DIDI: Diffusion-Guided Diversity for Offline Behavioral Generation

Jinxin Liu [*1,2] **Xinghong Guo** [*1] **Zifeng Zhuang** [1] **Donglin Wang** [1,3]

Abstract

In this paper, we propose a novel approach called **DI**ffusion-guided **DI**versity (**DIDI**) for offline behavioral generation. The goal of DIDI is to learn a diverse set of skills from a mixture of label-free offline data. We achieve this by leveraging diffusion probabilistic models as priors to guide the learning process and regularize the policy. By optimizing a joint objective that incorporates diversity and diffusion-guided regularization, we encourage the emergence of diverse behaviors while maintaining the similarity to the offline data. Experimental results in four decision-making domains (Push, Kitchen, Humanoid, and D4RL tasks) show that DIDI is effective in discovering diverse and discriminative skills. We also introduce skill stitching and skill interpolation, which highlight the generalist nature of the learned skill space. Further, by incorporating an extrinsic reward function, DIDI enables reward-guided behavior generation, facilitating the learning of diverse and optimal behaviors from sub-optimal data.

1. Introduction

Offline reinforcement learning (RL) has shown great promise in enabling agents to learn from past experiences without further interaction with the environment (Levine et al., 2020; Brandfonbrener et al., 2021; Janner et al., 2021). Naturally, it eliminates the need for time-consuming and costly online exploration and enables learning from pre-collected large datasets. However, one inherent requirement of this formulation is that the offline data must be labeled with rewards, which guide the learning process of the policy.

In practice, a significant portion of datasets is often collected without reward labels, posing challenges in learning a useful policy from such a reward-free dataset, particularly when the offline data is suboptimal or noisy.

Moreover, offline real-world data is typically collected from a mixture of different data-collecting policies, resulting in datasets that exhibit multimodality and diversity (Wang et al., 2022; Shafiullah et al., 2022; Chen et al., 2022). Learning directly from such datasets may lead to suboptimal performance or bias the learned policy towards a specific behavior. Recent studies have explored incorporating additional contextual variables (Zhou et al., 2021; Liu et al., 2023b) or utilizing powerful model architectures (Chi et al., 2023; Furuta et al., 2021) to learn a set of behaviors from the dataset. However, a major limitation of these methods is the lack of encouragement for the emergence of diverse behaviors. In practical applications, the ability to command the diversity of behaviors is often desired, rather than solely generating a set of optional behaviors.

To address these challenges, we propose a novel approach called DIDI (DIffusion-guided DIversity) for offline behavioral generation. The objective of DIDI is to learn optimal *and* diverse behaviors from a mixture of label-free offline data. To achieve controllable behavioral generation, we introduce a contextual policy that can be commanded to produce a specific behavior. The learning of the contextual policy is guided by a diffusion probabilistic model acting as a regularization prior.

Yet, supervising the contextual policy with diffusion model is not trivial, since we cannot directly obtain <contextual input, behavior target> pairs from the diffusion model. To this end, we draw inspiration from online unsupervised RL and employ a three-step process: First, we command the contextual policy to produce a pseudo-behavior. Then, we use the diffusion model to *relabel* the pseudo-behavior and obtain a proxy-target. Finally, using the relabeled *<contextual input, proxy-target>* , we can supervise the learning of the contextual policy. The key insight is that the relabeling (noising and denoising) process of the diffusion model allows us to obtain a relabeled proxy-target, effectively bringing it back to the offline data distribution and thus eliminating the potential out-of-distribution (OOD) issues in offline settings.

We empirically evaluate our DIDI approach in four decision-

[*]Equal contribution [1]School of Engineering, Westlake University, Hangzhou, China [2]Zhejiang University, Hangzhou, China [3]Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, China. Correspondence to: Jinxin Liu <liujinxin@westlake.edu.cn>, Donglin Wang <wangdonglin@westlake.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

making domains: Push, Kitchen, Humanoid, and D4RL tasks. Across various action/observation spaces, our results demonstrate that DIDI successfully discovers diverse and discriminative skills. Compared to alternative baselines, DIDI generates more diverse behaviors and achieves superior performance. Additionally, we showcase the generalist nature of the learned skill space by illustrating skill stitching and interpolation. Finally, assuming the availability of extrinsic rewards, we show that DIDI can produce diverse and optimal behaviors.

The contribution of this paper can be summarized as follows: **1)** We propose DIDI, a novel approach for offline behavioral generation that utilizes a diffusion probabilistic model as a prior to guiding the learning of a contextual policy. **2)** Through extensive experiments, we demonstrate the effectiveness of DIDI in discovering diverse and discriminative skills, as well as generating optimal behaviors with new extrinsic rewards. **3)** Furthermore, we illustrate the generalist nature of the learned skill space through skill stitching and interpolation.

2. Related Work

This work resides at the intersection of offline reinforcement learning (RL), unsupervised skill learning, and diffusion probabilistic models. In this section, we provide a concise overview of the related work in these domains.

2.1. Offline Reinforcement Learning

Offline reinforcement learning (RL) refers to the setting where the agent learns from a fixed reward-labeled dataset without further interaction with the environment. To address the potential out-of-distribution (OOD) problem, recent works have introduced various designs to ensure the learned policy aligns with offline data distribution. These designs range from policy constraints (Fujimoto & Gu, 2021; Wu et al., 2019; 2022) to value regularization (Kumar et al., 2020a; Kostrikov et al., 2021; Liu et al., 2023c), and from iterative optimization (Kumar et al., 2019; Zhuang et al., 2023; Yu et al., 2021) to non-iterative frameworks (Emmons et al., 2021; Chen et al., 2021; Liu et al., 2023b; Lai et al., 2023; Zhuang et al., 2024). However, these methods are limited to learning a single policy and rely on extrinsic reward labels. Alternatively, some works propose using expert demonstrations (Zolna et al., 2020; Liu et al., 2023a; Kim et al., 2021; Liu et al., 2023d; Sun et al., 2023) or human preferences (Shin et al., 2021; Kang et al., 2023) to guide offline policy learning. However, both of them require additional extrinsic supervision (using rewards, demonstrations, or preferences), which may not be available in practice. Moreover, the learned policy in these methods tends to be biased towards a single behavior, especially when the dataset contains multiple behaviors. In contrast, our method can learn a diverse set of behaviors from the dataset without requiring any extrinsic supervision.

2.2. Unsupervised Skill Learning

Unsupervised skill learning aims to learn a set of skills from unlabeled interactions. Typically, unsupervised skill learning methods can be categorized into two groups: online setting and offline formulation. In online RL, skill learning is often formulated through the lens of empowerment (Laskin et al., 2022; Liu et al., 2022; Tian et al., 2021a; Achiam et al., 2018; Eysenbach et al., 2018; Sharma et al., 2019; Tian et al., 2021b; Liu et al., 2021), which seeks to maximize the mutual information between the skill and the induced behavior. In contrast, offline skill learning is often formulated as a behavioral cloning problem (Furuta et al., 2021; Chi et al., 2023), which directly fits a policy from the dataset. However, such a formulation inherently imposes burdens to the fitting model, especially when the offline data is multi-modal or noisy. Also in the offline setting, our DIDI method explicitly inherits the empowerment objective and uses a diffusion probabilistic model to guide the empowerment optimization. Compared with existing offline learning methods that heavily rely on the fitting model itself, one key difference is that we explicitly introduce a behavioral diversity objective, actively encouraging the emergence of diversity.

2.3. Diffusion Probabilistic Models

Diffusion probabilistic models have emerged as powerful tools for modeling complex data distributions, surpassing previous methods in terms of sample quality and diversity (Tevet et al., 2022; Zhang et al., 2022; Zhu et al., 2023). For decision-making tasks, recent works have shown that diffusion models can be used to model the policy/value network (Wang et al., 2022; Hansen-Estruch et al., 2023), serve as a planner (Janner et al., 2022; Liang et al., 2023; Mishra et al., 2023), or act as a data synthesizer (Lu et al., 2023; Yu et al., 2023; Chen et al., 2023; He et al., 2023). In our work, we employ a diffusion probabilistic model as a prior to guiding the learning of contextual policy, akin to (but extending beyond) the role of a data synthesizer.

3. Preliminaries

3.1. Offline Reinforcement Learning

Throughout this work, we consider reinforcement learning (RL) in the framework of Markov decision process (MDP) specified by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, p_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action space respectively, $\mathcal{T}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ denotes the transition dynamics, $r(\mathbf{s}_t, \mathbf{a}_t)$ denotes the reward function, $p_0(\mathbf{s}_0)$ denotes the initial state distribution, and γ denotes the discount

factor. In an environment (MDP), the goal of RL is to learn a (stationary) policy $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ that maximizes the expected discounted return $\mathcal{J}(\pi_\theta) = \mathbb{E}_{\pi_\theta(\boldsymbol{\tau})}[R(\boldsymbol{\tau})]$, where $R(\boldsymbol{\tau}) := \sum_{t=0}^{T} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)$ denotes the discounted return of a trajectory $\boldsymbol{\tau} := (\mathbf{s}_0, \mathbf{a}_0, \ldots, \mathbf{s}_T, \mathbf{a}_T)$, and we overload notation $\pi_\theta(\boldsymbol{\tau})$ to represent the probability of trajectory $\boldsymbol{\tau}$ generated by running policy $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ in the environment with transition dynamics $\mathcal{T}$: $\pi_\theta(\boldsymbol{\tau}) = \pi_\theta \otimes \mathcal{T} := p_0(\mathbf{s}_0)\pi_\theta(\mathbf{a}_0|\mathbf{s}_0)\prod_{t=0}^{T-1}\mathcal{T}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi_\theta(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})$.

In offline RL (Levine et al., 2020), the agent only has access to a static dataset $\mathcal{D} := \{\boldsymbol{\tau}|\boldsymbol{\tau} \sim \pi_\mathcal{D}(\boldsymbol{\tau})\}$ generated by one or more behavioral policies $\pi_\mathcal{D}$, and cannot interact further with the environment. Thus, model-based offline RL algorithms propose that we can learn a proxy transition model $\hat{\mathcal{T}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) = \arg\max_{\hat{\mathcal{T}}} \mathbb{E}_\mathcal{D}\left[\log \hat{\mathcal{T}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\right]$ and optimize the regularized objective:

$$\max_{\pi_\theta, \hat{\mathcal{T}}} \mathbb{E}_{\pi_{\theta,\hat{\mathcal{T}}}(\boldsymbol{\tau})}\left[R(\boldsymbol{\tau}) + \log \pi_\mathcal{D}(\boldsymbol{\tau}) - \log \pi_{\theta,\hat{\mathcal{T}}}(\boldsymbol{\tau})\right], \quad (1)$$

where $\pi_{\theta,\hat{\mathcal{T}}}(\boldsymbol{\tau}) = \pi_\theta \otimes \hat{\mathcal{T}}$ denotes the trajectory distribution generated by running policy $\pi_\theta(\mathbf{a}|\mathbf{s})$ in the proxy environment $\hat{\mathcal{T}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$. Briefly speaking, the additional regularization $(\log \pi_\mathcal{D}(\boldsymbol{\tau}) - \log \pi_{\theta,\hat{\mathcal{T}}}(\boldsymbol{\tau}))$ in Equation 1 represents a KL divergence, encouraging the new rollout trajectory $\pi_{\theta,\hat{\mathcal{T}}}(\boldsymbol{\tau})$ does not deviate heavily from the offline data distribution $\pi_\mathcal{D}(\boldsymbol{\tau})$, and such divergence can be replaced by other measure instances.

3.2. Emergence of Online Diverse Behaviors

In this work, we consider learning a latent-conditioned policy $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{z})$, where the latent variable $\mathbf{z} \in \mathbb{R}^d$ is drawn from a prior (skill) distribution $\mathbf{z} \sim p(\mathbf{z})$. Then we define the joint latent-variable trajectory distribution (in online environment) $\pi_\theta(\boldsymbol{\tau}, \mathbf{z}) = p(\mathbf{z})\pi_\theta(\boldsymbol{\tau}|\mathbf{z})$, where $\pi_\theta(\boldsymbol{\tau}|\mathbf{z}) = p_0(\mathbf{s}_0)\pi_\theta(\mathbf{a}_0|\mathbf{s}_0, \mathbf{z})\prod_{t=0}^{T-1}\mathcal{T}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi_\theta(\mathbf{a}_{t+1}|\mathbf{s}_{t+1}, \mathbf{z})$. To encourage the emergence of diverse behaviors, we maximize the mutual information between trajectories and latent variables (Eysenbach et al., 2018; Sharma et al., 2019) along with the return (Kumar et al., 2020b):

$$\max_{\pi_\theta} I(\boldsymbol{\tau}; \mathbf{z}) + \mathbb{E}_{p(\mathbf{z}),\pi_\theta(\boldsymbol{\tau}|\mathbf{z})}[R(\boldsymbol{\tau})]$$
$$= \mathbb{E}_{p(\mathbf{z}),\pi_\theta(\boldsymbol{\tau}|\mathbf{z})}[\log p(\mathbf{z}|\boldsymbol{\tau}) - \log p(\mathbf{z}) + R(\boldsymbol{\tau})],$$

If we remove the above $R(\boldsymbol{\tau})$ term, it corresponds to the pure unsupervised RL objective. To optimize above objective, we can approximate $p(\mathbf{z}|\boldsymbol{\tau})$ with a learned discriminator network $q_\phi(\mathbf{z}|\boldsymbol{\tau})$ and derive the evidence lower bound:

$$\max_{\pi_\theta, q_\phi} \mathbb{E}_{p(\mathbf{z}),\pi_\theta(\boldsymbol{\tau}|\mathbf{z})}[\log q_\phi(\mathbf{z}|\boldsymbol{\tau}) - \log p(\mathbf{z}) + R(\boldsymbol{\tau})]. \quad (2)$$

Besides the trajectory-level diversity as described above, we can alternatively choose to maximize the mutual information between next-states and skills: $\max I(\mathbf{s}_{t+1}; \mathbf{z}|\mathbf{s}_t)$, encouraging different skills, $\pi(\mathbf{a}_t|\mathbf{s}_t, \cdot)$, to visit different states, or to maximize $I(\mathbf{s}_T; \mathbf{z})$, encouraging different skills to reach different final states.

3.3. Diffusion Probabilistic Model

Diffusion models are a class of likelihood-based models that generate samples by gradually removing noise from a signal. Specifically, denoising diffusion probabilistic models (DDPMs, Ho et al. (2020)) generate samples $\boldsymbol{\tau} := \boldsymbol{\tau}^0$ by reversing a Gaussian noising process[1] (*superscript* $n \in [0, N]$ *denotes the number of diffusion iteration*):

$$\begin{aligned} p(\boldsymbol{\tau}^N) &= \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ p_\theta(\boldsymbol{\tau}^{n-1}|\boldsymbol{\tau}^n) &= \mathcal{N}(\boldsymbol{\tau}^{n-1}|\mu_\theta(\boldsymbol{\tau}^n, n), \Sigma_\theta(\boldsymbol{\tau}^n, n)), \\ p_\theta(\boldsymbol{\tau}^0) &= \int p(\tau^N) \prod_{n=1}^{N} p_\theta(\boldsymbol{\tau}^{n-1}|\boldsymbol{\tau}^n) \mathrm{d}\boldsymbol{\tau}^{1:N}, \end{aligned} \quad (3)$$

where $\mu_\theta(\boldsymbol{\tau}^n, n)$ and $\Sigma_\theta(\boldsymbol{\tau}^n, n)$ are learned neural networks. In general, we learn θ by maximizing a variational lower bound of the log likelihood $\mathbb{E}_{\pi_\mathcal{D}(\boldsymbol{\tau}^0)}\left[\log p_\theta(\boldsymbol{\tau}^0)\right]$:

$$\mathbb{E}_{q(\boldsymbol{\tau}^0, \boldsymbol{\tau}^{1:N})}\left[\log p_\theta(\boldsymbol{\tau}^0, \boldsymbol{\tau}^{1:N}) - \log q(\boldsymbol{\tau}^{1:N}|\boldsymbol{\tau}^0)\right], \quad (4)$$

where the expectation $q(\boldsymbol{\tau}^0, \boldsymbol{\tau}^{1:N}) := \pi_\mathcal{D}(\boldsymbol{\tau}^0) q(\boldsymbol{\tau}^{1:N}|\boldsymbol{\tau}^0)$ is a forward Gaussian noising process. According to a predefined schedule $\beta_1, \ldots, \beta_N$, DDPM sets the forward Gaussian noising process with $q(\boldsymbol{\tau}^{1:N}|\boldsymbol{\tau}^0) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{\tau}^n|\sqrt{1-\beta_n}\boldsymbol{\tau}^{n-1}, \beta_n \mathbf{I})$. With $\alpha_n := 1 - \beta_n$ and $\bar{\alpha}_n := \prod_{i=1}^{n} \alpha_i$, we can write the marginal distribution $q(\boldsymbol{\tau}^n|\boldsymbol{\tau}^0) = \mathcal{N}(\boldsymbol{\tau}^n|\sqrt{\bar{\alpha}_n}\boldsymbol{\tau}^0, (1-\bar{\alpha}_n)\mathbf{I})$ and express $\boldsymbol{\tau}^n$ as: $\boldsymbol{\tau}^n(\boldsymbol{\tau}^0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_n}\boldsymbol{\tau}^0 + (1-\bar{\alpha}_n)\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In implementation, instead of directly predict $\boldsymbol{\tau}^0$ (maximizing Equation 4), we can alternatively predict the noise $\boldsymbol{\epsilon}$ added to $\boldsymbol{\tau}^0$, and learn the diffusion parameters θ with

$$\min_{\boldsymbol{\epsilon}_\theta} \mathbb{E}_{n, \boldsymbol{\tau}^n, \boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{\tau}^n, n)\|^2\right], \quad (5)$$

where the expectation $\mathbb{E}_{n, \boldsymbol{\tau}^n, \boldsymbol{\epsilon}}$ is specified by Equation 4 (Ho et al., 2020). At testing/inference, we can then use Equation 3 to conduct the generative process by progressively denoising a noisy input starting from $\boldsymbol{\tau}^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

4. Diffusion-Guided Behavioral Diversity

As described in Section 3.2, encouraging diversity (for the contextual policy $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{z})$) requires us to keep track of the mutual information between trajectories (or states) and latent variables $\mathbf{z}$. In online RL, we can model the conditional distribution $\pi_\theta(\boldsymbol{\tau}|\mathbf{z})$ with the actual rollout in the environment, *e.g.*, $\pi_\theta(\boldsymbol{\tau}|\mathbf{z}) = \pi_\theta \otimes \mathcal{T}$. However, we

[1]Note that in this paper, we use *superscript* $n \in [0, N]$ to denote the number of (forward or reverse) diffusion iterations, and use *subscript* $t \in [0, T]$ to denote the time-step along a trajectory.

consider this problem in the context of an offline RL setting that prohibits the online interaction with the environment.

Following the model-based offline RL formulation, one straightforward solution is to learn an additional dynamics model $\hat{\mathcal{T}}$ that substitutes the rollout distribution with $\pi_{\theta,\hat{\mathcal{T}}}(\boldsymbol{\tau}|\mathbf{z}) = \pi_\theta \otimes \hat{\mathcal{T}}$. In implementation, such solution (learning $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t, \mathbf{z})$ with Equation 2 over the proxy $\hat{\mathcal{T}}$) can easily exploit the inaccuracies of the learned $\hat{\mathcal{T}}$ and encounter unreasonable behaviors. Thus, typical model-based offline RL algorithms incorporate additional uncertainty estimation or conservatism into the policy training (Yu et al., 2020; Kidambi et al., 2020), while such uncertainty estimation or conservatism conflicts with our diversity objective.

In this work, we advocate a single network for both policy learning and dynamics modeling, avoiding compounding rollout errors over an additional proxy dynamics model. We formulate such network as a sequence model, $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$, where $\boldsymbol{\tau}_t := [\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \dots, \mathbf{s}_T, \mathbf{a}_T]$. Here, the subscript t of trajectory $\boldsymbol{\tau}_t$ indicates the starting time of the trajectory. At testing, we then execute the first action $\mathbf{a}_t$ of the predicted $\boldsymbol{\tau}_t \sim \pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$ at state $\mathbf{s}_t$ given skill $\mathbf{z}$. For simplicity of notation, we also denote such action selection by $\mathbf{a}_t \sim \pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$.

4.1. Planning with Diffusion Models

Following Diffuser (Janner et al., 2022), one can first learn a diffusion model $\pi_\psi(\boldsymbol{\tau}_t|\mathbf{s}_t)$ to approximate the offline data distribution $\pi_\mathcal{D}(\boldsymbol{\tau}_t)$ with Equation 5. Then, at inference, we can model RL as conditional sampling over the learned diffusion model. Specifically, to encourage diverse behaviors, we approximate the reverse process (Equation 3) with

$$\begin{aligned} \pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z}) &:= p_\psi(\boldsymbol{\tau}_t^0|\mathbf{s}_t, \mathbf{z}), \\ p_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n, \mathbf{s}_t, \mathbf{z}) &= \mathcal{N}\left(\boldsymbol{\tau}_t^{n-1}|\mu_\psi + g\Sigma_\psi, \Sigma_\psi\right), \end{aligned} \quad (6)$$

where μ_θ and Σ_θ are the corresponding parameters in Equation 3, and $g := \nabla_{\boldsymbol{\tau}_t^n}\left(q_\phi(\mathbf{z}|\boldsymbol{\tau}_t^n) + R(\boldsymbol{\tau}_t^n)\right)$ denotes the diversity and reward-maximizing guidance. Intuitively, adding the gradient g to each reverse step will encourage the final $\boldsymbol{\tau}_t^0$ incorporating both the diversity (q_ϕ) and reward (R) information, and towards the desired behavior.

However, iterating the reverse denoising process specified by Equation 6 leaves two main questions unanswered: **1)** It remains unclear how to obtain a pre-trained skill discriminator q_ϕ to guide the denoising process. Note that training q_ϕ is inherently different from the training of $R(\boldsymbol{\tau}^n)$ that is orthogonal to the diffusion training. To derive an optimal discriminator $q_\phi(\mathbf{z}|\boldsymbol{\tau}_t)$, we need a meaningful trajectory $\boldsymbol{\tau}_t$, which in turn depends on the performance of $q_\phi(\mathbf{z}|\boldsymbol{\tau}_t)$ that will guide the conditional diffusion sampling (Equation 6). **2)** Performing action inference requires performing iterative denoising process (Figure 1 *top*), which hinders their use in some real-time tasks. Even though Janner et al. (2022) propose the warm-starting diffusion for faster planning, compared to the standard single forward-pass though inference networks, which still requires a large number of forward and backward (calculating gradient g) computations in all. To relax the burden of inference time, setting too small denoising step will suffer a clear performance degradation.

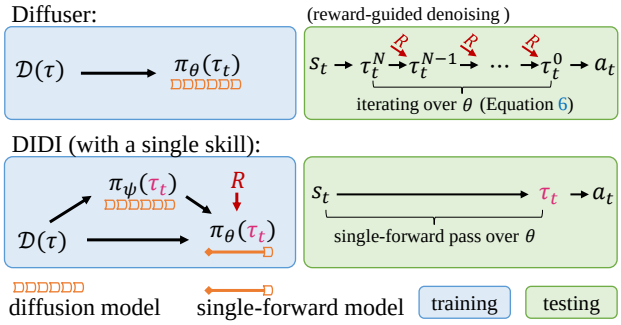

Figure 1. Comparison between Diffuser (Janner et al., 2022) and our DIDI (with a single skill, *i.e.*, assuming $p(\mathbf{z}) = \delta(\mathbf{z})$).

In the next subsection, we will discuss how we can address the above two challenges by incorporating pre-trained diffusion models (as priors) into the diversity-guided and reward-maximizing objective, and deriving a well-shaped policy network $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$ that can directly produce decision action through a single forward-pass at inference, as shown in Figure 1 *bottom*.

4.2. Diffusion Probabilistic Models as Priors

Recall that our goal is learning diverse skills/behaviors, formulated by contextual policy $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$, in the offline RL setting. We first lay out a general formulation for such an objective by combining Equations 1 and 2: (next, we will mark $\boldsymbol{\tau}_t$ with pink to emphasize that $\boldsymbol{\tau}_t$ is the output of our learning policy $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$, different from the offline $\boldsymbol{\tau}_t \sim \pi_\mathcal{D}(\boldsymbol{\tau}_t)$ used to train the diffusion prior ψ as in Equation 8.)

$$\max_{\pi_\theta, q_\phi} \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}[\underbrace{\log q_\phi(\mathbf{z}|\boldsymbol{\tau}_t) - \log p(\mathbf{z})}_{\text{emergence of diversity}} + R(\boldsymbol{\tau}_t) + \underbrace{\log \pi_\mathcal{D}(\boldsymbol{\tau}_t) - \log \pi_\theta(\boldsymbol{\tau}_t)}_{\text{offline regularization}}], \quad (7)$$

where the expectation $\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t} := \mathbb{E}_{p(\mathbf{z}),\pi_\mathcal{D}(\mathbf{s}_t)\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})}$.

Learning diffusion priors. For the offline regularization term in Equation 7, we first approximate the offline data distribution $\pi_\mathcal{D}(\cdot)$ with a forward KL objective (be similar in spirit to Fujimoto & Gu (2021)) that learns a unconditional[2]

[2]The corresponding conditioned diffusion is $\pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n, \mathbf{z})$ that can not be trained directly because the "label" $q_\phi(\mathbf{z}|\boldsymbol{\tau}_t^n)$ is not accessible in the offline RL setting, as described in Section 4.1.

diffusion generative model $\pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n)$ by maximizing:

$$\max_{\pi_\psi} \mathbb{E}_{\boldsymbol{\tau}_t\sim\pi_\mathcal{D}(\boldsymbol{\tau}_t)}\left[\log \pi_\psi(\boldsymbol{\tau}_t)\right], \tag{8}$$

and implement it with Equations 4 and 5. Compared to the popular offline RL methods, such prior learning wrt π_ψ is similar in spirit to the typical behavior-policy pre-training step, and approximating with the forward KL also resembles the general BC loss for behavior-policy modeling.

Note that here we denote the diffusion parameters by ψ instead of θ, emphasizing that π_ψ is only used during policy training (*as priors*) and it will *not* serve as an inference network at testing. Prior diffusion-based RL (Janner et al., 2022) conducts inference directly over the diffusion model, which will hinder the learning of diverse behaviors (discriminator q_ϕ, Section 4.1), and encounter expensive inference time at testing, as described in Figure 1 *top*.

Incorporating priors into skill learning. In essence, given offline data $\boldsymbol{\tau}_t^0 := \boldsymbol{\tau}_t \sim p_\mathcal{D}(\boldsymbol{\tau})$, diffusion model π_ψ is trained by introducing *new latent variables* $\boldsymbol{\tau}_t^{1:N}$ (Equation 4) that is specified by a simple diffusion forward (noising) process $q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)$. Here we show how to incorporate such latent variables $\boldsymbol{\tau}_t^{1:N}$ into Equation 7 and use the pre-trained diffusion model π_ψ to regularized the contextual policy $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})$.

Assuming trajectory $\boldsymbol{\tau}_t \sim \pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})$ involves latent variables $\mathbf{v}_t$, *i.e.*, $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z}) = \int_{\mathbf{v}_t}\pi_\theta(\boldsymbol{\tau}_t,\mathbf{v}_t|\mathbf{s}_t,\mathbf{z})\mathrm{d}\mathbf{v}_t$, we can rewrite Equation 7 as:

$$\begin{aligned}&\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\log q_\phi(\mathbf{z}|\boldsymbol{\tau}_t) - \log p(\mathbf{z}) + R(\boldsymbol{\tau}_t)\right] + \\ &\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\mathbb{E}_{q(\mathbf{v}|\boldsymbol{\tau}_t)}\left[\log \pi_\mathcal{D}(\boldsymbol{\tau}_t,\mathbf{v}_t) - \log \pi_\theta(\boldsymbol{\tau}_t,\mathbf{v}_t|\mathbf{s}_t,\mathbf{z})\right]\right],\end{aligned} \tag{9}$$

where $\pi_\theta(\boldsymbol{\tau}_t,\mathbf{v}_t|\mathbf{s}_t,\mathbf{z}) = \pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})q(\mathbf{v}_t|\boldsymbol{\tau}_t)$. Then, we can specify the output of $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})$ as the instance of $\boldsymbol{\tau}_t^0$ that can render a diffusion forward (noising) process $q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)$ in diffusion probabilistic models, *e.g.*, $\boldsymbol{\tau}_t^0 := \boldsymbol{\tau}_t$, and thus we can formulate $q(\mathbf{v}_t|\boldsymbol{\tau}_t)$ as the corresponding noising process, *e.g.*, setting $\mathbf{v}_t := \boldsymbol{\tau}_t^{1:N}$.

More importantly, such formulation naturally allows us to replace the distribution $\pi_\mathcal{D}(\boldsymbol{\tau}_t^0,\mathbf{v}_t)$ in Equation 9 with our pre-trained diffusion prior π_ψ (Equation 8):

$$\pi_\mathcal{D}(\boldsymbol{\tau}_t,\mathbf{v}_t) := \pi_\mathcal{D}(\boldsymbol{\tau}_t^0,\boldsymbol{\tau}_t^{1:N}) \leftarrow p(\boldsymbol{\tau}_t^N)\prod_{n=1}^N \pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n).$$

Then, analogously to Ho et al. (2020), we can derive the second expectation in Equation 9 as

$$-\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t,\boldsymbol{\tau}_t^{1:N}}\left[D_\mathrm{KL}\left(\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})\|\pi_\psi(\boldsymbol{\tau}_t^0|\boldsymbol{\tau}_t^1)\right) + \sum_{n>1} D_\mathrm{KL}(q(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n,\boldsymbol{\tau}_t)\|\pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n))\right], \tag{10}$$

Algorithm 1 Diffusion-Guided Diversity (DIDI)

Require: offline dataset $\pi_\mathcal{D}(\boldsymbol{\tau}_t)$ and skill distribution $p(\mathbf{z})$. Initialize diffusion prior π_ψ, reward network R, skill discriminator $q_\phi(\mathbf{z}|\boldsymbol{\tau}_t^n)$ and contextual policy $\pi_\theta(\boldsymbol{\tau}_t^n|\mathbf{s}_t,\mathbf{z})$.

1: Train the diffusion prior π_ψ with Equation 8.
2: **while** not converged **do**
3: Sample $\mathbf{z}\sim p(\mathbf{z})$, $\mathbf{s}_t\sim\pi_\mathcal{D}(\boldsymbol{\tau}_t)$, and $n\sim[0,N]$.
4: Learn $q_\phi(\mathbf{z}|\boldsymbol{\tau}_t^n)$ and $\pi_\theta(\boldsymbol{\tau}_t^n|\mathbf{s}_t,\mathbf{z})$ with $\mathcal{J}_\mathrm{DIDI}$.
5: **end while**

Return: contextual policy $\mathbf{a}_t\sim\pi_\theta(\boldsymbol{\tau}_t^n|\mathbf{s}_t,\mathbf{z})$.

where $\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t,\boldsymbol{\tau}_t^{1:N}} := \mathbb{E}_{p(\mathbf{z}),\pi_\mathcal{D}(\mathbf{s}_t)\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t)}$.

Intuitively, such an objective suggests that we can use a pre-learned diffusion model π_ψ to guide/regularize the learning policy $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})$, thus avoiding the out-of-distribution issues in offline RL settings. Under the diffusion structure, it not only facilitates learning a single forward policy network (Figure 1 *bottom*, somewhat like knowledge distillation[3]), but also *retains its flexibility of combining with the other learning objectives*, *e.g.*, incorporating the mutual information objective in Equation 2.

Combining Equation 10 (after reparametrization in Appendix A) and the first expectation in Equation 9, we obtain

$$\begin{aligned}\mathcal{J}_\mathrm{DIDI}(q_\phi,\pi_\theta) :=& \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\log q_\phi(\mathbf{z}|\boldsymbol{\tau}_t) - \log p(\mathbf{z}) + R(\boldsymbol{\tau}_t)\right] \\ &- \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\mathbb{E}_{n,\boldsymbol{\tau}_t^n,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\psi(\boldsymbol{\tau}_t^n,n)\|^2\right]\right],\end{aligned}$$

where $\boldsymbol{\tau}_t^n = \sqrt{\bar{\alpha}_n}\boldsymbol{\tau}_t + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. Then, using the gradient back-propagated through the pre-trained diffusion prior ψ, we can learn the contextual policy $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})$ and discriminator $q_\phi(\mathbf{z}|\boldsymbol{\tau}_t^n)$ cooperatively. To summarize, we provide the pseudo-code of DIDI in Algorithm 1.

4.3. Variational Auto-Encoders as Priors

To gain more insight wrt our objective $\mathcal{J}_\mathrm{DIDI}$, we can also take variational auto-encoders (VAE) as priors for Equations 7 and 9. Then, we can first learn a VAE prior — $q_\mathrm{enc}(\mathbf{v}_t|\boldsymbol{\tau}_t)$ and $p_\mathrm{dec}(\boldsymbol{\tau}_t|\mathbf{v}_t)$ to approximate the offline data distribution $\pi_\mathcal{D}(\boldsymbol{\tau}_t)$ by maximizing:

$$\mathbb{E}_{\pi_\mathcal{D}(\boldsymbol{\tau}_t)q_\mathrm{enc}(\mathbf{v}_t|\boldsymbol{\tau}_t)}\left[\log p_\mathrm{dec}(\boldsymbol{\tau}_t|\mathbf{v}_t) - \log\frac{q_\mathrm{enc}(\mathbf{v}_t|\boldsymbol{\tau}_t)}{p(\mathbf{v}_t)}\right],$$

where $p(\mathbf{v}_t)$ is a fixed prior. Similar to Equation 10, we can derive an alternative form for the second expectation in Equation 9:

$$-\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t,\mathbf{v}_t}\left[D_\mathrm{KL}\left(\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z})\|q_\mathrm{dec}(\boldsymbol{\tau}_t|\mathbf{v}_t)\right)\right], \tag{11}$$

where $\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t,\mathbf{v}_t} := \mathbb{E}_{p(\mathbf{z}),\pi_\mathcal{D}(\mathbf{s}_t),\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t,\mathbf{z}),q_\mathrm{enc}(\mathbf{v}_t|\boldsymbol{\tau}_t)}$. Intuitively, Equation 11 says that we can use the pre-trained

[3] Compared to prior diffusion methods, we learn $\boldsymbol{\tau}_t^n$ and set π_ψ fixed, while DDPM learns π_ψ and samples $\boldsymbol{\tau}_t^n$ from fixed dataset.

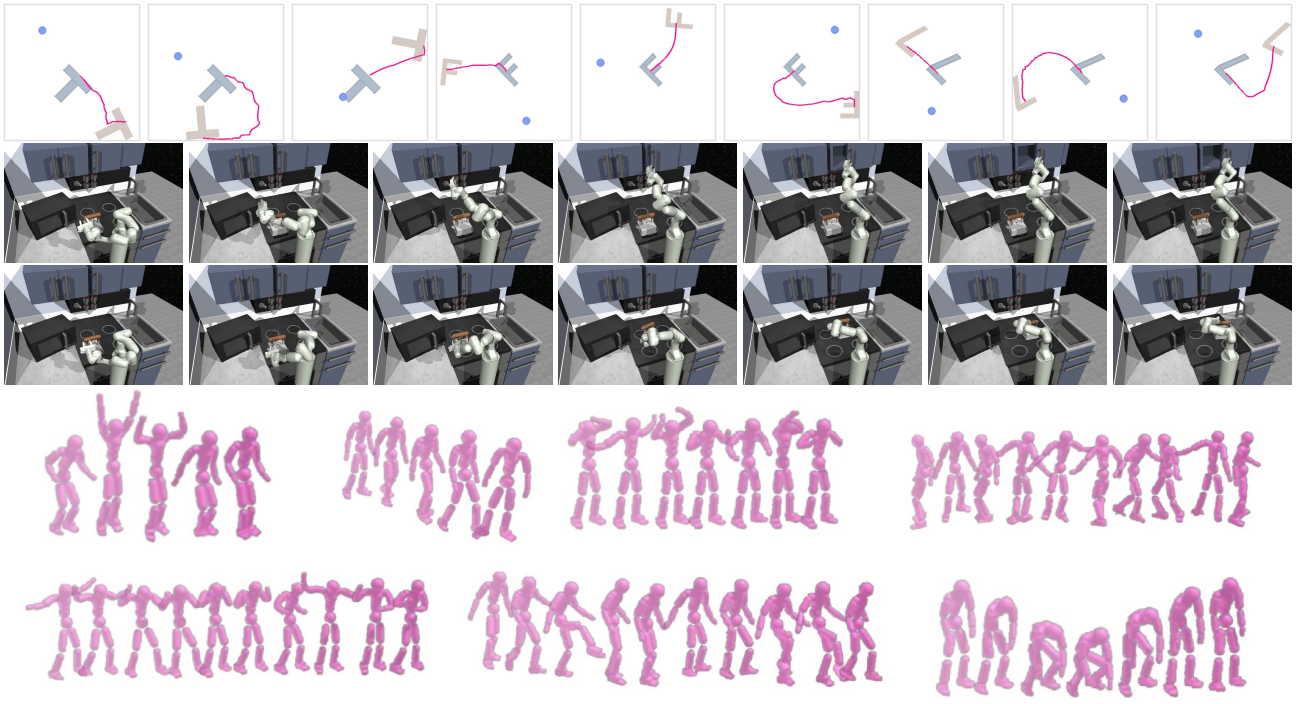

Figure 2. **Discovered diverse skills in three domains.** We can see that in the Push domain, blocks are pushed to different positions. In the Kitchen domain, the robotic arm executes distinct actions. In the Humanoid domain, the agent exhibits different movements and navigates in different directions (the color progression from light to dark indicates the movement progress of the humanoid).

decoder $q_{\text{dec}}(\boldsymbol{\tau}_t|\mathbf{v}_t)$ as a target to learn the contextual policy $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$, but it creates a chicken-and-egg problem that the target *fully* depends on the output of the learning policy itself (*left* sub-diagram, see below):

$$\begin{pmatrix}\mathbf{z}\\ \mathbf{s}_t\end{pmatrix} \xrightarrow{\pi_\theta} \boldsymbol{\tau}_t \xrightarrow{q_{\text{enc}}} \mathbf{v}_t \xrightarrow{q_{\text{dec}}} \boldsymbol{\tau}_t \qquad \begin{pmatrix}\mathbf{z}\\ \mathbf{s}_t\end{pmatrix} \xrightarrow{\pi_\theta} \boldsymbol{\tau}_t \quad p(\mathbf{v}_t) \xrightarrow{q_{\text{dec}}} \boldsymbol{\tau}_t$$

As a target for training π_θ | As a target for training π_θ

where q_{enc} and q_{dec} are fixed during the learning of π_θ.

To sidestep this chicken-and-egg problem (leading to unstable training and bad local optima), one can sample $\mathbf{v}_t$ directly from the "fixed prior" $p(\mathbf{v}_t)$ that was used to train the VAE, *i.e.*, replacing $q_{\text{enc}}(\mathbf{v}_t|\boldsymbol{\tau}_t)$ with $p(\mathbf{v}_t)$ as shown in the above diagram (*right*). But one has to keep in mind that, essentially, the general objective of $\mathcal{J}_{\text{DIDI}}$ is trying to cluster offline trajectories into different behavior patterns, which are specified by the contextual variables $\mathbf{z}$. Sampling $\mathbf{v}_t$ from a "fixed prior" and using this as the target for learning π_θ will cause the policy $\pi_\theta(\boldsymbol{\tau}_t|\mathbf{s}_t, \mathbf{z})$ to fail to capture diverse behaviors, because the target $(q_{\text{dec}}(\boldsymbol{\tau}_t|\mathbf{v}_t), \mathbf{v}_t \sim p(\mathbf{v}_t))$ will not exhibit diversity according to different $\mathbf{z}$, and as a result, it will collapse to a single behavior (pursued in normal offline RL). Thus, to encourage behavior diversity, we cannot arbitrarily separate the chicken-and-egg connection.

Going back to Equation 10, we can find that setting $\mathbf{v}_t = \boldsymbol{\tau}_t^{1:N}$ can naturally *yield a balance between the emergence of diversity and the training stability*: **1)** $\mathbf{v}_t := \boldsymbol{\tau}_t^{1:N}$ is conditioned on the policy's output $\boldsymbol{\tau}_t$ according to $q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t)$, which thus reserves the chicken-and-egg connection and does not sacrifice the diversity, and **2)** training stability would benefit from large diffusion steps N, owing that $\boldsymbol{\tau}_t^N$ approximates a fixed Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

5. Experiments

In our experiments, we aim to answer the following questions[4]: **1)** Can DIDI discover diverse and discriminative skills from a mixture of (label-free) offline data? **2)** How does DIDI compare to other methods? **3)** As the key point of DIDI is to distill a mixture of behaviors into a low-dimensional skill space, what kind of capabilities can this skill space empower us with? **4)** If an extrinsic reward function is available, can DIDI learn diverse and optimal behaviors from sub-optimal offline data? **5)** What are the benefits of learning a diverse set of behaviors for downstream tasks?

To answer the above questions, we validate our DIDI in four decision-making domains: Push, Kitchen, Humanoid (as shown in Figure 2), and D4RL (Fu et al., 2020) tasks. The Push task, derived from IBC (Florence et al., 2022), is planning the trajectory to moving a block in a platform with

[4]The code for our implementation is available at https://github.com/huey0528/icml24didi.

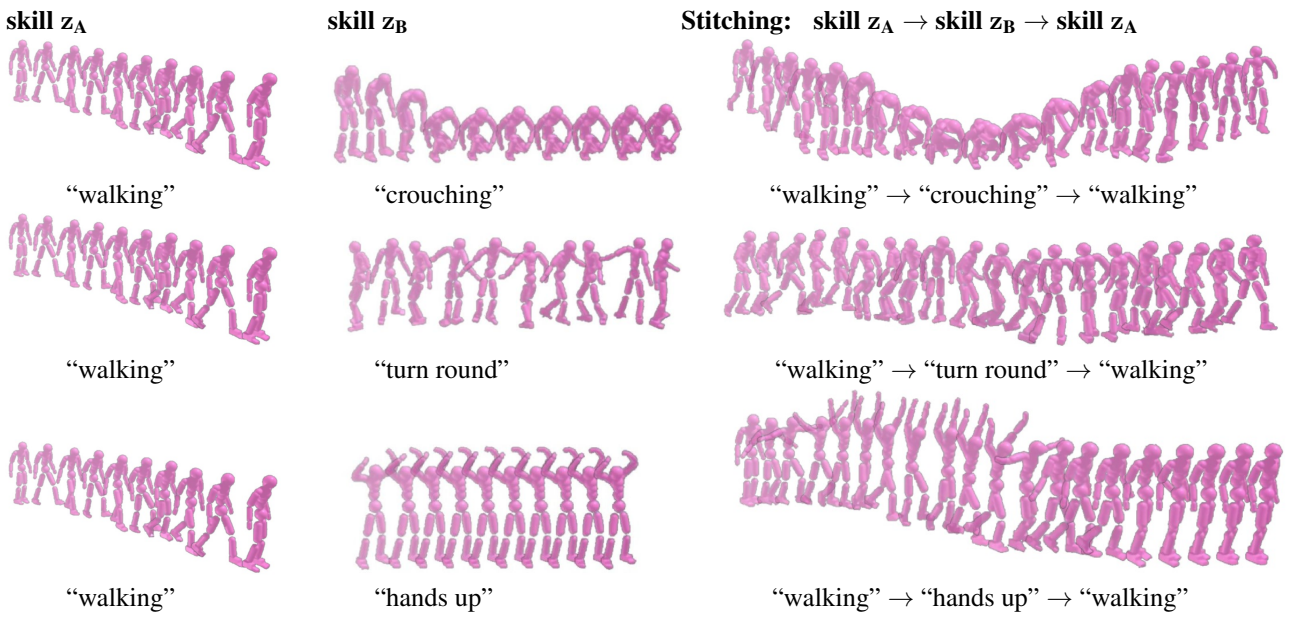

Figure 3. **Skill stitching:** (*1st row*) "walking" → "crouching" → "walking", (*2nd row*) "walking" → "turn round" → "walking", and (*3rd row*) "walking" → "hands up" → "walking". In the diagram, we show (*left*) a "walking forward" skill and (*middle*) a "crouching" / "turn round" / "hands up" skill, and we find that when the robot is walking forward and we suddenly switch to the "crouching" / "turn round" / "hands up" skill, the robot is able to naturally switch the behaviors (*right*). Then, we proceeded to "walking forward" and the robot could switch back to walking forward. The color progression from light to dark indicates the movement progress of the humanoid.

a circular end-effector. The Kitchen task (Gupta et al., 2019) describes the interaction between Franka and seven objects and includes a dataset of 566 human demonstrations. The Humanoid task inherits from PHC (Luo et al., 2023), which focuses on the attainment of high-quality motion imitation. The D4RL task, as introduced by Fu et al. (2020), provides a comprehensive suite of benchmark environments designed for offline RL. Specifically, we utilize the Gym-MuJoCo tasks, which involve continuous control environments such as HalfCheetah, Hopper, and Walker2d, to evaluate our DIDI framework's performance (given extrinsic rewards).

5.1. Emgerence of Behavioral Diversity

In this section, we investigate the ability of our DIDI approach to discover diverse and discriminative skills from a mixture of label-free offline data.

Qualitatively, we visualize the discovered behaviors in Figure 2. First, we apply DIDI to the Push domain. We can observe that DIDI can learn diverse behaviors that push the block to various positions, demonstrating that DIDI can effectively learn diverse behaviors in this simple task. To further evaluate the generality of DIDI, we extend its application to higher-dimensional state and action spaces in the Kitchen and Humanoid tasks. In the Kitchen task, DIDI learns a variety of skills such as opening cabinets and picking up kettle. Similarly, in the Humanoid task, DIDI learns skills like jumping, walking forward, and crouching. These skills cover a wide range of behaviors and showcase the ability of DIDI to discover diverse actions in complex tasks.

Table 1. **Comparison to the behavioral diversity, which qualifies the variance of the motions across all action types.**

	Push T	Push F	Push 7	Kitchen
k-means-DI	6.8	10.4	8.1	8.7
VAE-DI	2.5	6.5	7.7	8.3
VAE-DI-fixed	1.6	7.1	5.4	9.0
DIDI	7.2	12.2	10.2	9.8

5.2. Comparison with Alternative Baselines

To further evaluate the effectiveness of our DIDI approach, we compare it with alternative baselines in terms of skill discovery and performance across Push (including T-shape, F-shape, and 7-shape) and Kitchen domains.

Specifically, we compare the diversity of skills discovered by DIDI with those obtained by alternative methods: k-means-DI, VAE-DI, and VAE-DI-fixed. For k-means-DI, we first use k-means to partition the offline data and then learn a skill policy for each partition; for VAE-DI implementation, we first learn a VAE prior and then use such a prior to guiding the contextual policy, as described in Equation 11; for VAE-DI-fixed, we sample the latent variable $\mathbf{v}_t$ from the fixed prior $p(\mathbf{v}_t)$, and use $q_{\text{dec}}(\mathbf{v}_t)$ as a target for training π_θ.

We analyze the range of behaviors captured by each method

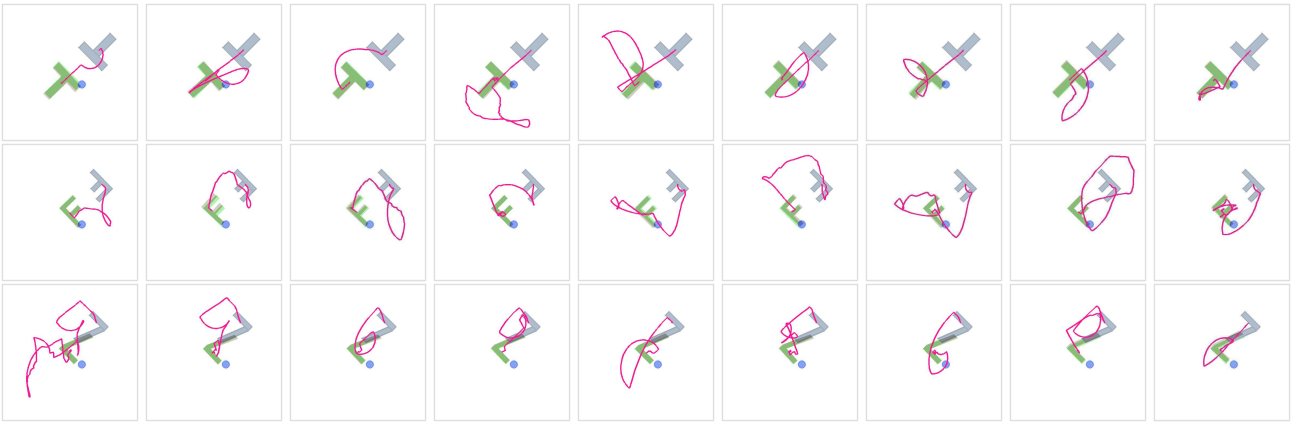

Figure 4. **Discovered diverse and optimal skills in the Push domain (with T-shape, F-shape, and 7-shape blocks).** The green block represents the starting point, the gray block represents the target, and the red curve represents the motion trajectory. We can observe that in all tasks, green blocks successfully move to the target positions and display different movement trajectories simultaneously.

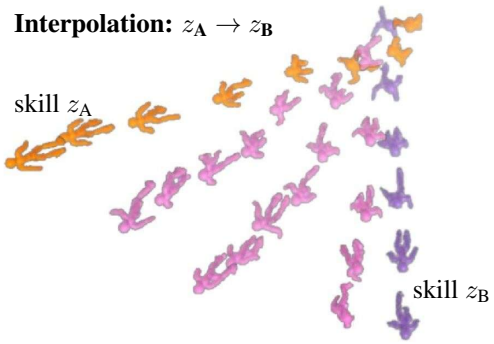

Figure 5. **Visualization of skill interpolation.** We visualize the top view of skill z_A and skill z_B (Humanoid domain). We can see that by interpolating the skill space, we can obtain the interpolated skills that lie between the movement directions of skills z_A and z_B.

and assess the diversity of the learned skills. We use the variance of the motions across all action types as our diversity evaluation metric (Guo et al., 2022). We show our diversity scores in Table 1. The results of the experiment prove the effectiveness of DIDI, achieving higher diversity scores. This evaluation provides insights into the ability of DIDI to discover diverse and discriminative skills from a mixture of label-free offline data.

5.3. Generalist Skill Space: Stitching and Interpolation

Moreover, we also find that our learned skill space tends to be a generalist. For the distilled contextual policy $\pi_\theta(\cdot|\mathbf{s}_t, \mathbf{z})$, their abilities as generalists make them well-suited for skill stitching and interpolation.

Skill stitching refers to sequentially combining different skills to perform complex tasks or solve novel problems. As shown in Figure 3, we can observe that our agents were able to stitch together skills seamlessly: (*1st row*) "walking" $\rightarrow$ "crouching" $\rightarrow$ "walking", (*2nd row*) "walking" $\rightarrow$ "turn round" $\rightarrow$ "walking", and (*3rd row*) "walking" $\rightarrow$ "hands up" $\rightarrow$ "walking". And importantly, when we directly command π_θ from skill z_A to skill z_B, our learned contextual policy is able to adaptively make adjustments to the actions, without leading to a collapsing behavioral breakdown due to a sudden skill switch. Such ability demonstrates the versatility and adaptability of our contextual policy.

Skill interpolation is another important aspect of our generalist skill space. It involves smoothly generating behaviors that lie in learned skills. Our agents also exhibit the capability to interpolate skills, allowing them to perform actions and generate new skill behaviors that were not explicitly encountered during training. In Figure 5, we visualize the top view of skill z_A and skill z_B (Humanoid). We can see that by interpolating the skill space, we can obtain the interpolated skills. Thus, such flexibility in skill interpolation showcases the generalization power of our learned skill space.

Furthermore, the generalist nature of our skill space enables the acquisition of new skills through a process of skill refinement and expansion. Specifically, this process involves refining existing skills to meet new requirements and expanding the learned behaviors. As we will show in the next subsection, by leveraging the existing repertoire of learned skills, our agents can adapt and learn new skills. The ability to continuously learn and expand the skill space will contribute to the overall adaptability of our agents.

5.4. Reward-Guided Behavior Generation

In addition to the generalist nature of our skill space, our DIDI approach also facilitates reward-guided behavior generation. By incorporating an extrinsic reward function, we can leverage DIDI to learn diverse *and* optimal behaviors even from sub-optimal offline data.

Table 2. **Success rates of Push tasks with obstacles after fine-tuning.** The score before the arrow ("→") represents fine-tuning a single optimal policy, while the score after the arrow represents fine-tuning our DIDI's behaviors.

	RL (10 ep)	RL (50 ep)	IL (1 demo)	IL (5 demos)
Push T	0.06 → 0.40	0.18 → 0.58	0.02 → 0.52	0.14 → 0.82
Push F	0.04 → 0.36	0.10 → 0.68	0.12 → 0.48	0.24 → 0.74
Push 7	0.10 → 0.44	0.12 → 0.60	0.10 → 0.50	0.30 → 0.88

Table 3. **Quantitative results on D4RL tasks.** Diff.: Diffuser. ha: halfcheetah. ho: hopper. wa: walker2d. -m: -medium. -me: -medium-expert. -mr: -medium-replay.

	CQL	DT	Diff.	DIDI		
				skill1	skill2	skill3
ha-me	91.6	86.8	88.9	83.0	81.9	86.6
ho-me	105.4	107.6	103.3	101.2	102.1	101.9
wa-me	108.8	108.1	106.9	107.5	105.6	106.9
ha-m	44.0	42.6	42.8	43.0	39.7	42.3
ho-m	58.5	67.6	74.3	70.4	74.1	72.8
wa-m	72.5	74.0	79.6	80.3	78.3	79.1
ha-mr	45.5	36.6	37.7	34.3	33.2	34.8
ho-mr	95.0	82.7	93.6	91.5	87.8	96.1
wa-mr	77.2	66.6	70.6	68.9	70.3	68.1
Avg.	77.6	74.7	77.5	75.6	74.8	75.7

In Push domain, we introduce an additional goal-reaching reward function (in Figure 4, the gray blocks represent the corresponding goal state). Then, we combine both the diversity objective, diffusion regularization, and the goal-reaching reward function to train the contextual policy. We show the learned behaviors in Figure 4. We can find that all the learned behaviors can reach the goal state, meanwhile providing a rich set of options for the agent to explore and adapt.

Furthermore, we conduct experiments on D4RL tasks, as shown in Table 3. The results indicate that our approach, guided by extrinsic rewards, can learn diverse and optimal skills that achieve comparable performance to standard offline RL methods like CQL and Decision Transformer.

In Figures 7, 8 and 9 (appendix), we visualize the learned skills by DIDI, VAE-DI, and Diffuser on D4RL tasks. The visualizations show that the skills learned by DIDI exhibit a high level of diversity compared to those learned by VAE-DI and Diffuser. These results provide further quantitative comparisons, demonstrating that our approach achieves high levels of diversity while maintaining competitive performance with the guidance of extrinsic rewards.

5.5. Diverse Behaviors for Downstream Tasks

To demonstrate the practical benefits (of diverse behaviors) on downstream tasks, we conduct tests in two types of downstream settings: fine-tuning in a downstream sparse-reward RL setting and fine-tuning in the few-shot imitation learning (IL) setting. In three Push tasks (Push T, Push F, and Push 7), we randomly introduced obstacles into the environment that could block the robot's path to push the blocks. In this way, we randomly set up 50 downstream tasks (*i.e.*, 50 random obstacle/goal configurations).

We fine-tune the learned (single) optimal policy parameters (standard offline RL) for both downstream sparse-reward RL and few-shot imitation learning settings. However, we only fine-tuned the skill embeddings (*i.e.*, contextual variables z) in our DIDI method, keeping the contextual policy parameters fixed. Our experimental results are shown in Table 2, where the online fine-tuning setting provides results for 10 and 50 interactions (episodes) with the environment, and the few-shot imitation fine-tuning setting provides results for 1 and 5 demonstrations. In the table, the score before the arrow ("→") in each experimental result represents the (fine-tuned) single optimal policy, while the score after the arrow ("→") represents the (fine-tuned) DIDI's policies, where the score indicates the success rate of Push tasks (with obstacles). We can see that our DIDI method outperforms the baseline (a single policy) in all tested downstream tasks, demonstrating the benefits of our approach (learning diverse behaviors) in enhancing performance in downstream tasks.

6. Conclusion

In this paper, we propose a novel approach called DIDI for offline behavioral generation, aiming to learn diverse behaviors from a mixture of label-free offline data. To control the behavioral generation, we introduce a contextual policy that can be commanded to produce specific behaviors. We then use a diffusion probabilistic model as a prior to guide the learning of the contextual policy. We also compare our approach with the use of variational auto-encoders (VAEs) as priors. We find that our DIDI method better balances diversity (reserving the chicken-and-egg connection) and training stability (approximating a fixed Gaussian prior).

Experimental results in four decision-making domains demonstrate the effectiveness of DIDI in discovering diverse and discriminative skills, and generating optimal behaviors from sub-optimal offline data. We also show that DIDI can be used to stitch skills and interpolate between skills, which demonstrates the generalist nature of the learned skill space.

Acknowledgements

The authors would like to express their gratitude to the anonymous reviewers for their valuable comments and suggestions, which have greatly improved the quality of this work. This work was supported by the National Science and Technology Innovation 2030 - Major Project (Grant No. 2022ZD0208800), and NSFC General Program (Grant No. 62176215).

Impact Statement

The potential broader impact of our work lies in several aspects. Firstly, DIDI addresses the challenge of learning from suboptimal or noisy offline data without reward labels, which is a common scenario in real-world applications. By enabling agents to learn from pre-collected large datasets, DIDI eliminates the need for time-consuming and costly online exploration. This can significantly reduce the burden on data collection and accelerate the development of RL applications in various domains.

Secondly, DIDI introduces a controllable behavioral generation approach by incorporating a contextual policy that can be commanded to produce specific behaviors. This controllability is particularly important in practical applications where the ability to command the diversity of behaviors is desired. By allowing users to specify desired behaviors, DIDI opens up possibilities for personalized and adaptive agent behavior in areas such as robotics, gaming, and autonomous systems.

Thirdly, DIDI contributes to the advancement of the field of machine learning by proposing a novel combination of diffusion probabilistic models and unsupervised RL. This integration provides a principled framework for incorporating prior knowledge and regularization into the learning process, leading to improved performance and interpretability of the learned policies. This can have implications for various machine learning tasks beyond offline behavioral generation, such as imitation learning, transfer learning, and model-based RL.

References

Achiam, J., Edwards, H., Amodei, D., and Abbeel, P. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Brandfonbrener, D., Whitney, W., Ranganath, R., and Bruna, J. Offline rl without off-policy evaluation. *Advances in neural information processing systems*, 34:4933–4946, 2021.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Chen, X., Ghadirzadeh, A., Yu, T., Gao, Y., Wang, J., Li, W., Liang, B., Finn, C., and Zhang, C. Latent-variable advantage-weighted policy optimization for offline rl. *arXiv preprint arXiv:2203.08949*, 2022.

Chen, Z., Kiami, S., Gupta, A., and Kumar, V. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Florence, P., Lynch, C., Zeng, A., Ramirez, O. A., Wahid, A., Downs, L., Wong, A., Lee, J., Mordatch, I., and Tompson, J. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168. PMLR, 2022.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Furuta, H., Matsuo, Y., and Gu, S. S. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*, 2021.

Guo, C., Zuo, X., Wang, S., Liu, X., Zou, S., Gong, M., and Cheng, L. Action2video: Generating videos of human 3d actions. *International Journal of Computer Vision*, 130 (2):285–315, 2022.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

Hansen-Estruch, P., Kostrikov, I., Janner, M., Kuba, J. G., and Levine, S. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

He, H., Bai, C., Xu, K., Yang, Z., Zhang, W., Wang, D., Zhao, B., and Li, X. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *arXiv preprint arXiv:2305.18459*, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Kang, Y., Shi, D., Liu, J., He, L., and Wang, D. Beyond reward: Offline preference-guided policy optimization. *arXiv preprint arXiv:2305.16217*, 2023.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Kim, G.-H., Seo, S., Lee, J., Jeon, W., Hwang, H., Yang, H., and Kim, K.-E. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020a.

Kumar, S., Kumar, A., Levine, S., and Finn, C. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33:8198–8210, 2020b.

Lai, Y., Liu, J., Tang, Z., Wang, B., Hao, J., and Luo, P. Chipformer: Transferable chip placement via offline decision transformer. In *International Conference on Machine Learning*, pp. 18346–18364. PMLR, 2023.

Laskin, M., Liu, H., Peng, X. B., Yarats, D., Rajeswaran, A., and Abbeel, P. Cic: Contrastive intrinsic control for unsupervised skill discovery. *arXiv preprint arXiv:2202.00161*, 2022.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Liang, Z., Mu, Y., Ding, M., Ni, F., Tomizuka, M., and Luo, P. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. *arXiv preprint arXiv:2302.01877*, 2023.

Liu, J., Shen, H., Wang, D., Kang, Y., and Tian, Q. Unsupervised domain adaptation with dynamics-aware rewards in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28784–28797, 2021.

Liu, J., Wang, D., Tian, Q., and Chen, Z. Learn goal-conditioned policy with intrinsic motivation for deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7558–7566, 2022.

Liu, J., He, L., Kang, Y., Zhuang, Z., Wang, D., and Xu, H. Ceil: Generalized contextual imitation learning. *arXiv preprint arXiv:2306.14534*, 2023a.

Liu, J., Zhang, H., Zhuang, Z., Kang, Y., Wang, D., and Wang, B. Design from policies: Conservative test-time adaptation for offline policy optimization. *arXiv preprint arXiv:2306.14479*, 2023b.

Liu, J., Zhang, Z., Wei, Z., Zhuang, Z., Kang, Y., Gai, S., and Wang, D. Beyond ood state actions: Supported cross-domain offline reinforcement learning. *arXiv preprint arXiv:2306.12755*, 2023c.

Liu, J., Zu, L., He, L., and Wang, D. Clue: Calibrated latent guidance for offline reinforcement learning. In *Conference on Robot Learning*, pp. 906–927. PMLR, 2023d.

Lu, C., Ball, P. J., and Parker-Holder, J. Synthetic experience replay. *arXiv preprint arXiv:2303.06614*, 2023.

Luo, Z., Cao, J., Kitani, K., Xu, W., et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10895–10904, 2023.

Mishra, U. A., Xue, S., Chen, Y., and Xu, D. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pp. 2905–2925. PMLR, 2023.

Shafiullah, N. M., Cui, Z., Altanzaya, A. A., and Pinto, L. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35: 22955–22968, 2022.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Shin, D., Brown, D. S., and Dragan, A. D. Offline preference-based apprenticeship learning. *arXiv preprint arXiv:2107.09251*, 2021.

Sun, Z., He, B., Liu, J., Chen, X., Ma, C., and Zhang, S. Offline imitation learning with variational counterfactual reasoning. *Advances in Neural Information Processing Systems*, 36, 2023.

Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., and Bermano, A. H. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

Tian, Q., Liu, J., Wang, G., and Wang, D. Unsupervised discovery of transitional skills for deep reinforcement learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021a.

Tian, Q., Wang, G., Liu, J., Wang, D., and Kang, Y. Independent skill transfer for deep reinforcement learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 2901–2907, 2021b.

Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

Wu, J., Wu, H., Qiu, Z., Wang, J., and Long, M. Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 31278–31291, 2022.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Yu, T., Xiao, T., Stone, A., Tompson, J., Brohan, A., Wang, S., Singh, J., Tan, C., Peralta, J., Ichter, B., et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., and Liu, Z. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

Zhou, W., Bajracharya, S., and Held, D. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, pp. 1719–1735. PMLR, 2021.

Zhu, Z., Zhao, H., He, H., Zhong, Y., Zhang, S., Yu, Y., and Zhang, W. Diffusion models for reinforcement learning: A survey. *arXiv preprint arXiv:2311.01223*, 2023.

Zhuang, Z., Lei, K., Liu, J., Wang, D., and Guo, Y. Behavior proximal policy optimization. *arXiv preprint arXiv:2302.11312*, 2023.

Zhuang, Z., Peng, D., Liu, J., Zhang, Z., and Wang, D. Reinformer: Max-return sequence modeling for offline rl. *arXiv preprint arXiv:2405.08740*, 2024.

Zolna, K., Novikov, A., Konyushkova, K., Gulcehre, C., Wang, Z., Aytar, Y., Denil, M., de Freitas, N., and Reed, S. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.

A. Additional Derivation

Below is a derivation of our objective

$$\mathcal{J}_{\text{DIDI}} := \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\log q_\phi(\mathbf{z}|\boldsymbol{\tau}_t) - \log p(\mathbf{z}) + R(\boldsymbol{\tau}_t)\right] + \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\mathbb{E}_{n,\boldsymbol{\tau}_t^n,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi(\boldsymbol{\tau}_t^n, n)\|^2\right]\right],$$

which incorporating a pre-trained diffusion model $\boldsymbol{\epsilon}_\psi$ into learning single-forward policy $\pi_\theta(\boldsymbol{\tau}_t^n|\mathbf{s}_t,\mathbf{z})$.

Starting from the objective in Equation 9 in the main paper,

$$\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\log q_\phi(\mathbf{z}|\boldsymbol{\tau}_t) - \log p(\mathbf{z}) + R(\boldsymbol{\tau}_t)\right] + \underbrace{\mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\mathbb{E}_{q(\mathbf{v}|\boldsymbol{\tau}_t)}\left[\log \pi_{\mathcal{D}}(\boldsymbol{\tau}_t,\mathbf{v}_t) - \log \pi_\theta(\boldsymbol{\tau}_t,\mathbf{v}_t|\mathbf{s}_t,\mathbf{z})\right]\right]}_{\mathcal{J}_{\text{Eq-9-II}}}. \quad \text{(Equation 9)}$$

we can rewrite the second expectation

$$\begin{aligned}
&\mathcal{J}_{\text{Eq-9-II}} \\
&= \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t}\left[\mathbb{E}_{q(\mathbf{v}_t|\boldsymbol{\tau}_t)}\left[\log \pi_{\mathcal{D}}(\boldsymbol{\tau}_t,\mathbf{v}_t) - \log \pi_\theta(\boldsymbol{\tau}_t,\mathbf{v}_t|\mathbf{s}_t,\mathbf{z})\right]\right] \\
&= \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t^0}\left[\mathbb{E}_{q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)}\left[\log \pi_{\mathcal{D}}(\boldsymbol{\tau}_t^0,\boldsymbol{\tau}_t^{1:N}) - \log \pi_\theta(\boldsymbol{\tau}_t^0,\boldsymbol{\tau}_t^{1:N}|\mathbf{s}_t,\mathbf{z})\right]\right] && (\boldsymbol{\tau}_t^0 := \boldsymbol{\tau}_t,\ \mathbf{v}_t := \boldsymbol{\tau}_t^{1:N}) \\
&= \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t^0}\left[\mathbb{E}_{q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)}\left[\log p(\boldsymbol{\tau}_t^N)\prod_{n=1}^N \pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n) - \log \pi_\theta(\boldsymbol{\tau}_t^0|\mathbf{s}_t,\mathbf{z})\prod_{n=1}^N q(\boldsymbol{\tau}_t^n|\boldsymbol{\tau}_t^{n-1})\right]\right] && (\pi_{\mathcal{D}} \leftarrow \pi_\psi) \\
&= \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t^0}\left[\mathbb{E}_{q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)}\left[\log p(\boldsymbol{\tau}_t^N) + \sum_{n=2}^N \log\frac{\pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n)}{q(\boldsymbol{\tau}_t^n|\boldsymbol{\tau}_t^{n-1})} + \log\frac{\pi_\psi(\boldsymbol{\tau}_t^0|\boldsymbol{\tau}_t^1)}{q(\boldsymbol{\tau}_t^0|\boldsymbol{\tau}_t^1)} - \log \pi_\theta(\boldsymbol{\tau}_t^0|\mathbf{s}_t,\mathbf{z})\right]\right] \\
&= \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t^0}\left[\mathbb{E}_{q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)}\left[\log p(\boldsymbol{\tau}_t^N) + \sum_{n=2}^N \log\frac{\pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n)}{q(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n,\boldsymbol{\tau}_t^0)}\cdot\frac{q(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^0)}{q(\boldsymbol{\tau}_t^n|\boldsymbol{\tau}_t^0)} + \log\frac{\pi_\psi(\boldsymbol{\tau}_t^0|\boldsymbol{\tau}_t^1)}{q(\boldsymbol{\tau}_t^0|\boldsymbol{\tau}_t^1)} - \log \pi_\theta(\boldsymbol{\tau}_t^0|\mathbf{s}_t,\mathbf{z})\right]\right] \\
&= \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t^0}\left[\mathbb{E}_{q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)}\left[\log\frac{p(\boldsymbol{\tau}_t^N)}{q(\boldsymbol{\tau}_t^N|\boldsymbol{\tau}_t^0)} + \sum_{n=2}^N \log\frac{\pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n)}{q(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n,\boldsymbol{\tau}_t^0)} + \log \pi_\psi(\boldsymbol{\tau}_t^0|\boldsymbol{\tau}_t^1) - \log \pi_\theta(\boldsymbol{\tau}_t^0|\mathbf{s}_t,\mathbf{z})\right]\right] \\
&= \mathbb{E}_{\mathbf{z},\mathbf{s}_t,\boldsymbol{\tau}_t^0}\left[\mathbb{E}_{q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)}\left[-D_{\text{KL}}\left(q(\boldsymbol{\tau}_t^N|\boldsymbol{\tau}_t^0)\|p(\boldsymbol{\tau}_t^N)\right) - \sum_{n=2}^N D_{\text{KL}}\left(q(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n,\boldsymbol{\tau}_t^0)\|\pi_\psi(\boldsymbol{\tau}_t^{n-1}|\boldsymbol{\tau}_t^n)\right)\right.\right. \\
&\qquad\qquad \left.\left. - D_{\text{KL}}\left(\pi_\theta(\boldsymbol{\tau}_t^0|\mathbf{s}_t,\mathbf{z})\|\pi_\psi(\boldsymbol{\tau}_t^0|\boldsymbol{\tau}_t^1)\right)\right]\right]
\end{aligned}$$

In the implementation, we set $q(\boldsymbol{\tau}_t^{1:N}|\boldsymbol{\tau}_t^0)$ as a Gaussian noising process, thus the first KL term $D_{\text{KL}}\left(q(\boldsymbol{\tau}_t^N|\boldsymbol{\tau}_t^0)\|p(\boldsymbol{\tau}_t^N)\right)$ is a constant and can be ignored.

B. Additional Results

In Figure 6, we provide more discovered diverse skills in the Push, Kitchen, and Humanoid domains. By applying DIDI across different tasks, we observe its flexibility and robustness in learning. In the Push domain, the variation in block positions highlights DIDI's effectiveness in handling straightforward tasks. Moving to the Kitchen domain, the range of learned skills, from manipulation of objects to intricate interactions with the environment, further illustrates its versatility. Finally, in the Humanoid domain, the development of complex motor skills such as jumping and crouching underscores DIDI's potential in navigating and performing in high-dimensional, dynamic environments.

In Figures 7, 8 and 9, we visualize the learned skills by DIDI, VAE-DI, and Diffuser on D4RL tasks. We can see that the learned skills (by DIDI) exhibit a high level of diversity (compared to VAE-DI and Diffuser). These results provide further quantitative comparisons, demonstrating that our approach achieves high levels of diversity while maintaining competitive performance with the guidance of extrinsic rewards.

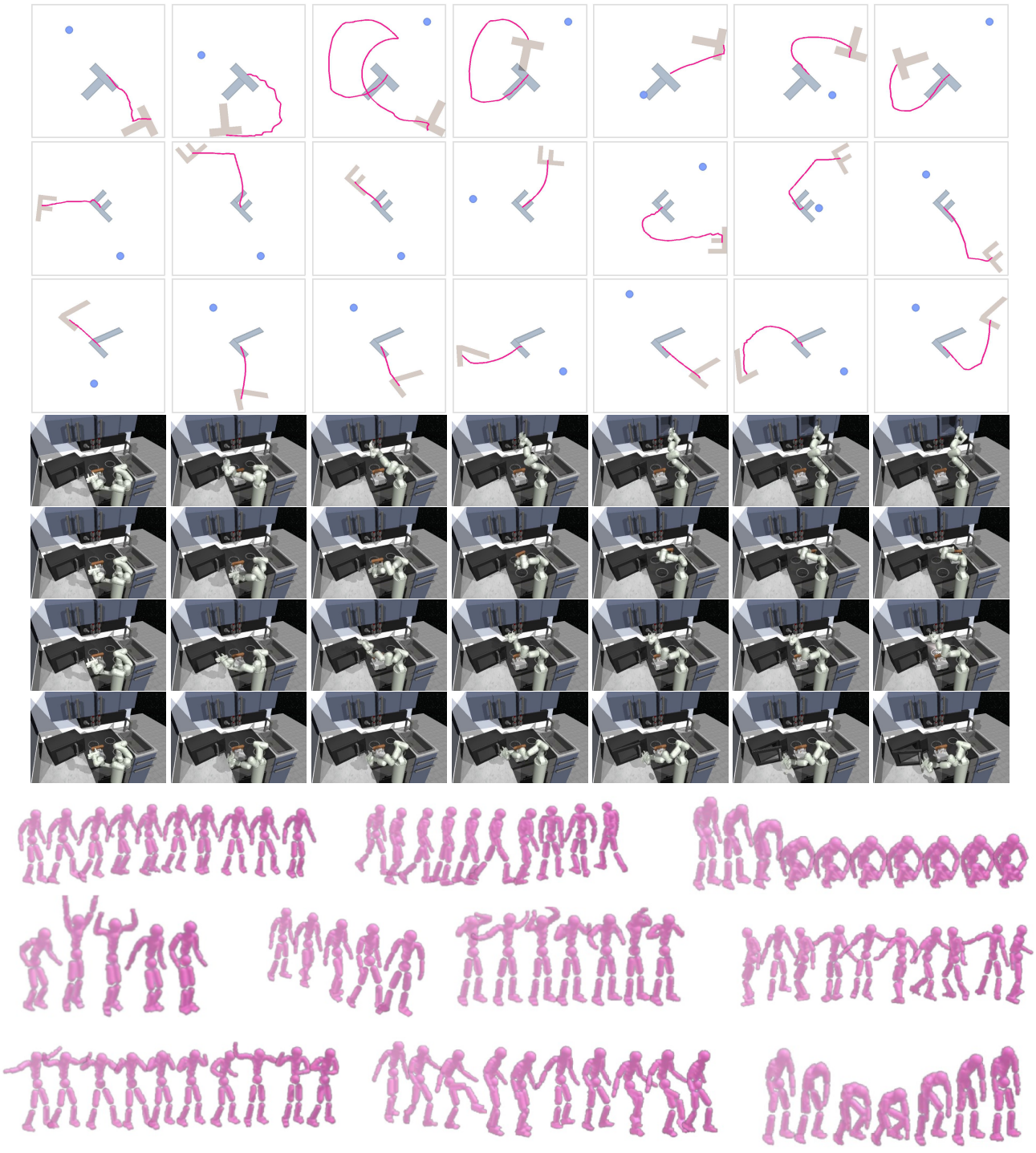

Figure 6. **More discovered diverse skills in three domains.** We can see that in the Push domain, blocks are pushed to different positions. In the Kitchen domain, the robotic arm executes distinct actions. In the Humanoid domain, the agent exhibits different movements and navigates in different directions (the color progression from light to dark indicates the movement progress of the humanoid).

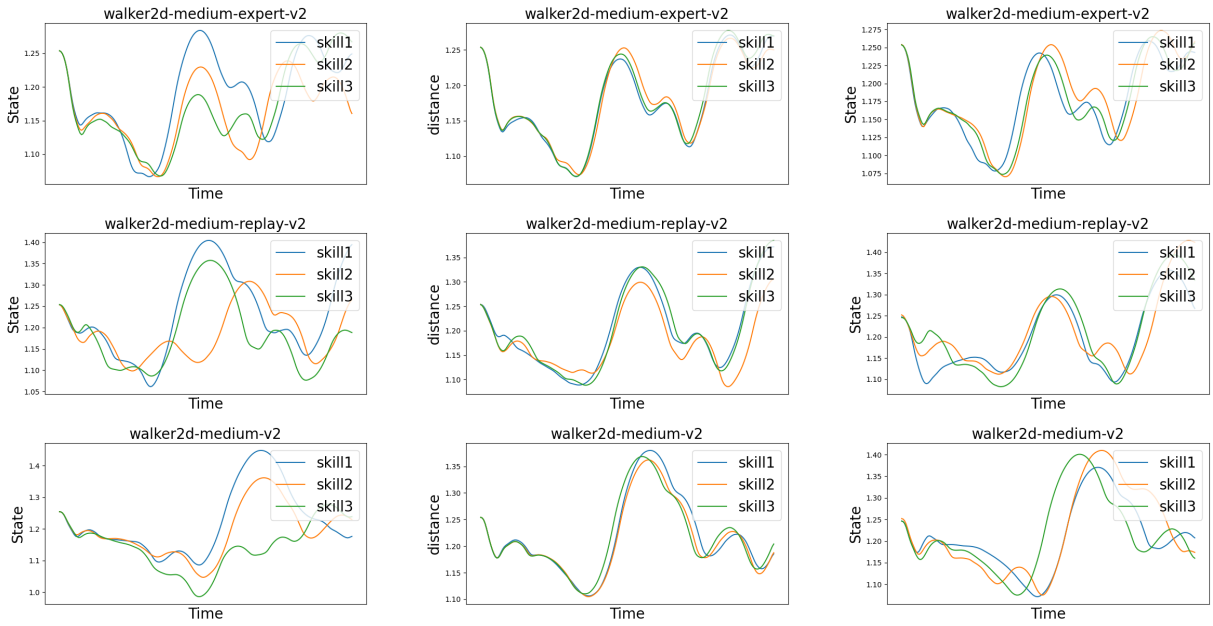

Figure 7. **Skill (Diversity) Visualization.** The figures illustrate the learned skills by DIDI, VAE-DI, and Diffuser on the D4RL walker2d domain. The skills demonstrated by DIDI exhibit a high level of diversity compared to VAE-DI and Diffuser. Each row represents a different task (walker2d-medium-expert-v2, walker2d-medium-replay-v2, walker2d-medium-v2), and each column compares the performance of the three methods. The Y-axis represents the state, and the X-axis represents time.

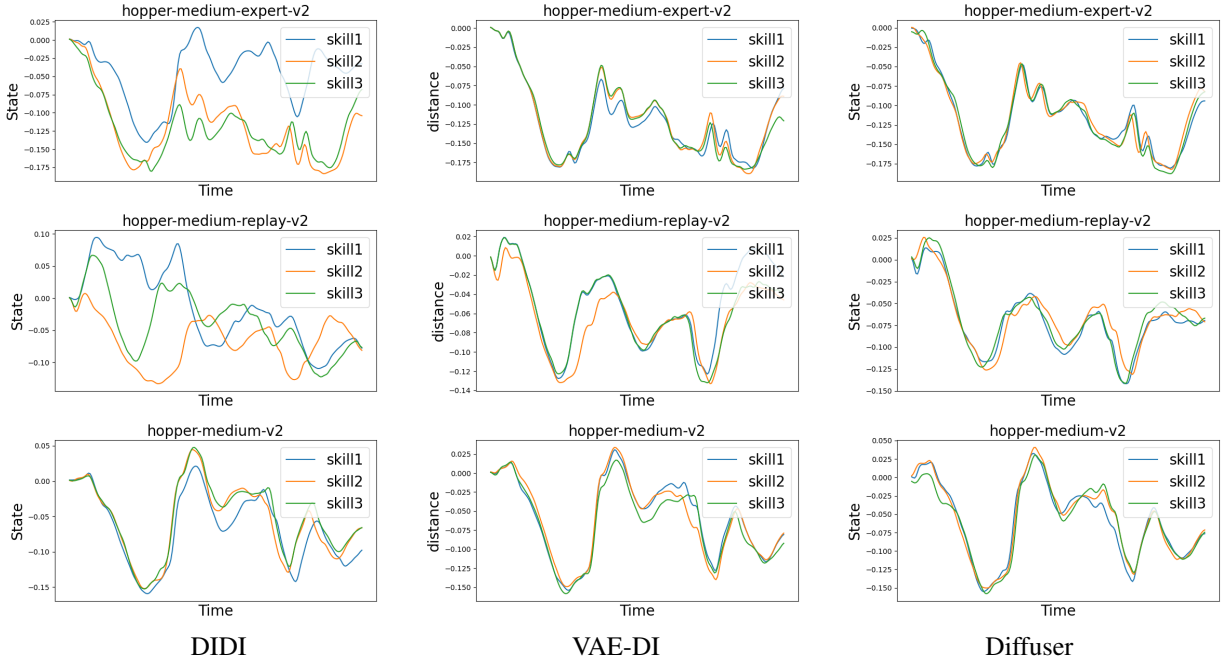

Figure 8. **Skill (Diversity) Visualization.** The figures illustrate the learned skills by DIDI, VAE-DI, and Diffuser on the D4RL hopper domain. The skills demonstrated by DIDI exhibit a high level of diversity compared to VAE-DI and Diffuser. Each row represents a different task (hopper-medium-expert-v2, hopper-medium-replay-v2, hopper-medium-v2), and each column compares the performance of the three methods. The Y-axis represents the state, and the X-axis represents time.

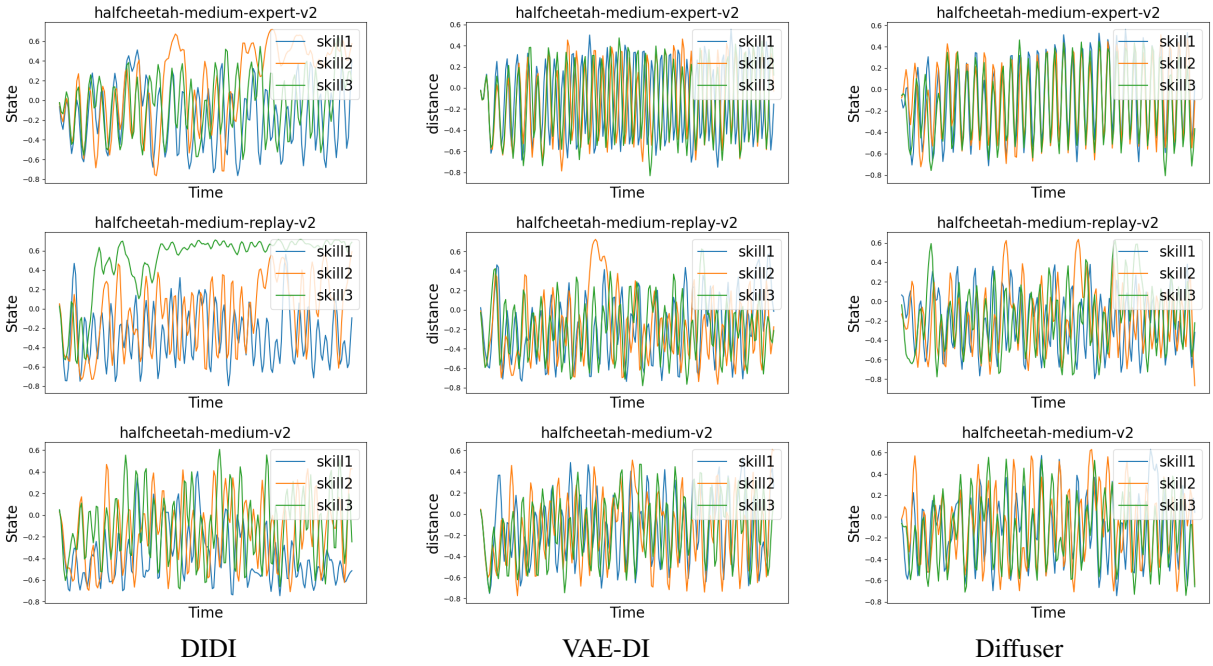

Figure 9. **Skill (Diversity) Visualization.** The figures illustrate the learned skills by DIDI, VAE-DI, and Diffuser on the D4RL halfcheetah domain. The skills demonstrated by DIDI exhibit a high level of diversity compared to VAE-DI and Diffuser. Each row represents a different task (halfcheetah-medium-expert-v2, halfcheetah-medium-replay-v2, halfcheetah-medium-v2), and each column compares the performance of the three methods. The Y-axis represents the state, and the X-axis represents time.