the Lagrangian function $\mathcal{L}(\delta_\theta, \lambda)$. Let $\lambda \geq 0$ be the Lagrange multiplier associated with the trust-region constraint [78]:

$$\mathcal{L}(\delta_\theta, \lambda) = \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\beta(\cdot|s)}\left[Q_\phi(s, a + \delta_\theta(s, a))\right] - \lambda\left(\mathbb{E}_{s,a}\left[\frac{1}{2}\delta_\theta(s, a)^\top I(s, a)\delta_\theta(s, a)\right] - \epsilon\right), \tag{55}$$

where $I(s, a) = \nabla_a \log \pi_\beta \nabla_a \log \pi_\beta^\top$ denotes the Fisher information matrix at the state-action pair (s, a). The optimal parameters are found by solving the minimax problem: $\min_{\lambda\geq 0} \max_\theta \mathcal{L}(\theta, \lambda)$.

Inner Problem (Primal Update). For a fixed dual variable λ, the inner problem aims to update the policy parameters θ to maximize the Lagrangian. Unlike traditional KL-constrained methods that require complex approximations, our quadratic reformulation allows for stable and efficient optimization via automatic differentiation:

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathbb{E}_{s,a}\left[Q_\phi(s, a + \delta_\theta(s, a)) - \frac{\lambda}{2}\delta_\theta(s, a)^\top I(s, a)\delta_\theta(s, a)\right], \tag{56}$$

where η_θ is the learning rate. By leveraging the reparameterization trick through the transport map $T_s(a) = a + \delta_\theta(s, a)$, the gradient flows directly from the critic and the Fisher-based penalty into the residual network.

Dual Update. The dual variable λ is updated to enforce the trust-region constraint. Given the current residual policy δ_θ, λ is updated by minimizing the dual objective:

$$\lambda \leftarrow \mathrm{ReLU}\left(\lambda + \eta\, \mathbb{E}_{s,a}\left[\frac{1}{2}\delta_\theta(s, a)^\top I(s, a)\delta_\theta(s, a) - \epsilon\right]\right) \tag{57}$$

In practice, to ensure that λ remains non-negative, we typically optimize $\log \lambda$. This dual update automatically adjusts the "penalty strength" of the Fisher metric, ensuring the policy refinement stays within the provable neighborhood of the behavioral policy support.

Further Discussion of the Lagrangian (55). The inner optimization problem in (55) admits an insightful analytical structure. By applying a first-order Taylor expansion to the critic $Q_\phi(s, a + \delta_\theta)$ around the behavioral action a, the objective becomes locally quadratic in δ:

$$\max_{\delta_\theta} \mathbb{E}_{s,a}\left[\nabla_a Q_\phi(s, a)^\top \delta_\theta(s, a) - \frac{\lambda}{2}\delta_\theta(s, a)^\top I(s, a)\delta_\theta(s, a)\right]. \tag{58}$$

Taking the derivative with respect to δ gives the normal equation

$$\lambda I(s, a)\delta = \nabla_a Q_\phi(s, a).$$

Since the local Fisher matrix $I(s, a)$ may be positive semidefinite, the pseudo-inverse solution

$$\delta^*(s, a) \approx \frac{1}{\lambda} I(s, a)^\dagger \nabla_a Q_\phi(s, a)$$

should be interpreted as the minimum-norm stationary solution when $\nabla_a Q_\phi(s, a) \in \mathrm{Range}(I(s, a))$. If this condition fails, the undamped objective may be unbounded along the null space of $I(s, a)$. In practice, this motivates using a small damped metric

$$I_\rho(s, a) = I(s, a) + \rho I_d, \qquad \rho > 0,$$

which yields the well-defined update

$$\delta^*_\rho(s, a) \approx \frac{1}{\lambda} I_\rho(s, a)^{-1} \nabla_a Q_\phi(s, a). \tag{59}$$

This expression reveals that the optimal policy refinement is effectively a pointwise Natural Gradient update in the action space, where the gradient of the critic is preconditioned by the inverse of local Fisher information matrix.

This closed-form solution suggests a significant opportunity for accelerating offline RL training. Traditionally, policy refinement requires iterative gradient-based optimization (as shown in the Primal Update). However, (59) implies that if the score function (and thus $I(s,a)$) is efficiently estimated, one could potentially bypass the inner loops of policy training entirely. By directly computing δ^* and updating the policy via a single supervised learning step, or even performing "on-the-fly" refinement during inference, the computational bottleneck of policy-critic coordination could be drastically reduced. We believe that leveraging this metric-aware analytical solution to design training-free or one-step refinement paradigms is a promising frontier for real-time offline RL applications.

C.5 Deviated Optimal Solution Gap under Isotropic Metric.

We analyze the suboptimality induced by replacing the anisotropic metric $I(s,a)$ (or practical computation with $I_\rho(s,a)$ for one-step computation) with an isotropic identity matrix. Under the quadratic approximation in (55), the objective takes the general linear form with the quadratic constraint:

$$\mathcal{L}(\delta_\theta, \lambda) = g^\top \delta - \frac{\lambda}{2}\delta^\top M \delta, \tag{60}$$

where $g = \nabla_a Q_\phi(s,a)$ and $M = I_\rho(s,a)$.

The optimal solution under the anisotropic metric is:

$$\delta^*_{\text{ani}} = \frac{1}{\lambda} M^\dagger g, \tag{61}$$

yielding the optimal value:

$$\mathcal{L}^*_{\text{ani}} = \frac{1}{2\lambda} g^\top M^\dagger g. \tag{62}$$

In contrast, replacing M with the identity matrix leads to the isotropic solution:

$$\delta^*_{\text{iso}} = \frac{1}{\lambda} g, \tag{63}$$

with objective value:

$$\mathcal{L}^*_{\text{iso}} = \frac{1}{2\lambda} g^\top g. \tag{64}$$

We define the optimality gap as:

$$\Delta(s,a) = \mathcal{L}^*_{\text{ani}} - \mathcal{L}^*_{\text{iso}} = \frac{1}{2\lambda}\left(g^\top M^\dagger g - g^\top g\right). \tag{65}$$

To further characterize this gap, consider the eigendecomposition $M = U\Lambda U^\top$, where $\Lambda = \text{diag}(\lambda_i)$ and $U = [u_1, \cdots, u_d]$ ($\{u_i\}_{i=1}^d$ is the orthogonal basis). Then the gap can be written as:

$$\Delta(s,a) = \frac{1}{2\lambda}\sum_i (u_i^\top g)^2 \left(\lambda_i^{-1} - 1\right). \tag{66}$$

This expression reveals that the suboptimality arises from ignoring the local curvature encoded in $M = I_\rho(s,a)$. Specifically, directions with small eigenvalues ($\lambda_i < 1$) are under-emphasized by the isotropic update, while directions with large eigenvalues ($\lambda_i > 1$) are over-amplified. As a result, the isotropic solution fails to properly balance exploration across directions, leading to degraded objective improvement.

In the special case where $M = I_d$, the gap vanishes. However, in general offline RL settings where the data distribution induces highly anisotropic geometry, this gap can be significant. This highlights the importance of anisotropic metric-aware updates for achieving optimal policy refinement.

D Implementation

Algorithm 1: FiDec

Function $I(s, a)$
  return $\frac{\left(t_\varepsilon v_\beta(t_\varepsilon, s, a) - a\right)\left(t_\varepsilon v_\beta(t_\varepsilon, s, a) - a\right)^\top}{(1-t_\varepsilon)^2}$ `// Fisher information matrix`
end

Function $\mu_\beta(s, z)$
  for $t = 0$ **to** $M - 1$ **do**
    $z \leftarrow z + \frac{1}{M} v_\beta\left(\frac{t}{M}, s, z\right)$
  end
  return z `// Behavioral policy`
end

Function $T_s(\mu_\beta(s, z))$
  Sample $z \sim \mathcal{N}(0, I_d)$;
  return $a + \delta_\theta(s, \mu_\beta(s, z))$ `// Transport map`
end

while *not converged* **do**
  Sample minibatch $\{(s, a, r, s')\} \sim \mathcal{D}$;
  `// Step (1): Update critic` Q_ϕ
  Sample $z \sim \mathcal{N}(0, I_d)$;
  $a' \leftarrow T_{s'}(\mu_\beta(s', z))$;
  Update ϕ by minimizing TD error:

$$\mathbb{E}\left[\left(Q_\phi(s, a) - r - \gamma \bar{Q}(s', a')\right)^2\right]$$

  `// Step (2): Train vector field` v_β
  Sample $x^0 \sim \mathcal{N}(0, I_d)$;
  $x^1 \leftarrow a$;
  Sample $t \sim \mathrm{Unif}[0, 1]$;
  $x^t \leftarrow (1 - t)x^0 + tx^1$;
  Update β by minimizing:

$$\mathbb{E}_{t\sim\mathrm{Unif}[0,1], s, a=x^1\sim\mathcal{D}, x^0\sim\mathcal{N}(0,I_d)}\left[\|v_\beta(t, s, x^t) - (x^1 - x^0)\|_2^2\right]$$

  `// Step (3): Train transport map` T
  $\delta_\theta(s, a) \leftarrow T(s, \mu_\beta(s, z)) - \mu_\beta(s, z)$;
  Threshold ϵ ;
  Update θ by maximizing:

$$\mathbb{E}_{s\sim\mathcal{D},\, z\sim\mathcal{N}(0,I_d)}\left[Q_\phi(s, T_s(\mu_\beta(s, z))) - \lambda\left(\frac{1}{2}\delta_\theta(s, a)^\top I(s, a)\delta_\theta(s, a) - \epsilon\right)\right]$$

  `// Step (4): Update dual variable` λ
  Step size η ;

$$\lambda \leftarrow \mathrm{ReLU}\left(\lambda + \eta\, \mathbb{E}_{s\sim\mathcal{D},\, a\sim\pi_\beta(\cdot|s))}\left[\frac{1}{2}\delta_\theta(s, a)^\top I(s, a)\delta_\theta(s, a) - \epsilon\right]\right)$$

end
return transport map T_s

E More Experimental Results.

E.1 Benchmarking Tasks

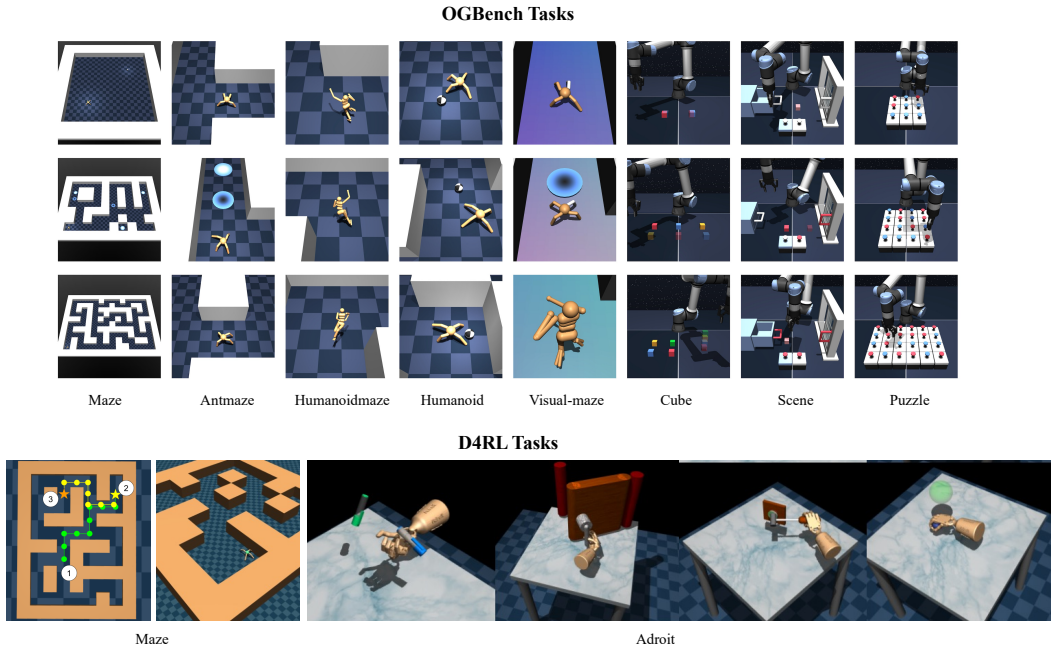

Figure 6: Overview of benchmark tasks. Our evaluation spans a diverse set of environments from OGBench [5] (Maze, Antmaze, Humanoidmaze, Humanoid, Visual-maze, Cube, Scene, Puzzle) and D4RL [39] (Maze and Adroit), covering a wide range of challenges including navigation, locomotion, dexterous manipulation, and vision-based control.

E.2 Architectures and Hyperparameters

Table 5 summarizes the main training configuration of our FiDec. We retain the same DeFlow/FQL-style backbone settings for optimization, network scale, critic update, flow integration, and evaluation protocol, while introducing the Fisher estimation time t_ε as the key additional hyperparameter.

Table 5: Hyperparameters and configuration summary for the FiDec/DeFlow backbone.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
discount γ (default)	0.99	target update rate τ	0.005
actor / critic learning rate	3×10^{-4}	batch size	256
evaluation interval	10^5	evaluation episodes	50
replay buffer size (minimum)	2×10^6	save interval	5×10^5
actor hidden dims	$(512, 512, 512, 512)$	critic hidden dims	$(512, 512, 512, 512)$
critic layer norm	True	actor layer norm	False
flow steps	10	flow time sampling	$\mathrm{Unif}[0, 1]$
double critic ensemble	2	gradient clipping	5.0
Q-normalization	True	use Lagrange tuning	True
BC / trust coefficient init.	10.0	fix BC flow online	False

Notes. This table reports the shared backbone configuration together with the Fisher-specific settings used in the final runs. Following the latest wandb-confirmed runtime configuration, we use $t_\varepsilon = 0.8$. The replay buffer size is reported as a configured minimum, since the actual buffer allocation is implemented as $\max(\texttt{buffer_size}, \texttt{dataset.size} + 1)$.

Table 6 summarizes the model structure and computation flow. The BC flow policy, refinement head, and double-critic backbone remain aligned with the DeFlow/FQL-style implementation. The

key component is a paradigm shift from an isotropic penalty to a local Fisher information matrix, estimated from the velocity field of flow.

Table 6: Architecture and computation summary of the FiDec.

Module	**Sub-Module**	**Structure / Computation**	**Output Dimension**
BC Flow Policy $v_\beta(s, x_t, t)$	Input	Observation, corrected flow sample, and time scalar	$d_s + d_a + 1$
	Backbone	MLP with hidden dims $(512, 512, 512, 512)$	512
	Update objective	Standard BC flow-matching loss	1
	Output	Velocity field in action space	d_a
Transport map $T_s(a) = a + \delta_\theta(s, a)$	Input	Observation and base action $a = \mu_\beta(s, a)$	$d_s + d_a$
	Backbone	MLP with hidden dims $(512, 512, 512, 512)$	512
	Role	Predict residual refinement $\delta_\theta(s, a)$	d_a
	Output action	$\mu_\beta(s, z) + \delta_\theta(s, a)$	d_a
Reward Critic $Q_\phi(s, a)$	Input	Observation-action pair (s, a)	$d_s + d_a$
	Backbone	Value network with hidden dims $(512, 512, 512, 512)$	512
	Critic normalization	LayerNorm enabled	–
	Ensemble	Double critic	2
	Output	Scalar Q estimate	1
Fisher Metric Estimation	Base action sampling	$a = \mathrm{ODESolve}(v_\beta, s, z)$, $z \sim \mathcal{N}(0, I)$	d_a
	Velocity query	Reuse BC flow network $v_\beta(s, x_t, t_\varepsilon)$	d_a
	Score estimate	$\mathbf{s}_{t_\varepsilon} = (t_\varepsilon v_t - a)/(1 - t_\varepsilon)$	d_a
	Fisher penalty	$\frac{1}{2}\delta(s, a)^\top I(s, a)\delta(s, a)$ with trace-normalized $I(s, a)$	1
Constraint Control	Lagrange parameter	Learnable scalar $\log \lambda$	1
	Trust-region control	Fisher-induced quadratic metric corresponding to KL divergence approximation	1
	Actor objective	BC flow loss + quadratic penalty + Q-maximization term	1

Notes. The BC flow policy and refinement head share the same four-layer MLP scale as the DeFlow/FQL-style backbone. The main architectural difference is not a larger network, but the replacement of the isotropic residual penalty by a trace-normalized Fisher quadratic form estimated at $t_\varepsilon = 0.8$.

Table 7: Task-specific hyperparameters for offline RL. Prior columns follow the original task-specific settings reported in FQL and reused by DeFlow. We append our FiDec variant as the final column. DeFlow explicitly states that Q-normalization is enabled; we follow the same paper-level setting here. The only task-sensitive additional hyperparameter for the FiDec variant is the Fisher estimation time t_ε, fixed to 0.8 in our final experiments.

Task(s)	IQL α	ReBRAC (α_1, α_2)	IDQL N	SRPO β	CAC η	FAWAC α	FBRAC α	IFQL N	FQL α	DeFlow δ	Fisher-Decorator t_ε
antmaze-large-navigate-singletask-task{1,2,3,4,5}-v0	10	(0.003, 0.01)	32	0.3	1	3	3	32	10	0.01	0.8
antmaze-giant-navigate-singletask-task{1,2,3,4,5}-v0	10	(0.003, 0.01)	32	0.3	1	3	10	32	10	0.01	0.8
humanoidmaze-medium-navigate-singletask-task{1,2,3,4,5}-v0	10	(0.01, 0.01)	32	0.3	0.03	3	30	32	30	0.001	0.8
humanoidmaze-large-navigate-singletask-task{1,2,3,4,5}-v0	10	(0.01, 0.01)	32	0.3	1	3	30	32	30	0.001	0.8
antsoccer-arena-navigate-singletask-task{1,2,3,4,5}-v0	1	(0.01, 0.01)	32	0.03	1	10	30	64	10	0.01	0.8
cube-single-play-singletask-task{1,2,3,4,5}-v0	1	(1, 0)	32	0.03	0.003	1	100	32	300	0.001	0.8
cube-double-play-singletask-task{1,2,3,4,5}-v0	0.3	(0.1, 0)	32	0.1	0.3	0.3	100	32	300	0.001	0.8
scene-play-singletask-task{1,2,3,4,5}-v0	10	(0.1, 0.01)	32	0.1	0.3	0.3	100	32	300	0.001	0.8
puzzle-3x3-play-singletask-task{1,2,3,4,5}-v0	10	(0.3, 0.01)	32	0.1	0.01	0.3	100	32	1000	0.0005	0.8
puzzle-4x4-play-singletask-task{1,2,3,4,5}-v0	3	(0.3, 0.01)	32	0.1	0.01	0.3	300	32	1000	0.0005	0.8
antmaze-umaze-v2 antmaze-umaze-diverse-v2 antmaze-medium-play-v2 antmaze-medium-diverse-v2	-	-	-	-	0.01	3	10	32	10	0.015	0.8
antmaze-large-play-v2	-	-	-	-	4.5	3	1	32	3	0.015	0.8
antmaze-large-diverse-v2	-	-	-	-	3.5	3	1	32	3	0.015	0.8
pen-human-v1	-	-	32	0.03	0.003	0.03	30000	32	10000	0.01	0.8
pen-cloned-v1	-	-	32	0.1	0.003	0.3	10000	32	10000	0.01	0.8
pen-expert-v1	-	-	32	0.1	0.03	0.1	30000	32	3000	0.01	0.8
door-human-v1	-	-	32	0.01	0.03	1	30000	32	30000	0.001	0.8
door-cloned-v1	-	-	32	0.03	0.03	1	10000	128	30000	0.001	0.8
door-expert-v1	-	-	32	0.01	0.03	3	30000	32	30000	0.001	0.8
hammer-human-v1	-	-	128	0.1	0.03	3	30000	32	30000	0.001	0.8
hammer-cloned-v1	-	-	32	0.1	0.003	0.03	10000	32	10000	0.001	0.8
hammer-expert-v1	-	-	32	0.03	0.03	3	30000	32	30000	0.001	0.8
relocate-human-v1	-	-	32	0.03	0.01	0.3	30000	128	10000	0.001	0.8
relocate-cloned-v1	-	-	64	0.03	0.01	0.1	3000	32	30000	0.001	0.8
relocate-expert-v1	-	-	32	0.01	0.003	1	30000	32	30000	0.001	0.8
visual-cube-single-play-singletask-task1-v0	1	(1, 0)	-	-	-	-	100	32	300	0.001	0.8
visual-cube-double-play-singletask-task1-v0	0.3	(0.1, 0)	-	-	-	-	100	32	100	0.001	0.8
visual-scene-play-singletask-task1-v0	10	(0.1, 0.01)	-	-	-	-	100	32	100	0.001	0.8
visual-puzzle-3x3-play-singletask-task1-v0	10	(0.3, 0.01)	-	-	-	-	100	32	300	0.0005	0.8
visual-puzzle-4x4-play-singletask-task1-v0	3	(0.3, 0.01)	-	-	-	-	300	32	300	0.0005	0.8

Notes. We follow the task-specific hyperparameter conventions used in FQL and DeFlow, and append our algorithm (FiDec) as an additional column. For the FiDec, the only task-sensitive additional hyperparameter is the Fisher estimation time t_ε, which is fixed to 0.8 across all offline tasks in our final experiments. Other Fisher-specific constants are treated as fixed implementation details and are therefore omitted from the main comparison table. For the hyperparameter ϵ, we keep it the same as in DeFlow for a fair comparison.

E.3 Additional Results

The additional examples in Figure 7 further illustrate that this advantage is not restricted to a single landscape geometry. In Figure 7(a), the asymmetric bimodal landscape with an off-center saddle shows that isotropic regularization still induces either averaging across modes or interpolation through low-value regions, whereas FiDec concentrates refinement on the favorable high-value mode. Figure 7(b) considers a curved support manifold with a detached hotspot: FQL is attracted toward the unsupported high-value region, DeFlow remains overly conservative on the support corridor, and FiDec advances farther along the valid manifold toward better actions. Figure 7(c) presents a more complex multi-lobe landscape, where the central basin and eccentric valley make isotropic updates even more ambiguous; again, FiDec preserves the meaningful modal structure while steering mass toward high-value regions. Finally, Figure 7(d) shows a crescent-like support geometry with multiple high-value tendencies, where FiDec consistently follows the data geometry and achieves support-preserving improvement, while FQL is prone to off-support attraction and DeFlow tends to under-exploit the available high-value directions. Taken together, these four examples highlight that the benefit of FiDec comes from geometry-aware anisotropic refinement rather than from any task-specific coincidence of the original two examples.

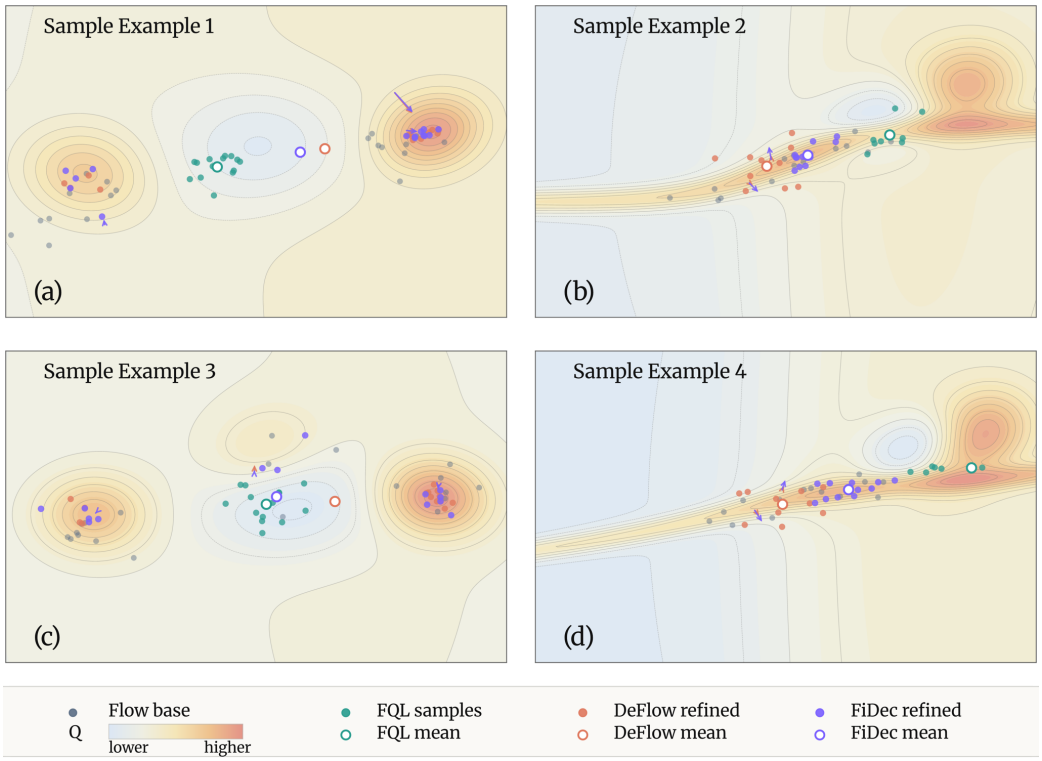

Figure 7: Additional examples of isotropic and anisotropic policy refinement. Panels (a) and (c) consider multimodal landscapes. Under isotropic regularization, FQL tends to average across modes, while DeFlow often interpolates through low-value regions or remains insufficiently biased toward favorable modes. In contrast, FiDec preserves multimodality while shifting mass toward higher-value regions by following the local geometry induced by the Fisher information matrix. Panels (b) and (d) consider policies supported on thin manifolds with detached or weakly supported high-value regions. In these cases, FQL is more easily attracted toward unsupported areas and exhibits out-of-distribution drift, whereas DeFlow remains comparatively conservative along the support and may fail to reach better regions. FiDec instead follows the support geometry and refines the policy toward higher-value valid regions. Together, these additional examples show that the advantage of FiDec is not tied to a single landscape, but persists across diverse multimodal and manifold-supported refinement scenarios.

Table 8: Full offline RL results corresponding to Table 2. We report all 73 OGBench and D4RL offline tasks. (*) indicates the default task in each environment.

	Gaussian Policies			Diffusion Policies			Flow Policies					
Task	BC	IQL	ReBRAC	IDQL	SRPO	CAC	FAWAC	FBRAC	IFQL	FQL	DeFlow	**FiDec**
antmaze-large-navigate-singletask-task1-v0 (*)	0 ±0	48 ±9	**91 ±10**	0 ±0	0 ±0	42 ±7	1 ±1	70 ±20	24 ±17	80 ±8	**87±3**	**88±2**
antmaze-large-navigate-singletask-task2-v0	6 ±3	42 ±6	**88 ±4**	14 ±8	4 ±4	1 ±1	0 ±1	35 ±12	8 ±3	57 ±10	75±6	80±2
antmaze-large-navigate-singletask-task3-v0	29 ±5	72 ±7	51 ±18	26 ±8	3 ±2	49 ±10	12 ±4	83 ±15	52 ±17	**93 ±3**	**94±4**	**90±3**
antmaze-large-navigate-singletask-task4-v0	8 ±3	51 ±9	**84 ±7**	62 ±25	45 ±19	17 ±6	10 ±3	37 ±18	18 ±8	80 ±4	62±23	**85±4**
antmaze-large-navigate-singletask-task5-v0	10 ±3	54 ±22	**90 ±2**	2 ±2	1 ±1	55 ±6	9 ±5	76 ±8	38 ±18	83 ±4	**86±4**	**89±2**
antmaze-giant-navigate-singletask-task1-v0 (*)	0 ±0	0 ±0	**27 ±22**	0 ±0	0 ±0	0 ±0	0 ±0	0 ±1	0 ±0	4 ±5	6±2	15±1
antmaze-giant-navigate-singletask-task2-v0	0 ±0	1 ±1	16 ±17	0 ±0	0 ±0	0 ±0	0 ±0	4 ±7	0 ±0	9 ±7	18±13	**19±4**
antmaze-giant-navigate-singletask-task3-v0	0 ±0	0 ±0	**34 ±22**	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±1	0±0	0±0
antmaze-giant-navigate-singletask-task4-v0	0 ±0	0 ±0	5 ±12	0 ±0	0 ±0	0 ±0	0 ±0	9 ±4	0 ±0	14 ±23	0±0	**21±14**
antmaze-giant-navigate-singletask-task5-v0	1 ±1	19 ±7	**49 ±22**	0 ±1	0 ±0	0 ±0	0 ±0	6 ±10	13 ±9	16 ±28	0±0	10±12
humanoidmaze-medium-navigate-singletask-task1-v0 (*)	1 ±0	32 ±7	16 ±9	1 ±1	0 ±0	38 ±19	6 ±2	25 ±8	69 ±19	19 ±12	35±4	**80±5**
humanoidmaze-medium-navigate-singletask-task2-v0	1 ±0	41 ±9	18 ±16	1 ±1	1 ±1	47 ±35	40 ±2	76 ±10	85 ±11	**94 ±3**	76±2	**92±1**
humanoidmaze-medium-navigate-singletask-task3-v0	6 ±2	25 ±5	36 ±13	0 ±1	2 ±1	**83 ±18**	19 ±2	27 ±11	49 ±49	74 ±18	75±3	**84±7**
humanoidmaze-medium-navigate-singletask-task4-v0	0 ±0	0 ±1	**15 ±16**	1 ±1	1 ±1	5 ±4	1 ±1	1 ±2	1 ±1	3 ±4	11±2	5±1
humanoidmaze-medium-navigate-singletask-task5-v0	2 ±1	66 ±4	24 ±20	1 ±1	3 ±3	91 ±5	31 ±7	63 ±9	**98 ±2**	**97 ±2**	88±4	**97±1**
humanoidmaze-large-navigate-singletask-task1-v0 (*)	0 ±0	3 ±1	2 ±1	0 ±0	0 ±0	1 ±1	0 ±0	0 ±1	6 ±2	**7 ±6**	5±1	6±1
humanoidmaze-large-navigate-singletask-task2-v0	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0±0**	**0±0**
humanoidmaze-large-navigate-singletask-task3-v0	1 ±1	7 ±3	8 ±4	3 ±1	1 ±1	2 ±3	1 ±1	10 ±2	**48 ±10**	11 ±7	13±5	17±1
humanoidmaze-large-navigate-singletask-task4-v0	1 ±0	1 ±0	1 ±1	0 ±0	0 ±0	0 ±1	0 ±0	0 ±0	1 ±1	2 ±3	5±1	**7±4**
humanoidmaze-large-navigate-singletask-task5-v0	0 ±1	1 ±1	2 ±2	0 ±0	0 ±0	0 ±0	0 ±0	1 ±1	0 ±0	1 ±3	0±0	**10±13**
antsoccer-arena-navigate-singletask-task1-v0	2 ±1	14 ±5	0 ±0	44 ±12	2 ±1	1 ±3	22 ±2	17 ±3	61 ±25	77 ±4	84±4	**89±4**
antsoccer-arena-navigate-singletask-task2-v0	2 ±2	17 ±7	0 ±1	15 ±12	3 ±1	0 ±0	8 ±1	8 ±2	75 ±3	88 ±3	87±2	**94±2**
antsoccer-arena-navigate-singletask-task3-v0	0 ±0	6 ±4	0 ±0	0 ±0	0 ±0	8 ±19	11 ±5	16 ±3	14 ±22	**61 ±6**	56±5	56±2
antsoccer-arena-navigate-singletask-task4-v0 (*)	1 ±0	3 ±2	0 ±0	0 ±1	0 ±0	0 ±0	12 ±3	24 ±4	16 ±9	39 ±6	43±6	**48±6**
antsoccer-arena-navigate-singletask-task5-v0	0 ±0	2 ±2	0 ±0	0 ±0	0 ±0	0 ±0	9 ±2	15 ±4	0 ±1	36 ±9	**41±5**	32±5
cube-single-play-singletask-task1-v0	10 ±5	88 ±3	89 ±5	**95 ±2**	89 ±7	77 ±28	81 ±9	73 ±33	79 ±4	**97 ±2**	91±2	**96±1**
cube-single-play-singletask-task2-v0	3 ±1	85 ±8	92 ±4	**96 ±2**	82 ±16	80 ±30	81 ±9	83 ±13	73 ±3	**97 ±2**	**96±1**	**96±1**
cube-single-play-singletask-task3-v0	9 ±3	91 ±5	93 ±3	**99 ±1**	**96 ±2**	**98 ±1**	87 ±4	82 ±12	88 ±4	**98 ±2**	**97±1**	**98±1**
cube-single-play-singletask-task4-v0	2 ±1	73 ±6	**92 ±3**	**93 ±4**	70 ±18	**91 ±2**	79 ±6	79 ±20	79 ±6	**94 ±3**	87±2	89±2
cube-single-play-singletask-task5-v0	3 ±3	78 ±9	87 ±8	**90 ±6**	61 ±12	80 ±20	78 ±10	76 ±33	77 ±7	**93 ±3**	80±6	**90±1**
cube-double-play-singletask-task1-v0 (*)	8 ±3	27 ±5	45 ±6	39 ±19	7 ±6	21 ±8	21 ±7	47 ±11	35 ±9	61 ±9	**67±10**	**69±10**
cube-double-play-singletask-task2-v0 (*)	0 ±0	1 ±1	7 ±3	16 ±10	0 ±0	2 ±2	2 ±1	22 ±12	9 ±5	36 ±6	53±3	**58±9**
cube-double-play-singletask-task3-v0	0 ±0	0 ±0	4 ±1	17 ±8	0 ±1	3 ±1	1 ±1	4 ±2	8 ±5	22 ±5	40±8	**51±13**
cube-double-play-singletask-task4-v0	0 ±0	0 ±0	1 ±1	0 ±1	0 ±0	0 ±1	0 ±0	0 ±1	1 ±1	5 ±2	3±2	**6±1**
cube-double-play-singletask-task5-v0	0 ±0	4 ±3	4 ±2	1 ±1	0 ±0	3 ±2	2 ±1	2 ±2	17 ±6	19 ±10	28±3	**32±7**
scene-play-singletask-task1-v0	19 ±6	94 ±3	**95 ±2**	**100 ±0**	94 ±4	**100 ±1**	87 ±8	**96 ±8**	**98 ±3**	**100 ±0**	**98±1**	**100±0**
scene-play-singletask-task2-v0 (*)	1 ±1	12 ±3	50 ±13	33 ±14	2 ±2	50 ±40	18 ±8	46 ±10	0 ±0	**76 ±9**	59±5	72±5
scene-play-singletask-task3-v0	1 ±1	32 ±7	55 ±16	**94 ±4**	4 ±4	49 ±16	38 ±9	78 ±14	54 ±19	**98 ±1**	88±4	**96±1**
scene-play-singletask-task4-v0	2 ±2	0 ±1	3 ±3	4 ±3	0 ±0	0 ±0	6 ±1	4 ±4	0 ±0	5 ±1	8±3	**12±3**
scene-play-singletask-task5-v0	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0±0**	**0±0**
puzzle-3x3-play-singletask-task1-v0	5 ±2	33 ±6	**97 ±4**	52 ±12	89 ±5	**97 ±2**	25 ±9	63 ±19	94 ±3	90 ±4	**99±1**	**96±2**
puzzle-3x3-play-singletask-task2-v0	1 ±1	4 ±3	1 ±1	0 ±1	0 ±1	0 ±0	4 ±2	2 ±2	1 ±2	16 ±5	0±0	**37±2**
puzzle-3x3-play-singletask-task3-v0	1 ±1	3 ±2	3 ±1	0 ±0	0 ±0	0 ±0	1 ±0	1 ±1	0 ±0	10 ±3	0±0	**28±3**
puzzle-3x3-play-singletask-task4-v0 (*)	1 ±1	2 ±1	2 ±1	0 ±0	0 ±0	0 ±0	1 ±1	2 ±2	0 ±0	16 ±5	3±3	**24±1**
puzzle-3x3-play-singletask-task5-v0	1 ±0	3 ±2	5 ±3	0 ±0	0 ±0	0 ±0	1 ±1	2 ±2	0 ±0	16 ±3	20±10	**30±2**
puzzle-4x4-play-singletask-task1-v0	1 ±1	12 ±2	26 ±4	**48 ±5**	24 ±9	44 ±10	1 ±2	32 ±9	**49 ±9**	34 ±8	7±1	38±3
puzzle-4x4-play-singletask-task2-v0	0 ±0	7 ±4	12 ±4	14 ±5	0 ±1	0 ±0	0 ±1	5 ±3	4 ±4	**16 ±5**	3±1	15±2
puzzle-4x4-play-singletask-task3-v0	0 ±0	9 ±3	15 ±3	34 ±5	21 ±10	29 ±12	1 ±1	20 ±10	**50 ±14**	18 ±5	4±2	8±2
puzzle-4x4-play-singletask-task4-v0 (*)	0 ±0	5 ±2	10 ±3	**26 ±6**	7 ±4	1 ±1	0 ±0	5 ±1	21 ±11	11 ±3	4±1	6±2
puzzle-4x4-play-singletask-task5-v0	0 ±0	4 ±1	7 ±3	**24 ±11**	1 ±1	0 ±0	0 ±1	4 ±3	2 ±2	7 ±3	2±1	9±1
antmaze-umaze-v2	55	77	**98**	94	**97**	66 ±5	90 ±6	94 ±3	92 ±6	**96 ±2**	**97±3**	**99±1**
antmaze-umaze-diverse-v2	47	54	84	80	82	66 ±11	55 ±7	82 ±9	62 ±12	**89 ±5**	77±3	79±3
antmaze-medium-play-v2	0	66	**90**	84	81	49 ±24	52 ±12	77 ±7	56 ±15	78 ±7	78±7	83±2
antmaze-medium-diverse-v2	1	74	84	85	75	0 ±1	44 ±15	77 ±6	60 ±25	71 ±13	77±3	**91±4**
antmaze-large-play-v2	0	42	52	64	54	0 ±0	10 ±6	32 ±21	55 ±9	**84 ±7**	79±4	**82±6**
antmaze-large-diverse-v2	0	30	64	68	54	0 ±0	16 ±10	20 ±17	64 ±8	**83 ±4**	78±4	**86±3**
pen-human-v1	71	78	**103**	76 ±10	69 ±7	64 ±8	67 ±5	77 ±7	71 ±12	53 ±6	65±7	41±6
pen-cloned-v1	52	83	**103**	64 ±7	61 ±7	56 ±10	62 ±10	67 ±9	80 ±11	74 ±11	72±6	94±12
pen-expert-v1	110	128	**152**	140 ±6	134 ±4	103 ±9	118 ±6	119 ±7	139 ±5	142 ±6	136±5	132±2
door-human-v1	2	3	-0	6 ±2	3 ±3	5 ±2	2 ±1	4 ±2	**7 ±2**	0 ±0	0±0	0±0
door-cloned-v1	-0	**3**	0	0 ±0	0 ±0	1 ±0	0 ±1	0 ±0	2 ±2	2 ±1	0±0	0±0
relocate-cloned-v1	-0	0	**2**	-0 ±0	-0 ±0	-0 ±0	-0 ±0	1 ±1	-0 ±0	-0 ±0	0±0	0±0
door-expert-v1	**105**	**107**	**106**	**105 ±1**	**105 ±0**	98 ±3	**103 ±1**	**104 ±1**	**104 ±2**	**104 ±1**	32±17	88±2
hammer-human-v1	**3**	2	0	2 ±1	1 ±1	2 ±0	2 ±1	2 ±1	**3 ±1**	1 ±1	0±0	1±0
hammer-cloned-v1	1	2	5	2 ±1	2 ±1	1 ±1	1 ±0	2 ±1	2 ±1	**11 ±9**	6±3	7±2
hammer-expert-v1	127	**129**	**134**	125 ±4	127 ±0	92 ±11	118 ±3	119 ±9	117 ±9	125 ±3	29±12	92±12
relocate-human-v1	**0**	**0**	**0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0 ±0**	**0±0**	**0±0**
relocate-expert-v1	**108**	**106**	**108**	**107 ±1**	**106 ±2**	93 ±6	**105 ±3**	**105 ±2**	**104 ±3**	**107 ±1**	68±17	97±5
visual-cube-single-play-singletask-task1-v0[1]	-	70 ±12	**83 ±6**	-	-	-	-	55 ±8	49 ±7	81 ±12	71±7	**86±2**
visual-cube-double-play-singletask-task1-v0[1]	-	**34 ±23**	4 ±4	-	-	-	-	6 ±2	8 ±6	21 ±11	7±4	15±7
visual-scene-play-singletask-task1-v0[1]	-	**97 ±2**	**98 ±4**	-	-	-	-	46 ±4	86 ±10	**98 ±3**	**99±1**	**98±1**
visual-puzzle-3x3-play-singletask-task1-v0[1]	-	7 ±15	88 ±4	-	-	-	-	7 ±2	**100 ±0**	94 ±1	**97±1**	**96±1**
visual-puzzle-4x4-play-singletask-task1-v0[1]	-	0 ±0	26 ±6	-	-	-	-	0 ±0	8 ±15	**33 ±6**	22±7	23±1

[1] Due to the high computational cost of pixel-based tasks, only six strong methods are benchmarked on the visual OGBench tasks, following FQL/DeFlow.

Table 9: Offline-to-online RL results. Results are averaged over 8 seeds.

Task	IQL	ReBRAC	Cal-QL	RLPD	IFQL	FQL	DeFlow	**FiDec**
humanoidmaze-medium-navigate-singletask-v0	21 ±13→16 ±8	16 ±20→1 ±1	0 ±0→0 ±0	0 ±0→8 ±10	56 ±35→82 ±20	12 ±7→22 ±12	36±7→97±3	**82±4→99±2**
antsoccer-arena-navigate-singletask-v0	2 ±1→0 ±0	0 ±0→0 ±0	0 ±0→0 ±0	0 ±0→0 ±0	26 ±15→39 ±10	28 ±8→86 ±5	40±12→87±1	**51±12→91±4**
cube-double-play-singletask-v0	0 ±1→0 ±0	6 ±5→28 ±28	0 ±0→0 ±0	0 ±0→0 ±0	12 ±9→40 ±5	**40 ±11→92 ±3**	49±6→83±7	61±9→87±2
scene-play-singletask-v0	14 ±11→10 ±9	**55 ±10→100 ±0**	1 ±2→50 ±53	**0 ±0→100 ±0**	0 ±1→60 ±39	82 ±11→100 ±1	35±19→100±0	**65±14→100±0**
puzzle-4x4-play-singletask-v0	5 ±2→1 ±1	8 ±4→14 ±35	0 ±0→0 ±0	0 ±0→100 ±1	23 ±6→19 ±33	8 ±3→38 ±52	3±1→100±0	**7±3→100±0**
antmaze-umaze-v2	77→96	98→75	77→100	0 ±0→98 ±3	94 ±5→96 ±2	97 ±2→99 ±1	88±2→99±1	**97±1→100±0**
antmaze-umaze-diverse-v2	60→64	**74→98**	32→98	0 ±0→94 ±5	69 ±20→93 ±5	79 ±16→100 ±1	**79±9→99±1**	**62±6→100±0**
antmaze-medium-play-v2	72→90	88→98	**72→99**	0 ±0→98 ±2	52 ±19→93 ±2	77 ±7→97 ±2	78±4→98±0	83±2→98±2
antmaze-medium-diverse-v2	64→92	85→99	62→98	0 ±0→97 ±2	44 ±26→89 ±4	55 ±19→97 ±3	66±6→97±2	**81±2→98±2**
antmaze-large-play-v2	38→64	68→32	32→97	0 ±0→93 ±5	64 ±14→80 ±5	66 ±40→84 ±30	**67±10→95±2**	**71±10→98±2**
antmaze-large-diverse-v2	27→64	67→72	44→92	0 ±0→94 ±3	69 ±6→86 ±5	75 ±24→94 ±3	72±3→94±2	**76±3→95±2**
pen-cloned-v1	84→102	74→138	-3→3	3 ±2→120 ±10	77 ±7→107 ±10	**53 ±14→149 ±6**	67±7→130±4	83±6→143±7
door-cloned-v1	1→20	**0→102**	-0→0	0±0→102 ±7	3 ±2→50 ±15	0 ±0→102 ±5	1±0→99±1	**0±0→104±2**
hammer-cloned-v1	1→57	**7→125**	0→0	**0 ±0→128 ±29**	4 ±2→60 ±14	0 ±0→127 ±17	4±3→120±1	7±3→125±4

Table 10: Ablation (full table for Figure 5) on the perturbed time t_ε over OGBench-5. Results are reported as mean $\pm$ std over three runs for each task.

Env	DeFlow (isotropic L^2 constraint)	t_ε=0.70	t_ε=0.75	t_ε=0.80	t_ε=0.85	t_ε=0.90	t_ε=0.95
antsoccer-arena-navigate-singletask-task4	0.43±0.08	0.54±0.11	0.52±0.03	0.49±0.04	0.47±0.05	0.36±0.05	0.03±0.01
cube-double-play-singletask-task2	0.50±0.03	0.57±0.13	0.51±0.03	0.58±0.02	0.47±0.06	0.45±0.04	0.62±0.06
humanoidmaze-medium-navigate-singletask-task1	0.35±0.07	0.64±0.08	0.77±0.08	0.76±0.07	0.56±0.12	0.16±0.07	0.02±0.02
puzzle-4x4-play-singletask-task4	0.04±0.01	0.03±0.02	0.03±0.00	0.05±0.01	0.07±0.02	0.09±0.01	0.04±0.02
scene-play-singletask-task2	0.51±0.06	0.61±0.01	0.62±0.09	0.70±0.07	0.71±0.06	0.73±0.01	0.77±0.04
Avg	**0.36±0.17**	**0.48±0.23**	**0.49±0.25**	**0.51±0.25**	**0.45±0.21**	**0.36±0.23**	**0.30±0.33**