

# Robust Policy Learning via Offline Skill Diffusion

Woo Kyung Kim, Minjong Yoo, Honguk Woo\*

Department of Computer Science and Engineering, Sungkyunkwan University  
 {kwk2696, mjyoo2, hwoo}@skku.edu

## Abstract

Skill-based reinforcement learning (RL) approaches have shown considerable promise, especially in solving long-horizon tasks via hierarchical structures. These skills, learned task-agnostically from offline datasets, can accelerate the policy learning process for new tasks. Yet, the application of these skills in different domains remains restricted due to their inherent dependency on the datasets, which poses a challenge when attempting to learn a skill-based policy via RL for a target domain different from the datasets' domains. In this paper, we present a novel offline skill learning framework DuSkill which employs a guided Diffusion model to generate versatile skills extended from the limited skills in datasets, thereby enhancing the robustness of policy learning for tasks in different domains. Specifically, we devise a guided diffusion-based skill decoder in conjunction with the hierarchical encoding to disentangle the skill embedding space into two distinct representations, one for encapsulating domain-invariant behaviors and the other for delineating the factors that induce domain variations in the behaviors. Our DuSkill framework enhances the diversity of skills learned offline, thus enabling to accelerate the learning procedure of high-level policies for different domains. Through experiments, we show that DuSkill outperforms other skill-based imitation learning and RL algorithms for several long-horizon tasks, demonstrating its benefits in few-shot imitation and online RL.

## 1 Introduction

Skill-based learning demonstrates the potentials in accelerating the adaptation to complex long-horizon tasks by leveraging pretrained skill representations on behavior patterns from the offline datasets. However, existing approaches in skill-based reinforcement learning (RL) (e.g., Pertsch, Lee, and Lim 2020; Pertsch et al. 2021) and skill-based few-shot imitation learning (e.g., Hakhamaneshi et al. 2022; Du et al. 2023) often operate under the premise that the target domain for a downstream task was present during skill pretraining. Thus, policies learned with the pretrained skills might lead to sub-optimal performance, particularly when the target domain diverges from the domains of the given datasets.

\*Honguk Woo is the corresponding author.  
 Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

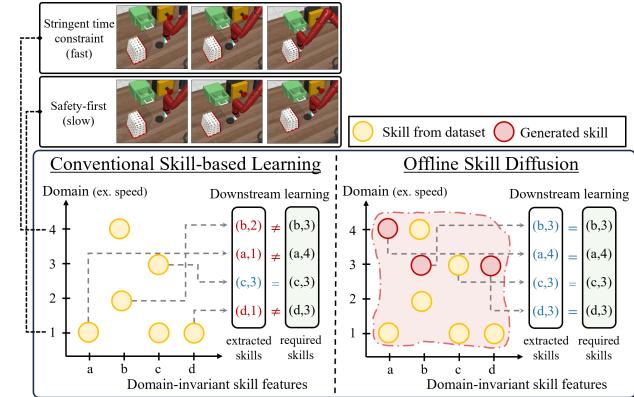


Figure 1: Concept of Offline Skill Diffusion: When a downstream task belongs to the domain different from those of the training datasets, conventional skill-based learning approaches struggle in learning and choosing suitable skills. In contrast, our offline skill diffusion expands the skill diversity that goes beyond the training datasets, enabling the execution of compatible skills for the downstream task. The skills are discretely represented for visual illustration.

As shown in the left side of Figure 1, where each small circle denotes a specific skill, conventional skill learning approaches might experience low performance in the high-level policy for a downstream task, if the task calls for skills that differ from the pretrained ones. For instance, suppose that robotic manipulation skills are learned from the datasets in the safety-first domain; then, they might heavily lean towards “slow” speed manipulation. In that case, a high-level policy learned with these skills might fail to adapt efficiently to the downstream task that involves stringent time constraints. These situations often arise in the environment encompassing diverse domains, as a single task can require different skills depending on the domain it is in. The dependency of conventional skill-based learning approaches on the specificity of the datasets exacerbates these challenges. Moreover, it is practically difficult to obtain comprehensive datasets that span all potential skills for diverse domains.

To tackle these challenges in skill-based learning, we take a novel approach, offline skill diffusion, aiming to broaden skill diversity beyond mere imitation from the

datasets. Given that diffusion models have been recognized for their efficacy in generating human-like images with conditional values (Rombach et al. 2022; Ho and Salimans 2022; Preechakul et al. 2022), we leverage a guided diffusion model for the skill decoder to generate diverse skills. The right side of Figure 1 illustrates the benefits of our framework approach, where the red-colored dotted quadrangle represents an expanded skill set encompassing all the skills that are required to solve the downstream task in a different domain.

In this paper, we present the DuSkill framework, designed to generate diverse skills for downstream tasks that can belong to the domains different from the source domains in the training datasets. Recognizing that certain aspects of a skill remain consistent despite domain variations, we view each skill as a composition of domain-invariant and domain-variant features. Then, we employ a guided diffusion-based decoder along with a hierarchical domain encoder so as to effectively disentangle each skill into two separate embedding spaces. The hierarchical domain encoder systematically segments skills into domain-invariant and domain-variant embeddings by conditioning only the lower encoder on domain variations. With two distinct embeddings, the conditional generation process of the guided diffusion-based decoder enables distinct modulation of each embedding, thereby facilitating the execution of a wide range of skills in different domains.

For downstream tasks in different domains, we train a high-level policy which produces both domain-variant and domain-invariant embeddings. The high-level policy operates alongside the frozen guided diffusion-based decoder, which encompasses the necessary knowledge for generating a broad range of skills, adaptable to various domains. As such, our proposed framework stands apart from existing skill-based learning approaches (Pertsch, Lee, and Lim 2020; Pertsch et al. 2021), as it enables the generation of diverse skills that extend beyond the training datasets.

The contributions of our work are summarized as follows.

- We present the DuSkill framework, which facilitates robust skill-based policy learning for downstream tasks in different domains.
- We develop the offline skill diffusion model, incorporating the hierarchical domain encoder and guided diffusion-based decoder. The model enables the diverse skill generation that extends beyond the training datasets.
- We test the framework with long-horizon tasks in various domains, demonstrating its capability to adapt to different domains in both few-shot imitation and online RL settings.

## 2 Preliminaries and Problem Formulation

Given the training datasets  $\mathcal{D} = \{\tau_i\}_{i \leq n}$ , where each trajectory  $\tau_i$  in the datasets is represented as a sequence of state and action pairs  $\{(s_t, a_t)\}_{t \leq H}$ , the objective of skill representation learning is to accelerate the adaptation to long-horizon tasks by leveraging pretrained skill representations. Here, a skill is defined as a sequence of  $h$  consecutive actions  $\mathbf{a} = \{a_t, \dots, a_{t+h}\}$  (Pertsch, Lee, and Lim 2020;

Hakhamaneshi et al. 2022). Through the joint learning of both a skill encoder  $q(z|\mathbf{a})$  and a skill decoder  $\epsilon(\mathbf{a}|z)$  on the datasets, we are able to obtain a skill embedding  $z \in \mathcal{Z}$ . The learning objective involves optimizing the evidence lower bound (ELBO), which consists of a reconstruction term and a regularization term,

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{\mathbf{a} \sim \mathcal{D}} [-\log \epsilon(\mathbf{a}|z) + \beta D_{\text{KL}}(p(z)||q(z|\mathbf{a}))] \quad (1)$$

where  $D_{\text{KL}}$  is the Kullback-Leibler (KL) divergence,  $p(z)$  is a prior following a unit Gaussian distribution  $\mathcal{N}(0, I)$  and  $\beta$  is a hyperparameter for regularization (Higgins et al. 2016). Then, this skill representation is leveraged to accelerate the downstream task adaptation.

**Problem Formulation.** In our problem formulation, we operate under the premise that training datasets  $\mathcal{D}$  are available. As skills required for a task might vary depending on the domain it belongs to, we further assume that the datasets are collected from multiple source domains. The variations across the domains are parameterized by  $\omega \in \Omega$ , and this parameter information is incorporated in the datasets. Then, we aim to tackle downstream tasks in a range of domains distinct from the source domains. Successfully achieving this requires more than merely imitating the skills present in the given datasets, due to the varied nature of the tasks across different domains.

We formulate a downstream task as a goal-conditioned Markov decision process (MDP)  $\mathcal{M}$  combined with its domain  $\Omega$ . The goal-conditioned MDP  $\mathcal{M}$  is denoted as  $(S, A, P, r, \mathcal{G}, \gamma, \mu_0)$  where  $s \in S$  is a state space,  $a \in A$  is an action space,  $\mathcal{G}$  is a goal space,  $P : S \times A \times \Omega \rightarrow [0, 1]$  is a transition probability,  $r : S \times A \times \mathcal{G} \times \Omega \rightarrow \mathbb{R}$  is a reward function,  $\gamma \in [0, 1]$  is a discount factor, and  $\mu_0 : S \rightarrow [0, 1]$  is an initial state distribution. The domain variations may affect either the reward function or the transition probability in an MDP, while the goal space remains consistent. For instance, on the top of Figure 1, a task may have the same goal space as slide puck in the goalposts, but the domain related to the time constraint results in different reward functions.

Then, our objective is to maximize the discounted cumulative sum of rewards for downstream tasks in different domains, through a high-level policy  $\pi(z|s)$ ,

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{z \sim \pi(\cdot|s), \mathbf{a} \sim \epsilon(\cdot|z)} \left[ \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) \right] \quad (2)$$

where  $\epsilon(\mathbf{a}|z)$  is a skill decoder and  $T$  is the maximum length for an episode.

## 3 Approach

### 3.1 DuSkill Framework

To facilitate the diverse skill generation, we propose the DuSkill framework consisting of two main phases: (i) the offline skill diffusion phase, and (ii) the downstream policy learning phase, as illustrated in Figure 2.

In the offline skill diffusion phase, we employ a guided diffusion-based decoder in conjunction with a hierarchical domain encoder to disentangle skills into domain-invariant and domain-variant embeddings. Specifically, the hierarchical domain encoder consists of a domain-invariant encoder

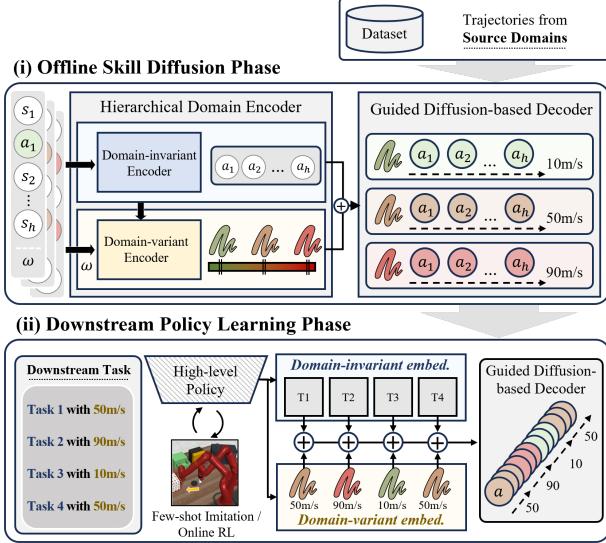


Figure 2: Offline Skill Diffusion and Downstream Policy Learning in DuSkill: (i) In the offline skill diffusion phase, a skill is decomposed into domain-invariant and domain-variant embeddings, and then they are combined through the guided diffusion based decoder to generate diverse skills. (ii) In the downstream policy learning phase, a high-level policy is learned for a task in different domains either through few-shot imitation or online RL.

and a domain-variant encoder. The domain-invariant encoder processes a sequence of state and action, resulting in domain-invariant embedding. Meanwhile, the domain-variant encoder takes the domain-invariant embedding and the domain parameterization as input to produce the domain-variant embedding. In this way, these encoders play distinct roles, in which domain-invariant encoder is responsible for encapsulating features necessary to reconstruct fundamental action sequences pertinent to achieving goals, while the domain-variant encoder is tailored to capture the features related to domain variations. To effectively disentangle skill features via the hierarchical domain encoder, we utilize a guided diffusion-based decoder with two components, where one is conditioned on the domain-invariant embedding, while the other focuses on the domain-variant embedding. This conditional generation mechanism facilitates a distinct influence of the domain-invariant and domain-variant embeddings on the separate segments of generated action sequences. By employing the hierarchical domain encoder and guided diffusion-based decoder, our framework is capable of generating diverse action sequences that encompass different combinations of the domain-invariant and domain-variant features.

In the downstream policy learning phase, we exploit the disentangled embeddings via a high-level policy, which produces both domain-invariant and domain-variant embeddings. These embeddings are then fed into the frozen guided diffusion-based decoder, which generates executable skills. In this phase, we consider few-shot imitation and online RL

adaptation scenarios, where the high-level policy adapts to the tasks in different domains through either fine-tuning on a limited number of trajectories or online RL interactions.

### 3.2 Offline Skill Diffusion

**Hierarchical domain encoder.** To disentangle domain-invariant and domain-variant features from skills, we introduce a hierarchical encoding approach. This enables learning in two distinct embedding spaces: the domain-invariant embedding space  $\mathcal{Z}_\rho$  and the domain-variant embedding space  $\mathcal{Z}_\sigma$ . Specifically, we employ a domain-invariant encoder  $q_\rho$  which maps a sequence of states and actions to the domain-invariant embedding. We also use a domain-variant encoder  $q_\sigma$  which maps the domain-invariant embedding and the domain parameterization  $\omega$  to the domain-variant embedding, i.e.,

$$z_\rho \sim q_\rho(\mathbf{s}, \mathbf{a}), z_\sigma \sim q_\sigma(z_\rho, \omega) \quad (3)$$

where  $\mathbf{s} = \{s_t, \dots, s_{t+h}\}$  is a sequence of states,  $\mathbf{a} = \{a_t, \dots, a_{t+h}\}$  is a sequence of actions,  $z_\rho \in \mathcal{Z}_\rho$  is the domain-invariant embedding, and  $z_\sigma \in \mathcal{Z}_\sigma$  is the domain-variant embedding. To optimize the encoders and the skill decoder  $\epsilon$ , we employ the evidence lower bound (ELBO) loss using (1), similar to (Pertsch et al. 2021; Hakhamaneshi et al. 2022), i.e.,

$$\begin{aligned} \mathcal{L}_{\text{HVAE}} = & \mathbb{E}_{(\mathbf{s}, \mathbf{a}) \sim \mathcal{D}} \left[ - \prod_{t=0}^h \log \epsilon(a_t | s_t, z_\rho, z_\sigma) \right. \\ & + \beta_\rho D_{\text{KL}}(p(z_\rho) || q_\rho(z_\rho | \mathbf{s}, \mathbf{a})) \\ & \left. + \beta_\sigma D_{\text{KL}}(p(z_\sigma | z_\rho) || q(z_\sigma | z_\rho, \mathbf{s}, \mathbf{a})) \right] \end{aligned} \quad (4)$$

where  $q(z_\sigma | z_\rho, \mathbf{s}, \mathbf{a})$  is a naive modeling for the domain-variant encoder, and  $\beta_\rho$  and  $\beta_\sigma$  are regularization hyperparameters. The prior  $p(z_\sigma)$  and  $p(z_\rho | z_\sigma)$  are set to be unit Gaussian. To disentangle skill features, we modify the regularization term of the domain-variant encoder in (4) as

$$\mathcal{L}_{\text{aspect}} = \mathbb{E}_{(\mathbf{s}, \mathbf{a}) \sim \mathcal{D}} \left[ \log \frac{q(z_\sigma | z_\rho, \mathbf{s}, \mathbf{a})}{q_\sigma(z_\sigma | z_\rho, \omega)} \right]. \quad (5)$$

This loss term enables the domain-variant encoder to construct the distinct embedding space  $\mathcal{Z}_\sigma$ , capturing domain-variant features conditioned on the domain-invariant embedding. By combining (4) and (5), we rewrite the loss as

$$\begin{aligned} \mathcal{L}_{\text{DHVAE}} = & \mathbb{E}_{(\mathbf{s}, \mathbf{a}) \sim \mathcal{D}} \left[ - \prod_{t=0}^h \log \epsilon(a_t | s_t, z_\rho, z_\sigma) \right. \\ & + \beta_\rho D_{\text{KL}}(p(z_\rho) || q_\rho(z_\rho | \mathbf{s}, \mathbf{a})) \\ & \left. + \beta_\sigma D_{\text{KL}}(p(z_\sigma | z_\rho) || q_\sigma(z_\sigma | z_\rho, \omega)) \right]. \end{aligned} \quad (6)$$

For downstream policy learning, we employ a domain-invariant prior  $p_\rho$  and a domain-variant prior  $p_\sigma$ .

$$z_\rho \sim p_\rho(s_t), z_\sigma \sim p_\sigma(z_\rho) \quad (7)$$

The domain-invariant prior  $p_\rho$  is conditioned on the current state  $s_t$ , facilitating the selection of suitable domain-invariant embedding, while the domain-variant prior  $p_\sigma$

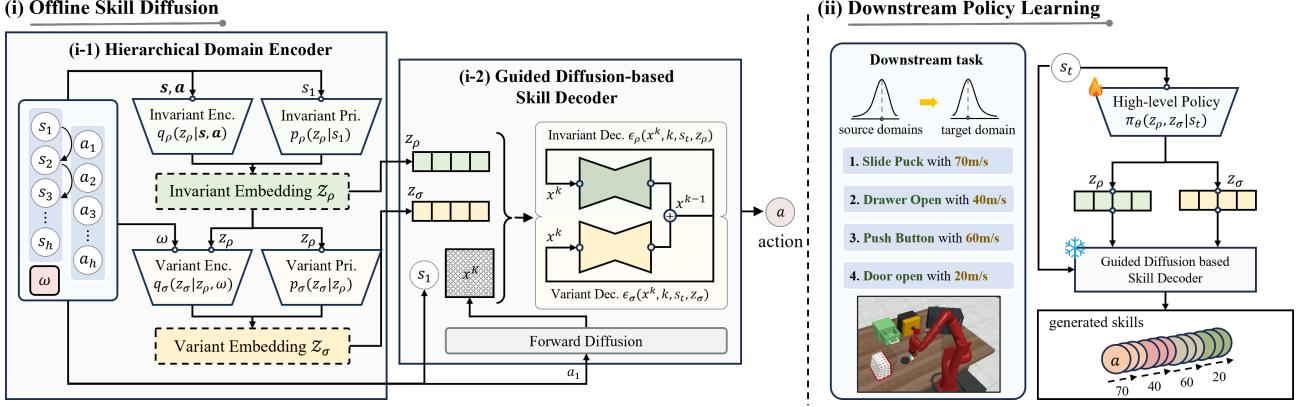


Figure 3: DuSkill Framework: In (i-1), the hierarchical domain encoder disentangles the domain-invariant and domain-variant embeddings. At the same time, the domain-invariant and domain-variant priors are jointly learned with these encoders. For diverse skill generation, the encoders are trained in conjunction with the guided diffusion-based decoder in (i-2). Here, the domain-invariant decoder and domain-variant decoder are responsible for reconstructing actions based on the domain-invariant and domain-variant embeddings, respectively. In (ii), the high-level policy is learned to solve the task in different domain either through few-shot imitation or online RL.

provides a prior distribution over the domain-variant embedding. By designing a domain-variant prior conditioned solely on the domain-invariant embedding, our model achieves the flexibility to explore a wide range of parameterizations across different domains. These priors are jointly trained with the hierarchical domain encoder by minimizing the KL divergence between each respective encoder.

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{(\mathbf{s}, \mathbf{a}) \sim \mathcal{D}} [D_{\text{KL}}(p_{\rho}(z_{\rho} | s_t) || q_{\rho}(z_{\rho} | \mathbf{s}, \mathbf{a})) + D_{\text{KL}}(p_{\sigma}(z_{\sigma} | z_{\rho}) || q_{\sigma}(z_{\sigma} | z_{\rho}, \omega))] \quad (8)$$

**Guided diffusion-based decoder.** To generate diverse skills from domain-invariant embedding  $z_{\rho}$  and domain-variant embedding  $z_{\sigma}$ , we employ the diffusion model (HO, Jain, and Abbeel 2020; Pearce et al. 2023; Wang, Hunt, and Zhou 2023; Ajay et al. 2023). In particular, we adopt the denoising diffusion probabilistic model (DDPM) (HO, Jain, and Abbeel 2020) to represent our skill decoder. The decoder reconstructs an action  $a_t$  from a noisy input  $x^K \sim \mathcal{N}(0, I)$  by sequentially predicting  $x^{K-1}, x^{K-2}, \dots, x^0 (= a_t)$ , with each iteration being a marginally denoised version of its predecessor. During the training phase, the noisy input  $x^k$  is generated by progressively adding the Gaussian noise to the original action  $a_t (= x^0)$  over  $K$  steps as

$$x^k = \sqrt{\bar{\alpha}^k} a_t + \sqrt{1 - \bar{\alpha}^k} \eta \quad (9)$$

where  $\eta \sim \mathcal{N}(0, I)$  and  $\bar{\alpha}^k$  is a variance schedule. As our skill decoder is designed to predict the noise  $\eta$ , the reconstruction loss in (6) becomes an  $L_2$  distance between the decoder’s output and the noise.

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{k \sim [1, K], \eta \sim \mathcal{N}(0, I)} [\|\epsilon(x^k, k, s_t, z_{\sigma}, z_{\rho}) - \eta\|_2^2] \quad (10)$$

To further align with the objective in (6), we slightly modify the classifier-free guidance (Ho and Salimans 2022) to divide the decoder into two separate parts: domain-invariant decoder  $\epsilon_{\rho}(x^k, k, s_t, z_{\rho})$  and domain-variant decoder  $\epsilon_{\sigma}(x^k, k, s_t, z_{\sigma})$ . Thus, our decoder is redefined as

#### Algorithm 1: Offline Skill Diffusion

**Input:** Training Datasets  $\mathcal{D}$ , total denoise step  $K$ , guidance weight  $\delta$ , hyperparameters  $\beta_{\rho}, \beta_{\sigma}$

- 1: Initialize encoders  $q_{\rho}, q_{\sigma}$ , priors  $p_{\rho}, p_{\sigma}$ , decoders  $\epsilon_{\rho}, \epsilon_{\sigma}$
- 2: **while** not converge **do**
- 3:     Sample a batch  $\{(\mathbf{s}, \mathbf{a}, \omega)\}_i \sim \mathcal{D}$
- 4:     Update  $q_{\rho}$  and  $q_{\sigma}$  using  $\mathcal{L}_{\text{DHVAE}}$  in (6)
- 5:     Update  $p_{\rho}$  and  $p_{\sigma}$  using  $\mathcal{L}_{\text{prior}}$  in (8)
- 6:     Update  $\epsilon_{\rho}$  and  $\epsilon_{\sigma}$  using  $\mathcal{L}_{\text{rec}}$  in (10)
- 7: **end while**
- 8: **return**  $q_{\rho}, q_{\sigma}, p_{\rho}, p_{\sigma}, \epsilon_{\rho}, \epsilon_{\sigma}$

the combination of both a domain-invariant decoder and a domain-variant decoder, i.e.,

$$\epsilon(x^k, k, s_t, z_{\rho}, z_{\sigma}) := (1 - \delta)\epsilon_{\rho}(x^k, k, s_t, z_{\sigma}) + \delta\epsilon_{\sigma}(x^k, k, s_t, z_{\rho}) \quad (11)$$

where  $\delta > 0$  serves as a guidance weight that determines the degree of adjustment towards the domain-variant decoder. This approach allows our domain-invariant decoder to reconstruct actions that consistently execute the designated task across domain features. Simultaneously, the domain-variant decoder is seamlessly incorporated by generating guidance that encapsulates domain-variant features.

In our framework, we employ the hierarchical domain encoder to establish distinct embedding spaces for domain-invariant and domain-variant features, optimized by the loss defined in (6). Concurrently, the respective priors for the downstream task are jointly trained, optimized by the loss in (8), as depicted in Figure 3 (i-1). Furthermore, to effectively disentangle the domain-invariant and domain-variant embeddings, we jointly train the guided diffusion-based decoder in conjunction with the encoders, optimized by the loss (10), as illustrated in Figure 3 (i-2). Algorithm 1 lists

the learning procedures of DuSkill.

### 3.3 Downstream Policy Learning

For efficient policy learning on downstream tasks, we adopt a hierarchical learning scheme akin to other skill-based approaches (Pertsch, Lee, and Lim 2020; Hakhamaneshi et al. 2022). In this scheme, the higher-level policy is employed to produce the skill embedding as output rather than directly generating executable actions. Specifically, we train a policy  $\pi(z_\rho, z_\sigma|s_t)$ , designed to align with the hierarchical domain encoder, which generates domain-invariant and domain-variant embeddings,

$$\pi(z_\rho, z_\sigma|s_t) = \pi_\rho(z_\rho|s_t) \cdot \pi_\sigma(z_\sigma|z_\rho). \quad (12)$$

Subsequently, these embeddings are fed into the learned guided diffusion-based decoder, which remains frozen to decode them into a sequence of executable actions. To decode an action from the guided diffusion-based decoder, the process starts with sampling a noisy input  $x^K \sim \mathcal{N}(0, I)$ . Then, the decoder iteratively denoises the input while conditioning the decoder on the disentangled embeddings to generate an action,

$$x^{k-1} = \frac{1}{\sqrt{\alpha^k}} \left( x^k - \frac{1 - \alpha^k}{\sqrt{1 - \alpha^k}} \epsilon(x^k, k, s_t, z_\rho, z_\sigma) \right) + \zeta^k \eta \quad (13)$$

where  $\eta \sim \mathcal{N}(0, I)$ ,  $\zeta^k$  and  $\alpha^k$  are parameters for variance schedule. As it is empirically observed that the low-temperature sampling leads to improved performance, similar to (Ajay et al. 2023), we set  $\zeta^k = 0$ .

For downstream task adaptation, we explore two different scenarios: few-shot imitation and online RL adaptation. For the few-shot imitation, we initialize the high-level policy with the learned domain-invariant and domain-variant priors, and then we fine-tune the high-level policy using (10) along with the learned guided diffusion-based decoder. Likewise, for online RL adaptation, we adopt the soft actor-critic (SAC) algorithm, where the learned priors guide the high-level policy as in (Pertsch, Lee, and Lim 2020). In both scenarios, we freeze the decoder and solely fine-tune the high-level policy. Even in the absence of decoder updates, our framework manages to attain robust performance on downstream tasks in different domains, as demonstrated in Section 4.2. Figure 3 (ii) illustrates the procedure of downstream policy learning.

## 4 Evaluations

### 4.1 Experiment Settings

**Environments.** For evaluation, we use the multi-stage Meta-World, which is implemented based on the Meta-World simulated benchmark (Yu et al. 2019). Each multi-stage task is composed of a sequence of existing Meta-World tasks (sub-tasks). In these multi-stage tasks, an agent is required to maneuver a robotic arm to complete a series of sub-tasks, such as slide puck, close drawer, and etc. To emulate different domains in the environment, we deliberately add variations to either reward functions or transition dynamics. Specifically, we modify the reward function by setting time constraints

(i.e., speed domains) and by incorporating the energy usage consideration (energy domains). Furthermore, to emulate varying conditions of transition dynamics in the environment, we manipulate kinematic parameters such as wind (wind domains).

**Datasets.** For datasets, we implement several rule-based expert policies tailored to each domain-specific environment. For offline training datasets, we collect 20 trajectories for each source domain (of 6 ~ 16 domains). For few-shot imitation datasets, we collect another 3 trajectories for each target domain.

**Baselines.** For baselines, we implement several imitation learning and skill-based RL algorithms.

- BC (Esmaili, Sammut, and Shirazi 1998) is a widely used supervised behavior cloning method. A policy is learned on the training datasets and then fine-tuned for the downstream tasks.
- SPiRL (Pertsch, Lee, and Lim 2020) is a state-of-art skill-based RL algorithm that employs a hierarchical structure to embeds skills into a latent space, thereby accelerating the downstream adaptation.
- SPiRL-c is a variant of SPiRL that uses the closed-loop skill decoder, used in (Pertsch et al. 2021).
- FIST (Hakhamaneshi et al. 2022) is a state-of-art few-shot skill-based imitation algorithm that employs a semi-parametric approach for skill determination.

To obtain few-shot imitation, the baselines such as BC, SPiRL and SPiRL-c are pretrained on the training datasets and then fine-tuned on the few-shot imitation datasets.

### 4.2 Few-shot Imitation Performance

Table 1 compares the few-shot imitation performance in rewards achieved by our framework (DuSkill) and other baselines (BC, SPiRL, SPiRL-c, FIST). Among the baselines, those denoted with an asterisk (\*) signify that both the high-level policy and the decoder are fine-tuned, while those without an asterisk indicate fine-tuning solely of the high-level policy. An important distinction is that DuSkill exclusively focuses on fine-tuning the high-level policy. Based on the domain disparities between the datasets and the downstream tasks, we categorize them into four different levels. At the source level, downstream tasks remain same with the source domains. As we move to higher levels, the domain disparities become more pronounced.

As shown, the baselines exhibit a notable decline in performance as the domain dissimilarity increases from source to level 4, where even the most competitive baseline FIST achieves an average degradation of 25.3%. In contrast, DuSkill consistently maintains robust performance across all domains and levels, achieving a small degradation of 7.6% at average.

In this experiment, SPiRL-c demonstrates relatively low performance, primarily because its decoder can only generate skills present in its training data. Consequently, this poses a challenge in attaining robust performance when solely relying on fine-tuning the high-level policy. Meanwhile, SPiRL-c\* is expected to yield higher performance

Domain	Level	BC	SPiRL*	SPiRL-c	SPiRL-c*	FIST*	DuSkill
Speed	Source	2.66 $\pm$ 0.73	2.16 $\pm$ 0.63	3.62 $\pm$ 0.35	3.72 $\pm$ 0.19	3.98 $\pm$ 0.01	<b>3.99 <math>\pm</math> 0.00</b>
	Level 1	2.71 $\pm$ 0.28	1.83 $\pm$ 0.50	3.41 $\pm$ 0.50	3.51 $\pm$ 0.25	3.86 $\pm$ 0.05	<b>3.92 <math>\pm</math> 0.07</b>
	Level 2	2.62 $\pm$ 0.54	2.16 $\pm$ 0.62	3.33 $\pm$ 0.44	3.24 $\pm$ 0.23	3.51 $\pm$ 0.12	<b>3.83 <math>\pm</math> 0.06</b>
	Level 3	2.22 $\pm$ 0.23	2.22 $\pm$ 0.49	2.87 $\pm$ 0.47	3.12 $\pm$ 0.22	3.28 $\pm$ 0.19	<b>3.81 <math>\pm</math> 0.06</b>
Energy	Source	1.49 $\pm$ 0.35	0.67 $\pm$ 0.07	2.83 $\pm$ 0.39	1.60 $\pm$ 0.33	3.08 $\pm$ 0.19	<b>3.85 <math>\pm</math> 0.08</b>
	Level 1	1.26 $\pm$ 0.32	0.56 $\pm$ 0.07	1.90 $\pm$ 0.45	1.19 $\pm$ 0.13	2.53 $\pm$ 0.25	<b>3.87 <math>\pm</math> 0.09</b>
	Level 2	0.90 $\pm$ 0.15	0.47 $\pm$ 0.08	1.77 $\pm$ 0.26	0.91 $\pm$ 0.19	1.85 $\pm$ 0.34	<b>3.71 <math>\pm</math> 0.10</b>
	Level 3	0.53 $\pm$ 0.09	1.69 $\pm$ 0.25	0.89 $\pm$ 0.18	2.02 $\pm$ 0.24	1.04 $\pm$ 0.20	<b>3.67 <math>\pm</math> 0.14</b>
Wind	Source	3.32 $\pm$ 0.41	3.78 $\pm$ 0.10	2.83 $\pm$ 0.39	3.92 $\pm$ 0.10	<b>4.00 <math>\pm</math> 0.00</b>	3.98 $\pm$ 0.02
	Level 1	2.99 $\pm$ 0.52	3.14 $\pm$ 0.54	1.90 $\pm$ 0.45	3.71 $\pm$ 0.29	<b>3.89 <math>\pm</math> 0.15</b>	3.78 $\pm$ 0.13
	Level 2	2.19 $\pm$ 0.42	2.62 $\pm$ 0.33	1.77 $\pm$ 0.26	3.24 $\pm$ 0.24	3.24 $\pm$ 0.24	<b>3.48 <math>\pm</math> 0.21</b>
	Level 3	2.41 $\pm$ 0.45	2.63 $\pm$ 0.46	0.89 $\pm$ 0.18	3.04 $\pm$ 0.29	3.04 $\pm$ 0.29	<b>3.44 <math>\pm</math> 0.18</b>

Table 1: Few-shot Imitation Performance in Multi-stage Meta-World: The performance of the baselines and DuSkill is measured in achieved rewards. For each domain, we categorize domain disparity between the training datasets and downstream tasks into four different levels. The baselines marked with an asterisk (\*) indicate that both the high-level policy and decoder are fine-tuned, while the baselines without an asterisk and DuSkill only fine-tune the high-level policy. Each is evaluated with 3 random seeds, and the highest performance is highlighted in bold.

than SPiRL-c as it fine-tunes both the high-level policy and the decoder; yet, in some cases, SPiRL-c surpasses SPiRL-c\*. This is because fine-tuning the entire model with a few samples might cause a covariant shift, a phenomenon commonly observed in the few-shot imitation studies (Hakhamaneshi et al. 2022; Nasiriany et al. 2022). FIST\* adopts a different strategy, involving the fine-tuning of the entire model along with the utilization of a semi-parametric method to retrieve the future state  $s_{t+H}$  that it aims to reach from the training datasets. While the semi-parametric method leads to improved performance for tasks in source domains, FIST\* is prone to fail in producing skills for downstream tasks in different domains. This is because the training datasets do not cover the skills required for different domains. In contrast, DuSkill disentangles the domain-invariant and -variant features to effectively generate the skills through the guided diffusion-based decoder. This allows for robust performance in few-shot adaptation across different domains, through fine-tuning solely the high-level policy.

### 4.3 Analysis

**Online RL.** Table 2 compares online RL adaptation performance in reward achieved by our framework (DuSkill) and other baselines (BC+SAC, SPiRL, SPiRL-c). Both the baselines and DuSkill fine-tune the high-level policy via the SAC algorithm. Here, we categorize domain disparities into source and target domains, where target domains correspond to the level 3 in Table 1. For more stable learning in target domains, we warm-up the high-level policy with a single trajectory for DuSkill and other baselines. The performance gap between SPiRL-c and DuSkill is not remarkable in source domain tasks, as expected. In contrary, DuSkill exhibits superior performance compared to the baselines for tasks in different domains, outperforming SPiRL-c by 89.16%. This highlights the capability of our

guided diffusion-based decoder in generating diverse skills that extend beyond the limitations of the given datasets.

Level	BC+SAC	SPiRL-c	DuSkill
Source	0.00 $\pm$ 0.00	4.00 $\pm$ 0.00	<b>4.00 <math>\pm</math> 0.00</b>
Target	0.00 $\pm$ 0.00	0.36 $\pm$ 0.02	<b>3.32 <math>\pm</math> 0.20</b>

Table 2: Online RL Performance in Speed Domain

**Embedding Visualization.** Here, we verify the efficacy of our DuSkill framework in disentangled embeddings. Figure 4 visualizes the domain-invariant and domain-variant embeddings generated by DuSkill separately in two distinct speed domains (fast and slow). In the figure, the labels T1 to T4 correspond to sub-tasks numbered from 1 to 4. Regarding the domain-invariant embeddings, we observe that the identical tasks are paired together, thereby establishing the domain-invariant knowledge. On the other hand, the domain-variant embeddings are grouped with respect to the domains, implying the encapsulation of domain-variant knowledge. This indicates the effectiveness of our offline skill diffusion process, which disentangles domain-invariant and variant features.

**Sample Efficiency.** Figure 5 shows the performance with respect to samples (or timesteps) used by DuSkill and the baselines (SPiRL-c, SPiRL-c\*, FIST) for downstream policy learning in few-shot imitation and online RL scenarios. For the few-shot imitation learning, we test with different numbers of few-shot trajectories (1  $\sim$  20). As shown in Figure 5(a), DuSkill exhibits robust performance with only a 4.89% drop from 1 to 20 trajectories, whereas the most competitive baseline FIST shows a notable performance drop of 13.96%. Furthermore, as shown in Figure 5(b), DuSkill efficiently adapts to the downstream task in online settings, enhancing performance with only 50k samples, while SPiRL-c

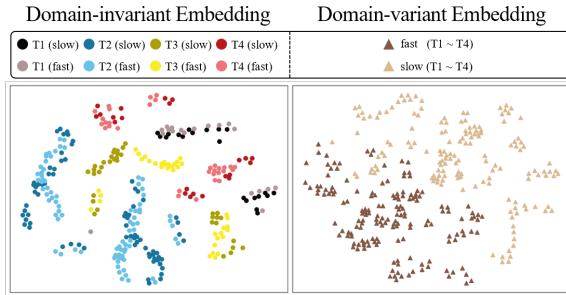


Figure 4: Visualization of Domain-invariant and Domain-variant embeddings

rarely achieves improvement with those samples.

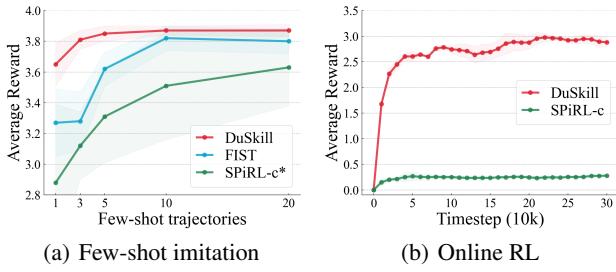


Figure 5: Sample Efficiency of Downstream Policy Learning

**Ablation on Skill Diffusion.** Table 3 provides an ablation study of DuSkill, focusing on the impact of the hierarchical embedding structure and guided diffusion-based decoder. In this study, we conduct few-shot imitation scenarios with two ablated variants. DU is a variant of SPIRL-c, utilizing a naive diffusion model for the skill decoder, while HDU employs the hierarchical embedding structure like DuSkill and a naive diffusion model for the skill decoder. The results show that the combination of hierarchical domain encoder and guided diffusion based decoder in DuSkill yields improved performance compared to the other variants, showing  $8.77 \sim 8.92\%$  average performance gains. This is not surprising since the guided diffusion based decoder promotes the learning of disentangled representations, as the domain-invariant and domain-variant decoders are conditioned on each embedding to generate executable actions effectively. Therefore, it is crucial to employ both hierarchical embedding structure and guided diffusion based decoder for achieving better few-shot imitation learning performance.

Domain	DU	HDU	DuSkill
Speed	$3.59 \pm 0.19$	$3.49 \pm 0.68$	<b><math>3.81 \pm 0.06</math></b>
Energy	$3.16 \pm 0.35$	$3.39 \pm 0.36$	<b><math>3.67 \pm 0.14</math></b>
Wind	$3.19 \pm 0.17$	$3.08 \pm 0.36$	<b><math>3.44 \pm 0.18</math></b>

Table 3: Performance by Encoder and Decoder Types

## 5 Related Work

**Skill-based Learning.** To leverage offline datasets for long-horizon tasks, hierarchical skill representation learning techniques have been investigated in the context of online RL (Pertsch, Lee, and Lim 2020; Pertsch et al. 2021) and imitation learning (Hakhamaneshi et al. 2022; Nasiriany et al. 2022; Du et al. 2023). Pertsch, Lee, and Lim (2020) proposed the hierarchical skill learning structure to accelerate the downstream task adaptation by guiding a high-level policy with learned skill priors. Meanwhile, Hakhamaneshi et al. (2022) exploited a semi-parametric approach within this hierarchical skill structure, focusing on the few-shot imitation. In our work, we also utilize skill embedding techniques, but we tackle the challenge of adapting to downstream tasks in different domains. Unlike the prior work, our DuSkill adapts a robust generative model such as diffusion with the technique of disentangled skill embeddings, enabling the effective generation of diverse skills in offline.

**Diffusion for RL.** Given the remarkable success of diffusion models in the field of computer vision (HO, Jain, and Abbeel 2020; Rombach et al. 2022), their application has been extended to sequential decision-making problems in recent years. Pearce et al. (2023) utilized diffusion models to imitate human demonstrations, capitalizing on their capacity to represent highly multi-modal data. Wang, Hunt, and Zhou (2023) adopted diffusion models in the context of offline RL to implement policy regularization. Furthermore, Liang et al. (2023) leveraged diffusion models to generate diverse synthetic trajectories on limited training data, aiming to have self-evolving offline RL for goal-conditioned RL tasks. Recently, Ajay et al. (2023) proposed a general framework for sequential decision-making problems using diffusion models. It allows for dynamic recombination of behaviors at test time by conditioning the diffusion models on several factors such as returns, constraints, or skills. To the best of our knowledge, our DuSkill framework is the first to integrate a diffusion model with skill embedding techniques, providing a novel hierarchical RL method to generate diverse skills on limited datasets and achieving robust performance for downstream tasks in different domains.

## 6 Conclusion

In this work, we presented the DuSkill framework, a novel approach designed to bridge the gap between the given datasets and downstream tasks that exist within different domains. We devised the offline skill diffusion, which employs the guided diffusion-based decoder in conjunction with the hierarchical encoders to effectively disentangle domain-invariant and -variant features from skills. This enables the generation of diverse skills capable of addressing tasks in different domains. Our framework stands apart from existing skill-based learning approaches, which are typically limited to adapt within the domains encountered during skill pre-training. In our future work, we aim to extend our framework to address challenging cross-domain situations with significant domain shifts, such as entirely different tasks, robot embodiment variations, or different simulation environments.

## Acknowledgments

We would like to thank anonymous reviewers for their valuable comments and suggestions. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-01045, 2022-0-00043, 2019-0-00421, 2021-0-00875) and by the National Research Foundation of Korea (NRF) grant funded by the MSIT (No. RS-2023-00213118).

## References

Ajay, A.; Du, Y.; Gupta, A.; Tenenbaum, J. B.; Jaakkola, T. S.; and Agrawal, P. 2023. Is Conditional Generative Modeling All You Need for Decision-Making? In *Proceedings of the 11th International Conference on Learning Representations*.

Du, M.; Nair, S.; Sadigh, D.; and Finn, C. 2023. Behavior Retrieval: Few-Shot Imitation Learning by Querying Unlabeled Datasets. arXiv:2304.08742.

Esmaili, N.; Sammut, C.; and Shirazi, G. M. 1998. Behavioural cloning in control of a dynamic system. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics.*, 2904–2909.

Hakhamaneshi, K.; Ruihan Zhao, A. Z.; Abbeel, P.; and Laskin, M. 2022. Hierarchical Few-Shot Imitation with Skill Transition Models. In *Proceedings of the 10th International Conference on Learning Representations*.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C. P.; Glorot, X.; Botvinick, M. M.; Mohamed, S.; and Lerchner, A. 2016. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *Proceedings of the 4th International Conference on Learning Representations*.

HO, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Proceedings of the 34th Conference on Neural Information Processing System*.

Ho, J.; and Salimans, T. 2022. Classifier-Free Diffusion Guidance. arXiv:2207.12598.

Liang, Z.; Mu, Y.; Ding, M.; Ni, F.; Tomizuka, M.; and Luo, P. 2023. AdaptDiffuser: Diffusion Models as Adaptive Self-evolving Planners. In *Proceedings of the 40th International Conference on Machine Learning*.

Nasiriany, S.; Gao, T.; Mandlekar, A.; and Zhu, Y. 2022. Learning and Retrieval from Prior Data for Skill-based Imitation Learning. In *Proceedings of the 6th Conference on Robot Learning*, 2181–2204.

Pearce, T.; Rashid, T.; Kanervisto, A.; Bignell, D.; Sun, M.; Georgescu, R.; Macua, S. V.; Tan, S. Z.; Momennejad, I.; Hofmann, K.; and Devlin, S. 2023. Imitating Human Behaviour with Diffusion Models. In *Proceedings of the 11th International Conference on Learning Representations*.

Pertsch, K.; Lee, Y.; and Lim, J. J. 2020. Accelerating Reinforcement Learning with Learned Skill Priors. In *Proceedings of the 4th Conference on Robot Learning*, 188–204.

Pertsch, K.; Lee, Y.; Wu, Y.; and Lim, J. J. 2021. Demonstration-Guided Reinforcement Learning with Learned Skills. In *Proceedings of the 5th Conference on Robot Learning*, 729–739.

Preechakul, K.; Chathee, N.; Wizadwongs, S.; and Suwanjanakorn, S. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 10609–10619.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 10674–10685.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2023. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *Proceedings of the 11th International Conference on Learning Representations*.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2019. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Proceedings of the 3rd Conference on Robot Learning*, 1094–1100.