

Learning Hierarchical Skill Policies with Offline Quality-Diversity Reinforcement Learning

Tanachai Anakawat¹, Takayuki Osa², and Tatsuya Harada^{1,2}

¹The University of Tokyo ²RIKEN Center for Advanced Intelligence Project
Tokyo, Japan

Abstract—Recent studies investigate how to leverage pre-collected datasets to improve the policy performance and sample efficiency of RL. One promising approach to achieve this goal is to employ a two-stage strategy: In the first stage, diverse skills are extracted as a low-level policy from a given dataset, and a high-level policy is trained to solve a specific task in the second stage. Typically, extraction of the low-level policy is performed based on unsupervised learning such as trajectory VAE. However, a limitation of this approach is that the quality of the low-level policy highly depends on the quality of the dataset. To address this issue, we introduce QDOS (Quality-Diversity Offline Skill learning), a unified pipeline for robust offline-to-online learning. Our approach incorporates an Advantage-Weighted Quality-Diversity pretraining objective, which weights the skill extraction and diversity objectives by the estimated advantage of each trajectory segment. This approach allows the model to extract diverse and high-value skills. By providing robust and task-relevant skill representations, QDOS significantly improves the quality of the embedded skill space used by the low-level policy. We further integrate this with a dual dataset reuse strategy, where offline data is used both for skill pretraining and for populating the online replay buffer via pseudo-labeling. Experiments demonstrate that QDOS significantly outperforms strong baselines in structured manipulation tasks and unstructured locomotion tasks, confirming its ability to accelerate exploration and improve final returns in challenging sparse-reward domains.

I. INTRODUCTION

Deep reinforcement learning (RL) has achieved remarkable success in solving complex control tasks, ranging from playing video games to manipulating robotic arms [1], [2]. Despite these achievements, the standard online RL paradigm requires a large amount of interactions with the environment to learn effective policies. This sample inefficiency makes it prohibitively expensive and often unsafe for real-world applications, such as robotics, where data collection is costly and potentially dangerous.

To address these limitations, Offline RL has emerged as a promising alternative, enabling agents to learn policies entirely from static, pre-collected datasets without online interaction [3]. In parallel, unsupervised skill discovery methods aim to extract reusable behaviors, or “skills,” from large, unlabeled datasets. Recent methods such as OPAL [4] and SPiRL [5] encode primitive skills into a latent space and train a high-level policy that leverages these low-level policies to solve complex tasks. This temporal abstraction reduces the effective horizon of the task, simplifying the learning of long-horizon problems.

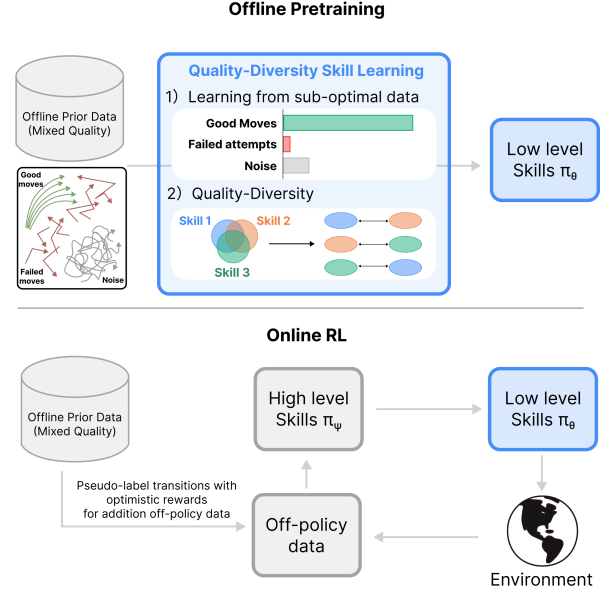


Fig. 1. Overview of QDOS Framework. (1) **Offline Pretraining**: The agent learns low-level skills (π_θ) from a mixed-quality dataset containing good moves, failed attempts, and noise. We employ an Advantage-Weighted Quality-Diversity objective to filter out sub-optimal data (failed attempts and noise) while extracting diverse, high-quality skills. (2) **Online RL with Dual Dataset Reuse**: The same offline data is reused to populate the high-level replay buffer via pseudo-labeling, enabling the high-level policy to utilize prior knowledge for efficient online exploration.

Recent work has begun to bridge these two fields in the “offline-to-online” setting, where an agent is pretrained on offline data and then fine-tuned online interacting with the environment. Approaches such as SUPE [6] demonstrate that reusing offline data for both skill extraction and as a replay buffer for the high-level policy can dramatically accelerate exploration, resulting in higher performance after online RL.

However, a critical challenge remains: *how to identify and extract useful behaviors when the offline data contains a mixture of optimal and suboptimal actions?* Real-world datasets often contain optimal trajectories entangled with noise, failed attempts, and irrelevant behaviors. Standard trajectory VAEs, as used in previous methods, treat all data segments equally, aiming to minimize the reconstruction error over the entire dataset. When the offline data is noisy, these methods learn a “cluttered” skill space where useful skills are entangled with useless ones, making it difficult for

the high-level policy to select effective actions.

This leads to a trade-off in current approaches. Purely diversity-driven methods (e.g., maximizing mutual information) encourage distinct behaviors but fail to distinguish between distinct *useful* behaviors and distinct *useless* ones. Conversely, purely reward-driven or behavior-cloning approaches often collapse to a single mode, failing to capture the diversity of solutions necessary for robust downstream exploration. Consequently, there is a lack of a unified framework that can filter out noise while preserving a diverse set of high-quality skills suitable for accelerating online learning.

In this work, we propose QDOS (Quality-Diversity Offline Skill learning), a unified framework for robust offline-to-online learning, as illustrated in Fig. 1. Our method addresses the mixed-quality dilemma by introducing skill learning based on unsupervised multi-solution discovery using offline RL. The objective function for our skill learning incorporates both the expected return and a skill-diversity term, thereby yielding diverse and high-quality behaviors. This approach enables the learning of a skill space composed of useful primitives. We integrate this into an offline-to-online framework that reuses offline data twice: first for skill extraction, and second for populating the online replay buffer via “pseudo-labeling,” where offline trajectories are annotated with inferred skills and optimistic reward estimates following SUPE [6]. This allows the high-level policy to perform off-policy learning on the extensive offline dataset from the start.

We extensively evaluate our method on diverse and complex domains, including *antsoccer*, *kitchen*, and *humanoidmaze*. Our experiments demonstrate that our method significantly outperforms state-of-the-art baselines (such as SUPE [6]) in terms of sample efficiency, asymptotic performance, and goal-finding speed, particularly in tasks requiring the composition of diverse, dynamic behaviors.

II. RELATED WORK

A. Unsupervised Skill Discovery

Unsupervised skill discovery aims to learn reusable behaviors without extrinsic rewards, typically by maximizing an intrinsic objective. Early online methods focused on maximizing the mutual information between latent codes and states [7], [8] or regularizing option policies [9]. These methods encourage the agent to visit diverse states and learn distinguishable behaviors.

In the offline setting, methods like OPAL [4] and SPiRL [5] adapt these ideas by using trajectory VAEs to learn skills from fixed demonstration datasets. They compress short trajectory segments into a latent variable z , allowing a high-level policy to operate in this latent space. However, these methods typically discard the offline data after the pretraining phase. Furthermore, they do not account for data quality; they try to reconstruct all trajectories, including suboptimal ones, which can degrade the quality of the learned skill space when the dataset is noisy.

B. Offline Reinforcement Learning

Offline RL focuses on learning optimal policies from static datasets. The primary challenge is distribution shift, where the learned policy visits states outside the dataset support, leading to overestimation of values. Algorithms like CQL [10] and IQL [11] address this by constraining the policy or value function. While effective for learning from fixed data, these methods can be overly conservative, which hinders exploration when the agent is allowed to interact with the environment online.

C. Offline-to-Online Reinforcement Learning

Offline-to-online RL leverages offline data to accelerate online learning. A common strategy is to initialize an RL agent with a policy pretrained offline and then fine-tune it online [12], [13]. Recent work like SUPE [6] proposes a more integrated approach, reusing offline data for both skill learning and online replay. By pseudo-labeling offline trajectories with inferred skills and rewards, SUPE allows the high-level policy to learn from the rich offline dataset. However, SUPE treats all offline data equally during skill learning, inheriting the brittleness of standard VAEs when dealing with mixed-quality data. Our method improves upon this by applying advantage weighting during skill discovery to ensure that only beneficial but diverse behaviors are captured and reused.

D. Quality Diversity RL

Quality-Diversity (QD) optimization is a paradigm that aims to generate large collections of diverse solutions that are all high-performing, contrasting with pure optimization which seeks a single global optimum. This concept was introduced by the Generative and Developmental Systems community [14], [15] with algorithms such as Novelty Search with Local Competition and MAP-Elites. QD algorithms operate in a behavioral or feature space rather than the genotypic space, attempting to fill the behavior space with high-performing solutions even if they are not global peaks in the fitness landscape. In robotics, QD has been used to create repertoires of behaviors [16] and for damage adaptation. It has also been applied to engineering design [17] and video game level generation [18].

Recently, these ideas have been adapted to Reinforcement Learning. DiveOff [19] introduces a QD perspective to offline RL, aiming to discover multiple solutions from a single task. DiveOff employs a variational objective that maximizes the mutual information between latent skills and trajectories to ensure diversity, while simultaneously regularizing the latent space to maintain high quality. Specifically, it uses a coordinate ascent approach to prevent mode collapse and ensure that the learned skills cover the manifold of useful behaviors. Our work builds upon this by explicitly integrating advantage-weighting into the QD objective, ensuring that diversity is sought specifically among high-value trajectories.

E. Limitations of Current Approaches in Mixed-Quality Data

Despite the promise of data reuse, existing methods like SUPE and OPAL rely on the assumption that the offline dataset consists of coherent, segmentable behaviors. In real-world scenarios, datasets are often unstructured, containing a mix of optimal trajectories, exploratory noise, and failed attempts. Standard trajectory VAEs minimize reconstruction error over the entire dataset, forcing the model to allocate representational capacity to noise. This results in a “cluttered” skill space where distinct latent codes map to indistinguishable or useless behaviors. Our work addresses this critical gap by introducing an advantage-weighted objective that extracts high-quality segments and learns a latent skill space that corresponds to high-advantage behaviors.

III. PRELIMINARIES

A. Problem Formulation

We consider a Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r, \rho\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition dynamics, γ is the discount factor, r is the reward function, and ρ is the initial state distribution.

We assume access to a static offline dataset $\mathcal{D} = \{\tau_i\}_{i=1}^N$, where each trajectory τ_i consists of a sequence of state-action-reward tuples $\tau_i = (s_0, a_0, r_0, \dots, s_T, a_T, r_T)$. Crucially, while rewards are present, the dataset is *unlabeled* regarding skill definitions. There are no segmentations or labels z indicating the active primitive behavior. The dataset contains both good and bad actions mixed together; the agent must infer which behaviors are worth learning and reusing.

The goal is two-fold:

- 1) **Skill Discovery:** Learn a low-level latent-conditioned policy $\pi_\theta(a | s, z)$ and prior $p(z)$ that capture diverse, useful functional behaviors while filtering out noise and suboptimal data.
- 2) **Hierarchical Adaptation:** Learn a high-level policy $\pi_\psi(z | s)$ that selects these skills to maximize expected return $\eta(\pi) = \mathbb{E}[\sum \gamma^t r_t]$ during online interaction.

B. Limitations of Current Approaches

SUPE (Skills from Unlabeled Prior data for Exploration) [20] uses a standard VAE to embed skills from the offline dataset and reuses this data to populate the online high-level replay buffer. While SUPE bridges offline skill learning and online RL, it implicitly assumes that the extracted skills are meaningful. Standard trajectory VAEs treat all segments as valid signals. In noisy datasets, the VAE reconstructs noise, resulting in a “cluttered” library where distinct latents map to indistinguishable or useless behaviors. This leads to unreliable pseudo-labeling, causing the high-level policy to receive inconsistent signals and fail to learn robust strategies. This mixed-quality issue is illustrated in Fig. 2.

IV. METHOD

In this section, we detail our proposed framework, QDOS, which consists of two main phases: (1) Advantage-Weighted

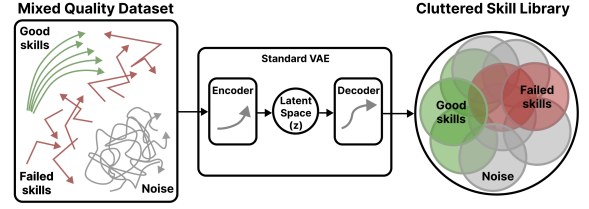


Fig. 2. The Mixed-Quality Dilemma: Standard VAEs encode all behaviors equally, entangling noise with useful skills.

Quality-Diversity Skill Pretraining, and (2) Online Exploration with Trajectory Skills. The overall procedure is summarized in Algorithm 1.

A. Advantage-Weighted Quality-Diversity Pretraining

Our skill learning framework is inspired by DiveOff [19], which introduces a Quality-Diversity approach to offline RL. While the primary goal of DiveOff is to discover multiple diverse solutions within a static dataset, our objective is to *reuse* these diverse, high-performing solutions to accelerate *online* fine-tuning. By embedding these skills into a latent space, we provide the high-level policy with a structured action space that facilitates efficient exploration.

To effectively extract useful behaviors from mixed-quality datasets, we propose an Advantage-Weighted VAE framework. Standard VAEs minimize the reconstruction error over all data segments uniformly. This forces the model to dedicate capacity to reconstructing noise and suboptimal behaviors, cluttering the latent space. Instead, we propose to weight the learning objective by the estimated advantage of each trajectory segment.

1) *Advantage Estimation via IQL:* We first estimate the advantage of each trajectory segment. We segment the offline trajectories into fixed-length segments $\tau_{[H]} = \{s_0, a_0, \dots, s_{H-1}, a_{H-1}\}$ of length H . To estimate the quality of these segments, we train a value function $V_\psi(s)$ on the offline dataset using Implicit Q-Learning (IQL) [11]. IQL learns a value function that approximates the upper expectile of the return distribution, effectively capturing the value of the best policy supported by the data without querying out-of-distribution actions. The IQL value function is trained by minimizing the expectile regression loss:

$$L_V(\psi) = \mathbb{E}_{(s,a) \sim \mathcal{D}} [L_2^\tau(Q_{\hat{\psi}}(s, a) - V_\psi(s))], \quad (1)$$

where $L_2^\tau(u) = |\tau - \mathbb{I}(u < 0)|u^2$ is the expectile loss with $\tau \in (0.5, 1)$ and $Q_{\hat{\psi}}$ denotes the target critic network. This allows us to estimate the potential return from any state in the dataset. The advantage of a segment starting at state s_0 can then be estimated. We compute a weight $w(\tau_{[H]})$ for each segment:

$$w(\tau_{[H]}) \propto \exp\left(\frac{A(\tau_{[H]})}{\lambda}\right), \quad (2)$$

where λ is a temperature parameter controlling the sharpness of the weight. High-advantage segments receive significantly higher weights, effectively focusing the VAE on learning

from the “good” parts of the dataset while ignoring the “bad” parts. This filtering is crucial for preventing the skill space from being polluted by noise.

2) *Weighted VAE Objective*: We learn the skill space using a Variational Autoencoder (VAE). The encoder $f_\theta(z | \tau_{[H]})$ maps a trajectory segment to a latent variable z , and the decoder (low-level policy) $\pi_\theta(a | s, z)$ reconstructs the actions. We also learn a state-dependent prior $p_\theta(z | s_0)$ to capture the distribution of skills available at a given state. The weighted VAE objective is defined as:

$$\mathcal{L}_{\text{VAE}}^{\text{AW}}(\theta) = \beta D_{\text{KL}}(f_\theta(z | \tau_{[H]}) \parallel p_\theta(z | s_0)) - \mathbb{E}_{z \sim f_\theta} \left[w(\tau_{[H]}) \sum_{t=0}^{H-1} \log \pi_\theta(a_t | s_t, z) \right]. \quad (3)$$

This objective ensures that the model prioritizes the reconstruction of high-value behaviors. The KL-divergence term regularizes the learned posterior towards the prior, ensuring a smooth latent space.

3) *Mutual Information Maximization*: To prevent mode collapse (where the policy ignores the latent code) and ensuring diversity among the learned skills, we maximize the mutual information $I(z; \tau)$ between the latent skill z and the generated trajectory τ . Following variational information maximization principles [19], [21], we utilize a variational lower bound:

$$I(z; \tau) \geq \mathbb{E}_{z \sim p(z), \tau \sim \pi_\theta(\cdot | z)} [\log q_\theta(z | \tau)] + H(z), \quad (4)$$

where $q_\theta(z | \tau)$ is a parameterized posterior distribution. Since the entropy $H(z)$ is constant for a fixed prior, maximizing mutual information is equivalent to maximizing the expected log-likelihood of the posterior. To maximize the entropy $H(z)$, a natural choice for the prior $p(z)$ is the uniform distribution. In practice, we sample random skills $z_{\text{rand}} \sim p(z)$, generate short auxiliary rollouts with the current policy π_θ , and minimize the reconstruction error of the latent skill from these generated trajectories. Crucially, we also weight this diversity objective by the trajectory advantage. This ensures that we encourage diversity specifically among high-quality behaviors, rather than encouraging diverse failure modes. The expected log-likelihood of the posterior can be maximized by minimizing the following mean-squared L2 norm:

$$\mathcal{L}_{\text{info}}(\theta) = \frac{1}{B} \sum_{i=1}^B w^{(i)} \left\| \hat{z}_{\text{rand}}^{(i)} - z_{\text{rand}}^{(i)} \right\|_2^2, \quad (5)$$

where z_{rand} is sampled from the prior, and $\hat{z}_{\text{rand}}$ is encoded from the rollout of $\pi_\theta(z_{\text{rand}})$.

The total pretraining objective is a weighted combination:

$$\mathcal{L}_\theta = \mathcal{L}_{\text{VAE}}^{\text{AW}} + \alpha_{\text{loss}} \mathcal{L}_{\text{info}}. \quad (6)$$

B. Online Exploration with Trajectory Skills

Following pretraining, the low-level skill policy π_θ is frozen. We then introduce a high-level policy $\pi_\psi(z | s)$ that operates in the latent skill space, selecting a new skill z every H environment steps.

Algorithm 1 QDOS: Advantage-Weighted Quality-Diversity Skill Pretraining and Online RL

1: **Inputs**: dataset $\mathcal{D}$, segment length H , batch size B , replay buffer $\mathcal{D}_{\text{replay}} \leftarrow \emptyset$, temperature λ , mutual-information weight α .
2: **Phase 1: Advantage-Weighted Skill Pretraining**
3: **for** each training step **do**
4: Sample segments $\{\tau_{[H]}^{(i)}\}_{i=1}^B \sim \mathcal{D}$.
5: Encode posterior $q_\theta(z | \tau_{[H]}^{(i)})$ and prior $p_\theta(z | s_0^{(i)})$; sample $z^{(i)}$.
6: Update IQL critics Q_ψ, V_ψ to estimate values.
7: Compute advantages $A_\psi(s_0^{(i)}, z^{(i)}) = Q_\psi(s_0^{(i)}, z^{(i)}) - V_\psi(s_0^{(i)})$.
8: Compute weights $w^{(i)} \propto \exp(A_\psi / \lambda)$ (normalized).
9: **Weighted VAE loss**:

$$\mathcal{L}_{\text{VAE}}^{\text{AW}} = \frac{1}{B} \sum_{i=1}^B w^{(i)} [\beta D_{\text{KL}} - \mathbb{E}_z [\log \pi_\theta]].$$

10: **Mutual-information loss**: Sample $z_{\text{rand}}^{(i)}$, rollout π_θ , encode $\hat{z}_{\text{rand}}^{(i)}$.

$$\mathcal{L}_{\text{info}} = \frac{1}{B} \sum_{i=1}^B w^{(i)} \left\| \hat{z}_{\text{rand}}^{(i)} - z_{\text{rand}}^{(i)} \right\|_2^2.$$

11: Update θ via $\mathcal{L}_\theta = \mathcal{L}_{\text{VAE}}^{\text{AW}} + \alpha \mathcal{L}_{\text{info}}$.
12: **end for**
13: **Phase 2: Online RL with Pseudo-Labeling**
14: **Initialize**: High-level policy $\pi_\psi(z | s)$ and $\mathcal{D}_{\text{offline}} = \emptyset$.
15: **for** each offline segment $\tau_{[H]}^{(i)} \in \mathcal{D}$ **do**
16: Infer $\hat{z}^{(i)} \sim q_\theta(z | \tau_{[H]}^{(i)})$ and optimistic reward $\hat{r}^{(i)}$.
17: Add $(s_0^{(i)}, \hat{z}^{(i)}, \hat{r}^{(i)}, s_H^{(i)})$ to $\mathcal{D}_{\text{offline}}$.
18: **end for**
19: **for** each online interaction step **do**
20: Sample $z \sim \pi_\psi(z | s)$, execute $\pi_\theta(a | s, z)$ for H steps.
21: Store transition in $\mathcal{D}_{\text{replay}}$.
22: Update π_ψ using $\mathcal{D}_{\text{replay}} \cup \mathcal{D}_{\text{offline}}$ with off-policy RL (e.g., IQL).
23: **end for**

To accelerate the learning of this high-level policy, we employ the dual dataset reuse strategy from SUPE [6]. We “pseudo-label” the offline dataset $\mathcal{D}$ to create a replay buffer of high-level transitions. For each offline segment $\tau_{[H]}$, we: 1. Infer the skill label $\hat{z} \sim q_\theta(z | \tau_{[H]})$ using the pretrained encoder. 2. Compute an optimistic reward estimate $\hat{r}$ using an upper-confidence bound (UCB) estimator. We use Random Network Distillation (RND) [22] to add an exploration bonus to the reward predicted by a learned reward model.

These pseudo-labeled transitions $(s_0, \hat{z}, \hat{r}, s_H)$ are added to the online replay buffer. This allows the high-level agent to perform off-policy learning (using an algorithm like RLPD [12] or SAC [23]) on the rich offline data from the very beginning of the online phase. The combination of high-quality, advantage-weighted skills and optimistic exploration bonuses guides the agent towards promising regions of the state space, significantly accelerating exploration.

C. Implementation Notes

We implement QDOS using PyTorch. For skill pretraining, we use the Adam optimizer with a learning rate of 3×10^{-4} . The segment horizon is set to $H = 20$ for all tasks. The advantage weights are computed using a value function trained via IQL on the offline dataset. The expectile parameter τ is fixed at 0.9 for all environments, while the temperature parameter λ is tuned per environment (typically $\lambda \in [0.3, 1.0]$). The mutual information weight α_{loss} is also a hyperparameter, with values in $\{0.1, 0.2, 0.3\}$ tested.

During online adaptation, we employ an entropy-regularized actor-critic (SAC) for the high-level policy. The

replay buffer is initialized with the pseudo-labeled offline data, and we maintain a 50/50 sampling ratio between online and offline data during high-level updates to ensure stable learning. All experiments are averaged over 5 random seeds.

V. EXPERIMENTS

A. Experimental Setup

We evaluate QDOS on four challenging sparse-reward domains that require long-horizon planning and precise control:

- **antmaze**: A locomotion task where a quadrupedal ant must navigate a maze to reach a goal. We use the large maze layout.
- **kitchen**: A sequential manipulation task where a robotic arm must interact with multiple objects (e.g., microwave, kettle, switch) in a kitchen environment. We use the mixed and partial datasets.
- **humanoidmaze**: A high-dimensional locomotion task where a humanoid agent must navigate a maze. This task is significantly harder than AntMaze due to the stability requirements of the humanoid.
- **antsoccer**: A complex task where an ant must push a soccer ball to a goal location. This requires both locomotion and object manipulation skills.

We compare our method against several strong baselines:

- **SUPE** [6]: Our primary baseline, a state-of-the-art offline-to-online method that uses trajectory skills without advantage weighting.
- **ExPLORe** [24]: A method that uses exploration bonuses on offline data but learns a flat (non-hierarchical) policy.
- **Trajectory skills** [4], [5]: A baseline using standard trajectory-based skill learning without advantage weighting or diversity maximization.
- **IQL** [11]: An offline RL method that learns a value function using expectile regression to avoid out-of-distribution actions.
- **BC**: Behavior Cloning, which learns a policy by maximizing the likelihood of actions in the offline dataset. For the learning curves, we compare against Diffusion BC with Jump-Start RL (JSRL) [25], a strong baseline that initializes the policy with BC and fine-tunes it online.

We evaluate performance based on the normalized return (success rate for maze/soccer tasks, number of completed subtasks for Kitchen) and sample efficiency (number of steps to reach the goal).

B. Overall Results

QDOS consistently outperforms baselines across all evaluated domains. The quantitative results are summarized in Table I, and the training dynamics are shown in Fig. 3.

In the highly complex *antsoccer-arena* environment, QDOS achieves a normalized return of **0.80**, which is a substantial improvement over SUPE’s 0.27. This task requires fine-grained control to manipulate the ball, and standard skill discovery methods likely fail to capture these precise

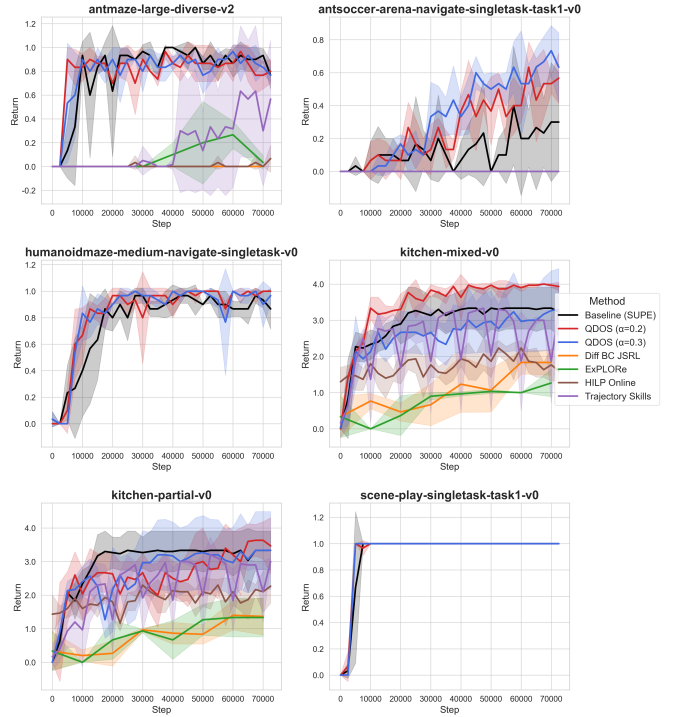


Fig. 3. Learning curves across environments. QDOS is shown in red ($\alpha = 0.2$) and blue ($\alpha = 0.3$); Baseline (SUPE), ExPLORe, Trajectory Skills, HILP Online, and Diff BC JSRL are shown in black, green, purple, brown, and orange, respectively.

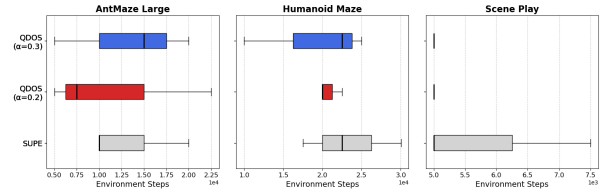


Fig. 4. Training steps to first goal finding. QDOS (Red) finds the goal significantly faster than baselines, indicating more efficient exploration.

interactions amidst the noise of general locomotion. Our advantage-weighting ensures that the skills focus on the moments where the ball is successfully moved.

In *kitchen-mixed*, QDOS attains a perfect score of **4.00**, effectively stitching together fragmented subtasks from the dataset to solve the full sequential task. The baseline SUPE achieves only 3.40, indicating it struggles to compose the necessary skills or fails to learn some subtasks entirely.

In *humanoidmaze*, our method achieves perfect performance (**1.00**), slightly outperforming SUPE. This confirms that even in high-dimensional state spaces, filtering for high-quality skills leads to more robust locomotion.

C. Goal Finding Efficiency

A key advantage of our method is the speed at which it discovers solutions. In sparse-reward tasks, the time to reach the first goal is a critical metric for exploration efficiency. We analyzed the number of environment steps required to reach the goal for the first time.

TABLE I

NORMALIZED EVALUATION RETURNS COMPARING BC, IQL, SUPE, AND QDOS ACROSS VARIOUS DOMAINS. RESULTS FOR SUPE AND QDOS ARE AVERAGED OVER 3 SEEDS.

Environment	BC [11] [26]	IQL [11] [26]	SUPE	QDOS ($\alpha = 0.2$)	QDOS ($\alpha = 0.3$)
antmaze-large	41 ± 7	64 ± 10	83.0 ± 21.0	86.0 ± 6.0	96.0 ± 6.0
antsoccer-arena	5 ± 1	50 ± 2	27.0 ± 31.0	73.0 ± 31.0	80.0 ± 10.0
humanoidmaze-medium	8 ± 2	27 ± 2	97.0 ± 6.0	97.0 ± 6.0	100.0 ± 0.0
kitchen-mixed	51.5	51.0	85 ± 13.25	100.00 ± 0.00	83.25 ± 28.75
kitchen-partial	38.0	46.3	83.25 ± 14.50	85.75 ± 24.75	82.50 ± 30.25
scene-task1	5 ± 1	51 ± 4	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0

TABLE II

TRAINING STEPS TO REACH THE GOAL FOR THE FIRST TIME (MEAN $\pm$ STD).

Environment	Method	Steps
antmaze-large	SUPE	13334 ± 5774
	QDOS ($\alpha = 0.2$)	11668 ± 9465
humanoidmaze	SUPE	23334 ± 6292
	QDOS ($\alpha = 0.3$)	19168 ± 8036
scene	SUPE	5834 ± 1443
	QDOS ($\alpha = 0.2$)	5001 ± 0

As shown in Table II and Fig. 4, QDOS consistently finds the goal faster than SUPE. For instance, in the *scene* task, QDOS reaches the goal in the minimum possible time (5001 steps), while SUPE takes significantly longer (5834 steps). In *humanoidmaze*, our method reduces the exploration time by over 4000 steps. This accelerated exploration is a direct result of the cleaner skill space: the high-level policy does not waste time exploring with noisy or ineffective skills, allowing it to traverse the environment more efficiently.

D. Hyperparameter Sensitivity

We analyzed the sensitivity of our method to the mutual information weight α_{loss} , which balances the reconstruction quality and the diversity of the skills. We observed a task-dependent trade-off.

For complex, dynamic tasks like *antsoccer*, a higher diversity weight ($\alpha_{\text{loss}} = 0.3$) yields the best performance. The solution space for manipulating a soccer ball is large, and encouraging a diverse range of interaction skills increases the likelihood of finding a successful strategy.

Conversely, for structured manipulation tasks like *kitchen*, a moderate diversity weight ($\alpha_{\text{loss}} = 0.2$) performs better. In these tasks, the optimal behaviors are more constrained (e.g., opening a microwave requires a specific motion). Excessive diversity can lead to the learning of task-irrelevant variations that distract the policy. This suggests that QDOS is flexible and can be tuned to the specific needs of the domain.

E. Failure Case

We extended our hyperparameter analysis to high-quality datasets such as *cube-single* and *cube-double* that only require pick-and-place motion. As illustrated in Fig. 5, we observed that reducing the mutual information weight α_{loss} causes our method to converge towards the performance of

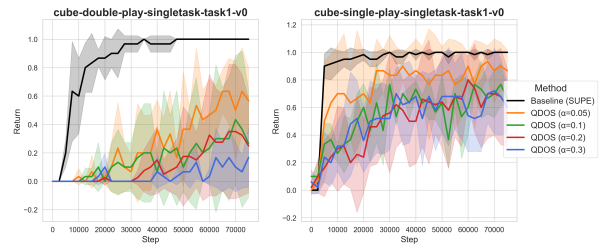


Fig. 5. Experimental results on the cube datasets. Reducing the diversity weight α_{loss} leads to performance closer to the SUPE baseline, indicating that explicit diversity enforcement is less critical for high-quality datasets.

the baseline. This trend reinforces our finding that for datasets with high-quality, structured demonstrations, the need for enforcing diversity via mutual information is diminished.

VI. VISUALIZATION OF THE MODEL

To better understand the properties of the learned skill space and how they contribute to performance, we conducted qualitative visualizations of the latent space and the resulting agent behaviors.

A. Latent Space Structure

We analyzed the structure of the learned latent space by projecting the latent variables from the offline dataset into a 2D space using t-SNE, as shown in Fig. 6. The visualization compares the latent space usage of QDOS (Orange) and SUPE (Blue) across *kitchen*, *antmaze*, and *antsoccer* tasks. We observe that the latent variables for QDOS appear to be distributed more broadly across the space compared to SUPE. This trend suggests that QDOS might be utilizing the latent space more extensively to represent a diverse set of skills, whereas SUPE seems to use the latent space in a more limited or concentrated manner.

B. Behavior Visualization

We also visualized the rollout trajectories of the learned skills in the *kitchen* and *antmaze* domains. In *kitchen*, the learned skills correspond to distinct, semantically meaningful object interactions. For example, specific latent codes consistently trigger the “open microwave” or “move kettle” behaviors. The pseudo-labeling process accurately assigns these skills to the corresponding offline segments, enabling the high-level policy to compose them effectively. As shown in Fig. 7, QDOS reaches the goal at $t = 22$ in the *kitchen-mixed* trajectory, and the trajectory pattern in

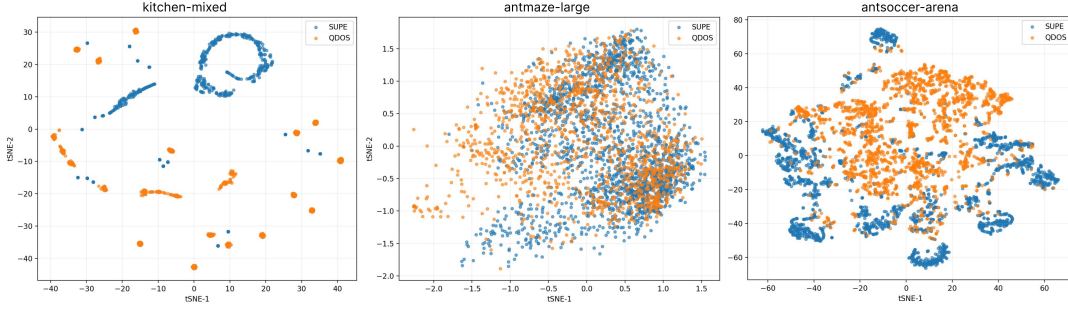


Fig. 6. t-SNE visualization (top) and skill usage timeline (bottom) across *kitchen*, *antmaze*, and *antsoccer*. Orange points represent QDOS ($\alpha = 0.3$ for *antsoccer* and *antmaze*, $\alpha = 0.2$ for *kitchen*), and Blue points represent SUPE. The t-SNE plots indicate that QDOS tends to utilize a broader region of the latent space, suggesting the capture of more diverse behaviors compared to the more restricted latent usage of SUPE.

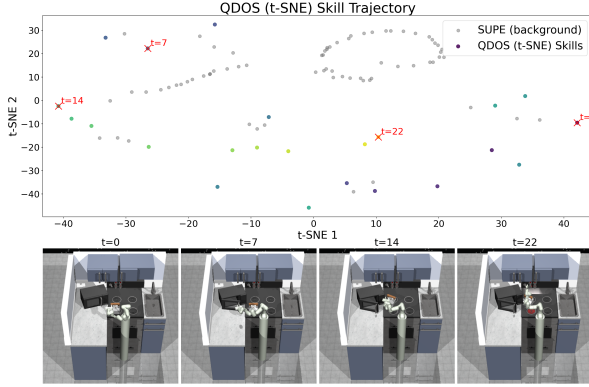


Fig. 7. QDOS skill trajectory visualization for *kitchen-mixed*. Red crosses mark selected timesteps ($t = 0, 7, 14, 22$) with corresponding scene snapshots.

latent space is clearly structured, which is consistent with effective skill composition.

In *antmaze*, the skills represent coherent directional movements (e.g., “move north”, “turn left”). By continuously manipulating the continuous latent variable z (e.g., interpolating between two latent codes), we observed smooth transitions between behaviors, such as gradually changing the movement direction or speed. This confirms that the latent space captures the underlying manifold of useful motions, providing a smooth and navigable action space for the high-level policy. This smoothness is crucial for the stability of the high-level RL training.

VII. CONCLUSION

We presented QDOS, a unified framework for Quality-Diversity offline-to-online Reinforcement Learning. By integrating advantage weighting into the skill pretraining objective, QDOS effectively addresses the challenge of learning from mixed-quality offline datasets. It distills robust, task-aligned skills while filtering out noise and failures, overcoming the limitations of standard unsupervised skill discovery methods.

We demonstrated that combining this robust skill learning with a dual dataset reuse strategy, using offline data for both skill extraction and high-level initialization, leads to signifi-

cant performance gains. Empirical results on challenging domains like *antsoccer*, *kitchen*, and *humanoidmaze* show that QDOS outperforms state-of-the-art baselines in terms of asymptotic return, sample efficiency, and exploration speed.

These findings highlight the importance of quality-aware representation learning in offline RL. Future work will explore adaptive methods to automatically tune the quality-diversity trade-off and investigate mechanisms for fine-tuning the low-level skills during online interaction to further adapt to novel situations.

APPENDIX

A. Environment Details

We evaluate our method on three distinct domains, each presenting unique challenges for skill discovery and hierarchical control.

1) *State-based Locomotion*: *antmaze*, *humanoidmaze*, and *antsoccer* involve controlling complex agents to navigate mazes or manipulate objects.

- *antmaze*: A standard benchmark from D4RL [27] where a quadrupedal ant must navigate a large maze. We use the large maze layout, which requires long-horizon planning.
- *humanoidmaze*: A domain from OGBench [26] featuring a high-dimensional humanoid agent. The agent must balance while navigating, making the skill space significantly more complex than AntMaze.
- *antsoccer*: A multi-stage task where an ant must navigate to a soccer ball and push it to a goal. This requires composing locomotion skills with object interaction dynamics.

2) *State-based Manipulation*: *kitchen*, *cube*, and *scene* focus on robotic manipulation.

- *kitchen*: A sequential manipulation task from D4RL [27] where a 7-DOF Franka robot must manipulate multiple objects (microwave, kettle, burner, switch) in sequence. We use the mixed dataset, which contains various subtasks being performed, but the 4 target subtasks are never completed in sequence together. We also use the partial dataset, which includes other tasks being performed, but there are sub-trajectories where the 4 target subtasks are completed in sequence.

- **cube:** A manipulation domain from OGBench [26] focusing on pick-and-place tasks. We use **cube-single** which requires lifting a single cube, and **cube-double** which involves manipulating two cubes.
- **scene:** A domain from OGBench [26] requiring the composition of atomic behaviors like locking/unlocking and opening drawers.

B. Hyperparameters

We provide the detailed hyperparameters used for the VAE skill pretraining and the high-level online RL agent in Table III and Table IV, respectively.

TABLE III
HYPERPARAMETERS FOR VAE TRAINING.

Parameter Name	Value
Batch size	256
Optimizer	Adam
Learning rate	3×10^{-4}
GRU Hidden Size	256 (antmaze, kitchen) 512 (antsoccer, humanoidmaze, scene)
GRU Layers	2 (antmaze, kitchen) 3 (antsoccer, humanoidmaze, scene)
KL Coefficient (β)	0.1 (antmaze, humanoid, kitchen) 0.2 (scene)
Latent Dimension (z)	8
Segment Length (H)	20

TABLE IV
HYPERPARAMETERS FOR THE HIGH-LEVEL ONLINE RL AGENT.

Parameter	Value
Batch size	256
Discount factor (γ)	0.99 (AntMaze, Kitchen) 0.995 (Humanoid, Scene, Soccer)
Optimizer	Adam
Learning rate	3×10^{-4}
Critic ensemble size	10
UTD Ratio	20
Network Width	256 (AntMaze, Kitchen) 512 (Humanoid, Soccer, Scene)
Network Depth	3 hidden layers
RND coefficient (α)	8.0

C. Acknowledgement

This work was partially supported by JST Moonshot R&D Grant Number JPMJPS2011, CREST Grant Number JPMJCR2015 and Basic Research Grant (Super AI) of Institute for AI and Beyond of the University of Tokyo. Takayuki Osa was supported by JSPS KAKENHI Grant Number JP25K03176.

D. Use of Large Language Models (LLMs)

We utilized large language models to refine the clarity and grammar of the text throughout this manuscript. All scientific content, data, and figures are the original work of the authors.

REFERENCES

- [1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: <http://dx.doi.org/10.1038/nature14236>
- [2] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.
- [3] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.
- [4] A. Ajay, A. Kumar, P. Agrawal, S. Levine, and O. Nachum, "Opal: Offline primitive discovery for accelerating offline reinforcement learning," in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=V69LGWJ0IIN>
- [5] K. Pertsch, Y. Lee, and J. Lim, "Accelerating reinforcement learning with learned skill priors," in *Conference on Robot Learning*. PMLR, 2021, pp. 188–204.
- [6] M. Wilcoxon, Q. Li, K. Frans, and S. Levine, "Leveraging skills from unlabeled prior data for efficient online exploration," *arXiv preprint arXiv:2410.18076*, 2025.
- [7] K. Gregor, D. J. Rezende, and D. Wierstra, "Variational intrinsic control," *arXiv preprint arXiv:1611.07507*, 2016.
- [8] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *arXiv preprint arXiv:1802.06070*, 2018.
- [9] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.
- [10] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.
- [11] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit Q-learning," *arXiv preprint arXiv:2110.06169*, 2021.
- [12] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine, "Efficient online reinforcement learning with offline data," in *International Conference on Machine Learning*. PMLR, 2023, pp. 1577–1594.
- [13] S. Lee, Y. Seo, K. Lee, P. Abbeel, and J. Shin, "Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble," in *Conference on Robot Learning*. PMLR, 2022, pp. 1702–1712.
- [14] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, 2011.
- [15] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.
- [16] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.
- [17] A. Gaier, A. Asteroth, and J.-B. Mouret, "Data-efficient design exploration through surrogate-assisted illumination," *Evolutionary Computation*, vol. 26, no. 3, pp. 381–410, 2018.
- [18] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, "Talakat: Bullet hell generation through constrained map-elites," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 1047–1054.
- [19] T. Osa and T. Harada, "Discovering multiple solutions from a single task in offline reinforcement learning," *Proceedings of the 41st International Conference on Machine Learning*, 2024.
- [20] Q. Li, M. Wilcoxon, K. Frans, and S. Levine, "Skills from unlabeled prior data for efficient exploration," *arXiv preprint arXiv:2410.18076*, 2024.
- [21] D. Barber and F. Agakov, "The im algorithm: a variational approach to information maximization," *Advances in neural information processing systems*, vol. 16, no. 320, p. 201, 2004.
- [22] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.
- [23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.
- [24] Q. Li, J. Zhang, D. Ghosh, A. Zhang, and S. Levine, "Accelerating exploration with unlabeled prior data," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [25] I. Uchendu, T. Xiao, Y. Lu, B. Zhu, M. Yan, J. Simon, M. Bennice, C. Fu, C. Ma, J. Jiao, *et al.*, "Jump-start reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 34 556–34 583.
- [26] S. Park, K. Frans, B. Eysenbach, and S. Levine, "Ogbench: Benchmarking offline goal-conditioned rl," *ArXiv*, 2024.
- [27] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.