Enhancing Sub-Optimal Trajectory Stitching: Spatial Composition RvS for Offline RL

Sheng Zang
Nanyang Technological University
Singapore
Institute for Infocomm Research,
A*STAR, Singapore
zang0015@e.ntu.edu.sg

Zhiguang Cao
Singapore Management University
Singapore
zgcao@smu.edu.sg

Bo An
Nanyang Technological University
Singapore
Skywork AI
Singapore
boan@ntu.edu.sg

Senthilnath Jayavelu
Institute for Infocomm Research,
A*STAR, Singapore
j_senthilnath@i2r.a-star.edu.sg

Xiaoli Li
Institute for Infocomm Research,
A*STAR, Singapore
Nanyang Technological University
Singapore
xlli@i2r.a-star.edu.sg

ABSTRACT

Reinforcement learning via supervised learning (RvS) has been known as a burgeoning paradigm for offline reinforcement learning (RL). While return-conditioned RvS (RvS-R) predominates across a wide range of datasets pertaining to the offline RL tasks, recent findings suggest that goal-conditioned RvS (RvS-G) outperforms in specific sub-optimal datasets where trajectory stitching is crucial for achieving optimal performance. However, the underlying reasons for this superiority remain insufficiently explored. In this paper, employing didactic experiments and theoretical analysis, we reveal that the proficiency of RvS-G in stitching trajectories arises from its adeptness in generalizing to unknown goals during evaluation. Building on this insight, we introduce a novel RvS-G approach, Spatial Composition RvS (SC-RvS), to enhance its ability to generalize to unknown goals. This, in turn, augments the trajectory stitching performance on sub-optimal datasets. Specifically, by harnessing the power of advantage weight and maximum-entropy regularized weight, our approach adeptly balances the promotion of optimistic goal sampling with the preservation of a nuanced level of pessimism in action selection compared to existing RvS-G methods. Extensive experimental results on D4RL benchmarks show that our SC-RvS performed favorably against the baselines in most cases, especially on the sub-optimal datasets that demand trajectory stitching.

KEYWORDS

Offline Reinforcement Learning; Goal Conditioned Reinforcement Learning via Supervised Learning; Sub-optimal Trajectory Stitch

ACM Reference Format:
Sheng Zang, Zhiguang Cao, Bo An, Senthilnath Jayavelu, and Xiaoli Li. 2025. Enhancing Sub-Optimal Trajectory Stitching: Spatial Composition RvS for Offline RL. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Offline reinforcement learning (RL) [26] is a methodology that aims at learning the policy of an agent based on pre-collected static data, bypassing the need for real-time interactions with the environment. Recent studies have revealed that the incorporation of supervised learning can significantly enhance the performance of offline RL, which is achieved without relying on the temporal difference (TD) learning [20, 22]. This approach, known as reinforcement learning via supervised learning (RvS) [10], transforms the traditional RL problem into a conditional imitation learning problem. In cases where sequential models like Transformers are employed, it turns into a conditional sequence generation problem [7, 18]. This reformulation can be extended to pre-training models [23, 24, 27], designed to solve a wide range of decision-making problems [25, 37], wherein the underlying models are usually conditioned on rewards.

Nonetheless, those models, identified as reward-conditioned methods (RvS-R), always fall short in one of the crucial capabilities of the offline RL, i.e., learning optimal policies by stitching sub-optimal trajectories from datasets [27], which hinders their applications into practice. Without TD learning, RvS-R cannot achieve temporal compositionality through dynamic programming and empirical findings [10] indicated that RvS-R is essentially an implict reward-based trajectory filtering. Consequently, it motivates and expedites an inclination to shift the focus towards goal-conditioned RvS (RvS-G). Prior work [10] has highlighted that RvS-G exhibits superior performance compared to RvS-R in certain tasks consisting of sub-optimal trajectories. However, what accounts for this, and what is the pivotal aspect of RvS-G that enables it to effectively combine sub-optimal trajectories to attain optimal performance, are rarely explored.

In addressing these questions, we first undertake a preliminary experiment in a point-mass environment to evaluate the trajectory stitching capability of RvS-G and other established offline RL algorithms. We find that the capability of RvS-G in stitching trajectories

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

can be regarded as its proficiency in generalizing to unknown goals. This is because that trajectory stitching is crucial exclusively for reaching unknown goals, while existing state-goal pairs only require straightforward behavioral cloning (BC) or optimization for a shorter trajectory path using importance weights. Building on this, we conduct a theoretical analysis to find the significant component of RvS-G to perform well in evaluating unknown goals compared to the expert policy, instead of solely relying on the neural network's generalization ability and latent space similarity.

Then motivated by theoretical insights, we propose a new RvS-G method called Spatial Composition RvS (SC-RvS) to enhance the trajectory stitching capability of RvS-G while remaining aligned and pessimistic about the dataset. In addition to goal relabeling, which expands the size of the state-goal dataset, our method leverages the advantage weight and maximum-entropy weight to effectively address the trajectory stitching problem at multiple levels. Remarkably, our SC-RvS achieves as a more generalized form that unifies existing RvS-G methods, providing a theoretical basis for the strong performance observed in vanilla RvS-G [10] or when simply incorporating goal relabeling [3]. Furthermore, we observe that the trade-off between trajectory stitching and reducing extrapolation errors can be somewhat mitigated in the context of RvS-G, allowing us to manage pessimism and generalization simultaneously. In essence, we can concurrently promote optimistic goal sampling while maintaining a degree of pessimism in action selection.

We conduct comprehensive experimental evaluations on D4RL benchmarks [12], where SC-RvS with efficient and effective trajectory stitching capability, performs favorably across diverse datasets against baselines.

2 RELATED WORK

Offline Reinforcement Learning. Offline RL lacks in addressing the distributional shift between behavior and action policies [26], which may result in inaccurate predictions for value queries of unseen actions. Potential solutions to address this issue include imposing constraints on the learning policy [13, 33] or assigning low estimated values to the unseen actions [20, 22]. Some alternatives are achieved through weighted imitation learning [5, 8], where trajectories with high returns are filtered and reserved or segments of trajectories with relative superiority are selected for imitation. A different perspective tackles this issue by conditioning it on specific information, formalized within the framework of reinforcement learning via supervised learning (RvS) [10]. Predominantly, most of the widely used RvS methods are conditioned on rewards (RvS-R) [21, 39], with recent advancement [7, 14] harnessing Transformer models for reward-conditioned sequence modeling. However, RvS-R struggles with trajectory stitching, and several works have proposed solutions to address this limitation, e.g., substituting return-to-go with expected values [44], adjusting the history length maintained at test time [42], and relabeling the return-to-go for each trajectory as the maximum total reward within a series of trajectories [27]. In contrast, our approach falls within the realm of goal-conditioned RvS (RvS-G), which excels in trajectory stitching, and we also incorporates weighted imitation learning. Unlike RvS-R, RvS-G has not garnered as much attention and is often discussed within the context of goal-conditioned RL [9, 28]. The first common RvS-G framework was introduced by [10], wherein goals are established by randomly selecting the near-terminal states or searching for the best ones through accessing the environment. However, such access is untenable within offline RL settings. Besides, Policy-guided Offline RL (POR) [43] focuses on next-state generalization by stitching neighboring states along the trajectory, which can be viewed as a special case of our SC-RvS approach. Furthermore, another recent approach, proposed by [3], employs waypoints as conditioned goals to better facilitate stitching, resembling a variant of SC-RvS with goal relabeling alone.

Goal-conditioned Reinforcement Learning. Within the realm of RL, goal-conditioned approaches are recognized for their ability to guide agents in achieving specific objectives. Among these, Hindsight Experience Replay (HER) [2] stands out as a crucial technique that adeptly addresses the challenge of sparse rewards. HER introduces an effective strategy that involves relabeling rewards and transitions based on unsuccessful trajectories. This approach can also be regarded as an implicit curriculum [36, 38], widely applicable in various goal-conditioned scenarios [9, 32, 40]. Particularly noteworthy is the synergy between HER and goal-conditioned imitation learning techniques [9], which empowers agents to learn from expert demonstrations. When human supervisor is absent, the task shifts to goal-conditioned RL, focusing on maximizing the discounted cumulative return. Some methods exclusively concentrate on the reward at the final step [16]. In the realm of offline goal-condition RL, current methods can also be categorized into policy regularization [29, 46] and value underestimation [6]. Additionally, hierarchical RL [32] offers hierarchical structures intertwined with goal-conditioning for the enhanced efficiency and exploration. Our proposed SC-RvS effectively aligns with these methodologies due to their shared essence of goal-conditioning. The difference lies in the broader applicability of SC-RvS. It extends beyond the confines of goal-conditioned tasks, operating without the prerequisite of prior task knowledge or expert demonstrations. Moreover, the central objective of our approach hinges on achieving compositionality in space through goal-conditioning for superior performance.

3 PRELIMINARIES

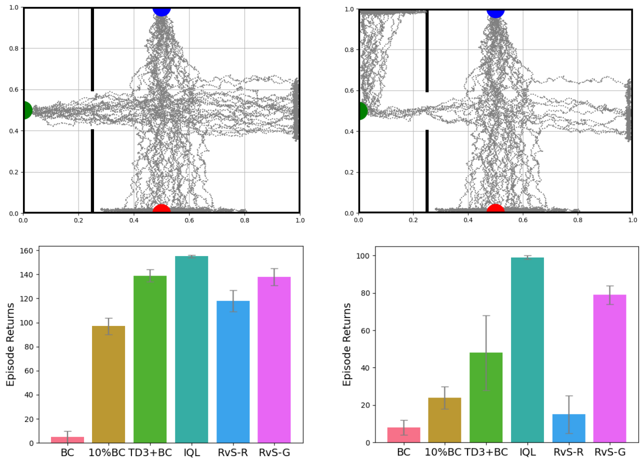
Markov Decision Process and Offline RL. The RL problem can be cast as a Markov Decision Process (MDP), and defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$. In this definition, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition probability; $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function and $\gamma \in (0, 1]$ is the discount factor; $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ denotes the policy, and the objective of RL is to learn an optimal policy $\pi(a|s)$ that maximizes $\mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)} [\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. For offline RL, during the training process the agent operates exclusively with a fixed dataset $\mathcal{D}$ without interaction with the environment, i.e. $(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}$.

RL via Supervised Learning. Traditional RL methods adopt techniques like policy gradient and temporal difference (TD) learning to train the policy. Whereas, reinforcement learning via supervised learning (RvS) offers an alternative yet effective perspective to perform conditioned behavior cloning. The objective of RvS is to derive an additional hindsight information conditioned policy π from the dataset $\mathcal{D}$ by maximizing $\mathbb{E}_{(a_t,s_t,\hat{R}(s_t))\sim\mathcal{D}}\left[\log \pi\left(a_t|s_t,\hat{R}(s_t)\right)\right]$ or $\mathbb{E}_{(a_t,s_t,G(s_t))\sim\mathcal{D}}\left[\log \pi\left(a_t|s_t,G(s_t)\right)\right]$. Here, $\hat{R}(s_t)$ refers to return-to-go and $G(s_t)$ represents goal states, which are the two primary variables utilized for conditioning.

4 TRAJECTORY STITCHING FOR RVS-G

In this section, we evaluate the trajectory stitching capabilities of various offline RL algorithms in a straightforward point-mass environment. Drawing insights from these practical observations, we deduce that the trajectory stitching capability of RvS-G mainly stems from its ability to generalize to unknown goals during evaluation. We then introduce a theoretical analysis to further understand this.

4.1 Didactic Example



(a) Stitch-easy

(b) Stitch-hard

Figure 1: Results on two datasets that require stitching. The red dots in the figure denote the goal, while the green and blue dots represent two distinct starting points. Specifically, the green dot represents the initial state during evaluation. The gray lines indicate trajectories, and the black represents obstructive walls.

Similar to previous work investigating the trajectory stitching capability of return-conditioned supervised learning [4], we construct two datasets, i.e., stitch-easy and stitch-hard, in a basic point-mass environment with sparse rewards, incorporating a constraining wall to shape the trajectories towards the goal. Visualizations of the datasets and corresponding results can be found in Figure 1.

In the stitch-easy dataset, we incorporate two types of trajectories: some trajectories move to the right from the initial state region but do not reach the goal, while others progress towards the goal from the upper side of the environment. In contrast, the stitch-hard dataset introduces a "hard" element with trajectories starting from the initial state and moving upwards, presenting a deliberate distraction for methods biased towards the behavior. The dominant action from the initial state now favors moving upward instead of towards the goal-reaching trajectories. To compare the trajectory stitching capabilities of various offline RL methods, we preliminarily examine six methods across three categories: behavior cloning (BC) and 10%BC [7], the latter of which utilizes the top 10% highest return trajectories; TD3+BC [13] and IQL [20], two representative TD-learning approaches; and vanilla RvS-R [10] and vanilla RvS-G [10] employing two-layer feedforward MLP policies.

4.2 Empirical Observation

In the stitch-easy dataset, RvS-R exhibits a remarkable performance although it is conditioned on an unknown return for the initial state during evaluation, as there is no positive return values for the green dot in Figure 1 (a). This suggests that RvS-R can align well with the behavior policy that moves to the right when conditioned on an unknown return, navigating the environment until it reaches a state (represented as the crossroads in Figure 1 (a)) where the return information serves as a crucial signal guiding the agent towards the goal. Similar analysis also applies to the performance of RvS-G, taking into account the unseen red dot for the green dot. However, in the stitch-hard dataset, RvS-R encounters difficulties and is even worse than 10%BC. This is because the return values of the two different trajectories both tend to negative infinity, and RvS-R is no longer capable of generalizing solely through the neural network's generalization ability. Consequently, the learned policy tends to default to the behavior. Given that actions from the initial state predominantly move upwards in Figure 1 (b), RvS-R follows this, leading to a low success rate. On the other hand, a similar issue appears to be mitigated with RvS-G. In this case, the two different trajectories from the initial state have different final goals, and RvS-G exhibits better generalization in the latent state space compared to the numerical space of return values. More specifically, in the stitch-easy task, we define a dataset $D = \{(s_0, s_2, s_4), (s_1, s_2, s_3)\}$, where s_0 is the green dot, s_1 is the blue dot, s_3 is the red dot, and s_2 is the intersection. The learning objective is to move from s_0 to s_3 by stitching partial trajectories. Using RvS-R, the partial trajectory (s_0, s_2) can be combined with (s_2, s_3) to form a complete trajectory. In contrast, the stitch-hard task adds a new trajectory (s_0, s_5), making the decision process more complex. With multiple actions from s_0, RvS-R fails due to ambiguity in action selection. However, RvS-G conditions on goals instead of returns, simplifying decision-making by focusing on the goal, with goal relabeling and a goal generator further easing the task.

4.3 Theoretical Analysis

However, we cannot solely depend on the generalization ability of the neural network itself and the similarity of the states in latent space. Therefore, we strive to identify the key factors that contribute to the strong performance of RvS-G during evaluation through theoretical analysis and attempt to make enhancements over vanilla RvS-G. It is noteworthy that our primary focus here lies in the performance of RvS-G at the evaluation stage, rather than emphasizing the state-goals already present in the offline training dataset. We posit the formulation that the state-goal pairs in the offline training dataset are denoted as $(s_t, g) \sim P^{\mathcal{D}}$. During evaluation, initial states and desired goals are sampled from distribution $P^{\mathcal{T}}$, and $P^{\mathcal{T}} \neq P^{\mathcal{D}}$, which indicates the goals sampled during evaluation are unknown. The objective here is to minimize the discrepancy between policy π and the expert policy π_E, denoted as $V^{\pi_E} - V^{\pi}$, across the evaluation dataset $P^{\mathcal{T}}$.

Lemma 4.1 (Performance Difference Lemma [19]). *For any π and π', it holds that*

$$V^{\pi} - V^{\pi'} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim P}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[A^{\pi'}(s,a)\right].$$

By Lemma 4.1, we have

$$\begin{aligned} V^{\pi_E} - V^{\pi} &= -\frac{1}{1-\gamma}\mathbb{E}_{(s,g)\sim P^{\mathcal{T}}}\mathbb{E}_{a\sim\pi(\cdot|s,g)}\left[A^{\pi_E}(s,g,a)\right] \\ &\le \frac{R_{\max}}{(1-\gamma)^2}\mathbb{E}_{(s,g)\sim P^{\mathcal{T}}}\left[\sqrt{2\mathrm{TV}(\pi(\cdot \mid s,g), \pi_E(\cdot \mid s,g))}\right], \end{aligned} \tag{1}$$

The detailed proof can be found in supplement material.

For simplicity, in the subsequent analysis, we denote $\mathbb{E}_{(s,g)\sim P^{\mathcal{T}}}\left[\sqrt{\mathrm{TV}(\pi(\cdot \mid s,g), \pi_E(\cdot \mid s,g))}\right]$ as $D_{\mathrm{TV}}^{\mathcal{T}}(\pi, \pi_E)$. D_{TV} is the Total Variation Distance. The upper bound of $D_{\mathrm{TV}}^{\mathcal{T}}(\pi_E, \pi)$ is:

$$\begin{aligned} D_{\mathrm{TV}}^{\mathcal{T}}(\pi_E,\pi) &= D_{\mathrm{TV}}^{\mathcal{T}}(\pi_E,\pi) + D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\pi) - D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\pi) \\ &\le D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\pi) + |D_{\mathrm{TV}}^{\mathcal{T}}(\pi_E,\pi) - D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\pi)| \\ &\le D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\pi) + \sum_{(s,g)} |P^{\mathcal{T}}(s,g) - P^{\mathcal{D}}(s,g)|. \end{aligned} \tag{2}$$

Lemma 4.2 (Generalization Bound for Finite ERM [30]). *Let $\mathcal{F}$ be a finite hypothesis space, and the loss function be bounded within $[a,b]$. If we utilize finite samples to minimize the empirical imitation loss $\hat{L}(f) = \frac{1}{m}\sum_i^m \mathcal{L}(f(x_i), y_i)$ instead of the expected one $L(f) = \mathbb{E}_{(x,y)}\mathcal{L}(f(x),y)$, with probability at least $1-\delta$, the imitation error can be bounded as:*

$$L(f) \le \hat{L}(f) + \sqrt{\frac{(b-a)^2(\log 2|\mathcal{F}| + \log\frac{1}{\delta})}{2m}}.$$

Typically, the true expert policy π_E is inaccessible, so we resort to imitating a surrogate policy $\hat{\pi}_E$. In this offline scenario, we rely on finite samples to estimate $D_{\mathrm{TV}}^{\mathcal{D}}(\hat{\pi}_E, \pi)$ and the imitation error can be bounded using Lemma 4.2. Note that the D_{TV} can be bounded in $[0,1]$. Hence, we can draw the conclusion that with a probability of at least $1-\delta$, the following inequality holds:

$$\begin{aligned} D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\pi) &\le D_{\mathrm{TV}}^{\mathcal{D}}(\hat{\pi}_E,\pi) + D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\hat{\pi}_E) \\ &\le \hat{D}_{\mathrm{TV}}^{\mathcal{D}}(\hat{\pi}_E,\pi) + \sqrt{\frac{\log 2|\Pi| + \log\frac{1}{\delta}}{2m}} + D_{\mathrm{TV}}^{\mathcal{D}}(\pi_E,\hat{\pi}_E), \end{aligned} \tag{3}$$

where m is the size of the training dataset and Π denotes a finite hypothesis space.

Combining (1), (2) and (3), we derive the final optimization objective:

$$\begin{aligned} V^{\pi_E} - V^{\pi} \le \frac{\sqrt{2}R_{max}}{(1-\gamma)^2}\Bigg[&\hat{D}_{\mathrm{TV}}^{\mathcal{D}}(\hat{\pi}_E,\pi) + \sqrt{\frac{\log 2|\Pi| + \log\frac{1}{\delta}}{2m}} \\ &+ D_{\mathrm{TV}}^{\mathcal{D}}(\hat{\pi}_E,\pi_E) + \sum_{(s,g)} |P^{\mathcal{T}}(s,g) - P^{\mathcal{D}}(s,g)|\Bigg]. \end{aligned} \tag{4}$$

Algorithm 1 SC-RvS for Offline RL

```
//Training
Input: offline dataset $\mathcal{D}$
Initialize the goal generator $\theta_h$,$\theta_{V^h}$, the policy $\theta_l$,$\theta_{V^l}$
for epoch = 1 to N do
    Sample transitions $(s, a, s', r) \sim \mathcal{D}$
    Make goal sampling with probability $P_{\mathrm{DWHER}}$ (Eq. (5))
    Update transitions $(s, a, s', r, g, d)$ in $\mathcal{D}$
    Update the value functions $\theta_{V^h}$ and $\theta_{V^l}$ using TD-loss
    Update the goal generator $\theta_h$ by maximizing Eq. (9)
    Update the policy $\theta_l$ by maximizing Eq. (7) or Eq. (8)
end for
//Evaluation
Input: initial state $s$, $done$=False
while not $done$ do
    Get goal $g$ from the goal generator: $g = argmax_g \pi^h(g|s)$
    Get action $a$ from the policy: $a = argmax_a \pi^l(a|s, g)$
    Exec $a$ and get $(s', r, done)$ from the env
    Set $s = s'$
end while
```

5 METHODOLOGY: SPATIAL COMPOSITION RVS

Motivated by Eq. (4) , we present the Spatial Composition RvS (SC-RvS). Our design choice is guided by theoretical insights, aiming to minimize the upper bound of Eq. (4), thus minimizing the discrepancy with the expert policy for reaching unknown goals during evaluation. SC-RvS can be seen as a generalized framework that unifies existing RvS-G methods and introduces targeted improvements to each component. This framework points out the most significant aspects of RvS-G for its trajectory stitching capabilities, while also taking into account other components that contribute to enhanced trajectory stitching capability.

Firstly, we emphasize that the most crucial aspect of RvS-G, which makes it particularly effective for the challenging Antmaze task requiring trajectory stitching, is the incorporation of goal relabeling.

Goal relabeling. It is important to note that the second term in the Eq. (4) is inversely proportional to the dataset size m. Hence, enlarging the state-goal dataset size through goal relabeling can lead to a more constrained upper bound, thereby minimizing the suboptimality of the policy during evaluation. While the Antmaze dataset contains many suboptimal trajectories, i.e., the state-goal pairs during evaluation are not included in the training set, the goal relabeling mitigates this issue by enabling agents to train on a more diverse set of state-goal pairs. Existing RvS-G methods [3, 10] mostly adopt Hindsight Experience Replay (HER) [2] to sample the future states along the same trajectory. Specifically, given a trajectory $(s_0, a_0, r_0, ..., s_t, a_t, r_t, ..., s_{t+d}, a_{t+d}, r_{t+d}, ...)$, we sample a future state s_{t+d} which is d steps after the current state s_t, and designate it as the goal-state for s_t, denoted as $g_t = s_{t+d}$. Here, d is constrained within the range $1 \le d \le T - t$, where T signifies the horizon of the trajectory. This suggests that any future states along this trajectory following s_t can be relabeled as goals.

However, in the original HER method, goals are selected from states achieved further along the trajectory, utilizing random curriculum heuristics. Due to the lack of a systematic strategy, this method struggled to manage the balance between near-reach and far-reach states, causing undesirable performance fluctuation across various tasks. The rationale behind this phenomenon can be interpreted as follows: training with nearby states improve the stability of the learning process and enhances adaptation to similar scenarios, while incorporating distant states improves sampling efficiency and promotes extrapolation. Nevertheless, excessive focus on either of them can result in sub-optimal learning, as indicated by the results of our experiments. We present a new goal sampling strategy that is different from vanilla HER, namely Distance-Weighted HER (DWHER). DWHER introduces a refined strategy to HER by incorporating the distance between the current state s_t and a future state s_{t+d}, represented as $D(s_t, s_{t+d}) = d$. This strategy places emphasis on closer states, resulting in a cohesive training approach that effectively integrates both nearby and distal states. For simplicity, in practice, DWHER is mainly regulated using two hyperparameters, i.e., λ and τ. In specific, λ is used to control the proportion of non-neighbor states in the goal states set, and τ is used to determine the length of horizon for experience replay. Consequently, the calculation of P_{DWHER} is updated as

$$P_{\text{DWHER}}(s_t, s_{t+d}) = \begin{cases} \frac{\min(1[d \leq \tau], \lambda)}{T-t}, & \text{if } d > 1 \\ 1 - \lambda, & \text{otherwise.} \end{cases} \tag{5}$$

Besides of this term, we derive other contributing components of our framework.

Weighted Imitation Learning. For the first term in Eq. (4), where the policy π is optimized to approximate the surrogate policy $\hat{\pi}_E$, we utilize a weighted behavior policy as the surrogate policy: $\hat{\pi}_E(a|s,g) \propto w(s,a,g)\pi_\beta(a|s,g)$. This choice is motivated by our reliance on vanilla RvS-G, which can be regarded as a variant of weighted imitation learning. The optimal policy π_g^E for a specific goal conditioning function $g(s)$ can be expressed using Bayesian rules as:

$$\begin{aligned} \pi_g^E(a|s) = P_\beta(a|s, g(s)) &= \frac{P_\beta(a|s) P_\beta(g(s)|s,a)}{P_\beta(g(s)|s)} \\ &= \pi_\beta(a|s) \frac{P_\beta(g(s)|s,a)}{P_\beta(g(s)|s)}. \end{aligned}$$

Here, π_β represents the behavior policy that generates the dataset. The weight $w(s,a,g)$ denotes the probability density of reaching the conditioned goal $g(s)$. Essentially, the RvS-G policy re-weights the behavior according to the distribution of future goal attainment. Then following [33, 41], we conduct weighted imitation learning on the offline data to minimize $\hat{D}_{\text{TV}}^{\mathcal{D}}(\hat{\pi}_E, \pi)$.

Maximum Entropy-Regularized Weight. The last term involving distribution shift, $\sum_{(s,g)} |P^{\mathcal{T}}(s,g) - P^{\mathcal{D}}(s,g)|$, poses challenges for minimization when there's no prior knowledge of the target goal distribution $P^{\mathcal{T}}$. Taking a Bayesian perspective [31], the agent should learn uniformly from achieved goals. Hence, we can re-weight the offline training dataset $\mathcal{D}$ to a uniform distribution, thereby reducing the upper-bound distribution shift. Achieving this involves multiplying the reciprocal of density, yet estimating density can be challenging. In our approach, we employ uncertainty as a proxy for density, as bootstrapped uncertainty is inversely related to density [47]. Specifically, we adopt tricks from recent work [34] and employ Random Network Distillation (RND) to calculate the uncertainty of the state-goal samples in $\mathcal{D}$ as $u(s,g)$.

RND consists of two neural networks: a fixed and randomly initialized *prior* network $\bar{f}_{\bar{\psi}}$, and a *predictor* network f_ψ which learns to predict the prior outputs on the training data:

$$u(s,g) = \|f_\psi(s,g) - \bar{f}_{\bar{\psi}}(s,g)\|_2^2, \tag{6}$$

where f_ψ learns to align embeddings with data points similar to those in the training dataset, while failing to predict on new data points. Thus, Eq. (6) provides a means to estimate the uncertainty associated with various state-goal pairs in the dataset. We can leverage this uncertainty measure as the maximum-entropy regularized weight to mitigate the disparity between the training and testing datasets, thereby enhancing the generalization to unknown goals.

Generally, RvS methods do not incorporate advantages obtained through TD learning. However, to completely reach the upper bound of the optimization objective, we can integrate advantage-weighted regression. This component is optional if we wish to strictly adhere to the principles of RvS.

Advantage Weight. Minimizing the third term $D_{\text{TV}}^{\mathcal{D}}(\hat{\pi}_E, \pi_E)$ can be reinterpreted as maximizing the expected value of the surrogate policy $\hat{\pi}_E$, since we know π_E has the highest expected value. Following [35, 41], advantage re-weighting $\hat{\pi}_E(a|s,g) \propto \exp(A(s,a,g)) \cdot \pi_\beta(a|s,g)$ leads to an improved expected value over π_β. Therefore, we can employ Exponential Advantage Weighting (EAW) to minimize the divergence from $\hat{\pi}_E$ to π_E. Here, the advantage $A(s,a,g)$ is computed using asymmetric loss following recent work [20].

In summary, the overall policy of SC-RvS is expressed as follows:

$$J(\theta_l) = \mathbb{E}_{(s,a,g)\sim\mathcal{D}}\left[u(s,g) \cdot \exp(A(s,a,g)) \cdot \log \pi_{\theta_l}^l(a|s,g)\right]. \tag{7}$$

And if not incorporating advantage weight, the optimization objective becomes

$$J(\theta_l) = \mathbb{E}_{(s,a,g)\sim\mathcal{D}}\left[u(s,g) \cdot \log \pi_{\theta_l}^l(a|s,g)\right]. \tag{8}$$

Here, the notation $\pi_{\theta_l}^l$ is introduced to distinguish the policy optimized to take actions. Besides, in order to make SC-RvS more applicable to non-goal-conditioned environments, we train an automated goal generator $g(s) = \pi_{\theta_h}^h$ separately to alleviate the need for manual selection of the appropriate conditioned goal during evaluation. The goal generator $\pi_{\theta_h}^h$ provides the policy $\pi_{\theta_l}^l$ with a conditioned goal for each evaluation moment.

$$\begin{aligned} J(\theta_h) = \mathbb{E}_{(s_t,g_t,d)\sim\mathcal{D}}\Big[\gamma^d \cdot \exp\Big(\sum_{i=t}^{t+d-1} r_i + \gamma V_h(g_t) - V_h(s_t)\Big)\cdot \\ \log \pi_{\theta_h}^h(g_t|s_t)\Big]. \end{aligned} \tag{9}$$

6 EXPERIMENTS

In this section, we design the experiments to answer the following research queries (RQ):
(RQ-1) How does the proposed SC-RvS compare against other existing offline RL algorithms?
(RQ-2) How do the different components of SC-RvS contribute to

Table 1: SC-RvS evaluation results compared with TD-learning methods on the Gym Locomotion domain with average performance and standard deviation over 5 trials with different seeds.

	Ensemble-free			Ensemble-based			
Dataset	TD3+BC	IQL	CQL	SAC-N	EDAC	RORL	SC-RvS
halfcheetah-medium-v2	48.3 ± 0.3	47.4 ± 0.2	46.9 ± 0.4	67.5 ± 1.2	65.9 ± 0.6	66.8 ± 0.7	51.4 ± 0.7
halfcheetah-medium-replay-v2	44.6 ± 0.5	44.2 ± 1.2	45.3 ± 0.3	63.9 ± 0.8	61.3 ± 1.9	61.9 ± 1.5	44.3 ± 1.5
halfcheetah-medium-expert-v2	90.7 ± 4.3	86.7 ± 5.3	95.0 ± 1.4	107.1 ± 2.0	106.3 ± 1.9	107.8 ± 1.1	93.1 ± 0.5
hopper-medium-v2	59.3 ± 4.2	66.2 ± 5.7	61.9 ± 6.4	100.3 ± 0.3	101.6 ± 0.6	104.8 ± 0.1	77.2 ± 5.2
hopper-medium-replay-v2	60.9 ± 18.8	94.7 ± 8.6	86.3 ± 7.3	101.8 ± 0.5	101.0 ± 0.5	102.8 ± 0.5	96.9 ± 7.3
hopper-medium-expert-v2	98.0 ± 9.4	91.5 ± 14.3	96.9 ± 15.1	110.1 ± 0.3	110.7 ± 0.1	112.7 ± 0.2	99.0 ± 3.3
walker2d-medium-v2	83.7 ± 2.1	78.3 ± 8.7	79.5 ± 3.2	87.9 ± 0.2	92.5 ± 0.8	102.4 ± 1.4	86.1 ± 2.3
walker2d-medium-replay-v2	81.8 ± 5.5	73.8 ± 7.1	76.8 ± 10.0	78.7 ± 0.7	87.1 ± 2.4	90.4 ± 0.5	70.2 ± 4.4
walker2d-medium-expert-v2	110.1 ± 0.5	109.6 ± 1.0	109.1 ± 0.2	116.7 ± 0.4	114.7 ± 0.9	121.2 ± 1.5	107.6 ± 4.5
Average	67.5	68.9	73.6	84.4	85.2	85.7	80.8

Table 2: SC-RvS evaluation results compared with TD-learning methods on the Antmaze domain with average performance and standard deviation over 5 trials with different seeds.

	Ensemble-free			Ensemble-based		
Dataset	TD3+BC	IQL	CQL	RORL	MSG	SC-RvS
antmaze-umaze-v2	78.6	87.5	74.0	97.7 ± 1.9	97.8 ± 1.2	92.4 ± 4.8
antmaze-umaze-diverse-v2	71.4	62.2	84.0	90.7 ± 2.9	81.8 ± 3.0	68.4 ± 3.8
antmaze-medium-play-v2	10.6	71.2	61.2	76.3 ± 2.5	89.6 ± 2.2	93.4 ± 3.7
antmaze-medium-diverse-v2	3.0	70.0	53.7	69.3 ± 3.3	88.6 ± 2.6	92.6 ± 6.6
antmaze-large-play-v2	0.2	39.6	15.8	16.3 ± 11.1	72.6 ± 7.0	77.3 ± 5.3
antmaze-large-diverse-v2	0.0	47.5	14.9	41.0 ± 10.7	71.4 ± 12.2	72.1 ± 7.5
Average	27.3	63.0	50.6	65.2	83.6	82.7

the overall performance of the algorithm?
(RQ-3) Can the proposed method exhibit compositionality in space for enhanced extrapolation, especially in sub-optimal datasets that can assess the trajectory stitching capability of the model?

6.1 Environmental Settings

We evaluate and analyze our proposed SC-RvS on the D4RL benchmark [12]. For our experiments, we select two tasks: AntMaze and Gym Locomotion.

AntMaze. This task contains a few optimal trajectories and requires stitching parts of the sub-optimal trajectories to get an optimal policy. The maze configurations consist of three sizes: umaze, medium, and large, and there are two modes to choose from: play and diverse. The "play" mode emphasizes training in stable environments, while the "diverse" mode prioritizes diversity to enhance exploration.

Gym Locomotion. This task involves many high-return trajectories that are near expert. It includes HalfCheetah, Hopper, and Walker datasets, each featuring different dataset settings. Here, we employ the medium, medium-replay, and medium-expert datasets, differentiated by the levels of policy optimality.

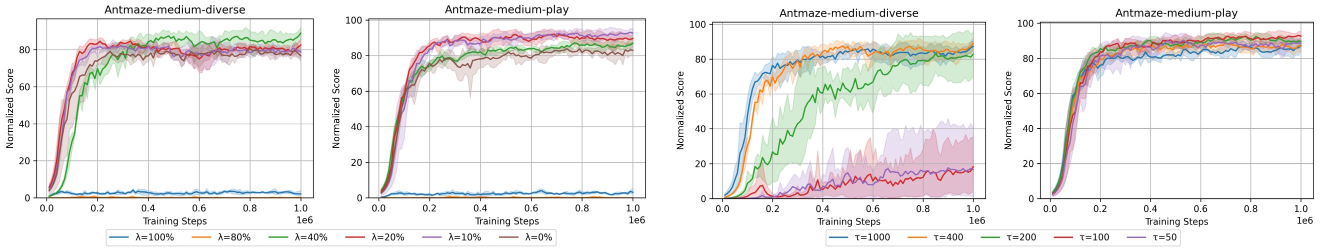
6.2 Baselines

We choose several popular state-of-the-art offline RL methods as baselines, and they can be categorized as follows:

- **TD-learning Approach:** this approach utilizes dynamic programming to realize temporal compositionality and achieve remarkable performance in offline RL. We choose the widely adopted CQL [22] ,TD3+BC [13], and IQL [20], which are ensemble-free and have demonstrated effectiveness on D4RL. In addition, some recent ensemble-based methods have yielded state-of-the-art scores on D4RL, especially on the Gym domain. We choose SAC-N & EDAC [1] and RORL [45] to compare in the Gym locomotion tasks. For Antmaze tasks, we opted for MSG [15] and RORL [45]. SAC-N and EDAC are not included due to the absence of public results in this domain, and their evaluation scores are zero according to our experiment.
- **Reward-conditioned Approach:** we also consider a comparison with Decision Transformer (DT) [7] and vanilla RvS-R [10] to showcase the benefits of goal-conditioning in the context of sub-optimal trajectory stitching capability. Several recent works have proposed solutions to address the limitations of DT in terms of its trajectory stiching capability and we also include them for a comparison here: Q-learning Decision Transformer (QDT) [44], Waypoint Transformer (WT) [3], Agentic Transformer (AT) [27] and Elastic Decision Transformer (EDT) [42].
- **Goal-conditioned Approach:** to substantiate the spatial composition capability of our approach, we additionally compare SC-RvS with other goal-conditioned methods, including vanilla RvS-G [10], POR [43] and WDT [3].
- **Goal-conditioned Reinforcement Learning Approach:** We also compare SC-RvS with several recent goal-conditioned reinforcement learning (GCRL) methods, including GCSL [16], WGCSL [46], GoFar [29], Goal-conditioned IQL (GCIQL), DWSL [17], CODA [11] and GCPC [48]. Strictly speaking, GCRL methods fall outside the scope of offline RL algorithms, as they use different datasets and baselines compared to those used in offline RL. However, to provide a fair and comprehensive comparison, we include results for Antmaze, since these methods do not apply to Gym-Locomotion tasks, which are not goal-conditioned.

Table 3: Results compared against reward-conditioned and goal-conditioned methods for D4RL datasets, showcasing average performance and standard deviation over 5 trials with different seeds. "-" indicates that the algorithm has no publicly available results for that dataset.

Dataset	10% BC	RvS	POR	DT	QDT	WT	AT	EDT	SC-RvS
halfcheetah-medium-v2	42.5	41.6	47.1 ± 0.3	42.6 ± 0.1	42.3 ± 0.4	43.0 ± 0.2	45.12 ± 0.34	42.5 ± 0.9	51.4 ± 0.7
hopper-medium-v2	56.9	60.2	75.4 ± 8.9	67.6 ± 1.0	66.5 ± 6.3	63.1 ± 1.4	70.45 ± 0.45	63.5 ± 5.8	77.2 ± 5.2
walker2d-medium-v2	75.0	71.7	82.8 ± 1.0	74.0 ± 1.4	67.1 ± 3.2	74.8 ± 1.0	88.71 ± 0.55	72.8 ± 6.2	86.1 ± 2.3
halfcheetah-medium-replay-v2	40.6	38.0	43.2 ± 0.2	36.6 ± 0.8	35.6 ± 0.5	39.7 ± 0.3	46.86 ± 0.33	37.8 ± 1.5	44.3 ± 1.5
hopper-medium-replay-v2	75.9	73.5	95.2 ± 10.4	82.7 ± 7.0	52.1 ± 20.1	88.9 ± 2.4	96.85 ± 0.41	89.0 ± 8.3	96.9 ± 7.3
walker2d-medium-replay-v2	62.5	60.6	63.1 ± 4.7	66.6 ± 3.0	58.2 ± 5.1	67.9 ± 3.4	92.32 ± 1.21	74.8 ± 4.9	70.2 ± 4.4
halfcheetah-medium-expert-v2	92.9	92.2	90.7 ± 1.8	86.8 ± 1.3	-	93.2 ± 0.5	95.81 ± 0.25	-	93.1 ± 0.5
hopper-medium-expert-v2	110.9	101.7	96.0 ± 3.0	107.6 ± 1.8	-	110.9 ± 0.6	115.92 ± 1.26	-	99.0 ± 3.3
walker2d-medium-expert-v2	109.0	106.0	101.2 ± 2.9	108.1 ± 0.2	-	109.6 ± 1.0	114.87 ± 0.56	-	107.6 ± 4.5
gym-avg-v2	74.0	71.7	77.2 ± 3.7	74.7 ± 1.8	-	76.8 ± 1.2	85.21	63.4	80.8 ± 3.3
antmaze-umaze-v2	62.8	65.4	86.5 ± 0.9	65.6	-	64.9 ± 6.1	-	-	92.4 ± 4.8
antmaze-umaze-diverse-v2	50.2	60.9	67.3 ± 4.1	51.2	-	71.5 ± 7.6	-	-	68.4 ± 3.8
antmaze-medium-play-v2	5.4	58.1	86.0 ± 3.4	1.0	-	62.8 ± 5.8	-	-	93.4 ± 3.7
antmaze-medium-diverse-v2	9.8	67.3	76.3 ± 2.9	0.6	-	66.7 ± 3.9	-	-	92.6 ± 6.6
antmaze-large-play-v2	0.0	32.4	57.5 ± 2.6	0.0	-	72.5 ± 2.8	-	-	77.3 ± 5.3
antmaze-large-diverse-v2	6.0	36.9	58.4 ± 4.8	0.2	-	72.0 ± 3.4	-	-	72.1 ± 7.5
antmaze-avg-v2	22.5	53.5	72 ± 3.1	19.8	-	68.4 ± 4.9	-	-	82.7 ± 5.3



(a) Effect of λ-induced goal set composition

(b) Impact of experience replay horizon length τ

Figure 2: Ablation Study of the proposed Distance-Weighted HER.

Table 4: Evaluation results of SC-RvS compared with GCRL methods on the Antmaze domain, with average performance and standard deviation calculated over 5 trials using different seeds.

Dataset	GCSL	WGCSL	GoFar	GCIQL	DWSL	CODA	GCPC	SC-RvS
antmaze-umaze-v2	64±2	90.8±2.8	91±1	91.6±4.0	71.2±4.2	94.8±1.3	71.2±1.3	92.4±4.8
antmaze-umaze-diverse-v2	59±1	55.6±15.7	86±3	88.8±2.2	74.6±2.8	72.8±7.7	71.2±6.6	68.4±3.8
antmaze-medium-play-v2	56±6	63.2±13.7	70±1	82.6±5.4	77.6±3.0	75.8±1.9	70.8±3.4	93.4±3.7
antmaze-medium-diverse-v2	60±3	46.0±12.6	63±4	76.2±6.3	74.8±9.3	84.5±5.2	72.2±3.4	92.6±6.6
antmaze-large-play-v2	17±5	0.6±1.3	40±7	40.0±16.2	15.2±7.7	60.0±7.6	78.2±3.2	77.3±5.3
antmaze-large-diverse-v2	12±3	2.4±4.3	45±8	29.8±6.8	19.0±2.8	36.8±6.9	80.6±3.9	72.1±7.5
Average	44.67	43.1	65.83	68	55.4	70.78	74.03	82.7

Note that the RvS score indicates vanilla RvS-G performance in Antmaze while it is based on vanilla RvS-R in Gym. The results of POR are obtained by executing 5 trials with different seeds for the code open-sourced by the author. For the rest of the methods, results are taken from their original papers. Our algorithm is trained through 1M gradient updates, during which we periodically evaluate the learned policy's performance every 5k steps for Gym Locomotion and every 10k steps for AntMaze. Besides, we conduct 50 evaluations for AntMaze and 10 evaluations for Gym Locomotion to derive the average scores.

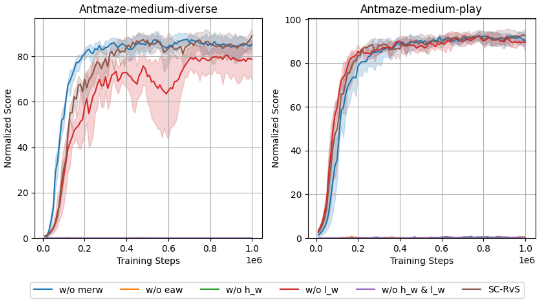


Figure 3: Ablation study for the different learning weights.

6.3 D4RL Results

The results compared to TD-learning approach are shown in Table 1 and Table 2. We observe that for the Gym Locomotion tasks, SC-RvS distinguishes itself among the ensemble-free methods, significantly surpassing them and attaining an average score comparable to that of ensemble-based methods. In the case of AntMaze tasks designed to evaluate stitching capability [12], SC-RvS is on par with the performance of the state-of-the-art ensemble-based method MSG and significantly outperforms all other baselines by a great margin except for umaze-diverse. We conjecture that this may stem from the excessive exploration in this dataset, causing substantial variations in the sampled goal space that hinder the convergence. When compared to other RvS-G and RvS-R variants, from Table 3, we can see that for the Gym Locomotion tasks, SC-RvS demonstrates competitive performance in a majority of the datasets. Overall we obtain the highest average score among all the RvS baselines. Notably, for the more sub-optimal Antmaze task, RvS-G shows remarkable superiority over RvS-R, and our SC-RvS, in turn, exhibits substantial improvement over existing RvS-G methods. And as shown in Table 4, even when compared to the GCRL methods, SC-RvS still achieves the best performance. This responds to (RQ-1), where the proposed SC-RvS performs better on the benchmark dataset in most cases.

6.4 Ablation of the Learning Weights

To answer (RQ-2) and assess the significance of each component of SC-RvS, we conduct ablation studies by systematically removing the different weights from the policy on the two medium datasets of Antmaze. Subsequently, we compare the results of these variants with the original model, as illustrated in Figure 3. We can see that H-EAW emerges as pivotal, as the model's viability significantly diminishes without it. Furthermore, MERW plays a crucial role in enhancing the algorithm's performance. Lastly, L-EAW contributes positively to model's performance, particularly in diverse tasks.

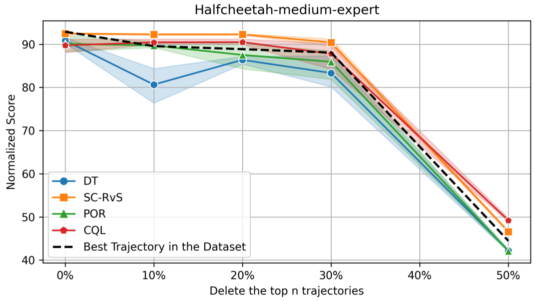


Figure 4: Experimental results regarding the impact of excluding the top n trajectories on different methods.

6.5 Ablation of the Goal Sampling Strategy

We conduct the ablation study on the two medium datasets of Antmaze. Figure 2 illustrates the effects of using different hyperparameters λ and τ. λ limits the proportion of far-reach states in the goal states set. From Figure 2(a), we can observe that the model struggles to undergo proper training when λ is set to 1 and 0.8. This result can also validate the effectiveness of our goal sampling stategy as when λ=1, it is similar to using vanilla HER for goal sampling. It can be explained that conditioning the agent solely on distant and challenging goals can hinder short-term transitions and impede training convergence. Consequently, picking a λ value of 0.4 or 0.1 becomes essential here to strike an optimal balance between goals at varying distances. Additionally, the performance becomes worse once λ is reduced to 0. This regression can be attributed to the overemphasis on near-reach states, which, in turn, curtails the model's extrapolation ability. The horizon length parameter τ also plays a crucial role in goal sampling. We find in Figure 2(b) that in environments with high diversity encouraging exploration (e.g., diverse mode), a larger τ value is needed. But for environments with fixed behavioral patterns (e.g., play mode), a smaller horizon of 100 proves sufficient during replay. The selection of λ and τ also holds relevance. Generally, a smaller λ is essential for larger τ, while larger λ values are more appropriate for shorter replay horizons.

6.6 Sub-optimal Trajectory Stitching Capability

While the performance on Antmaze effectively showcases our model's capability to stitch sub-optimal trajectories, we also conduct an additional experiment to provide further insight into this capability to address (RQ-3). Specifically, we select the Halfcheetah-medium-expert dataset here and progressively delete the top n trajectories with the highest returns to create sub-optimal datasets. As shown in Figure 4, both SC-RvS and CQL outperform the best trajectory in datasets across varying levels of optimality. Notably, when the top 30% of trajectories are gradually deleted, SC-RvS exhibited consistent performance, whereas POR and DT experience reduces significantly. This underscores our SC-RvS capability to stitch sub-optimal trajectories, which is comparable to the TD-learning approach CQL, albeit through different methodologies.

7 CONCLUSION

In this paper, we propose a novel Spatial Composition RvS (SC-RvS) that is designed to enhance the capability of sub-optimal trajectory stitching using the RvS-G framework. Through the introduction of Spatial RvS, we address the intricate balance between trajectory stitching and reducing extrapolation errors, showcasing promising results in experimental evaluations. This work contributes to advancing the understanding and effectiveness of RvS-G, particularly in scenarios where efficient and effective trajectory stitching is paramount. The proposed Spatial RvS algorithm holds promise for real-world applications, offering a valuable tool for policy learning from pre-collected data. In the future, we will further explore the impact of goal-conditioned methods in offline RL settings, integrating them with sequence modeling for pretraining, and seeking inspiration from approaches like inverse RL to further improve goal sampling.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. 2021. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems* 34 (2021), 7436–7447.

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. 2017. Hindsight experience replay. *Advances in neural information processing systems* 30 (2017).

[3] Anirudhan Badrinath, Yannis Flet-Berliac, Allen Nie, and Emma Brunskill. 2023. Waypoint Transformer: Reinforcement Learning via Supervised Learning with Intermediate Targets. *arXiv preprint arXiv:2306.14069* (2023).

[4] David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. 2022. When does return-conditioned supervised learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems* 35 (2022), 1542–1553.

[5] David Brandfonbrener, Will Whitney, Rajesh Ranganath, and Joan Bruna. 2021. Offline rl without off-policy evaluation. *Advances in neural information processing systems* 34 (2021), 4933–4946.

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

[7] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. In *Thirty-Fifth Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=a7APmM4B9d

[8] Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. 2020. Bail: Best-action imitation learning for batch deep reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 18353–18363.

[9] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. 2019. Goal-Conditioned Imitation Learning. In *Advances in Neural Information Processing Systems*. 15324–15335.

[10] Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. 2021. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751* (2021).

[11] Ying Fan, Jingling Li, Adith Swaminathan, Aditya Modi, and Ching-An Cheng. 2024. How to Solve Contextual Goal-Oriented Problems with Offline Datasets? *arXiv preprint arXiv:2408.07753* (2024).

[12] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219* (2020).

[13] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.

[14] Hiroki Furuta, Yutaka Matsuo, and Shixiang Shane Gu. 2021. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364* (2021).

[15] Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. 2022. Why So Pessimistic? Estimating Uncertainties for Offline RL through Ensembles, and Why Their Independence Matters. *arXiv preprint arXiv:2205.13703* (2022).

[16] Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach, and Sergey Levine. 2021. Learning to Reach Goals via Iterated Supervised Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rALA0Xo6yNJ

[17] Joey Hejna, Jensen Gao, and Dorsa Sadigh. 2023. Distance weighted supervised learning for offline interaction data. In *International Conference on Machine Learning*. PMLR, 12882–12906.

[18] Michael Janner, Qiyang Li, and Sergey Levine. 2021. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems* 34 (2021), 1273–1286.

[19] Sham Kakade and John Langford. 2002. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*. 267–274.

[20] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169* (2021).

[21] Aviral Kumar, Xue Bin Peng, and Sergey Levine. 2019. Reward-Conditioned Policies. *arXiv preprint arXiv:1912.13465* (2019).

[22] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).

[23] Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. 2022. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215* (2022).

[24] Jonathan N Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. 2023. Supervised Pretraining Can Learn In-Context Reinforcement Learning. *arXiv preprint arXiv:2306.14892* (2023).

[25] Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. 2022. Multi-game decision transformers. *Advances in Neural Information Processing Systems* 35 (2022), 27921–27936.

[26] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[27] Hao Liu and Pieter Abbeel. 2023. Emergent agentic transformer from chain of hindsight experience. *arXiv preprint arXiv:2305.16554* (2023).

[28] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. 2020. Learning Latent Plans from Play. In *Conference on Robot Learning*. PMLR, 1113–1132.

[29] Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. 2022. How Far I'll Go: Offline Goal-Conditioned Reinforcement Learning via f-Advantage Regression. *arXiv preprint arXiv:2206.03023* (2022).

[30] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of machine learning*. MIT press.

[31] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.

[32] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. 2018. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems* 31 (2018).

[33] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. 2020. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359* (2020).

[34] Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. 2023. Anti-exploration by random network distillation. *arXiv preprint arXiv:2301.13616* (2023).

[35] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177* (2019).

[36] Sebastien Racaniere, Andrew K Lampinen, Adam Santoro, David P Reichert, Vlad Firoiu, and Timothy P Lillicrap. 2019. Automated curricula through setter-solver interactions. *arXiv preprint arXiv:1909.12892* (2019).

[37] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. 2022. A generalist agent. *arXiv preprint arXiv:2205.06175* (2022).

[38] Zhizhou Ren, Kefan Dong, Yuan Zhou, Qiang Liu, and Jian Peng. 2019. Exploration via hindsight goal generation. *Advances in Neural Information Processing Systems* 32 (2019).

[39] Rupesh Kumar Srivastava, Pranav Shyam, Filipe Mutz, Wojciech Jaśkowski, and Jürgen Schmidhuber. 2019. Training Agents Using Upside-Down Reinforcement Learning. *arXiv preprint arXiv:1912.02877* (2019).

[40] Hao Sun, Zhizhong Li, Xiaotong Liu, Bolei Zhou, and Dahua Lin. 2019. Policy continuation with hindsight inverse dynamics. *Advances in Neural Information Processing Systems* 32 (2019).

[41] Qing Wang, Jiechao Xiong, Lei Han, Han Liu, Tong Zhang, et al. 2018. Exponentially weighted imitation learning for batched historical data. *Advances in Neural Information Processing Systems* 31 (2018).

[42] Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. 2023. Elastic decision transformer. *arXiv preprint arXiv:2307.02484* (2023).

[43] Haoran Xu, Li Jiang, Li Jianxiong, and Xianyuan Zhan. 2022. A policy-guided imitation approach for offline reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 4085–4098.

[44] Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. 2023. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*. PMLR, 38989–39007.

[45] Rui Yang, Chenjia Bai, Xiaoteng Ma, Zhaoran Wang, Chongjie Zhang, and Lei Han. 2022. RORL: Robust Offline Reinforcement Learning via Conservative Smoothing. *arXiv preprint arXiv:2206.02829* (2022).

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

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

[48] Zilai Zeng, Ce Zhang, Shijie Wang, and Chen Sun. 2024. Goal-conditioned predictive coding for offline reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).