

FROM SUBOPTIMAL TO OPTIMAL: ONE-STEP LATENT FLOW POLICIES FOR OFFLINE REINFORCEMENT LEARNING

A PREPRINT

Shenghua Lu*, Zixuan Yang*, Rongxu Zhao, Bohao Qu, Jifeng Hu†, Hechang Chen†

Jilin University

ABSTRACT

Offline reinforcement learning aims to learn effective policies from fixed datasets without environment interaction. Although generative models such as diffusion and flow matching enhance policy expressiveness, they typically start from uninformative Gaussian noise and generate actions through multi-step iteration, leading to high computational costs and failing to effectively leverage suboptimal trajectories in datasets. We propose MeanFlow2Better (MF2B), which reformulates offline RL as generative decision chunk optimization from current performance status to better performance status in a high-level latent decision space. MF2B maps trajectory chunks to high-level latent decision representations through an encoder, placing generation starting points within a structured decision space sculpted by behavioral data. The model learns a shared conditional mean velocity field in this space, constructing two flow mappings: from Gaussian-Induced Decision Sources toward positive behavioral decision regions, and from suboptimal decisions toward high-value decision regions. Both flow mappings complete decision refinement through one-step ODE integration over the unit time interval $\Delta t = 1$, requiring neither multi-step iterative sampling nor backpropagation through time. Furthermore, we propose hierarchical chunk-level advantage to guide the flow direction toward high-return regions, where the high-level decisions cover multiple low-level actions, leading to sequence length compression from T to $K \approx T/H$ and significantly lower computational overhead. On the D4RL Gym-MuJoCo benchmark, MF2B achieves an average score of 90.91, matching or surpassing strong baselines.

1 Introduction

Reinforcement learning (RL) has achieved remarkable progress across diverse domains, including robotics[Zhu et al., 2023, Hwangbo et al., 2019], games[Schrittwieser et al., 2020, Perolat et al., 2022], and fundamental science[Jumper et al., 2021, Ren et al., 2018]. In many real-world settings such as robotics[Zhu et al., 2023, Ma et al., 2026], healthcare[Yu et al., 2021a], and autonomous driving[Li et al., 2023], however, online interaction is expensive, unsafe, or infeasible. Offline RL addresses this by learning solely from pre-collected datasets[Huang et al., 2024], but faces two persistent challenges: limited policy expressiveness as behavioral distributions grow complex and multimodal[Kang et al., 2023], and the underutilization of suboptimal trajectories that constitute the majority of practical datasets.

Generative policies based on diffusion[Wang et al., 2022] and flow matching[Park et al., 2025] improve expressiveness, yet share two structural limitations. First, they generate behaviors through multi-step denoising or ODE integration, incurring high inference costs. Second, they model policies directly in raw action space, treating trajectories of different qualities as homogeneous samples[Fujimoto and Gu, 2021, Kostrikov et al., 2021] and offering no mechanism to lift suboptimal behaviors toward high-value regions.

We argue that suboptimal trajectories are not pure noise but starting points for improvement. Drawing on the neural null-space mechanism in motor cortex [Stavisky et al., 2017], we design Gaussian-Induced Decision Sources as a semantic and value null-space origin from which structured flow learning sculpts task-relevant decision representations.

*These authors contributed equally to this work

†Corresponding authors

Hierarchical behavior theory [Sutton et al., 1999a, Botvinick et al., 2009] motivates our chunk-level latent decision representation, and value correction principles [Sutton and Barto, 2018] inspire the hierarchical chunk-level advantage-guided flow mapping toward high-value regions.

We propose MF2B (MeanFlow2Better, MF2B), which reformulates offline RL as generative decision chunk optimization from current performance status to better performance status in a high-level latent decision space. A Chunk-Level Latent Decision Encoder compresses trajectory chunks into high-level latent representations, reducing sequence length from T to $K \approx T/H$. A shared conditional mean velocity field learns across two branches: one transporting Gaussian-Induced Decision Sources toward positive behavioral regions, another transporting suboptimal decisions toward high-value regions. Hierarchical chunk-level advantage estimation guides both branches toward high-return areas. Unlike multi-step diffusion or flow models, MF2B predicts the conditional mean velocity over the unit interval $\Delta t = 1$ and completes refinement in a single Euler step. Evaluation on the D4RL Gym-MuJoCo benchmark [Fu et al., 2020] demonstrates an average score of 90.91, surpassing all one-step flow policy baselines.

Our main contributions are:

- We propose MF2B, which reformulates offline RL as generative decision chunk optimization from suboptimal to high-value in a high-level latent decision space, realized through a dual-branch conditional mean velocity field with hierarchical chunk-level advantage guidance.
- We propose a one-step decision refinement mechanism based on conditional mean velocity fields, replacing multi-step ODE integration with a single Euler step over the unit interval $\Delta t = 1$.
- Evaluations on 9 D4RL Gym-MuJoCo tasks demonstrate an average score of 90.91, establishing state-of-the-art performance among one-step flow policies.

2 Related Work

Offline Reinforcement Learning. Reinforcement learning is inherently an online paradigm [Sutton et al., 1999b], and its online learning characteristics present limitations in scenarios where online data collection and learning cannot occur simultaneously, such as autonomous driving. Offline reinforcement learning enables agents to learn from pre-collected fixed datasets without collecting new data during the learning process [Fujimoto et al., 2019, Kumar et al., 2020, Yu et al., 2021b, Kumar et al., 2019, Hu et al., 2022, 2023].

Diffusion Policies in Offline RL. Offline RL aims to learn effective policies from fixed datasets without environment interaction [Levine et al., 2020], with early methods addressing distributional shift via conservative objectives such as CQL [Kumar et al., 2020], TD3+BC [Fujimoto and Gu, 2021], and IQL [Kostrikov et al., 2021]. However, these approaches often rely on unimodal Gaussian policies, which struggle to model complex, multi-modal action distributions. To address this, recent approaches have adopted diffusion models [Ho et al., 2020, Song et al., 2020] for offline RL. Diffusion Q-learning (DQL) [Wang et al., 2022] replaces Gaussian policies with diffusion models, modeling complex multimodal action distributions through multi-step denoising. IDQL [Hansen-Estruch et al., 2023] and EDP [Kang et al., 2023] improve upon DQL but still share the critical limitation of high sampling costs from multi-step denoising.

Flow Matching and Flow Policies. Flow matching is a generative modeling approach that learns time-varying velocity fields through regressing target vector fields, which define probability paths transporting samples from simple prior distributions to target data distributions, characterized by simple training and fast inference [Lipman et al., 2023] (see Section 3 for details). Recently, flow matching has attracted attention in reinforcement learning due to its strong expressiveness and faster inference. FQL [Park et al., 2025] uses flow matching policies to approximate behavior policies and distills flow policies for efficient policy learning and inference. FlowQ [Alles et al., 2025] learns a conditional velocity field for flow policies by modeling energy-guided probability paths as Gaussian trajectories. However, these methods typically rely on simple Gaussian priors and multi-step sampling from flow policies, limiting representation richness and bringing additional computational overhead. To address computational bottlenecks from multi-step generative policies, MeanFlowQL [Wang et al., 2026] reformulates MeanFlow [Geng et al., 2025a] as a residual mapping that directly transforms noise into actions. SSCP [Koirala and Fleming, 2026] uses enhanced flow matching objective functions to predict direct completion vectors from intermediate samples, achieving one-shot action generation.

Unlike the above methods, this paper introduces MF2B, achieving paradigm shifts in three dimensions. First, at the representation space level, existing flow policies operate directly in raw action space [Park et al., 2025, Alles et al., 2025], requiring learning highly nonlinear mappings from Gaussian noise to complex action distributions. MF2B compresses trajectory chunks into high-level latent decision representations through an encoder, reformulating flow matching as short-range approximately linear flow mappings within decision space, thereby establishing foundations for one-step generation. Second, at the generation mechanism level, FQL [Park et al., 2025] and FlowQ [Alles et al., 2025]

learn marginal velocity fields requiring multi-step ODE integration, while MF2B learns mean velocity fields[Geng et al., 2025a] directly predicting residual directions from starting points to endpoints, achieving one-step high-fidelity generation. Third, at the optimization objective level, existing methods indiscriminately fit dataset behavior frequencies, while MF2B introduces a dual-branch flow architecture: the Source-to-Positive Alignment Mean-Flow Branch establishes baseline mappings from Gaussian-Induced Decision Sources to dataset decisions, and the Quality-Uplift Mean-Flow Branch achieves value-guided flow mapping from suboptimal decisions to high-value decisions through hierarchical chunk-level advantage weighting, with both branches sharing the same conditional mean velocity field parameters. This design enables the model to extract flow mapping patterns leading from suboptimal trajectories to optimal policies, breaking through dataset quality ceilings. To our knowledge, this is the first offline reinforcement learning method combining mean flow matching with value-guided dual-branch flow mapping in high-level latent decision space to exploit suboptimal-to-optimal data capabilities, achieving unity of expressiveness and computational efficiency.

3 Preliminaries

3.1 Offline Reinforcement Learning

Reinforcement learning (RL) is typically formalized as a Markov Decision Process (MDP), defined by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s' | s, a)$ denotes the transition probability of moving from s to s' given action a , and $R(s, a)$ denotes the reward function. The discount factor $\gamma \in [0, 1)$ governs the trade-off between immediate and future rewards. In RL, the objective is to learn a policy $\pi_\theta(a | s)$, parameterized by θ , that maximizes the expected discounted return

$$\mathbb{E}_\pi \left[\sum_{h=0}^{\infty} \gamma^h R(s_h, a_h) \right]. \quad (1)$$

To support policy learning, the action-value (Q) function is defined:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{h=0}^{\infty} \gamma^h R(s_h, a_h) \mid s_0 = s, a_0 = a \right], \quad (2)$$

which measures the expected cumulative return starting from state s and action a under policy π .

Offline RL is a setting of RL where the agent does not interact with the environment but instead learns from a fixed dataset of transitions $\mathcal{D} = \{(s_h, a_h, s_{h+1}, r_h)\}$. The challenge lies in learning an optimal policy solely from this static dataset that often contains suboptimal behavior, without any further exploration.

3.2 Implicit Q-Learning

Implicit Q-Learning (IQL)[Kostrikov et al., 2021] is an offline reinforcement learning method that learns upper-bound estimates of value functions through expectile regression, thereby avoiding out-of-distribution action evaluation. Unlike traditional Q-learning that directly fits Q-value means, IQL’s core innovation lies in using asymmetric expectile loss functions to train the state-value network $V(s)$.

IQL maintains two Q-networks $Q_{\omega,1}(s, a)$ and $Q_{\omega,2}(s, a)$, a state-value network $V_\nu(s)$, and corresponding target Q-networks. The value network is learned through expectile regression:

$$\mathcal{L}_V = \mathbb{E} [|\tau_{\text{exp}} - \mathbb{I}(\delta < 0)| \delta^2], \quad (3)$$

where $\delta = \min(Q_{\bar{\omega},1}(s, a), Q_{\bar{\omega},2}(s, a)) - V_\nu(s)$, and $\tau_{\text{exp}} \in (0.5, 1)$ is the expectile parameter (typically 0.7). This asymmetric loss makes $V_\nu(s)$ learn as a high expectile of the behavior value distribution rather than a simple average, thereby providing more conservative value estimates.

Based on this value baseline, an advantage function can be defined:

$$A(s, a) = \min(Q_{\bar{\omega},1}(s, a), Q_{\bar{\omega},2}(s, a)) - V_\nu(s). \quad (4)$$

This advantage function can identify relatively high-value state-action pairs in the dataset, providing value guidance signals for policy learning without explicit policy improvement steps.

3.3 Flow Matching

Flow matching is a generative modeling approach that adopts simple and intuitive training objectives by regressing target vector fields, which define desired probability paths[Lipman et al., 2023]. Flow matching learns a time-varying vector field by defining an ordinary differential equation (ODE), whose solution transports samples from a simple prior distribution to the target data distribution.

Given a data distribution $p(x)$, the goal is to learn a time-varying velocity field $v_\theta(x_t, t) : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$ parameterized by θ , where $x_t \in \mathbb{R}^d$ denotes a d -dimensional data sample and $t \in [0, 1]$ denotes time. This velocity field defines a continuous flow that transports samples from a simple prior distribution (e.g., unit Gaussian) at $t = 0$ to the target data distribution $p(x)$ at $t = 1$.

Flow Path Construction and Velocity Field. Flow paths can be constructed as $x_t = \alpha_t x_0 + \beta_t \epsilon$ [Park et al., 2025, Lipman et al., 2023], where α_t and β_t are predefined time-varying functions (common choice: $\alpha_t = 1 - t$, $\beta_t = t$), $x_0 \sim p(x)$ is a data sample, and $\epsilon \sim \mathcal{N}(0, I)$ is noise. The corresponding *conditional velocity* is given by $v_t(x_t | x_0) = \frac{d}{dt} x_t = \alpha'_t x_0 + \beta'_t \epsilon$, where $'$ denotes the time derivative with respect to t .

Standard flow matching learns the *marginal velocity field* $v(x_t, t) \triangleq \mathbb{E}_{p_t(x_0|x_t)}[v_t(x_t|x_0)]$ via the Conditional Flow Matching (CFM) loss [Lipman et al., 2023], which requires multi-step ODE integration during generation. In contrast, the *mean velocity field*[Geng et al., 2025b] directly models the average velocity between two timesteps, defined as:

$$u(x_t, r, t) \triangleq \frac{1}{t-r} \int_r^t v(x_\tau, \tau) d\tau, \quad (5)$$

where u denotes the average velocity from time r to time t , and v denotes the marginal velocity. As $r \rightarrow t$, u converges to v . The advantage of the mean velocity field is that it can directly predict residual directions from starting points to endpoints, achieving one-step high-fidelity generation without iterative ODE integration. By learning the mean velocity field $u_\theta(x_t, r, t)$, samples can be directly generated through a one-step update $x_1 = x_0 + u_\theta(x_0, 0, 1)$ over the unit time interval $\Delta t = 1$.

Training Objective for Mean Velocity Field. The mean velocity field u_θ is optimized through the mean flow identity. Defining the average velocity as $u(x_t, r, t) = \frac{1}{t-r} \int_r^t v(x_\tau, \tau) d\tau$, by taking the time derivative and applying the chain rule, the mean flow identity can be obtained[Geng et al., 2025b]:

$$u(x_t, r, t) = v(x_t, t) - (t-r) (v(x_t, t) \cdot \partial_{x_t} u + \partial_t u), \quad (6)$$

where $\partial_{x_t} u$ and $\partial_t u$ denote partial derivatives of u with respect to x_t and t respectively. This identity relates the marginal velocity v to the mean velocity u , enabling construction of supervision targets for mean velocity through the known conditional velocity $v_t(x_t | x_1) = x_1 - \epsilon$. Training minimizes:

$$\mathcal{L}_{\text{MF}}(\theta) = \mathbb{E}_{\substack{0 \leq r < t \leq 1, \\ \epsilon \sim \mathcal{N}(0, I), \\ x_1 \sim p(x)}} \left[\|u_\theta(x_t, r, t) - u^*(x_t, r, t | x_1)\|_2^2 \right], \quad (7)$$

where $u^*(x_t, r, t | x_1)$ is computed from the conditional velocity through Equation (6).

Sampling and Generation with Mean Velocity Field. Given the mean velocity field $u_\theta(x_t, r, t)$, samples can be directly generated through a one-step update:

$$x_1 = x_0 + u_\theta(x_0, 0, 1), \quad (8)$$

where $x_0 = \epsilon \sim \mathcal{N}(0, I)$ is prior noise. The mean velocity field directly predicts residual directions from starting points to endpoints, achieving one-step high-fidelity generation without iterative ODE integration, significantly reducing inference costs.

4 MeanFlow2Better Method

MF2B reformulates offline reinforcement learning as a quality improvement problem in high-level latent decision space. Trajectory chunks are compressed into high-level latent decision representations through a Chunk-Level Latent Decision Encoder q_ϕ , a shared Conditional Latent Decision Mean-Flow Field v_θ learns conditional mean velocities across two supervisory branches, and a Hierarchical Action Decoder p_ψ recovers continuous action sequences from refined decisions. A Hierarchical Chunk-Level Advantage Critic $\mathcal{C}_\eta$ provides value guidance during training only. Throughout this section, we fix $t_0 = 0$ and $t_1 = 1$ as the start and end of the unit flow interval. Algorithm 1 summarizes the complete pipeline.

4.1 Chunk-Level High-Level Latent Decision Representation

Given an offline dataset $\mathcal{D} = \{\tau_i\}_{i=1}^N$, each trajectory $\tau_i = (s_0, a_0, r_0, \dots, s_{T-1}, a_{T-1}, r_{T-1})$ is segmented into consecutive chunks of length H :

$$\tau_k = (s_{k,0:H-1}, a_{k,0:H-1}), \quad (9)$$

yielding $K \approx T/H$ chunks per trajectory. The encoder q_ϕ maps each chunk to a distribution over high-level latent decisions:

$$q_\phi(\tau_k) = [\mu_k, \log \sigma_k^2] \in \mathbb{R}^{2d_z}, \quad (10)$$

and the full decision sequence is $Z = [z_1, \dots, z_K] \in \mathbb{R}^{K \times 2d_z}$. The mean-flow field operates on the concatenated $[\mu, \log \sigma^2]$ representation directly. The decoder samples via reparameterization, $z_k = \mu_k + \exp(\frac{1}{2} \log \sigma_k^2) \odot \epsilon_k$, and generates actions as $\hat{a}_{k,0:H-1} = p_\psi(s_{k,0:H-1}, z_k)$. This separation of high-level behavioral intention from low-level action execution allows the flow model to perform behavioral improvement in a compact, stable decision space rather than over raw action sequences.

4.2 High-Level Decision Source Construction

4.2.1 Gaussian-Induced Decision Source

A Gaussian pseudo-trajectory $\tau_G^- \sim \mathcal{N}(0, I)$ is drawn with the same temporal shape as real trajectories. **Inspired by the neural null-space mechanism** [Stavisky et al., 2017], we **explicitly set its reward, RTG, chunk-level RTG, and trajectory return to zero**, placing Gaussian-Induced Decision Sources at the zero point of the value dimension. After encoding:

$$Z_{\text{neg}} = q_\phi(\tau_G^-). \quad (11)$$

The Source-to-Positive Alignment Mean-Flow Branch transports Z_{neg} toward positive behavioral decision regions, establishing task-relevant structure from a value-neutral baseline unconstrained by any specific return interval in the dataset.

4.2.2 Suboptimal-to-High-Value Trajectory Pairing

Trajectories are partitioned into a high-value pool $\mathcal{I}_{\text{top}}$ and a suboptimal pool $\mathcal{I}_{\text{low}}$ by sorting on complete return $R_i = \sum_t r_{i,t}$. The high-value timestep coverage ratio $\rho_{\text{traj}} \in (0, 1]$ controls the timestep budget of $\mathcal{I}_{\text{top}}$. Suboptimal and high-value latent decisions are then obtained by encoding trajectories sampled from each pool:

$$Z_{\text{low}} = q_\phi(\tau_{\text{low}}), \quad Z_{\text{top}} = q_\phi(\tau_{\text{top}}). \quad (12)$$

The Quality-Uplift Mean-Flow Branch learns to transport Z_{low} toward Z_{top} , distilling high-value decision structure from existing suboptimal experiences.

4.3 Dual-Branch Conditional Latent Decision Mean Flow Mapping

Both branches share the Conditional Latent Decision Mean-Flow Field $v_\theta(Z, t_0, t_1, c)$ but differ in source, target, and condition. Return-sorted pairing ensures the model learns mean velocities aligned with return improvement rather than arbitrary intra-distribution variation.

Each branch receives chunk-level and global conditions. The local condition for the k -th chunk is the target trajectory’s return-to-go:

$$c_{\text{chunk}}^{(b,k)} = \sum_{t=kH}^{T-1} r_{\text{top},t}^{(b)}, \quad (13)$$

and the global condition is the return gap:

$$c_{\text{global}}^{(b)} = R_{\text{top}}^{(b)} - R_{\text{low}}^{(b)}. \quad (14)$$

For the Source-to-Positive Alignment branch, the pair is $(\tau_{\text{neg}}, \tau_{\text{pos}})$ with $R_{\text{neg}} = 0$, so $c_{\text{global}}^{(b)} = R_{\text{pos}}^{(b)}$. Both conditions are independently projected and injected into chunk tokens and a global condition token.

4.3.1 Source-to-Positive Alignment Mean-Flow Branch

The branch predicts the mean velocity transporting Z_{neg} to Z_{pos} over the unit interval $[t_0, t_1]$:

$$\hat{U}_{\text{flow}}^{(b)} = v_\theta(Z_{\text{neg}}^{(b)}, t_0, t_1, c_{\text{flow}}^{(b)}), \quad U_{\text{flow}}^{*(b)} = \frac{Z_{\text{pos}}^{(b)} - Z_{\text{neg}}^{(b)}}{t_1 - t_0}. \quad (15)$$

4.3.2 Quality-Uplift Mean-Flow Branch

The branch predicts the mean velocity transporting Z_{low} to Z_{top} over the unit interval $[t_0, t_1]$:

$$\hat{U}_{\text{main}}^{(b)} = v_{\theta} \left(Z_{\text{low}}^{(b)}, t_0, t_1, c_{\text{main}}^{(b)} \right), \quad U_{\text{main}}^{*(b)} = \frac{Z_{\text{top}}^{(b)} - Z_{\text{low}}^{(b)}}{t_1 - t_0}. \quad (16)$$

Refined high-level decisions are obtained by one-step integration over $[t_0, t_1]$:

$$Z_{\text{plus}}^{(b)} = Z_{\text{low}}^{(b)} + (t_1 - t_0) \hat{U}_{\text{main}}^{(b)}, \quad (17)$$

which are decoded to reconstruct action sequences for the target high-value trajectory. This branch directly instantiates the core hypothesis: learnable improvement directions exist between suboptimal and high-value behaviors in high-level decision space.

4.3.3 High-Level Decision Transformer Architecture

The mean-flow field v_{θ} is a Transformer with bidirectional multi-head self-attention over the chunk sequence, taking $Z \in \mathbb{R}^{K \times 2d_z}$ as input tokens. Unlike the causal encoder and decoder, the flow field uses full attention so every chunk token can attend to the complete high-level decision sequence, enabling globally coherent mean velocity prediction. The attention sequence length $K \approx T/H$ compresses the original T -step horizon into K decision units.

4.4 Hierarchical Chunk-Level Latent Decision Value Estimation and Advantage Guidance

To avoid indiscriminate fitting to dataset behavior frequency, a hierarchical chunk-level advantage critic provides value guidance. Following IQL [Kostrikov et al., 2021] (see Section 3), the critic operates on chunk-level transitions $(s_{b,k}, z_{b,k}, r_{b,k}^{\text{chunk}}, s_{b,k+1})$ rather than primitive actions or full trajectories, and is used only during training.

The Bellman target is:

$$y_{b,k} = r_{b,k}^{\text{chunk}} + \gamma_{\text{boot}} V_{\nu}(s_{b,k+1}), \quad (18)$$

with $\gamma_{\text{boot}} = 0.99$ and target networks maintained via Polyak averaging. The expectile value loss learns $V_{\nu}(s)$ as a high-expectile baseline of the behavioral decision value distribution. Chunk-level advantage is:

$$A_{b,k} = \min(Q_{\bar{\omega},1}(s_{b,k}, z_{b,k}), Q_{\bar{\omega},2}(s_{b,k}, z_{b,k})) - V_{\nu}(s_{b,k}). \quad (19)$$

Advantages are converted to exponential weights:

$$w_{b,k} \propto \exp(\beta A_{b,k}). \quad (20)$$

The Quality-Uplift branch uses a weight interpolated between $w_{b,k}$ and uniform to prevent extreme chunks from dominating flow supervision while preserving quality preference. The Source-to-Positive branch uses uniform weighting. Low-value chunks remain as source endpoints for the Quality-Uplift branch. The critic adjusts their training contribution rather than discarding them.

4.5 Joint Training Objective

The total loss is:

$$\mathcal{L} = \lambda_{\text{act}} \mathcal{L}_{\text{act}} + \lambda_{\text{main}} \mathcal{L}_{\text{main}} + \lambda_{\text{flow}} \mathcal{L}_{\text{flow}}. \quad (21)$$

The action reconstruction loss reweights samples by hierarchical chunk-level advantage:

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{b,k,t} \left[w_{b,k} \left\| \hat{a}_{bkt} - a_{bkt}^{\text{top}} \right\|_2^2 \right]. \quad (22)$$

To prevent flow supervision targets from drifting jointly with encoder parameters, stop-gradient is applied to both endpoints when constructing mean velocity targets. Let $\text{sg}(\cdot)$ denote the stop-gradient operator:

$$U_{\text{main}}^* = \frac{\text{sg}(Z_{\text{top}}) - \text{sg}(Z_{\text{low}})}{t_1 - t_0}, \quad U_{\text{flow}}^* = \frac{\text{sg}(Z_{\text{pos}}) - \text{sg}(Z_{\text{neg}})}{t_1 - t_0}. \quad (23)$$

This ensures $\mathcal{L}_{\text{main}}$ and $\mathcal{L}_{\text{flow}}$ update v_{θ} only through the predicted velocity branch and do not backpropagate into the encoder via the target construction path. The encoder receives end-to-end gradients through $\mathcal{L}_{\text{act}}$ and the decoding path from Z_{plus} .

Algorithm 1 MF2B Training and Inference

```

1: Input: Offline dataset  $\mathcal{D}$ , coverage ratio  $\rho_{\text{traj}}$ ; fix  $t_0 = 0, t_1 = 1$ 
2: Initialize:  $q_\phi, v_\theta, p_\psi, Q_{\omega,1}, Q_{\omega,2}, V_\nu$ 
3: Sort by return; partition  $\mathcal{I}_{\text{top}}$  and  $\mathcal{I}_{\text{low}}$ 
4: while not converged do
5:   Sample  $\tau_{\text{low}} \sim \mathcal{I}_{\text{low}}, \tau_{\text{top}} \sim \mathcal{I}_{\text{top}}, \tau_{\text{neg}} \sim \mathcal{N}(0, I), \tau_{\text{pos}} \sim \mathcal{D}$ 
6:   Encode:  $Z_{\text{low}}, Z_{\text{top}}, Z_{\text{neg}}, Z_{\text{pos}} \leftarrow q_\phi(\cdot)$ 
7:   Compute  $A_{b,k}, w_{b,k}$  via hierarchical chunk-level IQL critic
8:    $\hat{U}_{\text{flow}} \leftarrow v_\theta(Z_{\text{neg}}, t_0, t_1, c_{\text{flow}}); U_{\text{flow}}^* \leftarrow (\text{sg}(Z_{\text{pos}}) - \text{sg}(Z_{\text{neg}})) / (t_1 - t_0)$ 
9:    $\hat{U}_{\text{main}} \leftarrow v_\theta(Z_{\text{low}}, t_0, t_1, c_{\text{main}}); U_{\text{main}}^* \leftarrow (\text{sg}(Z_{\text{top}}) - \text{sg}(Z_{\text{low}})) / (t_1 - t_0)$ 
10:   $Z_{\text{plus}} \leftarrow Z_{\text{low}} + (t_1 - t_0) \hat{U}_{\text{main}}; \hat{A}_{\text{top}} \leftarrow p_\psi(S_{\text{top}}, Z_{\text{plus}})$ 
11:  Update  $\theta, \phi, \psi$ : minimize  $\lambda_{\text{act}} \mathcal{L}_{\text{act}} + \lambda_{\text{main}} \mathcal{L}_{\text{main}} + \lambda_{\text{flow}} \mathcal{L}_{\text{flow}}$ 
12:  Update  $Q_{\omega,1}, Q_{\omega,2}, V_\nu$ ; Polyak-update target networks
13: end while
14: // Inference: Online Decision Execution
15: Input: Trained  $q_\phi, v_\theta, p_\psi$ ; target return condition  $c$ 
16: for each episode do
17:   Reset environment; initialize observation buffer  $\mathcal{B} \leftarrow \{s_0\}$ 
18:   while episode not done do
19:     Construct source chunk  $\tau_{\text{source}} \leftarrow (s_{t-H+1:t}, a_{t-H+1:t-1})$  from buffer  $\mathcal{B}$ 
20:      $Z_{t_0} \leftarrow q_\phi(\tau_{\text{source}})$  {encode to high-level latent decision}
21:      $Z_{t_1} \leftarrow Z_{t_0} + (t_1 - t_0) v_\theta(Z_{t_0}, t_0, t_1, c)$  {one-step mean-flow refinement}
22:      $\hat{a}_{0:H-1} \leftarrow p_\psi(s_{t-H+1:t}, Z_{t_1})$  {decode  $H$  actions}
23:     for  $h = 0$  to  $H - 1$  do
24:       Execute  $\hat{a}_h$  in environment; observe  $s_{t+h+1}, r_{t+h}$ 
25:     end for
26:     Update buffer  $\mathcal{B}$  with new states  $s_{t+1:t+H}$ ;  $t \leftarrow t + H$ 
27:   end while
28: end for

```

The flow losses are normalized by the target velocity scale:

$$\mathcal{L}_{\text{main}} = \frac{\mathbb{E}_{b,k} \left[w_{b,k}^{\text{main}} \left\| \hat{U}_{\text{main}} - U_{\text{main}}^* \right\|_2^2 \right]}{\mathbb{E} \left[\left\| U_{\text{main}}^* \right\|_2^2 \right]}, \quad \mathcal{L}_{\text{flow}} = \frac{\mathbb{E}_{b,k} \left[\left\| \hat{U}_{\text{flow}} - U_{\text{flow}}^* \right\|_2^2 \right]}{\mathbb{E} \left[\left\| U_{\text{flow}}^* \right\|_2^2 \right]}. \quad (24)$$

4.6 One-Step High-Level Decision Refinement and Inference

Given a source trajectory at test time, inference reduces to:

$$Z_{t_0} = q_\phi(\tau_{\text{source}}), \quad (25)$$

$$Z_{t_1} = Z_{t_0} + (t_1 - t_0) v_\theta(Z_{t_0}, t_0, t_1, c), \quad (26)$$

$$\hat{a}_{0:T-1} = p_\psi(Z_{t_1}, s_{0:T-1}). \quad (27)$$

The mean-flow field performs a single forward pass to predict the full mean velocity over $[t_0, t_1]$, so no iterative ODE integration is required. The core of the method is not generating trajectories from scratch, but learning directional transport from existing suboptimal behaviors toward high-value behavioral regions in structured high-level decision space.

5 Experiments

5.1 Environmental Setup

We evaluate MF2B on 9 Gym-MuJoCo locomotion tasks from the D4RL benchmark [Fu et al., 2020], covering HalfCheetah, Hopper, and Walker2d under medium, medium-replay, and medium-expert dataset conditions.

Table 1: D4RL Gym-MuJoCo Benchmark Comprehensive Comparison.

Dataset	Regression-based Policies						Diffusion Policies						Flow Policies			
	BC	TD3-BC	IQL	CQL	MOReL	CAC	Diffuser	DD	EDP	IDQL	DQL	TCD	FQL	FlowQ	SSCP	MF2B
HalfCheetah ME	55.2	90.7	86.7	91.6	53.3	84.3	90.3±0.1	88.9±1.9	95.8±0.1	91.3±0.6	96.8±0.3	92.7±3.4	99.8±0.1	89.6±2.3	98.1±0.9	95.47±0.31
Hopper ME	52.5	98.0	91.5	105.4	108.7	100.4	107.2±0.9	110.4±0.6	110.8±0.4	110.1±0.7	111.1±1.3	112.6±1.0	86.2±1.3	103.5±7.0	110.9±0.9	119.74±1.24
Walker2d ME	107.5	110.1	109.6	108.8	95.6	110.4	107.4±0.1	108.4±0.1	110.4±0.0	110.6±0.0	110.1±0.3	111.3±0.7	100.5±0.1	109.1±0.2	111.1±0.1	117.78±0.95
HalfCheetah M	42.6	48.3	47.4	44.0	42.1	69.1	43.8±0.1	45.3±0.3	50.8±0.0	51.5±0.1	51.1±0.5	47.2±0.7	60.1±0.1	56.8±0.8	52.3±0.5	46.69±0.38
Hopper M	52.9	59.3	66.3	58.5	95.4	80.7	89.5±0.7	98.2±0.1	72.6±0.2	70.1±2.0	90.5±4.6	99.4±0.6	74.5±0.2	88.9±6.4	102.4±0.2	100.93±0.89
Walker2d M	75.3	83.7	78.3	72.5	77.8	83.1	79.4±1.0	79.6±0.9	86.5±0.2	88.1±0.4	87.0±0.9	82.1±1.8	72.7±0.8	84.9±0.4	84.2±0.9	93.20±1.44
HalfCheetah MR	36.6	44.6	44.2	45.5	40.2	58.7	36.0±0.7	42.9±0.1	44.9±0.4	46.5±0.3	47.8±0.3	40.6±1.4	51.1±0.1	50.3±0.9	44.4±1.0	43.81±0.30
Hopper MR	18.1	60.9	94.7	95.0	93.6	99.7	91.8±0.5	99.2±0.2	83.0±1.7	99.4±0.1	101.3±0.6	97.2±2.4	85.4±0.5	99.2±1.4	101.4±0.4	104.36±2.12
Walker2d MR	26.0	81.8	73.9	77.2	49.8	79.5	58.3±1.8	75.6±0.6	87.0±2.6	89.1±2.4	95.5±1.5	88.0±1.9	82.1±1.2	88.5±2.9	85.9±13.4	96.24±1.02
Total (Gym9)	466.7	677.4	692.6	698.5	656.5	765.9	703.7	748.5	741.8	756.7	791.2	771.1	712.3	770.8	788.3	818.22
Average (Gym9)	51.9	75.3	77.0	77.6	72.9	85.1	78.2	83.2	82.4	84.1	87.9	85.7	79.2	85.6	87.9	90.91

5.2 Evaluation Metrics

We report D4RL normalized scores: Normalized Score = $100 \times (\text{score} - \text{random score}) / (\text{expert score} - \text{random score})$, where 0 corresponds to a uniform random policy and 100 to a domain-specific expert. For evaluation, we report the average normalized return over 5 training seeds, with each model evaluated for 50 episodes per task.

5.3 Baseline Models

We compare MF2B with representative baselines across three categories: **regression-based policies** (BC, TD3-BC [Fujimoto and Gu, 2021], IQL [Kostrikov et al., 2021], CQL [Kumar et al., 2020], MOReL [Kidambi et al., 2020], ReBRAC [Tarasov et al., 2023], SRPO [Chen et al., 2024], CAC [Zhang et al., 2021], FAMAC, FERAC), **diffusion policies** (Diffuser [Janner et al., 2022], DD [Ajay et al., 2022], EDP [Kang et al., 2023], IDQL [Hansen-Estruch et al., 2023], DQL [Wang et al., 2022], TCD [Hu et al., 2026]), and **flow policies** (FQL [Park et al., 2025], IFQL, FlowQ [Alles et al., 2025], LaMPO [Liu et al., 2026], SSCP [Koirala and Fleming, 2026], Value Flows [Dong et al., 2026], FAN [Lee et al., 2026]).

6 Experimental Results

6.1 Main Performance Comparison

6.1.1 Gym-MuJoCo Locomotion Task Comprehensive Comparison

Table 1 presents the comprehensive comparison of MF2B with existing offline RL methods on 9 Gym-MuJoCo tasks. We categorize baseline models into three classes: regression-based policies, diffusion policies, and one-step flow policies. MF2B achieves an average score of 90.91 and total score of 818.22 on Gym9 tasks, performing particularly well on medium-replay datasets, which contain suboptimal and noisy trajectories posing challenges for standard policy learning. It achieves competitive performance among one-step flow policies.

From Table 1, MF2B achieves an average score of 90.91 on Gym9, surpassing all flow policies (FQL 79.2, FlowQ 85.6, SSCP 87.9) and most diffusion policies (DQL 87.9, TCD 85.7), demonstrating the efficiency of one-step conditional mean velocity field flow mapping on continuous control tasks.

7 Ablation Studies

7.1 Module Ablation Experiments

Module ablations are conducted on 9 Gym-MuJoCo tasks. Each variant removes or replaces exactly one mechanism while keeping all other settings identical to the complete model.

Direct High-Level Latent Decision Mapping (MF2B-DLM) Replaces conditional mean velocity transport with a direct mapping $\hat{Z}_{\text{top}} = f_{\text{DLM}}(Z_{\text{low}}, c_{\text{chunk}}, c_{\text{global}})$, testing whether the mean velocity flow formulation is necessary.

No Value Guidance (MF2B-NVG) Removes hierarchical chunk-level IQL advantage estimation and reweighting, testing whether value guidance is required to steer high-level latent decisions toward high-value regions.

Return-Ordered Uniform Pairing (MF2B-UP) Draws trajectory pairs uniformly from the full dataset while preserving return direction ($R_{\text{top}} \geq R_{\text{low}}$), removing the explicit high-value and suboptimal trajectory pools. Tests whether directional pool-based pairing provides supervision beyond simple return-ordered pairs.

No Source-to-Positive Alignment Mean-Flow Branch (MF2B-NAF) Sets the Source-to-Positive Alignment branch loss weight to zero, removing auxiliary supervision from Gaussian-Induced Decision Sources (Z_{neg}) to positive behavioral decisions (Z_{pos}) while retaining the Quality-Uplift Mean-Flow Branch. Tests whether the structure-forming branch is necessary for establishing task-relevant behavioral structure in the high-level latent decision space.

7.1.1 Module Ablation Experiment Results

Table 2 presents results on Gym-MuJoCo9. The complete MF2B (average 90.91) outperforms all variants. Key findings: (1) MF2B-DLM (-4.47) confirms the necessity of conditional mean velocity transport over direct target mapping. (2) MF2B-NVG (-4.07) shows chunk-level value guidance is critical for directing latent decisions toward high-value regions. (3) MF2B-UP (-5.71) demonstrates that explicit directional pool-based pairing provides stronger supervision than uniform return-ordered pairs. (4) **MF2B-NAF (-5.28)** validates that the Source-to-Positive Alignment Mean-Flow Branch—analogueous to the neural null-space mechanism [Stavisky et al., 2017]—establishes task-relevant topological structures and value gradient baselines that support subsequent quality-uplift transport.

Table 2: Module Ablation Experiment Results (Gym-MuJoCo 9 Tasks)

Dataset	MF2B-DLM	MF2B-NVG	MF2B-UP	MF2B-NAF	MF2B
HalfCheetah Medium	45.12 $\pm$ 0.25	46.09 $\pm$ 0.65	45.61 $\pm$ 0.38	44.99 $\pm$ 0.13	46.69 $\pm$ 0.38
HalfCheetah Medium-Replay	43.11 $\pm$ 0.42	42.33 $\pm$ 0.19	42.97 $\pm$ 0.26	42.37 $\pm$ 0.12	43.81 $\pm$ 0.30
HalfCheetah Medium-Expert	94.01 $\pm$ 0.22	94.25 $\pm$ 0.17	94.34 $\pm$ 0.31	94.74 $\pm$ 0.49	95.47 $\pm$ 0.31
Hopper Medium	81.57 $\pm$ 2.23	90.91 $\pm$ 3.83	92.74 $\pm$ 7.14	77.46 $\pm$ 1.17	100.93 $\pm$ 0.89
Hopper Medium-Replay	99.97 $\pm$ 3.37	93.10 $\pm$ 1.80	83.00 $\pm$ 13.69	97.28 $\pm$ 2.30	104.36 $\pm$ 2.12
Hopper Medium-Expert	117.75 $\pm$ 1.39	116.92 $\pm$ 0.72	116.71 $\pm$ 0.39	115.13 $\pm$ 0.51	119.74 $\pm$ 1.24
Walker2d Medium	88.37 $\pm$ 0.87	88.16 $\pm$ 0.43	86.15 $\pm$ 0.68	87.10 $\pm$ 0.35	93.20 $\pm$ 1.44
Walker2d Medium-Replay	89.94 $\pm$ 0.32	92.10 $\pm$ 0.66	90.76 $\pm$ 4.39	91.92 $\pm$ 1.42	96.24 $\pm$ 1.02
Walker2d Medium-Expert	114.74 $\pm$ 1.27	114.26 $\pm$ 1.37	111.12 $\pm$ 0.60	111.30 $\pm$ 0.20	117.78 $\pm$ 0.95
Total (Gym9)	774.58	778.12	763.40	767.29	818.22
Average (Gym9)	86.06	86.46	84.82	85.25	90.91

7.1.2 High-Level Latent Decision Space Distribution Analysis

To understand the geometric structure of the learned high-level latent decision space, we analyze four types of representations: low-return behavioral latent decisions (Z_{low}), high-return behavioral latent decisions (Z_{top}), Gaussian source negative endpoints (Z_{neg}), and source-to-positive positive endpoints (Z_{pos}). Note that all analyses operate on the model’s high-level latent representations, not the original action space.

Figure 1a shows t-SNE visualizations of the four distributions. Z_{pos} closely overlaps with Z_{top} , confirming that the Source-to-Positive Alignment branch targets align with high-value behavioral regions. Z_{neg} forms a clearly separated cluster, demonstrating that Gaussian-Induced Decision Sources provide non-trivial structure-forming training signals rather than copying the behavioral distribution. Z_{low} and Z_{top} maintain partial neighborhood continuity, supporting the shared high-level latent decision space hypothesis and confirming that the Quality-Uplift branch performs directional improvement within a common representation space rather than cross-domain generation. KDE analyses in Figure 1b quantitatively corroborate these findings across the first three latent dimensions.

The KDE plots in Figure 1b reveal three quantitative patterns across the first three latent dimensions. First, Z_{pos} and Z_{top} exhibit nearly identical density curves in all three dimensions—their peaks, scales, and tail shapes closely coincide—confirming that the Source-to-Positive Alignment branch successfully transports Gaussian sources into the high-value behavioral region. Second, Z_{neg} deviates significantly from Z_{top} in location, scale, and shape, with particularly pronounced differences in dimensions 1 and 3 where Z_{neg} extends further in the negative direction with heavier tails, validating that Gaussian-Induced Decision Sources occupy a semantically neutral region distinct from task-relevant behaviors. Third, Z_{low} retains substantial density overlap with Z_{top} in the major probability mass regions of all three dimensions, confirming that low-return and high-return behaviors share a common high-level latent decision space—the geometric premise that makes directional quality-uplift transport feasible.

Together, these results establish three conclusions: (1) the Source-to-Positive Alignment branch constructs a non-trivial structure-forming flow from a zero-value baseline to high-value behavioral regions, (2) low-return and high-

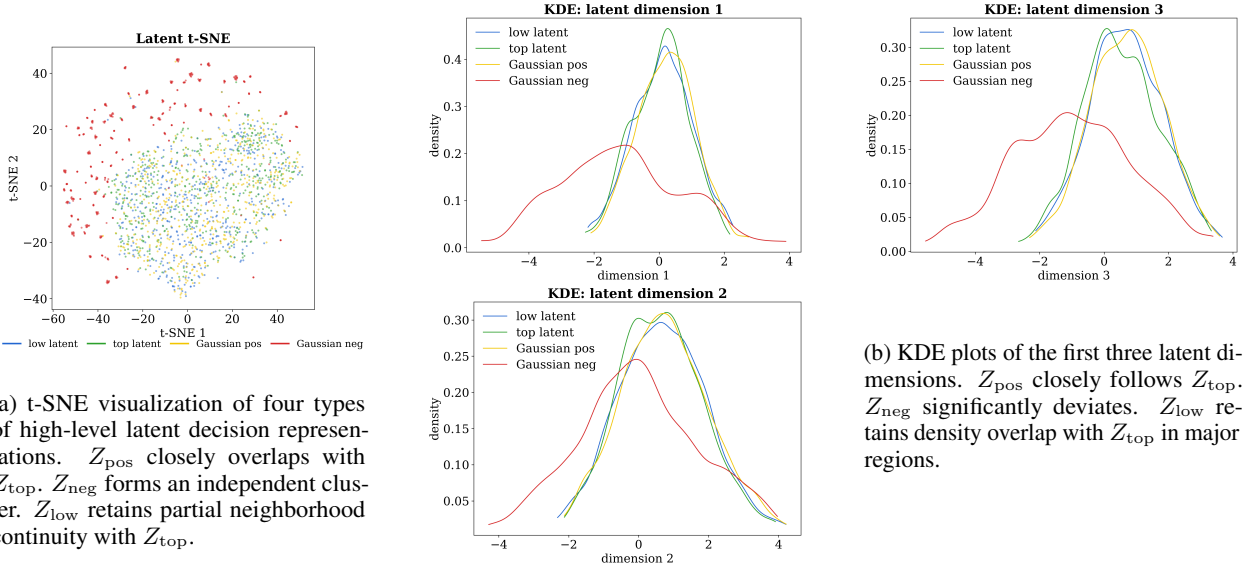


Figure 1: High-level latent decision space analysis. Left: t-SNE visualization of Z_{top} (green), Z_{low} (blue), Z_{pos} (yellow), and Z_{neg} (red). Right: KDE plots of the first three latent dimensions comparing the four distributions.

return behaviors share a common high-level latent decision space, providing the geometric premise for directional quality uplift, and (3) the dual-branch design achieves complementary roles—structure formation and directional improvement—within this shared space.

7.2 Hyperparameter Sensitivity Analysis

Sensitivity experiments are conducted on three medium-replay tasks (HalfCheetah, Hopper, Walker2d) to study high-level time scale and trajectory coverage ratio independently of data distribution changes.

7.2.1 Chunk Length and High-Level Context

Table 3 compares $(H, K_{\text{max}}) \in \{(5, 200), (10, 100), (20, 50)\}$ at approximately equal total primitive-step coverage. The configuration $H=10, K_{\text{max}}=100$ achieves the best balance. Both finer ($H=5$) and coarser ($H=20$) granularities slightly reduce performance, indicating that moderate temporal abstraction is most suitable for high-level latent decision flow mapping.

Table 3: Results for Different Chunk Lengths and Maximum High-Level Chunk Counts

Dataset	$H=5, K_{\text{max}}=200$	$H=10, K_{\text{max}}=100$	$H=20, K_{\text{max}}=50$
HalfCheetah MR	43.03 ± 0.35	43.81 ± 0.30	42.95 ± 0.70
Hopper MR	100.03 ± 1.39	104.36 ± 2.12	99.41 ± 2.62
Walker2d MR	91.73 ± 1.23	96.24 ± 1.02	93.28 ± 1.85

Table 4 fixes $H=10$ and varies $K_{\text{max}} \in \{25, 50, 100\}$. Performance degrades sharply at $K_{\text{max}}=25$ and $K_{\text{max}}=50$, confirming that the model requires sufficient high-level context to capture cross-stage task structure. $K_{\text{max}}=100$ covers the full 1000-step trajectory horizon.

Table 4: Results for Different K_{max} with Fixed $H = 10$

Dataset	$K_{\text{max}}=25$	$K_{\text{max}}=50$	$K_{\text{max}}=100$
HalfCheetah MR	11.74 ± 0.14	22.07 ± 0.22	43.81 ± 0.30
Hopper MR	25.69 ± 0.43	52.02 ± 0.45	104.36 ± 2.12
Walker2d MR	21.13 ± 0.86	42.58 ± 0.29	96.24 ± 1.02

7.2.2 Trajectory Coverage Ratio

Table 5 compares $\rho_{\text{traj}} \in \{0.1, 0.2, 0.5, 0.8\}$. Ratios of 0.1 and 0.5 both achieve performance close to the default $\rho_{\text{traj}}=0.2$, showing that MF2B can learn stable suboptimal-to-high-value flow mapping directions from small high-value timestep budgets. However, $\rho_{\text{traj}}=0.8$ causes sharp drops on Hopper and Walker2d, as an excessively large high-value pool reduces the directional contrast in suboptimal-to-high-value pairing.

Table 5: Results for Different High-Value Timestep Coverage Ratios

Dataset	$\rho=0.1$	$\rho=0.2$	$\rho=0.5$	$\rho=0.8$
HalfCheetah MR	42.32 $\pm$ 0.36	43.81 $\pm$ 0.30	43.00 $\pm$ 0.51	42.66 $\pm$ 0.30
Hopper MR	101.45 $\pm$ 2.33	104.36 $\pm$ 2.12	99.49 $\pm$ 3.48	66.24 $\pm$ 5.75
Walker2d MR	96.77 $\pm$ 2.34	96.24 $\pm$ 1.02	95.70 $\pm$ 2.17	85.92 $\pm$ 8.10

8 Conclusion

This paper proposes MF2B, a one-step policy optimization method for offline reinforcement learning in high-level latent decision space. MF2B reformulates offline RL as a generative policy optimization problem from suboptimal to optimal, mapping trajectory chunks to high-level decision representations through a Chunk-Level Latent Decision Encoder, and learning a shared conditional mean velocity field in this space. The method completes decision refinement through a dual-branch flow mapping mechanism: the first branch transports Gaussian-Induced Decision Sources toward positive behavioral decision regions, establishing task-relevant decision structures. The second branch transports suboptimal decisions toward high-value decision regions, distilling superior behavioral patterns from existing experiences. Hierarchical chunk-level advantage estimation further guides the flow mapping toward high-return regions. Unlike multi-step denoising diffusion policies and flow matching policies requiring multi-step ODE integration, MF2B directly predicts conditional mean velocity over the unit time interval $\Delta t = 1$ and completes decision refinement through one-step Euler integration, requiring neither iterative sampling nor backpropagation through time.

On 9 Gym-MuJoCo tasks from the D4RL benchmark, MF2B achieves an average score of 90.91, surpassing FlowQ (85.6) and SSCP (87.9), and significantly outperforming FQL (79.2). The high-level latent decision representation compresses sequence length from original action steps T to $K \approx T/H$, significantly reducing computational overhead in long-horizon tasks. The suboptimal-to-high-value transport mechanism effectively utilizes low-quality trajectories for policy improvement, making MF2B particularly effective in long-horizon continuous control tasks. Future research can extend MF2B to goal-conditioned RL and multi-task RL scenarios, learning conditional mean velocity fields to support diverse goal-oriented behaviors and further advancing the applicability of offline RL in real-world domains. Nevertheless, the high-level latent decision representations of MF2B are constrained by the diversity of the offline dataset, and excessively narrow behavioral support may limit the quality of the learned decision space. Addressing this through meta-learning initialization, data augmentation, or VLM-grounded latent priors remains an open direction, as does extending the framework to goal-conditioned RL and multi-task RL settings to support diverse goal-oriented behaviors.

References

- Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13344–13362, 2023.
- Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.
- Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.
- Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.
- John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
- Fang Ren, Logan Ward, Travis Williams, Kevin J Laws, Christopher Wolverton, Jason Hattrick-Simpers, and Apurva Mehta. Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments. *Science Advances*, 4(4):eaag1566, 2018.

- Hao Ma, Zhihan Luo, Tuan V Vo, Kaiwen Sima, and Tze-Yun Leong. Centralized reward agent for knowledge sharing and transfer in multi-task reinforcement learning. *Advances in Neural Information Processing Systems*, 38: 163080–163103, 2026.
- Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021a.
- Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3461–3475, 2023.
- Lili Huang, Baopu Dong, and Weinan Zhang. Efficient offline reinforcement learning with relaxed conservatism. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5260–5272, 2024.
- Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 67195–67212, 2023.
- 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.
- Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Proceedings of International Conference on Machine Learning*, 2025.
- Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.
- Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.
- Sergey D. Stavisky, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. Motor cortical visuomotor feedback activity is initially isolated from downstream targets in output-null neural state space dimensions. *Neuron*, 95(1): 195–208.e9, 2017. ISSN 0896-6273. doi: 10.1016/j.neuron.2017.05.023. URL <https://www.sciencedirect.com/science/article/pii/S0896627317304610>.
- Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999a.
- Matthew M. Botvinick, Yael Niv, and Andrew G. Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113(3):262–280, 2009. ISSN 0010-0277. doi: 10.1016/j.cognition.2008.08.011.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.
- 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.
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999b.
- Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.
- 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.
- Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in Neural Information Processing Systems*, 34:28954–28967, 2021b.
- Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- Yihao Hu, Haoran Wang, Xiangyu Zhan, Zhuoran Wang, Michael I Jordan, and Yang Yu. Policy regularization with dataset constraint for offline reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- Yihao Hu, Nathan Kallus, and Xiaojie Mao. Stochastic multi-armed bandits in constant space. *Journal of Machine Learning Research*, 24:1–33, 2023.
- 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.

- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- 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.
- 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.
- Yaron Lipman, Ricky T Q Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proceedings of International Conference on Learning Representations*, 2023.
- Moritz Alles, Nianchen Chen, Patrick van der Smagt, and Botond Cseke. Flowq: Energy-guided flow policies for offline reinforcement learning. *arXiv preprint arXiv:2505.14139*, 2025.
- Zihan Wang, Donglin Li, Yixuan Chen, Yiming Shi, Lei Bai, Tianhe Yu, and Yang Fu. One-step generative policies with q-learning: A reformulation of meanflow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 26751–26759, 2026.
- Zhijie Geng, Minglei Deng, Xiang Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025a.
- Pratyush Koirala and Cody Fleming. Flow-based single-step completion for efficient and expressive policy learning. In *Proceedings of International Conference on Learning Representations*, 2026.
- Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025b.
- Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:21810–21823, 2020.
- Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.
- Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. In *Proceedings of International Conference on Learning Representations*, 2024.
- Shangding Zhang, Yunpeng Jiang, Yu Pan, Weipeng Chen, Bosen Zhang, Mingchuan Zhang, and Jianye Hao. Cautious actor-critic. *arXiv preprint arXiv:2107.05217*, 2021.
- Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.
- Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.
- Jifeng Hu, Yanchao Sun, Sili Huang, Siyuan Guo, Hechang Chen, Li Shen, Lichao Sun, Yi Chang, and Dacheng Tao. Instructed diffuser with temporal condition guidance for offline reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(5):5346–5356, 2026. doi: 10.1109/TPAMI.2026.3652193.
- Tenglong Liu, Xin Xu, Yixing Lan, Jiacheng Huang, Xinglong Zhang, and Ling-Li Zeng. Latent meanflow policy optimization for offline reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2026. doi: 10.1109/TPAMI.2026.3720530.
- Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows. In *Proceedings of the 14th International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=2VyNYUVF2k>.
- Sungyoung Lee, Dohyeong Kim, Eshan Balachandar, Zelal Su Mustafaoglu, and Keshav Pingali. Towards efficient and expressive offline rl via flow-anchored noise-conditioned q-learning, 2026. URL <https://arxiv.org/abs/2605.01663>.