# SKILL-FLOW: LEARNING SKILL-BASED HIERARCHICAL POLICIES VIA FLOW MODELS FOR OFFLINE GOAL-CONDITIONED REINFORCEMENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We address the problem of learning policies for long-horizon goal-conditioned tasks with sparse rewards from offline datasets. Such problems are difficult for two primary reasons: sparse rewards in a long-horizon setting hinder value function learning, and limited data induces out-of-distribution (OOD) errors during policy learning. We tackle these challenges using a skill-based hierarchical framework with flow policies at two levels. The high-level policy selects temporally extended skills to guide long-horizon planning, while the low-level policy executes actions conditioned on the selected skill. Temporally extended skills alleviate sparse rewards by enabling the learning of goal-conditioned value functions, while hierarchical decomposition reduces error accumulation over long horizons. To mitigate OOD errors, we learn behavior-cloning flow policies over skills and actions, then steer their bounded input noise using learned value functions towards high value regions. This construction keeps the resulting policies within the support of the learned behavior-cloning policies. We evaluate our approach on the OGBench benchmark and demonstrate superior performance over state-of-the-art hierarchical and value learning based baselines.

## 1 INTRODUCTION

In a goal-conditioned reinforcement learning (GCRL) setting, tasks are specified by desired outcomes rather than handcrafted rewards, enabling general-purpose policies for diverse goals. Offline GCRL (OGCRL) extends this paradigm to fixed, reward-free trajectory datasets (Yang et al., 2023; Park et al., 2024a), which is appealing when real-world interaction is costly, time-consuming, or unsafe (Kumar et al., 2020). As a result, OGCRL is well suited to domains such as robotics, autonomous driving, logistics, healthcare, recommendation systems, and operations research, where historical or observational data are abundant but active exploration is infeasible or undesirable.

However, despite its conceptual appeal, offline GCRL remains fundamentally challenging in practice (Ghosh et al., 2023; Schaul et al., 2015). In long-horizon, sparse discounted-reward settings, learning accurate value functions is particularly challenging (Park et al., 2023; Ke et al., 2025; Giammarino et al., 2025a). In offline RL, limited data further renders sparse long-horizon value estimation vulnerable to extrapolation and overestimation errors under distribution shift (Kumar et al., 2020; Ke et al., 2025), leading to OOD errors during policy learning. A common strategy in offline GCRL is to reduce the effective task horizon through hierarchical RL (HRL) (Park et al., 2023; Giammarino et al., 2025b; Park et al., 2025b), thereby mitigating the challenges posed by long horizons and sparse rewards. These methods decompose complex goal-conditioned tasks into sequences of nearby subgoals and learn policies to reach them. Although HRL approaches have achieved strong performance in online settings (Levy et al., 2018; Nachum et al., 2018), their performance in offline GCRL remains limited (Park et al., 2023; 2025b; Giammarino et al., 2025a). Most existing offline GCRL methods employ unimodal policies and enforce Kullback–Leibler-based regularization during policy learning (Peng et al., 2019; Ma et al., 2022b) to mitigate OOD errors. However, these approaches can yield suboptimal policies due to limited policy expressivity. SHARSA and DSHARSA learn hierarchical flow policies by first modeling the behavior policy and subsequently performing rejection sampling from the BC samples (Park et al., 2025b), but the resulting method becomes prohibitive in high-dimensional systems.



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

We propose a skill-based hierarchical framework based on flow models to address the aforementioned challenges in offline GCRL. Inspired by prior work such as METRA (Park et al., 2024d), we define skills as latent variables inducing temporally extended fixed-length state transitions within the support of the dataset. Based on Park et al. (2023) and Park et al. (2024b), we hypothesize that the limitations of existing methods stem from both erroneous value learning and suboptimal policy extraction. We use a didactic example to show that the policy error bound can grow exponentially with the effective horizon in discounted sparse-reward settings, even under hierarchical decomposition. We then show that existing flow-based policy extraction methods without conservative statistical-distance regularization, such as FQL (Park et al., 2025c;b), suffer from OOD issues at both hierarchical levels. Motivated by these observations, we propose a skill-based hierarchical approach- `Skill-Flow`, with the following main contributions:

1. We introduce $N$-step implicit Q-learning with delayed high-level rewards ($N$-step IQL-D) for joint skill and value learning, reducing error compounding due to bootstrapping bias in long horizon sparse reward settings.
2. Our motivating example shows that OOD errors at either hierarchy level can impair goal reaching. We leverage expressive flow-based policies at both levels. In particular, we use *reflected flows with controlled noise manipulation to enforce support constraints to mitigate OOD issues at both the skill and action levels.*
3. Through extensive experiments on OGBench (Park et al., 2025a), we demonstrate the effectiveness of our approach across tasks with varying horizon lengths, dataset structures, and dimensionalities, outperforming existing state-of-the-art baselines. We provide ablations supporting the roles of hierarchicy, support and the proposed high-level value learning, and the benefits of low-level extraction.

## 2 Related Work

**Goal-Conditioned RL.** Reward design in classical RL is challenging and can induce unintended behaviors, while reward shaping remains vulnerable to reward hacking Everitt et al. (2017; 2021); Pan et al. (2022); Di Langosco et al. (2022). Goal-conditioned reinforcement learning (GCRL) mitigates this by formulating tasks as goal reaching and learning policies conditioned on desired goals from success indicators Kaelbling (1993); Schaul et al. (2015); Andrychowicz et al. (2017). This formulation supports generalization to new goals and has motivated efficient algorithms for sparse-feedback and offline settings, including contrastive and quasimetric approaches Eysenbach et al. (2022); Ma et al. (2022a); Wang et al. (2023).

**Offline GCRL.** Offline RL learns from fixed datasets without environment interaction Levine et al. (2020), making it suitable for settings where online data collection is expensive or unsafe, but it is prone to distributional shift and extrapolation error. Extending goal conditioning to this regime yields Offline GCRL, which further requires generalization to unseen state–goal pairs using only static data and no corrective feedback. These compounded challenges have motivated specialized approaches Chebotar et al. (2021); Yang et al. (2022; 2023); Mezghani et al. (2023); Sikchi et al. (2023); Park et al. (2024c;a) focused on improving stability and generalization.

**Value Learning.** Most offline GCRL methods focus on value learning, while policy-extraction errors remain comparatively underexplored. A key insight in GCRL is that the optimal goal-conditioned value corresponds to the shortest feasible path between a state and a goal, inducing a quasimetric structure over the state–goal space (Sontag, 1995; Liu et al., 2023). Several methods exploit this by constraining value functions to quasimetric mappings (Wang et al., 2023; Giammarino & Qureshi, 2026), but these approaches often require specialized architectures with additional training overhead and limited scalability to high-dimensional systems. Other works construct quasimetric value estimates using successor features learned via contrastive RL (Myers et al., 2025; 2024), though contrastive objectives can struggle in sparse-reward, long-horizon settings. Finally, Giammarino et al. (2025a) employ Hamilton–Jacobi–Bellman (HJB) PDE regularization in value estimation, but relies on structural assumptions about the dynamics that may not hold broadly across domains.

**Hierarchy in RL.** Temporal abstraction is a long-standing paradigm in RL in which policies are decomposed into high-level decision making over skills or subgoals and low-level controllers that execute them, thereby improving exploration and credit assignment Sutton et al. (1999); Vezhnevets

et al. (2017); Nachum et al. (2018). Recent work shows that such subgoal-based hierarchies are especially effective in mitigating the challenge of long task horizon in GCRL Park et al. (2024c; 2025b); Giammarino & Qureshi (2026). In our work, hierarchy plays a central role: we identify the limitations with value learning and policy extraction approaches in existing hierarchical approaches and address them.

# 3 BACKGROUND

**Offline GCRL.** In this paper, we use $\Delta(\mathcal{X})$ to denote the set of probability distributions over space $\mathcal{X}$. We consider a finite-horizon discounted Markov Decision Process (MDP) described by the tuple $(\mathcal{S}, \mathcal{G}, \mathcal{A}, \mathcal{T}, r, P_g, \rho_0, \gamma)$, where $\mathcal{S}$ is the state space, and $\mathcal{G}$ is the set of goals, which is assumed to be equivalent to the state space. $\mathcal{A} \subseteq \mathbb{R}^m$ is the set of actions, and $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability function. We define the sparse goal-conditioned reward function $r : \mathcal{S} \times \mathcal{G} \to \{-1, 0\}$ as $r(s, g) = -1$ when $s \neq g$ and $r(s, g) = 0$ otherwise. $P_g \in \Delta(\mathcal{G})$ is the distribution over goals. $\rho_0 \in \Delta(\mathcal{S})$ is the initial state distribution. $\gamma \in (0, 1]$ is the discount factor.

We assume the goal $g$ is sampled according to $P_g$ at the beginning of each episode. The objective is to maximize the expected sum of discounted rewards to successfully reach $g$. Formally, this objective is expressed as $J(\pi) = \mathbb{E}_{\tau \sim \rho^\pi(g)}[\sum_{k=0}^{k} \gamma^k r(s_k, g)]$ where $\pi : \mathcal{S} \times \mathcal{G} \to \Delta(\mathcal{A})$ is the goal-conditioned policy, and $\rho^\pi(g) = \rho_0(s_0)\pi(a_0|s_0, g)\mathcal{T}(s_1|s_0, a_0) \cdots \pi(a_k|s_k, g)$ denotes the trajectory distribution under policy $\pi(\cdot|\cdot, g)$. The optimal state-value function $V^*(s, g)$ and optimal action-value function $Q^*(s, a, g)$ are given by:

$$V^*(s, g) = \max_\pi V^\pi(s, g) = r(s, g) + \max_\pi \gamma \mathbb{E}_{a \sim \pi(\cdot|s,g), s' \sim \mathcal{T}(\cdot|s,a)} \left[V^\pi(s', g)\right] \tag{1}$$

$$Q^*(s, a, g) = \max_\pi \mathbb{E}\left[Q^\pi(s, a, g)\right] = r(s, g) + \max_\pi \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)} \left[V^\pi(s', g)\right] \tag{2}$$

where $s' \in \mathcal{S}$ denotes the next state sampled from the transition function $\mathcal{T}$. The optimal policy satisfies $\pi^*(s, g) = \arg\max_{\pi(s,g)} \mathbb{E}_{a \sim \pi}\left[Q^*(s, a, g)\right]$.

Preliminary details on flow matching can be found in Appendix A.

# 4 METHODOLOGY

In this section, we present our hierarchical flow-based skill framework. Our approach learns low-level policies for reachable skills from offline data and a high-level policy that composes them to solve long-horizon sparse-reward goal-reaching tasks. Skills are defined as latent variables inducing fixed-length state transitions within the support of dataset distribution conditioned on the current state and the selected goal. We define a latent variable $z \in \mathbb{R}^d$ and an encoder $\phi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^d$ mapping current state and state after $N$ steps to the latent space. Let $\pi_h : \mathcal{S} \times \mathcal{G} \to \Delta(\mathbb{R}^d)$ be the high-level policy and $\pi_\ell : \mathcal{S} \times \mathbb{R}^d \to \Delta(\mathcal{A})$ be the low-level policy conditioned on the currently selected skill $z$. The learning objective then becomes:

$$J(\pi_h, \pi_\ell) = \max_{\pi_h, \pi_\ell} \mathbb{E}_{\rho^{(\pi_h \circ \pi_\ell)}(g)} \left[\sum_{k=0}^{T} \gamma^t r\left(s_k, g\right)\right] = \max_{\pi_h, \pi_\ell} \mathbb{E}_{\rho^{(\pi_h \circ \pi_\ell)}(g)} \left[\sum_{k=0}^{K} \sum_{i=0}^{N-1} \gamma^{Nk+i} r\left(s_{Nk+i}, g\right)\right]$$

where $K = \frac{T}{N} - 1$ and $\rho^{(\pi_h \circ \pi_\ell)}(g)$ is the trajectory distribution produced by the composition of the hierarchical policies as a function of the goal $g$. Next, we present two motivating examples illustrating how our approach addresses the aforementioned challenges in OGCRL, followed by the details of our method. We provide the theoretical results with all proofs in Appendix B.

## 4.1 MOTIVATING EXAMPLES

### 4.1.1 VALUE LEARNING

As mentioned previously, both value function learning and policy extraction are challenging in offline GCRL. Li et al. (2026) show that one-step TD learning can introduce *bootstrapping bias*, where neural network estimation errors accumulate over the horizon in learned value estimates. This issue is

further exacerbated in sparse-reward settings, where the signal-to-noise ratio between the true value signal and estimation error is even lower Park et al. (2023), making flat policy learning ineffective.

Hierarchical decomposition is known to mitigate action-selection errors in long-horizon tasks by reducing the effective task horizon. In skill-based hierarchies, however, high-level value learning should rely on delayed rewards, since the contribution of a selected skill to task completion can only be evaluated after its execution. Assigning rewards at the end of the skill therefore provides a more informative learning signal for high-level value estimation.

We further motivate this using the following didactic example. We consider a 1-D chain environment, where states lie on the integer line, the goal is located at position $T$, and the distance to a state $s$ is defined as $d(s, g) = |T - s|$. At each step, the agent can move one unit to the left or right, and receives reward 0 upon reaching the goal and $-1$ otherwise as shown in Figure 7 in Appendix B. Under discounted sparse rewards, the optimal value depends on the distance to the goal as $V^*(s, g) = -G(|s - g|)$, where $G(d) = \sum_{t=0}^{d-1} \gamma^t = (1 - \gamma^d)/(1 - \gamma)$. We assume the learned value function contains multiplicative Gaussian noise, $\widehat{V}(s, g) = V^*(s, g)(1 + \sigma z_{s,g})$, where $z_{s,g} \sim \mathcal{N}(0, 1)$.

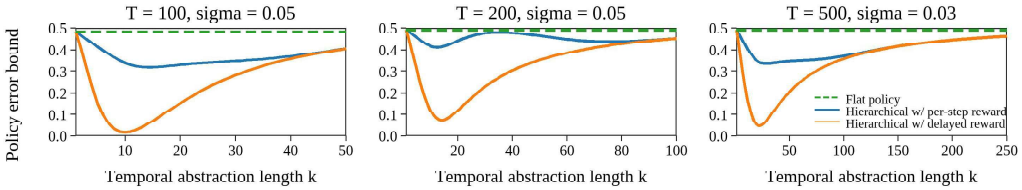


Figure 1: Comparison of **policy errors.** The error for hierarchical policies with delayed rewards is lower than hierarchical policies with per-step reward.

For hierarchical policies with length-N skills, the high-level value is $V^*(s, g) = -G(|s - g|)$ under per-step rewards and $V^*(s, g) = -G(H)$, $H = |s - g|/N$, under delayed rewards. The corresponding low-level value for executing a length-$k$ skill is $V^*(s, s + N) = -G(N)$. Based on these definitions, we derive policy error bounds for the discounted sparse-reward setting under flat policies and hierarchical policies with both per-step and delayed rewards; the derivations are provided in Appendix B.1. We plot the error bounds for different task horizons and temporal abstraction lengths in Figure 1. As shown, hierarchical policies (blue and orange) achieve lower error than the flat policy (green). Notably, the delayed-reward hierarchical policy provides a lower error than the per-step-reward hierarchical policy, as shown in Appendix B.1 and validated by the plots. This further motivates using delayed rewards for high-level value learning in the hierarchical setting.

### 4.1.2 Policy Learning

As discussed above, most OGCRL methods incorporate statistical-distance-based regularization in policy learning to tackle the OOD issue caused by extrapolation errors in the value function, since the value function is only updated on the support of the dataset. The policy learning objective in OGCRL is then defined as:

$$\max_{\pi_h} \mathbb{E}_{(s_k,g)\sim\mathcal{D},z_k\sim\pi_h}\left[Q_h^*(s_k, z_k, g)\right] \quad \text{s.t.} \quad D(\pi_h(\cdot|s_k, g) \,\|\, \pi_\beta(\cdot|s_k, g)) \leq \varepsilon, \tag{3}$$

where $Q_h^*$ is the optimal state-action (high-level) value at state $s_k$ conditioned on goal $g$, $D(\cdot\|\cdot)$ is some statistical distance, e.g., $D_{\text{KL}}$ for KL divergence or $D_{\text{W}_2}$ for the Wasserstein distance, and $\pi_\beta$ is the skill sampled under the behavior policy conditioned on the state $s_k$ and goal $g$. The majority of existing OGCRL approaches (Peng et al., 2019; Giammarino et al., 2025a; Park et al., 2023; Ma et al., 2022b), use a unimodal policy with KL-divergence-based policy regularization. However, Zhang et al. (2026) show that using the KL divergence can result in learning suboptimal policies based on the choice of $\epsilon$, while using the $\text{W}_2$ distance causes OOD errors.

Flow policies offer an expressive solution for modeling complex multimodal policy distributions; however, recent OGCRL methods using flow policies, such as FQL (Park et al., 2025b), can introduce OOD errors during policy learning/extraction. We provide a motivating example to illustrate this issue. We consider a 2-D navigation task with a point-mass agent following the `pointmaze` environment suite in OGBench (Park et al., 2025a). Further details about this example are provided in Appendix C.

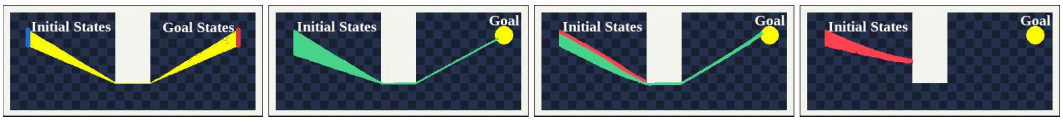


Figure 2: **Illustration of OOD errors in hierarchical policy extraction.** Left to right: collected dataset; evaluation using our method enforcing support-constraints for policy learning at both levels; evaluation using our method for policy learning at the high level and FQL at the low level; and evaluation using FQL extraction at the high level with our policy learning method at the low level. Successful trajectories are shown in green, and unsuccessful trajectories are shown in red.

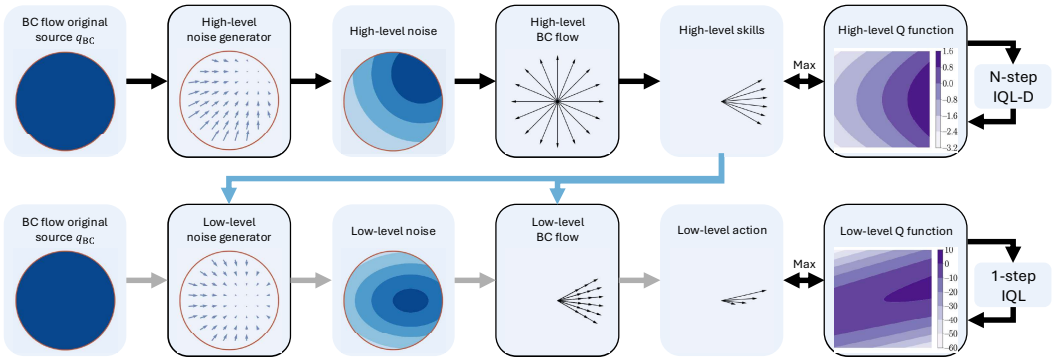


Figure 3: **`Skill-Flow` algorithm:** Our hierarchical approach consists of two levels. At the high level, we learn a behavior-cloned (BC) flow policy over skills from the dataset and apply $N$-step Implicit Q-Learning (IQL) with delayed rewards to learn the optimal Q-function. This Q-function is then used to train a noise generator that steers the BC flow policy toward actions that maximize the Q-value. At the low level, we learn skill-conditioned policies. A BC flow policy clones the conditional action distribution of the dataset, while a 1-step IQL objective is used to learn the skill-level Q-function, which in turn trains a noise generator to steer the BC policy toward Q-maximizing actions.

On the collected dataset shown in the left panel of Figure 2, we evaluate the learned policy under three settings. As shown, our method achieves a $100\%$ success rate (SR), with all trajectories reaching the goal state (green). In contrast, removing support constraints at either hierarchy level results in unsuccessful trajectories (red). Without high-level support constraints, the policy may select OOD skills for which the low-level policy is not trained, leading to extrapolation error. Without low-level support constraints, OOD actions can drive the agent to unsupported states, where subsequent high-level skill selection also relies on extrapolation which can cause errors.

**Remark 1.** OOD errors can originate at either the skill or action level and *compound across the hierarchy*. This motivates enforcing support constraints at both levels in our proposed method.

### 4.2 Our Approach

Next, we present our hierarchical approach, which incorporates delayed rewards and support-constrained policy optimization at both levels, as illustrated in Figure 3. We introduce parameters in the subscript when defining each loss optimized by our method. For example, the actual $Q_h$ value is parametrized as $Q_{\theta_h}$ in the learning objective.

#### 4.2.1 High-Level Policy Learning

The high-level policy learns skill-selection policies for long-horizon planning. We denote the corresponding state and state-action value functions by $V_h$ and $Q_h$, respectively. Motivated by the observations in 4.1.1, we use delayed rewards for high-level value learning. For each sampled training tuple, we form the skill target $z_k = \phi(s_k, s_{k+N})$, where $s_{k+N}$ is the state after $N$ transitions. Building on implicit Q-learning (IQL) (Kostrikov et al., 2021) , we introduce IQL with delayed rewards which we call $N$-step IQL-D, to learn the optimal Q-value $Q^*$ and subsequently the optimal

value $V^*$ via expectile regression as follows:

$$\mathcal{L}(V_{\theta_h}, \phi) = \mathbb{E}_{(s_k, s_{k+N}, g)\sim\mathcal{D}}\big[\, L_2^\tau\big(Q_{\theta_h}(s_k, z_k, g) - V_{\theta_h}(s_k, g)\big)\,\big] \tag{4}$$
$$\mathcal{L}(Q_{\theta_h}, \phi) = \mathbb{E}_{(s_k, s_{k+N}, g)\sim\mathcal{D}}\big[\, \big(r(s_{k+N}, g) + \gamma\, V_{\theta_h}(s_{k+N}, g) - Q_{\theta_h}(s_k, z_k, g)\big)^2\,\big],$$

where, the reward is delayed to step $k+N$. The expectile loss is defined as $L_2^\tau(u) = |\tau - \mathbf{1}(u<0)|u^2$, with $\tau > 0.5$. Note that the latent variable $z$, corresponding to skills, is learned jointly with the value functions. This enables learning latent representations that satisfy the Bellman optimality equations.

We next use the value estimates to extract a high-level policy. Motivated by the example in 4.1.2, we define the policy extraction objective as follows:

$$\max_{\pi_h} \mathbb{E}_{(s_k,g)\sim\mathcal{D}, z_k\sim\pi_h}\big[Q_h^*(s_k, z_k, g)\big] \quad \text{s.t.} \quad \operatorname{supp}(\pi_h(\cdot|s_k, g)) \subseteq \operatorname{supp}(\pi_\beta(\cdot|s_k, g)), \tag{5}$$

where $\pi_h(.|s_k, g)$ is the high-level policy over the skills. The constraint restricts skill selection to the support of the behavior policy $\pi_\beta$. We approximate this distribution with a BC policy $\pi_h^{\text{BC}}$ and constrain policy extraction to its learned support. Specifically, we use the latent noise steered policy optimization approach proposed in (Zhang et al., 2026) to learn the policy using two flow networks: (i) a behavior cloning (BC) flow policy network and (ii) a latent noise generation network to implement the high-level policy as illustrated in Figure 3.

Consider a bounded $d$-dimensional ball $\mathcal{B}_h^d \coloneqq \{x \in \mathbb{R}^d \mid \|x\| \le h\}$. We learn a BC policy over the set of skills in the dataset $\mathcal{D}$ collected using the behavior policy $\pi_\beta$ conditioned on the state $s$ and the goal $g$. We sample the noise $x_0 \sim \mathcal{U}(\mathcal{B}_h^d)$ uniformly from the ball, sample the skill $z$ from the dataset $\mathcal{D}$, and construct linear targets $x_t = (1-t)x_0 + tz$. Then, the BC objective reduces to learning the velocity field of the flow policy using the following objective:

$$\mathcal{L}_{\text{BC}}(v_{\theta_h}^{\text{BC}}) = \quad \mathbb{E}_{\substack{(s_k, s_{k+N}, g)\sim\mathcal{D} \\ x\sim\mathcal{U}(\mathcal{B}_h^d), t\sim\mathcal{U}[0,1]}} \left[\left\| v_{\theta_h}^{\text{BC}}(t,\, x_t; s_k, g) - (z - x_0)\right\|^2\right], \tag{6}$$

where $v_{\theta_h}^{\text{BC}} : [0,1] \times \mathbb{R}^d \times \mathcal{S} \times \mathcal{G} \to \mathbb{R}^d$ is the learned velocity field, and $z = \phi(s_k, s_{k+N})$. Let us define $\mu_h^{\text{BC}}(x_0; s, g) = \psi_h^{\text{BC}}(1, x_0; s, g)$ as the flow at time $t = 1$ corresponding to the learned velocity field $v_{\theta_h}^{\text{BC}}$. Applying this map to uniform noise $X \sim \mathcal{U}(\mathcal{B}_h^d)$ defines the BC policy $\pi_h^{\text{BC}}(\cdot \mid s, g)$.

To select higher-value skills, we replace the uniform BC input with a learned noise generator policy $\pi_h^{\text{NG}} : \mathcal{S} \times \mathcal{G} \to \mathbb{R}^d$ that outputs the noise for the BC flow policy network, i.e., $x_0 \sim \pi_h^{\text{NG}}$. Restricting its output to the same source ball gives the auxiliary objective:

$$\max_{\pi_h^{\text{NG}}} \mathbb{E}_{(s_k,g)\sim\mathcal{D}, x_0\sim\pi_h^{\text{NG}}}\big[Q^*(s_k, \mu_h^{\text{BC}}(x_0, s_k, g), g)\big] \quad \text{s.t.} \quad x_0 \in \mathcal{B}_h^d. \tag{7}$$

In practice, we train the noise generator to maximize this objective using the learned critic $Q_{\theta_h}$. We learn the policy $\pi_h^{\text{NG}}$ using another flow network, which we call the noise generator flow $\psi_h^{\text{NG}}$: $[0,1] \times \mathcal{B}_h^d \times \mathcal{S} \times \mathcal{G} \to \mathcal{B}_h^d$, and denote its associated velocity field as $v_h^{\text{NG}} : [0,1] \times \mathcal{B}_h^d \times \mathcal{S} \times \mathcal{G} \to \mathbb{R}^d$. Since we know the set $\mathcal{B}_h^d$, we can satisfy the constraint by *reflecting* the flow at each integration step if it leaves the ball. The *reflected* ODE (Xie et al., 2024) can be formally expressed as:

$$\begin{aligned} d\psi_h^{\text{NG}}(t, w_0; s, g) &= v_h^{\text{NG}}\big(t, \psi_h^{\text{NG}}(t, w_0; s, g); s, g\big)\, dt + dL_k, \\ \psi_h^{\text{NG}}(0, w_0; s, g) &= w_0, \end{aligned} \tag{8}$$

where $w_0 \sim \mathcal{U}(\mathcal{B}_h^d)$ is the sampled noise for the noise generator flow $\psi_h^{\text{NG}}$ and the reflection term $dL_k$ compensates for the outward velocity at $\partial\mathcal{B}_h^d$ by pushing the motion back inside $\mathcal{B}_h^d$. Commonly, the ODE is solved using Euler integration as follows:

$$w_{j+1} = w_j + v_h^{\text{NG}}(j\Delta t, w_j; s, g)\Delta t, \quad j \in \{0, \ldots, M-1\}, \tag{9}$$

where $M$ is the number of integration steps, $\Delta t = \frac{1}{M}$, and $w_1 = x_0$. Reflection is enforced by projecting $w_j$ back onto the ball. The reflected Euler integration step is given by:

$$\begin{aligned} w_{j+1} = {} & \mathbf{1}\{\tilde{w}_{j+1} \in \mathcal{B}_h^d\}\tilde{w}_{j+1} \\ & + (1 - \mathbf{1}\{\tilde{w}_{j+1} \in \mathcal{B}_h^d\})\,(\tilde{w}_{j+1} - \langle \hat{v}_h(j\Delta t, w_0; s, g)\Delta t, n_{j+1}\rangle n_{j+1})\,, \end{aligned}$$

where $\tilde{w}_{j+1} = w_j + \hat{v}_h(j\Delta t, w_0; s, g)\Delta t$ follows the original Euler step, $n_j = \tilde{w}_j / \|\tilde{w}_j\|$, and $\langle\cdot,\cdot\rangle$ is the inner product. Let us define $\mu_h^{\text{NG}}(w_0; s, g) = \psi_h^{\text{NG}}(1, w_0; s, g)$ for the reflected flow corresponding to the noise generator. Then the composition of the noise generator network with the BC flow network gives the latent variable $z = \mu_h^{\text{BC}}(\mu_h^{\text{NG}}(w_0; s, g); s, g)$.

**Remark 2.** The reflected flow keeps the generated noise in $\mathcal{B}_h^d$, satisfying Equation (7). With a continuous BC map, this gives $\operatorname{supp}\pi_h(\cdot \mid s, g) \subseteq \operatorname{supp}\pi_h^{\mathrm{BC}}(\cdot \mid s, g)$, following the BC support argument in Theorem 2 of Zhang et al. (2026).

#### 4.2.2 Low-Level Skill Policy Learning

In this section, we present the details of learning the low-level skill policy conditioned on the state $s$ and latent skill $z$. We use the sparse {-1,0} reward at each step and learn the optimal Q value using implicit Q-learning (Kostrikov et al., 2021) with the loss functions provided below:

$$\begin{aligned}\mathcal{L}(V_{\theta_\ell}) &= \mathbb{E}_{(s_k,a_k,s_{k+N},g)\sim\mathcal{D}}\big[\, L_2^\tau\big(Q_{\theta_\ell}(s_k,a_k,z_k) - V_{\theta_\ell}(s_k,z_k)\big)\,\big] \\ \mathcal{L}(Q_{\theta_\ell}) &= \mathbb{E}_{(s_k,a_k,s_{k+N},g)\sim\mathcal{D}}\big[\, \big(r(s_k,s_{k+N}) + \gamma\, V_{\theta_\ell}(s_{k+1},z_k) - Q_{\theta_\ell}(s_k,a_k,z_k)\big)^2\,\big]\end{aligned} \tag{10}$$

where $z_k = \phi(s_k, s_{k+N})$, is the same encoding as at the high level. Subsequently, the learned Q-value is used to learn the policy corresponding to the skills. Similar to the case of high-level policy learning, we enforce support constraint in the policy learning objective to tackle the OOD problem. The low-level policy learning problem is provided in the equations below:

$$\begin{aligned}&\max_{\pi_\ell} \mathbb{E}_{(s_k,s_{k+N},g)\sim\mathcal{D},a_k\sim\pi_l}\big[Q_l^*(s_k,a_k,z_k)\big] \\ &\text{s.t.} \quad \operatorname{supp}(\pi_l(\cdot|s_k,z_k)) \subseteq \operatorname{supp}(\pi_\beta(\cdot|s_k,z_k)),\end{aligned} \tag{11}$$

where $\operatorname{supp}\pi_\beta$ corresponds to the action support of the behavior policy $\pi_\beta$ conditioned on the state $s_k$ and skill $z_k$. The learned Q-value function $Q_{\theta_\ell}$ guides policy extraction within the learned BC support using the same idea of noise manipulation described for the high-level policy learning.

We learn a BC policy $\pi_\ell^{\mathrm{BC}}(\cdot|s_k,z_k)$ conditioned on the state $s_k$ and skill $z_k$. Given a ball $\mathcal{B}_l^n := \{x \in \mathbb{R}^n \mid \|x\| \le l\}$ as defined for the high-level policy learning case, we sample noise $x_0 \sim \mathcal{U}(\mathcal{B}_l^n)$ and actions $a_k$ from the dataset $\mathcal{D}$, construct linear targets as $x_t = (1-t)x_0 + ta_k$, and use the derivative $\frac{dx}{dt} = (a_k - x_0)$ to learn the flow policy vector field using the following objective:

$$\mathcal{L}_{\mathrm{BC}}(v_{\theta_\ell}^{\mathrm{BC}}) = \mathbb{E}_{\substack{(s_k,a_k,s_{k+N},g)\sim\mathcal{D},\\ x\sim\mathcal{U}(\mathcal{B}_l^n),\, t\sim\mathcal{U}[0,1]}}\Big[\big\|v_{\theta_\ell}^{\mathrm{BC}}\big(t,x_t;\, s_k,z_k\big) - (a_k - x_0)\big\|^2\Big]. \tag{12}$$

where $v_{\theta_\ell}^{\mathrm{BC}} : [0,1]\times\mathbb{R}^n\times\mathcal{S}\times\mathbb{R}^d \to \mathbb{R}^n$ is the velocity field, and $z_k = \phi(s_k, s_{k+N})$. Let us define $\mu_\ell^{\mathrm{BC}}(x_0; s_k, z_k) = \psi_\ell^{\mathrm{BC}}(1, x_0; s_k, z_k)$ as the BC flow. The distribution of its output under $x_0 \sim \mathcal{U}(\mathcal{B}_l^n)$ is $\pi_\ell^{\mathrm{BC}}(\cdot \mid s_k, z_k)$. Then, we formulate the auxiliary noise generation policy learning objective as follows:

$$\max_{\pi_\ell^{\mathrm{NG}}} \mathbb{E}_{(s_k,s_{k+N},g)\sim\mathcal{D},x_0\sim\pi_\ell^{\mathrm{NG}}}\big[Q^*(s_k,\mu_\ell^{\mathrm{BC}}(x_0,s_k,z_k),z_k)\big] \quad \text{s.t.} \quad x_0 \in \mathcal{B}_l^n. \tag{13}$$

We learn a reflected flow $\psi_\ell{}^{\mathrm{NG}} : [0,1]\times\mathcal{B}_l^n\times\mathcal{S}\times\mathbb{R}^d \to \mathcal{B}_l^n$. Let us define $\mu_\ell^{\mathrm{NG}}(w_0; s, z) = \psi_\ell^{\mathrm{NG}}(1, w_0; s, z)$ where $w_0 \sim \mathcal{U}(\mathcal{B}_l^n)$ is input noise to the noise generator flow. Then, the composition of the noise generator network with the BC flow gives the action $a = \mu_\ell^{\mathrm{BC}}(\mu_\ell^{\mathrm{NG}}(w_0; s, z); s, z)$. Under the conditions in Theorem 1, this design preserves the support of the learned low-level BC policy and the flow noise generator is trained to maximize the objective in Equation (11).

**Remark 3.** At each fixed $(s, z)$, a continuous low-level BC map and bounded noise similarly give $\operatorname{supp}\pi_\ell(\cdot \mid s, z) \subseteq \operatorname{supp}\pi_\ell^{\mathrm{BC}}(\cdot \mid s, z)$.

**Theorem 1.** *Given a hierarchical policy $\pi_h \circ \pi_\ell$ learned by noise manipulation using reflected flow, i.e. $a = \mu_\ell^{BC}(\mu_\ell^{NG}(w_0; s, z); s, z)$, where $z = \mu_h^{BC}(\mu_h^{NG}(w_0; s, g); s, g)$, it satisfies the support constraint in both levels i.e. $\operatorname{supp}\pi_h(.|s,g) \subseteq \operatorname{supp}\pi_\beta(\cdot|s,g)$ and $\operatorname{supp}\pi_\ell(\cdot|s,z) \subseteq \operatorname{supp}\pi_\beta(.|s,z)$.*

We present the pseudoscope of our algorithm in Appendix D.

## 5 Experiments

In this section, we conduct experiments to answer the following research questions:

(**Q1**) How does `Skill-Flow` perform in long-horizon tasks against other OGCRL algorithms?

(**Q2**) How does Skill-Flow perform across various types of datasets with different data quality?
(**Q3**) How important is the hierarchy compared to a flat flow policy using support constraints?
(**Q4**) How important is the N-step IQL-D loss compared to the standard IQL loss?
(**Q5**) How important is the support-constraint-based policy learning at both levels in our method?

Additional details for environments, baselines, implementation, hyperparameters, and full results, including more ablations, are provided in Section E.

### 5.1 Setup

**Environments.** We conduct experiments on navigation and manipulation environments from the OGBench (Park et al., 2025a) offline goal-conditioned RL benchmark. To answer **(Q1)**, we consider tasks with varying complexities and horizons. For locomotion, we use `point`, `ant`, and `humanoid` agents in `medium`, `large`, and `giant` mazes. For manipulation, we consider `cube`, `scene`, and `puzzle` tasks of varying complexity. To answer **(Q2)**, we evaluate three locomotion dataset types—`navigate`, `stitch`, and `explore`—corresponding to expert, stitched, and random trajectories, respectively. For manipulation, we use the `play` and `noisy` datasets, representing expert and noisy trajectories. For the main comparison, all methods are trained with three random seeds per environment, and the converged models are evaluated on three evaluation seeds.

**Baselines.** We compare Skill-Flow with *eight* baselines grouped into three categories: (i) value-learning methods, including Goal-Conditioned Implicit Q-Learning (GCIQL), Quasimetric Reinforcement Learning (QRL) Wang et al. (2023), and Temporal Metric Distillation (TMD) Myers et al. (2025); (ii) methods addressing OOD errors through value-function regularization, including Conservative Offline Goal-Conditioned Implicit V-Learning (CGCIVL) Ke et al. (2025); and (iii) hierarchical approaches, including Hierarchical Implicit Q-Learning (HIQL) Park et al. (2023), Hierarchical Flow Q-Learning HFQL Park et al. (2025b), SHARSA, and DSHARSA Park et al. (2025b).

### 5.2 Main Results

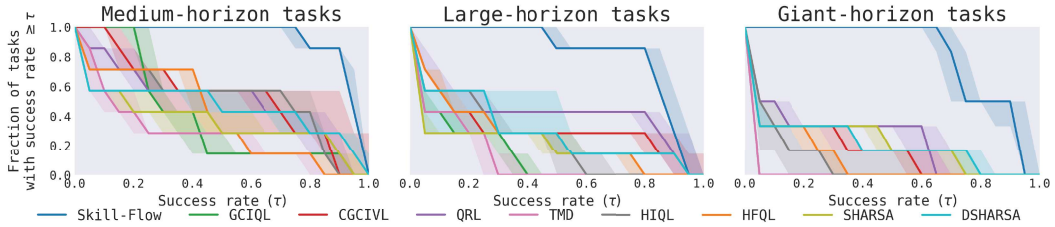


Figure 4: **Performance profile over tasks with different horizon lengths.** For a given success rate $\tau$ (x-axis), we show the fraction of tasks with a success rate $\geq \tau$ for each algorithm (see Agarwal et al. (2021) for details). Skill-Flow dominates all baselines in all tasks with different horizon lengths, especially in Giant-horizon tasks, where the baselines achieve much lower success rates.

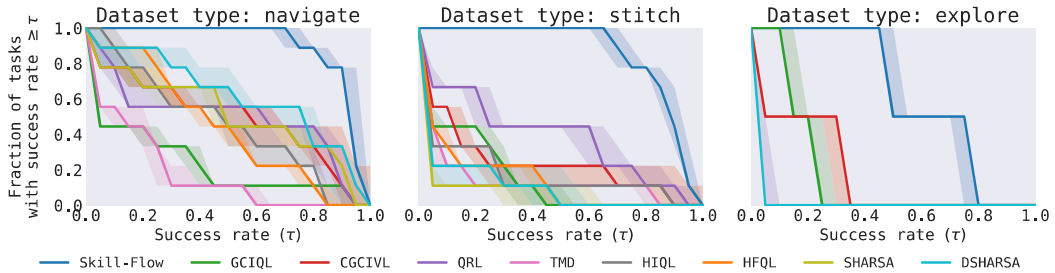


Figure 5: **Performance profile over tasks with different datasets.** Here we plot the performance profile with different dataset types, namely, `navigate` (expert trajectories), `stitch` (trajectory segments), and `explore` (random), in order of decreasing quality. Unlike baselines, Skill-Flow's performance does not decrease much with the decreasing dataset quality.

**(Q1) Skill-Flow achieves the best overall performance, especially in long-horizon tasks.** Following Agarwal et al. (2021), we plot the performance profile over `medium`, `long`, and `giant` navigation tasks in Figure 4. Skill-Flow performs significantly better on tasks across the horizon lengths. In addition, the performance gap between Skill-Flow and the baselines increases as the task horizon length increases, outlining the merit of our approach in long-horizon tasks.

**(Q2) Skill-Flow consistently achieves strong performance with different dataset quality.** We plot the performance profile across dataset types in Figure 5. While baseline performance varies significantly with dataset quality—e.g., DSHARSA, QRL, and CGCIVL perform second best on `navigate`, `stitch`, and `explore` datasets, respectively—Skill-Flow consistently achieves best performance.

### 5.3 Ablation Results

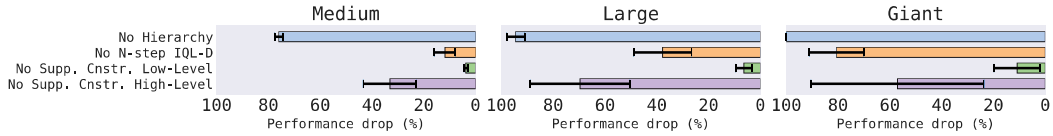


Figure 6: Ablation study results. We report the mean and std of the **percentage performance drop by averaging the performance drop across the `point`, `ant` and `humanoid` agents across the different horizon lengths**, namely `medium`, `large` and `giant`.

To study how each part of Skill-Flow contributes to the overall performance, we conduct the following experiments for the navigation environments with varying horizon lengths, with different agent types, namely, (`point`, `ant`, and `humanoid`) and different dataset qualities (`navigate`, `stitch` and `explore`). We present the results of the study in Figure 6.

**(Q3) `No Hierarchy`: hierarchy is important for long-horizon goal reaching.** This variant removes the hierarchy and learns a flat flow policy with one-step expectile regression for value learning and noise-steered policy extraction. Its zero success on all evaluated giant mazes in Table 8 shows that this flat design alone is insufficient for these long-horizon tasks, supporting the role of hierarchy in Skill-Flow.

**(Q4) `No N-step IQL-D`: the proposed high-level value learning improves long-horizon performance.** We retain the hierarchy, skill encoder, and both policy extractors, replacing only the high-level $N$-step IQL-D update with one-step IQL (Kostrikov et al., 2021). Success falls to zero on all three giant-stitch tasks, compared with 68–91% for Skill-Flow (Table 8). These results establish the benefit of the proposed high-level update within the same hierarchical architecture and extraction design.

**(Q5) `No Supp. Cnstr. Low-Level` and `No Supp. Cnstr. High-Level`: support constraints are important at both hierarchical levels.** We replace them with Wasserstein-$W_2$ regularization at at one level at a time . Both variants degrade performance, with removing the high-level support constraint causing the largest drop, consistent with the motivating example in Figure 2.

**Additional Evaluations.** We provide additional evaluations in Section F to support our algorithm design choices.

## 6 Conclusion and Future Works

We proposed Skill-Flow for addressing the challenges of long-horizon tasks with sparse reward in offline GCRL. Skill-Flow employs a skill-based hierarchy with delayed rewards for handling long decision horizons, flow-based policies at both hierarchical levels for enhanced expressiveness, and noise manipulation with reflected flows to mitigate the OOD issue. Through extensive experiments on 31 OGBench environments with tasks with different horizon lengths and dataset qualities, we show that Skill-Flow achieves the best performance compared to recent baselines. The performance gap between Skill-Flow and the baselines increases as the task horizon length increases and dataset quality decreases, suggesting Skill-Flow's capability of handling challenging tasks. A potential future research direction can be learning language grounded policies to improve generalizability.

### AI USE STATEMENT

In this work, we used generative AI tools for grammar checking and assistance with portions of the code. We reviewed all AI-assisted content and take full responsibility for the paper's text, claims, code, and results.

### ETHICS STATEMENT

This work does not introduce specific ethical concerns.

### REPRODUCIBILITY STATEMENT

We support reproducibility by providing the full method and training pseudocode, environment and dataset specifications, hyperparameters, compute setup, and per-task results in the paper and appendix. Experiments use publicly available OGBench datasets; we report results over three seeds and 50 evaluation runs per seed, with five-seed results for a subset of tasks.

## REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in Neural Information Processing Systems*, 30, 2017.

Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

Lauro Di Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 12004–12019. PMLR, 2022.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1705.08417*, 2017.

Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*, 198 (27):6435–6467, 2021.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Dibya Ghosh, Chethan A. Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, volume 202, pp. 11321–11339, 2023.

Vittorio Giammarino and Ahmed H. Qureshi. Goal reaching with eikonal-constrained hierarchical quasimetric reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Vittorio Giammarino, Ruiqi Ni, and Ahmed H Qureshi. Physics-informed value learner for offline goal-conditioned reinforcement learning. *arXiv preprint arXiv:2509.06782*, 2025a.

Vittorio Giammarino, Ruiqi Ni, and Ahmed H. Qureshi. Physics-informed value learner for offline goal-conditioned reinforcement learning. In *Proceedings of the Thirty-Ninth International Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025b. URL https://openreview.net/forum?id=LRYgQuz7kY. Poster.

Leslie Pack Kaelbling. Learning to achieve goals. In *International Joint Conference on Artificial Intelligence*, volume 2, pp. 1094–1098, 1993.

Kaiqiang Ke, Qian Lin, Zongkai Liu, Shenghong He, and Chao Yu. Conservative offline goal-conditioned implicit v-learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29591–29607. PMLR, July 2025. URL https://proceedings.mlr.press/v267/ke25a.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

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.

Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. In *International Conference on Learning Representations*, 2018.

Qiyang Li, Seohong Park, and Sergey Levine. Decoupled q-chunking. In *International Conference on Learning Representations*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric residual network for sample efficient goal-conditioned reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8799–8806, 2023.

Jason Yecheng Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. Offline goal-conditioned reinforcement learning via $f$-advantage regression. *Advances in Neural Information Processing Systems*, 35:310–323, 2022a.

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. How far i'll go: Offline goal-conditioned reinforcement learning via $f$-advantage regression. In *Advances in Neural Information Processing Systems*, NeurIPS, 2022b. URL https://arxiv.org/abs/2206.03023.

Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric, and Karteek Alahari. Learning goal-conditioned policies offline with self-supervised reward shaping. In *Conference on Robot Learning*, pp. 1401–1410. PMLR, 2023.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37076–37096. PMLR, 2024. URL https://proceedings.mlr.press/v235/myers24a.html.

Vivek Myers, Bill Chunyuan Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. *CoRR*, abs/2509.20478, 2025. URL https://arxiv.org/abs/2509.20478.

Ofir Nachum, Shixiang (Shane) Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Neural Information Processing Systems*, 2018.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34866–34891, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, 2024a.

Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is value learning really the main bottleneck in offline reinforcement learning? In *Advances in Neural Information Processing Systems (NeurIPS) 37*, pp. 79029–79056. Neural Information Processing Systems Foundation, Inc., 2024b. doi: 10.52202/079017-2509. URL https://proceedings.com/079017-2509.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36, 2024c.

Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. In *International Conference on Learning Representations*, 2024d.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025a.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes RL scalable. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2025b.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025c.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. URL https://arxiv.org/abs/1910.00177.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International Conference on Machine Learning*, pp. 1312–1320. PMLR, 2015.

Harshit Sikchi, Rohan Chitnis, Ahmed Touati, Alborz Geramifard, Amy Zhang, and Scott Niekum. Smore: Score models for offline goal-conditioned reinforcement learning. *arXiv preprint arXiv:2311.02013*, 2023.

Eduardo D Sontag. An abstract approach to dissipation. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 3, pp. 2702–2703. IEEE, 1995.

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, 1999.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, 2017.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pp. 36411–36430. PMLR, 2023.

Tianyu Xie, Yu Zhu, Longlin Yu, Tong Yang, Ziheng Cheng, Shiyue Zhang, Xiangyu Zhang, and Cheng Zhang. Reflected flow matching. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 54614–54634, 2024.

Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline rl. *arXiv preprint arXiv:2202.04478*, 2022.

Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What is essential for unseen goal generalization of offline goal-conditioned rl? In *International Conference on Machine Learning*, pp. 39543–39571. PMLR, 2023.

Songyuan Zhang, Oswin So, H M Sabbir Ahmad, Eric Yang Yu, Matthew Cleaveland, Mitchell Black, and Chuchu Fan. ReFORM: Reflected flows for on-support offline RL via noise manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=YvFsyRReeN.