The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Stitching Sub-trajectories with Conditional Diffusion Model for Goal-Conditioned Offline RL

Sungyoon Kim, Yunseon Choi, Daiki E. Matsunaga, Kee-Eung Kim
Kim Jaechul Graduate School of AI, KAIST
{sykim, yschoi, dematsunaga}@ai.kaist.ac.kr, kekim@kaist.ac.kr

Abstract

Offline Goal-Conditioned Reinforcement Learning (Offline GCRL) is an important problem in RL that focuses on acquiring diverse goal-oriented skills solely from pre-collected behavior datasets. In this setting, the reward feedback is typically absent except when the goal is achieved, which makes it difficult to learn policies especially from a finite dataset of suboptimal behaviors. In addition, realistic scenarios involve long-horizon planning, which necessitates the extraction of useful skills within sub-trajectories. Recently, the conditional diffusion model has been shown to be a promising approach to generate high-quality long-horizon plans for RL. However, their practicality for the goal-conditioned setting is still limited due to a number of technical assumptions made by the methods. In this paper, we propose SSD (Sub-trajectory Stitching with Diffusion), a model-based offline GCRL method that leverages the conditional diffusion model to address these limitations. In summary, we use the diffusion model that generates future plans conditioned on the target goal and value, with the target value estimated from the goal-relabeled offline dataset. We report state-of-the-art performance in the standard benchmark set of GCRL tasks, and demonstrate the capability to successfully stitch the segments of suboptimal trajectories in the offline data to generate high-quality plans.

1 Introduction

Offline Reinforcement Learning (Offline RL) (Lange, Gabel, and Riedmiller 2012; Levine et al. 2020), which aims to train agents from pre-collected datasets without any further interaction with the environment, has emerged as a promising paradigm in RL, in that it addresses the safety issue associated with exploring directly in the environment. One of the most important factors towards successfully training the agent is to *stitch* together the segments of sub-optimal behaviors in the dataset to generate high-quality behaviors (Kumar et al. 2022).

This stitching behavior becomes even more important when we consider Goal-Conditioned Reinforcement Learning (GCRL), where the agent aims to learn a policy that achieves specified goals. In contrast to traditional RL where the policy is solely conditioned on states, a GCRL policy is additionally conditioned on goals. Thus, a GCRL agent is tasked with learning a goal-conditioned policy prepared for diverse goals. In order to do so, it is essential to be able to stitch valuable parts of the behavior in the dataset to generate plans that constitute a policy that achieves a diverse set of goals. However, especially with sub-optimal behaviors in the dataset, the sparse nature of the rewards makes it challenging to accurately capture the valuable parts and integrate them appropriately.

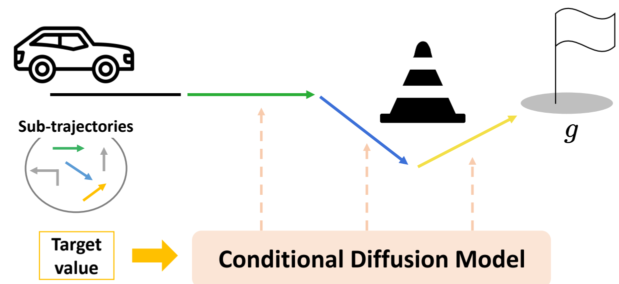


Figure 1: Our approach uses a diffusion model conditioned on the goal and return-to-go that generates sub-trajectories. These sub-trajectories are then be stitched together to achieve the goal.

The challenge originating from the sparse rewards becomes more pronounced when agents are tasked with long planning horizons, exemplified by many practical scenarios such as a service robot that needs to carry out a long sequence of complex manipulations to accomplish the goal. While current deep RL algorithms excel at learning policies for short-horizon tasks, they struggle to effectively reason over extended planning horizons (Shah et al. 2022). Recently, diffusion-based planning methods have been introduced as a promising approach to generate high-quality long-horizon plans (Janner et al. 2022; Ajay et al. 2023). However, some of the technical assumptions made by the methods make them hard to be used directly for GCRL. For example, Diffuser (Janner et al. 2022) relies on inpainting, where the last state in the trajectory is set to the goal in order to generate behaviors that terminate at the goal location. If the length of the plan is conservatively set to be longer than the length of the optimal plan, it would end up generating excessively long trajectories in the aforementioned tasks.

In this paper, we propose a novel approach named *SSD*

(Sub-trajectory Stitching with Diffusion) that utilizes the conditional diffusion model to promote behavior stitching for Offline GCRL, as illustrated in Figure 1. We begin with training the action-value function using multi-step goal chaining with relabeled goals, which extends the method in (Chebotar et al. 2021) to further enrich reward signals. We then use the estimated action value for conditioning the diffusion model with the *target value*, circumventing the need for optimal plan length (Janner et al. 2022), explicit subgoals (Florensa et al. 2018; Chane-Sane, Schmid, and Laptev 2021) or hierarchical architectures (Li et al. 2023). Regarding the conditional diffusion model, we propose an effective architecture for generating realistic trajectories, named Condition-Prompted-Unet. By incorporating transformer blocks instead of convolutional layers, we were able to improve conformity with the dynamics of the environment. Lastly, we empirically substantiate our approach and demonstrate state-of-the-art performance on Maze2D and Multi2D, where we also qualitatively showcase the stitching capabilities of SSD.

2 Preliminaries and Related Works

GCRL is formulated as goal-augmented MDP (GA-MDP) (Liu, Zhu, and Zhang 2022), represented by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \mathcal{G}, p_g, \psi, \mu_0, \gamma)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the probabilistic state transition function, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \rightarrow \mathbb{R}$ is the reward function, $\mathcal{G}$ is the set of goals, p_g is the probability distribution of the desired goal, $\psi : \mathcal{S} \rightarrow \mathcal{G}$ is the state-to-goal mapping, $\mu_0 \in \Delta(\mathcal{S})$ is the initial state distribution, and $\gamma \in [0, 1]$ is the discount factor. We consider a sparse binary reward function $R(s_t, a_t, g) \equiv \mathbf{1}(\psi(s_t) = g)$ which yields a reward of one if and only if the current state satisfies the goal.

Given a GA-MDP, we aim to find goal conditioned policy $\pi : \mathcal{S} \times \mathcal{G} \rightarrow \Delta(\mathcal{A})$ that maximizes the expected return:

$$J(\pi) = \mathbb{E}_{\substack{g \sim p_g(\cdot), s_0 \sim \mu_0(\cdot), \\ a_t \sim \pi(\cdot|s_t, g)}} [\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t, g)] \tag{1}$$

We assume the offline setting for GCRL where we aim to find the optimal policy from the dataset $\mathcal{D}$, comprised of trajectories collected by executing a potentially sub-optimal policy. Each trajectory in $\mathcal{D}$ is represented by $\tau = s_{0:T}||a_{0:T}||g$, which is the concatenation of state sequence $s_{0:T} = (s_0, \ldots, s_T)$, action sequence $a_{0:T} = (a_0, \ldots, a_T)$, and desired goal g, where T is the final time-step of the trajectory. We assume each trajectory terminates when it achieves the goal or surpasses a sufficiently large time-step limit. Thus, the action value $Q(s, a, g)$ represents the discounted probability of policy π achieving the goal g from the state s:

$$\begin{aligned} Q^\pi(s, a, g) &\equiv \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, g) | s_0 = s, a_0 = a, g] \\ &= (1 - \gamma) p^\pi(s_T = g | s_0 = s, a_0 = a) \end{aligned} \tag{2}$$

In our approach, we aim to learn from sub-trajectory segments of length h+1, denoted as $s_{t:t+h}||a_{t:t+h}$, with $t \in [0, T - h]$. We denote the reward function for sub-trajectories (Chebotar et al. 2021) as $R(s_{t:t+h}, a_{t:t+h}, g) \equiv \mathbf{1}(\psi(s_{t+h}) = g)$, and the space of all possible sub-trajectories as $\mathcal{SA}^h$.

2.1 Diffusion Models for RL

Diffusion models (Ho, Jain, and Abbeel 2020; Song and Ermon 2019; Song, Meng, and Ermon 2020; Song et al. 2021) are well known for effectively capturing complex distributions and generating diverse samples. As such, they have recently emerged as an effective approach for GCRL (Janner et al. 2022; Li et al. 2023). They have been shown to generate long-horizon plans without separate model learning, mitigating the issue of error accumulation often encountered in traditional planning algorithms. Additionally, their generating process can be controlled by specifying conditions such as optimality, goal and skill.

Denoising Diffusion Probabilistic Model (DDPM) Following the prior works (Janner et al. 2022; Ajay et al. 2023), we leverage the strong capabilities of DDPM (Ho, Jain, and Abbeel 2020) to generate sub-trajectories of states and actions $\mathbf{x}^0 \equiv s_{t:t+h}||a_{t:t+h}$[1]. DDPM is composed of (1) a predefined forward noising process q that incrementally adds noise to generate pure random noise $\mathbf{x}^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where the data is corrupted according to a pre-determined variance schedule $0 < \alpha_i < 1$, and (2) a trainable reverse denoising process p_θ that learns to recover $\mathbf{x}^0$ from $\mathbf{x}^N$, which yields the sampling distribution $\mathbf{x}^0 \sim p_\theta(\mathbf{x}^0)$.

$$\begin{aligned} q(\mathbf{x}^{i+1}|\mathbf{x}^i) &\equiv \mathcal{N}(\mathbf{x}^{i+1}; \sqrt{\alpha_i}\mathbf{x}^i, (1 - \alpha_i)\mathbf{I}), \\ p_\theta(\mathbf{x}^{i-1}|\mathbf{x}^i) &\equiv \mathcal{N}(\mathbf{x}^{i-1}; \mu_\theta(\mathbf{x}^i, i), \mathbf{\Sigma}_i) \end{aligned}$$

Then, the variational lower bound on log-likelihood is used as the objective function for training:

$$\begin{aligned} \mathbb{E}[-\log p_\theta(\mathbf{x}^0)] &\leq \mathbb{E}_q\big[-\log \frac{p_\theta(\mathbf{x}^{0:N})}{q(\mathbf{x}^{1:N}|\mathbf{x}^0)} \big] \\ &= \mathbb{E}_q\big[-\log p(\mathbf{x}^N) - \sum_{i \geq 1} \log \frac{p_\theta(\mathbf{x}^{i-1}|\mathbf{x}^i)}{q(\mathbf{x}^i|\mathbf{x}^{i-1})} \big] \\ &\equiv L(\theta). \end{aligned} \tag{3}$$

We can formulate a reparameterized version of the objective function for more efficient optimization. For details, we refer the readers to (Ho, Jain, and Abbeel 2020)

Diffuser Diffuser (Janner et al. 2022) is one of the early works on adopting diffusion models for RL, and showed promising results on generating high-quality plans in both offline RL and GCRL. Diffuser is underpinned by reframing RL as a conditional sampling problem by adopting the infer-to-control framework. More specifically, the reverse denoising process is reformulated as an instance of classifier-guided sampling (Sohl-Dickstein et al. 2015; Dhariwal and

[1]We use superscripts to denote the steps in the diffusion process, while subscripts to denote the time-steps in the trajectories.

Nichol 2021), where the guide is given by the return of the sub-trajectory

$$p_\theta(\mathbf{x}^{i-1}|\mathbf{x}^i, R(\mathbf{x}^0, g)) \approx \mathcal{N}(\mathbf{x}^{i-1}; \mu + \Sigma\nabla R, \Sigma) \quad (4)$$

where $\nabla R \equiv \nabla R(\mathbf{x}^0, g)|_{\mathbf{x}^0=\mu}$ is gradient of the return of the sub-trajectory $\mathbf{x}^0$, and μ and Σ are the parameters of the plain reverse process $p_\theta(\mathbf{x}^{i-1}|\mathbf{x}^i)$. It is worthwhile to note that when it comes to the goal-conditioned setting, Diffuser relies on inpainting to generate trajectories that achieves the goal at the final time-step, rather than using the goal-dependent value function $Q(s, a, g)$ to guide the sampling process.

2.2 Goal Relabeling Methods for GCRL

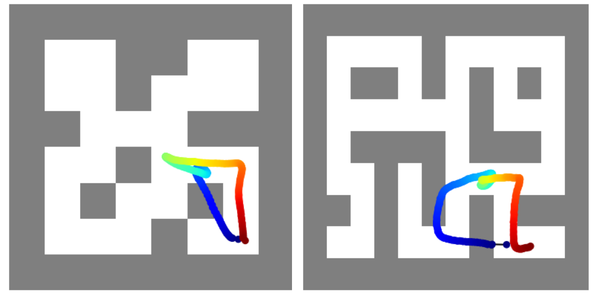
Goal-Conditioned Reinforcement Learning (GCRL) aims to train RL agents under a multi-goal setting (Kaelbling 1993; Schaul et al. 2015). Its offline counterpart aims to do so solely from the trajectory dataset without further interaction with the environment. One of the main challenges in offline GCRL is that, the reward signals in the dataset is extremely sparse especially when the data collection policy is sub-optimal and rarely achieves the goal. Thus, in order to capture valuable skills for achieving diverse goals, goal relabeling methods are commonly employed.

Hindsight Experience Replay (HER) Hindsight Experience Replay (Andrychowicz et al. 2017) involves replacing the specified goals by those corresponding to states actually visited during execution. For example, given trajectory $\tau = s_{0:T}||a_{0:T}||g$, replacing the goal g with the one corresponding to the final state $g' = \psi(s_T)$ yields a successful trajectory with non-zero reward. Through such goal relabeling, we can enrich the reward signal for learning the value function, allowing us to extract useful skills even from unsuccessful trajectories with zero reward. Although originally developed for the online RL setting, HER has shown to significantly improve the performance in the offline RL setting as well.

Actionable Model (AM) One shortcoming of HER is its constant generation of successful trajectories, resulting in biased training samples. Furthermore, the relabeled goals are confined to one of the states visited within the trajectory, which obstructs learning to achieve a goal present in other trajectories. More recently, AM (Chebotar et al. 2021) introduced a goal relabeling strategy that generates not only successful trajectories but also unsuccessful ones. This is done by relabeling with goal g' corresponding to the state arbitrarily selected from the whole dataset. In order to learn from unsuccessful trajectories where g' is sampled from those not appearing in the current trajectory, AM introduces a technique called *goal chaining* to examine whether g' is still achievable in some other trajectory.

This is performed by training the action-value function with the objective

$$L(\phi) = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D},g'\sim G}[(Q_\phi(s_t, a_t, g') - y(s_{t+1}, g'))^2], \quad (5)$$



(a) DD – Maze2d Medium (b) DD – Maze2d Large

Figure 2: Plot of sampled trajectory from Decision Diffuser (DD) (Ajay et al. 2023). It generates unrealistic trajectories that pass through the wall. The blue point is the initial point and the red point is the goal. left: Maze2D-Medium. right: Maze2D-Large.

with the goal relabeling strategy represented by G and the target y given by

$$y(s_{t+1}, g') = \begin{cases} 1 & \text{if } \psi(s_t) = g' \\ \gamma\mathbb{E}_{a\sim\pi}[Q_\phi(s_{t+1}, a, g')] & \text{otherwise.} \end{cases} \quad (6)$$

The key idea behind the target y is that even when the goal g' is not immediately achievable in the current trajectory, there might be some other trajectory that ultimately achieves the goal, and if so, it will be reflected into the action-value function of state s_{t+1} from that trajectory, thus effectively chaining the two trajectories through state s_{t+1}.

3 Method

In this section, we introduce SSD (Sub-trajectory Stitching with Diffusion), which utilizes the conditional diffusion model to address long-horizon tasks in offline GCRL. More concretely, SSD addresses two main challenges in offline GCRL: first, the sparsity of rewards makes it difficult to identify good sub-trajectories to stitch. Second, the conditional diffusion model, while holding promise in tackling long-horizon tasks, tends to generate unrealistic trajectories, as we show in Figure 2 for Decision Diffuser (DD) (Ajay et al. 2023). To overcome these challenges, SSD alternates between training (1) the goal-conditioned value function with multi-step goal chaining and (2) the value-conditioned diffusion model for generating sub-plans of length h. We provide details on each part in Sections 3.1 and 3.2, respectively. The full algorithm for our approach is provided in Section 3.3.

3.1 Value-Conditioned Diffusion Model

In order to generate more realistic sub-trajectories, we propose an architecture we name Condition-Prompted-Unet.

As shown in Figure 3, Condition-Prompted-Unet integrates the Unet architecture with transformer blocks. The transformer blocks provide the capacity to capture complex patterns of real-world sequential data, while the Unet structure facilitates preservation of spatial information. By re-

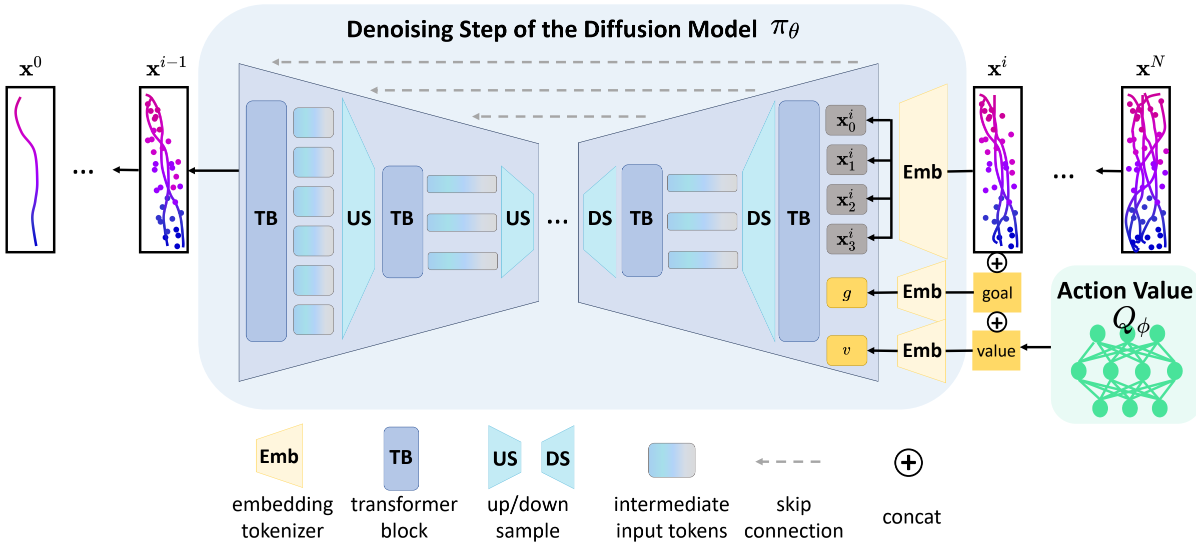


Figure 3: Architecture of the proposed Condition-Prompted-Unet model. Trajectory input is tokenized into state-action pairs. The goal and value are included as prompt tokens into the transformer layers, which constitute the Unet structure. Downsampling and Upsampling layers are integrated with each transformer block. The process is repeated for each reverse step, using the given goal and the trained action value as conditions.

placing convolutional blocks with transformer blocks, our approach aims to enhance the accuracy of the conditioning process. This hybrid architecture is designed to address the challenges of realistic trajectory generation in RL domains, ensuring compatibility with environment constraints (e.g. staying within the state space and following the transition probabilities of the MDP). The main motivation behind our model was that although the guidance model in Diffuser (Eqn. (4)) provides an attractive framework to leverage return-to-go for conditional generation of trajectory, the closed-form approximation therein was the main source of bottleneck for generating accurate, realistic trajectories. Instead, the approximation is directly learned in our model.

This conditional diffusion model π_θ is trained to generate sub-trajectories $\mathbf{x}^0 \equiv \mathbf{x}^0_{t:t+h}$ given the random noise $\mathbf{x}^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the target goal g, and the desired probability specified by the target value v, using the reparameterized version of the DDPM loss function in Eqn. (3) (Ho, Jain, and Abbeel 2020).

3.2 Multi-Step Goal Chaining

Another key component of our method is the multi-step goal chaining. Given that the diffusion model is trained with a batch of data that corresponds to sub-trajectories of length h+1, a natural goal relabeling strategy would be sampling one of the states within the sub-trajectory to make the sub-trajectory successful. However, sampling only within the sub-trajectory will incur a positive bias as in HER, thus we do so with probability 0.5, and select an arbitrary random state outside the sub-trajectory with probability 0.5. For the latter, we would need to perform goal chaining as in AM since it corresponds to learning from an unsuccessful sub-trajectory.

A straightforward target for training the action-value function can be derived as follows: given sub-trajectory $\mathbf{x} \equiv s_{t:t+h} \| a_{t:t+h}$ and relabeled goal g', we define target y by

$$y(\mathbf{x}, g') = \begin{cases} \gamma^{k-1} & \text{if } \exists k \in [0, \dots, h] : \psi(s_{t+k}) = g' \\ \max_{1 \le k' \le h} \gamma^{k'} \mathbb{E}_{a \sim \pi}[Q_\phi(s_{t+k'}, a, g')] & \text{o.w.} \end{cases}$$

This target has the following interpretation: if the relabeled goal happens to be achievable by sampling one of the states within the sub-trajectory, we use the discounted return γ^{k-1}. Otherwise, we attempt goal chaining at every subsequent state in the sub-trajectory and choose the best one.

Although this target is natural, we found that it incurs too much overestimation due to the max operator on the bootstrapped estimates in the offline setting. Thus, given that the sub-trajectory batch data is prepared for every time-step, we found it sufficient to attempt goal chaining only at time-step t+1. In other words, we define the target y by

$$y(\mathbf{x}, g') = \begin{cases} \gamma^{k-1} & \text{if } \exists k \in [0, \dots, h] : \psi(s_{t+k}) = g' \\ \gamma Q_\phi(s_{t+1}, a, g') & \text{otherwise,} \end{cases} \tag{7}$$

where $a \sim \pi$ is a sample from the target policy to approximate $Q_\phi(s_{t+1}, a, g') \approx \mathbb{E}_{a \sim \pi}[Q_\phi(s_{t+1}, a, g')]$.

3.3 Overall Algorithm

The pseudocode of the overall training procedure is shown in Algorithm 1. We use the same goal relabeling strategy G for training the critic Q_ϕ and the diffusion model π_θ, where we randomly select one of the states in the sub-trajectory

Algorithm 1: SSD (Training)

Require: offline data $\mathcal{D}$, diffusion model π_θ, critic Q_ϕ, number of training iterations K, horizon h, goal relabeling strategy G

for i in $1, ..., K$ **do**

  $\mathbf{x}_{t:t+h} = s_{t:t+h} \| a_{t:t+h} \sim \mathcal{D}$, $g' \sim G(\mathbf{x}_{t:t+h}, \mathcal{D})$ *// sample a sub-trajectory and its relabeled goal*

  ## Train action-value critic Q_ϕ

  Take a gradient descent update $\nabla_\phi L(\phi)$ using the target y in Eqn. (7), i.e.:

$$L(\phi) = \frac{1}{2} \left[Q_\phi(s_t, a_t, g') - y(s_{t:t+h}, a_{t:t+h}, g') \right]^2,$$
$$\text{where } y(s_{t:t+h}, a_{t:t+h}, g') = \begin{cases} \gamma^{k-1} & \exists k \in [0, \ldots, h] : \psi(s_{t+k}) = g' \\ \gamma Q_\phi(s_{t+1}, a \sim \pi_\theta, g') & \text{otherwise.} \end{cases}$$

  ## Train diffusion model π_θ

  if $\exists k \in [0, \ldots, h] : \psi(s_{t+k}) = g'$ **then**

    $\mathbf{x}_{t:t+h} = \mathrm{PAD}(s_{t:t+k}, h-k, s_{t+k}) \| \mathrm{PAD}(a_{t:t+k}, h-k, a_{t+k})$

    // Goal g' corresponds to state s_{t+k}, so replace subsequent entries by s_{t+k} and a_{t+k} since a successful sub-trajectory

  end if

  Take a gradient descent update $\nabla_\theta L(\theta)$ with $\mathbf{x}_{t:t+h} \sim \pi_\theta(\boldsymbol{\epsilon}, g', Q_\phi(s_t, a_t, g')), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ using the reparameterized version of Eqn. (3)

end for

return π_θ, Q_ϕ

with probability 0.5, and randomly select one of the states outside the sub-trajectory with probability 0.5, as described previously. Although we chose uniform probability distribution for simplicity, other choices of probabilities for the goal relabeling strategy are certainly possible for further optimization.

Finally, during the execution phase, we sample a trajectory $\mathbf{x}$ from π_θ every k steps with the first state inpainted as s_t, and use the first k sequence of $\mathbf{x}$ to take actions. This process is repeated until a goal is reached, whereby the episode terminates.

4 Experimental Results

In this section, we demonstrate the effectiveness of the proposed SSD approach in two different GCRL domains: Maze2D and Fetch. Our code is available publicly at https://github.com/rlatjddbs/SSD

4.1 Long-Horizon Planning Tasks

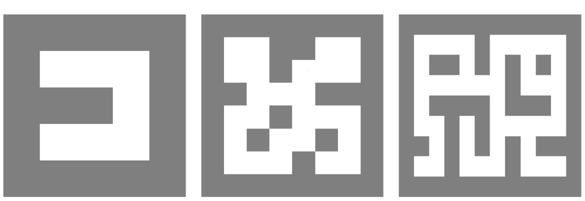


Figure 4: Layouts of Maze2d environment. left: Umaze, middle: Medium, and right: Large

Maze2D Maze2D is a popular benchmark for goal-oriented tasks, where a point mass agent navigates to a goal location. These tasks are intentionally devised to assess the stitching capability of offline RL algorithms. The environment encompasses three distinct layouts, each varying in difficulty and complexity, as depicted in Figure 4. Multi2D is the environment with the same layout with multi-task setting where the goal location changes in every episode.

We utilize the D4RL dataset (Fu et al. 2020), which is generated by a hand-designed PID controller as a planner, which produces a sequence of waypoints. The dataset comprises 1 million samples for Umaze, 2 million for Medium, and 4 million for Large.

Baselines Our main baseline for diffusion-based offline RL is Diffuser (Janner et al. 2022) which is described in detail in Section 2. For a fair comparison with our value-conditioned diffusion model, we also extended Diffuser to condition on the action value trained with HER and AM as goal relabeling strategies (Diffuser-HER and Diffuser-AM). Both methods use classifier guidance described in Eqn. (4). Decision Diffuser (DD) (Ajay et al. 2023) is another offline RL method based on the diffusion model, which uses classifier-free guidance and utilizes the inverse dynamics model for decision-making. Hierarchical Diffusion for offline decision MakIng (HDMI) (Li et al. 2023) employs a hierarchical diffusion model involving a planner that generates sub-goals. Finally, IQL (Kostrikov, Nair, and Levine 2022) is a state-of-the-art offline RL method which can be seen as advantage-weighted behavior cloning.

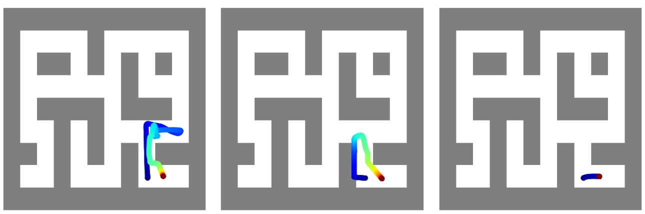
Results As we show in Table 1, SSD outperforms all baselines for all map sizes with a notable performance gap in larger maps and multi-task scenarios.

Overall, diffusion-based algorithms all outperform IQL, which demonstrates the benefit of trajectory-wise generation of actions.

Comparing Diffuser-HER and Diffuser-AM, the AM-based guidance exhibited superior performance in multi-task

Environment		IQL	Diffuser	Diffuser (HER)	Diffuser (AM)	DD	HDMI	SSD (Ours)
Maze2D	Umaze	47.4	113.9±1.8	125.4±0.4	121.4 ±1.8	116.2 ±2.7	120.1 ±2.5	**144.6**±3.4
Maze2D	Medium	34.9	121.5±2.7	130.3±2.3	127.2 ±3.1	122.3 ±2.1	121.8 ±1.6	**134.4**±6.1
Maze2D	Large	58.6	123.0±6.4	135.8±8.5	135.0 ±3.3	125.9 ±1.6	128.6 ±2.9	**183.5**±8.6
Single-task Average		47.0	119.5	130.5	127.9	121.5	123.5	**154.2**
Multi2D	Umaze	24.8	128.9±1.8	132.7±2.4	135.4 ±2.0	128.2 ±2.1	131.3 ±1.8	**158.2**±4.5
Multi2D	Medium	12.1	127.2±3.4	133.0±2.4	137.8 ±1.0	129.7 ±2.7	131.6 ±1.9	**155.2**±7.9
Multi2D	Large	13.9	132.1±5.8	139.2±6.6	145.7 ±7.7	130.5 ±4.2	135.4 ±2.5	**192.9**±8.5
Multi-task Average		16.9	129.4	135.0	139.6	129.5	132.8	**168.8**

Table 1: [Maze2D] Normalized scores of Maze2D environment. Goals are fixed in Maze2D scenarios, while goals are randomly sampled in Multi2D scenarios. Reported values are mean and standard error over 5 seeds. The bolded value represents the top-performing result.



(a) Diffuser (Janner et al.) (b) DD (Ajay et al.) (c) SSD (Ours)

Figure 5: Comparision of behaviors of diffusion based methods. (a), (b) and (c) are result of Diffuser (Janner et al. 2022), DD (Ajay et al. 2023), and SSD (Ours) respectively. The blue point is the initial point and the red point is the goal. All of them share the same initial point and the same goal.

scenarios where the goal switches in each episode, whereas the HER-based guidance demonstrated higher performance in simpler tasks with fixed goals. This is expected since the AM-guidance promotes stitching to improve handling of diverse goals. Also, we see that employing classifier guidance in Eqn. (4) on top of Diffuser improves the overall performance, especially when the guidance Q-value is trained based on the specified relabeling methods. Those variants of Diffuser show even outperforming other diffusion-based algorithms such as DD and HDMI.

While all the diffusion-based policies except for HDMI aim to generate single full-trajectory in evaluation time, SSD focuses on generating sub-trajectories and stitching them. This property removes the necessity to know the terminal time-step T a priori. Also, we show qualitative results in Figure 5 demonstrating that the baseline methods can produce unnecessary detouring trajectories instead of taking a more direct path as SSD does.

4.2 Robotic Manipulation Tasks

Baselines GCSL (Ghosh et al. 2021) combines behavior cloning with hindsight goal relabeling to make trajectories optimal. WGCSL (Yang et al. 2022) additionally uses a weighting scheme, by considering the importance of relabeled goals. GoFAR (Ma et al. 2022) is a *relabeling-free* method, formulating GCRL as a state-occupancy matching problem. DDPG refers to the DDPG (Lillicrap et al. 2016) with hindsight-relabeling (Andrychowicz et al. 2017) to learn a goal-conditioned critic.

Results We compare our approach to the aforementioned baseline methods in the Fetch environment, a robotic manipulation environment which includes FetchReach, FetchPush, FetchPickAndPlace, and FetchSlide. We use the offline dataset provided in Ma et al. (2022).

As our proposed method is formulated primarily to achieve the goal as soon as possible, we employ two quantitative metrics, namely success rate and discounted return, to rigorously evaluate and quantify its performance. As shown in Table 2, SSD achieves the state-of-the-art performance in average success rate across all tasks, by a large margin especially in FetchPush and FetchPickAndPlace. Also, SSD exhibits the state-of-the-art performance in average discounted return across all tasks, as shown in Table 3.

4.3 Performance Impact of Target Value v

Here, we provide additional results which show the performance across different values of v. When deployed to the environment, our algorithm chooses actions based on the target value v, akin to the return-to-go in Decision Transformer (Chen et al. 2021). The target value v represents the discounted probability of achieving a goal g from the current state. Since we want to achieve the goal as quickly as possible, we should assign a higher value to v at test time. However, an appropriate value for v depends on the complexity of the task. As shown in Figure 6, the target value for achieving the best performance decreases as the task complexity increases from Umaze to Large. This result implies that the target value conditioning is generally affecting the plan generation as desired.

5 Conclusion

In this work, we have presented a novel approach named SSD to address the challenges encountered in Offline GCRL, especially in long-horizon tasks. First, our proposed

Environment	**Supervised Learning**			**Q-value Based**		
	GCSL	WGCSL	GoFAR	AM	DDPG	SSD(Ours)
FetchReach	0.98 ±0.05	0.99 ±0.01	**1.0** ±0.0	**1.0** ±0.0	0.99 ±0.02	**1.0** ±0.0
FetchPickAndPlace	0.54 ±0.20	0.54 ±0.16	0.84 ±0.12	0.78 ±0.15	0.81 ±0.13	**0.9** ±0.03
FetchPush	0.72 ±0.15	0.76 ±0.12	0.88 ±0.09	0.67 ±0.14	0.65 ±0.18	**0.97** ±0.02
FetchSlide	0.17 ±0.13	0.18 ±0.14	0.18 ±0.12	0.11 ±0.09	0.08 ±0.11	0.1 ±0.07
Average	0.60	0.62	0.73	0.64	0.63	**0.74**

Table 2: [Fetch] Success rate in Fetch environment. Reported values are mean and standard error over 5 seeds. The bolded value represents the top-performing result.

Environment	**Supervised Learning**			**Q value Based**		
	GCSL	WGCSL	GoFAR	AM	DDPG	SSD(Ours)
FetchReach	20.91 ±2.78	21.9 ±2.13	28.2 ±0.61	**30.1** ±0.32	29.8 ±0.59	29.18 ±0.32
FetchPickAndPlace	8.94 ±3.09	9.84 ±2.58	**19.7** ±2.57	18.4 ±3.51	16.8 ±3.10	18.0 ±0.25
FetchPush	13.4 ±3.02	14.7 ±2.65	18.2 ±3.00	14.0 ±2.81	12.5 ±4.93	**19.84** ±0.50
FetchSlide	1.75 ±1.3	**2.73** ±1.64	2.47 ±1.44	1.46 ±1.38	1.08 ±1.35	2.6 ±0.94
Average	11.25	12.29	17.14	16.00	15.05	**17.40**

Table 3: [Fetch] Discounted return in Fetch environment. Reported values are mean and standard error over 5 different training seeds. The bolded value represents the top-performing result.

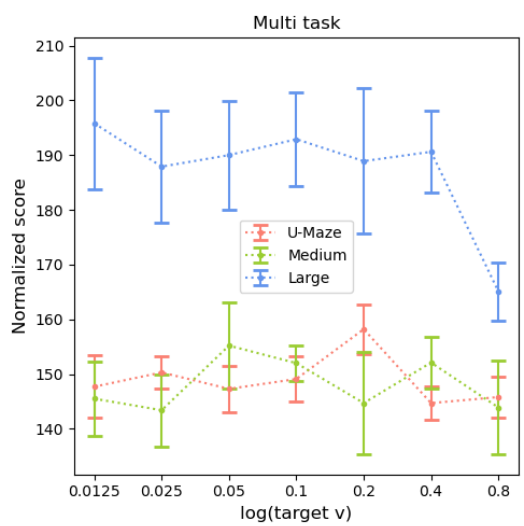


Figure 6: Plot of normalized scores across varying target values. Umaze domain achieves the highest score at $v = 0.2$, Medium domain at $v = 0.05$, and Large domain at $v = 0.0125$

approach offers effective solutions to overcome the issue of sparse rewards by generating sub-trajectories that are conditioned on action values. These action values are obtained through our novel hindsight relabeling technique, which enhances the stitching of trajectories. Second, our approach integrates the training of action values and conditioning within the diffusion training process, eliminating the need for separate guidance training. This integration minimizes approximation errors and leads to improved performance, as demonstrated in the Maze2D and Multi2D results. The experimental results validate the effectiveness of SSD and highlight its potential for practical applications in the field of GCRL.

Acknowledgments

This work was supported by the IITP grant funded by MSIT (No.2020-0-00940, Foundations of Safe Reinforcement Learning and Its Applications to Natural Language Processing; No.2022-0- 00311, Development of Goal-Oriented Reinforcement Learning Techniques for Contact-Rich Robotic Manipulation of Everyday Objects; No.2019-0-00075, AI Graduate School Program (KAIST)).

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 *The Eleventh International Conference on Learning Representations*.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Pieter Abbeel, O.; and Zaremba, W. 2017. Hindsight Experience Replay. In *Advances in Neural Information Processing Systems*, volume 30.

Chane-Sane, E.; Schmid, C.; and Laptev, I. 2021. Goal-Conditioned Reinforcement Learning with Imagined Subgoals. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 1430–1440. PMLR.

Chebotar, Y.; Hausman, K.; Lu, Y.; Xiao, T.; Kalashnikov, D.; Varley, J.; Irpan, A.; Eysenbach, B.; Julian, R. C.; Finn, C.; and Levine, S. 2021. Actionable Models: Unsupervised

Offline Reinforcement Learning of Robotic Skills. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 1518–1528. PMLR.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems*, volume 34, 15084–15097. Curran Associates, Inc.

Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 8780–8794.

Florensa, C.; Held, D.; Geng, X.; and Abbeel, P. 2018. Automatic Goal Generation for Reinforcement Learning Agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1515–1528. PMLR.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

Ghosh, D.; Gupta, A.; Reddy, A.; Fu, J.; Devin, C. M.; Eysenbach, B.; and Levine, S. 2021. Learning to Reach Goals via Iterated Supervised Learning. In *International Conference on Learning Representations*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, 6840–6851.

Janner, M.; Du, Y.; Tenenbaum, J. B.; and Levine, S. 2022. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 9902–9915. PMLR.

Kaelbling, L. P. 1993. Learning to Achieve Goals. In *International Joint Conference on Artificial Intelligence*.

Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

Kumar, A.; Hong, J.; Singh, A.; and Levine, S. 2022. Should I Run Offline Reinforcement Learning or Behavioral Cloning? In *International Conference on Learning Representations*.

Lange, S.; Gabel, T.; and Riedmiller, M. 2012. *Batch Reinforcement Learning*, 45–73. Berlin, Heidelberg: Springer Berlin Heidelberg.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *ArXiv*, abs/2005.01643.

Li, W.; Wang, X.; Jin, B.; and Zha, H. 2023. Hierarchical Diffusion for Offline Decision Making. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous Control with Deep Reinforcement Learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Liu, M.; Zhu, M.; and Zhang, W. 2022. Goal-Conditioned Reinforcement Learning: Problems and Solutions. arXiv:2201.08299.

Ma, Y. J.; Yan, J.; Jayaraman, D.; and Bastani, O. 2022. Offline Goal-Conditioned Reinforcement Learning via \$f\$-Advantage Regression. In *Advances in Neural Information Processing Systems*.

Schaul, T.; Horgan, D.; Gregor, K.; and Silver, D. 2015. Universal Value Function Approximators. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 1312–1320. Lille, France: PMLR.

Shah, D.; Toshev, A. T.; Levine, S.; and Ichter, B. 2022. Value Function Spaces: Skill-Centric State Abstractions for Long-Horizon Reasoning. In *International Conference on Learning Representations*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 2256–2265. Lille, France: PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising Diffusion Implicit Models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; and Ermon, S. 2019. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems*, volume 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Yang, R.; Lu, Y.; Li, W.; Sun, H.; Fang, M.; Du, Y.; Li, X.; Han, L.; and Zhang, C. 2022. Rethinking Goal-Conditioned Supervised Learning and its Connection to Offline RL. *arXiv preprint arXiv:2202.04478*.