

Are Expressive Models Truly Necessary for Offline RL?

Guan Wang^{*1}, Haoyi Niu^{*1}, Jianxiong Li¹, Li Jiang², Jianming Hu^{1†}, Xianyuan Zhan^{1,3,4†}

¹Tsinghua University

²McGill University

³Shanghai AI Laboratory

⁴Beijing Academy of Artificial Intelligence

imonenext@gmail.com, {nhy22@mails, hujm@mail, zhanxianyuan@air}.tsinghua.edu.cn

Abstract

Among various branches of offline reinforcement learning (RL) methods, goal-conditioned supervised learning (GCSL) has gained increasing popularity as it formulates the offline RL problem as a sequential modeling task, therefore bypassing the notoriously difficult credit assignment challenge of value learning in conventional RL paradigm. Sequential modeling, however, requires capturing accurate dynamics across long horizons in trajectory data to ensure reasonable policy performance. To meet this requirement, leveraging large, expressive models has become a popular choice in recent literature, which, however, comes at the cost of significantly increased computation and inference latency. Contradictory yet promising, we reveal that lightweight models as simple as shallow 2-layer MLPs, can also enjoy accurate dynamics consistency and significantly reduced sequential modeling errors against large expressive models by adopting a simple recursive planning scheme: recursively planning coarse-grained future sub-goals based on current and target information, and then executes the action with a goal-conditioned policy learned from data related with these sub-goal ground truths. We term our method as **Recursive Skip-Step Planning (RSP)**. Simple yet effective, RSP enjoys great efficiency improvements thanks to its lightweight structure, and substantially outperforms existing methods, reaching new SOTA performances on the D4RL benchmark, especially in multi-stage long-horizon tasks.

Code — https://github.com/imoneoi/RSP_JAX

Introduction

Offline reinforcement learning (RL) has emerged as a promising approach to solving many real-world tasks using logged experiences without extra costly or unsafe interactions with environments (Fujimoto, Meger, and Precup 2019; Levine et al. 2020; Zhan et al. 2022). Unfortunately, the absence of online interaction also poses the counterfactual reasoning challenge in out-of-distribution (OOD) regions, causing catastrophic failures in the conventional RL paradigm due to extrapolation error accumulation during value function learning using temporal difference (TD) updates (Fujimoto, Meger, and Precup 2019; Kumar et al. 2019; Li et al. 2023).

^{*}These authors contributed equally.

[†]Corresponding authors.

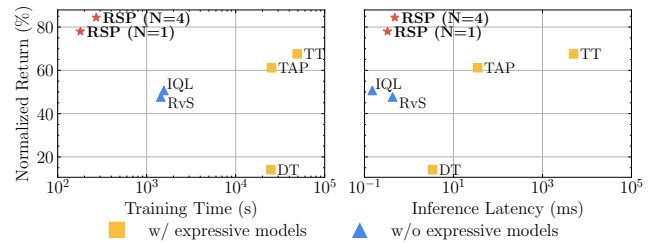


Figure 1: **Existing GCSL methods trade off score performances for slow training and inference.** However, RSP with recursion depth $N = 1$ significantly outperforms all baselines on Antmaze datasets while completing training in just 180 seconds and enjoying an inference latency of 0.33ms. $N = 4$ further enhances performance with minimal cost, as the inference time increases to 0.48 ms and the training time to 271s, remaining orders of magnitude lower than other methods that use expressive models.

To remedy this, Goal-Conditioned Supervised Learning (GCSL) has gained much attention as an alternative paradigm since it reformulates offline RL problems as sequential modeling tasks, where policy extraction can be learned in a supervised manner, thus bypassing problematic value update in conventional TD learning. However, this comes with the price of much higher requirements of accurately describing the data dynamics in sequential modeling, which naturally favors more expressive models. Recent advances in highly expressive models have yielded remarkable achievements across a diverse range of domains such as computer vision (Liu et al. 2021) and natural language processing (Vaswani et al. 2017), which have also been extended to offline RL to accurately capture policy distributions (Wang, Hunt, and Zhou 2023; Hu et al. 2023a; Hansen-Estruch et al. 2023; Ada, Oztop, and Ugur 2024; Wang et al. 2023; Chen et al. 2023), or reduce the accumulative compounding errors in sequential data modeling (Chen et al. 2021; Janner, Li, and Levine 2021; Janner et al. 2022; Ajay et al. 2022; Villafior et al. 2022; Jiang et al. 2023; Mazouze et al. 2023; Niu et al. 2024). Although they sometimes provide encouraging improvements over previous approaches, they also suffer from noticeable performance deterioration when they fail to accurately represent the behavior policy and/or environment dynamics in long-horizon tasks.

Moreover, the integration of expressive models in offline RL inevitably increases both the computational load and inference latency. This poses a significant challenge for many real-world applications that are latency-sensitive or resource-constrained. Besides, methods that employ such expressive models are notoriously data-hungry to train, making them impractical for scenarios with expensive samples (Zhan et al. 2022; Cheng et al. 2024). In Fig. 1, we compare key metrics of recent offline RL methods, including policy performance, training time, and inference latency. The use of large expressive models in prior offline RL methods such as DT (Chen et al. 2021), TAP (Zhang et al. 2022), TT (Janner, Li, and Levine 2021), and Diffuser (Janner et al. 2022) sometimes results in marginal performance gains. However, this incremental improvement is offset by the exponentially increased computational load and inference latency, particularly when compared to models that do not employ such expressive models. This prompts us to ask the question:

Are expressive models truly necessary for offline RL?

While prior attempts introduce highly expressive models to combat the accumulated compounding error in long-sequence modeling, these methods still operate in a *fine-grained* (step-by-step) manner (Chen et al. 2021; Wang, Hunt, and Zhou 2023; Janner et al. 2022), which is inherently tied to the temporal structure of the environment and is susceptible to rapidly accumulated approximation errors over longer horizons (Levine et al. 2020). In this study, we provide a novel perspective to avoid step-wise accumulated compounding error in sequential modeling while only relying on a lightweight model architecture (as simple as 2-layer MLPs). The core of our method is the Recursive Skip-step Planning (RSP) scheme, which employs an iterative two-phase approach to solve tasks: it recursively plans skip-step future sub-goals conditioned on the current and target information, and then executes a goal-conditioned policy based on these coarse-grained predictions. During the recursive planning phase, RSP bypasses step-wise sub-goal prediction and focuses more on longer-horizon outcomes through recursively skipping steps. In essence, RSP can generate long-horizon sub-goals with just a few planning steps, which uses exponentially fewer steps than what is required by previous fine-grained sequence modeling methods, therefore smartly bypassing the long-horizon compounding error issue while enjoying great computational efficiency.

RSP can be easily implemented using simple two-layer shallow MLPs for recursive skip-step dynamics models and the goal-conditioned policy. The entire learning process can be conducted in a supervised learning fashion, eliminating the need for complex stabilization tricks or delicate hyperparameter tuning in value learning. Notably, RSP not only exhibits great training efficiency but also enjoys very low inference complexity, while it achieves comparable or even superior performance against existing offline RL algorithms on D4RL benchmark (Fu et al. 2020), including those equipped with expressive models. This advantage is particularly evident in complex tasks such as *AntMaze* and *Kitchen*, demonstrating the effectiveness of the long-horizon modeling capability by adopting our recursive skip-step planning scheme.

Related Work

Model-free Offline RL. Most prior methods incorporate pessimism during training to alleviate the distributional shift problem (Levine et al. 2020). One solution is leveraging various behavior regularizations to constrain the learned policies close to the behavior policy in offline dataset (Fujimoto, Meger, and Precup 2019; Wu, Tucker, and Nachum 2019; Kumar et al. 2019; Fujimoto and Gu 2021; Li et al. 2023). Some other works introduce pessimism during policy evaluation by assigning low confidences or low values for the value functions in out-of-distribution (OOD) regions (Kumar et al. 2020; Kostrikov et al. 2021; Lyu et al. 2022; An et al. 2021; Bai et al. 2021; Niu et al. 2022). In-sample learning methods (Kostrikov, Nair, and Levine 2022; Xu et al. 2023, 2022; Garg et al. 2023; Xiao et al. 2023; Wang et al. 2024; Zheng et al. 2024) have recently emerged as an alternative, optimizing on samples exclusively from the offline dataset and thus eliminating any OOD value queries. Moreover, an alternative in-sample learning approach performs RL tasks via supervised learning by conditioning other available sources, also called Reinforcement Learning via Supervised Learning (RvS) (Kumar, Peng, and Levine 2019; Schmidhuber 2019; Emmons et al. 2022; Feng et al. 2022). These approaches are super stable and easy to tune compared to other methods.

Existing methods, however, rely on shallow MLPs to model their actors, performing well on simple tasks but falling short on more complex long-horizon planning tasks, such as *AntMaze* and *Kitchen* tasks. To combat this, one popular direction is to increase policy capacity by employing highly expressive models to accurately approximate the actor, which obtains certain degrees of performance gains (Wang, Hunt, and Zhou 2023; Hansen-Estruch et al. 2023; Ada, Oztop, and Ugur 2024; Chen et al. 2023; Lu et al. 2023). However, this marginal improvement comes at the cost of extra model complexity and exponentially increased computational burden, inevitably limiting their applications.

Sequential Modeling for Offline RL. Different from traditional TD methods, this avenue treats offline RL problems as general sequence modeling tasks, with the motivation to generate a sequence of actions or trajectories that attain high rewards. In this formulation, shallow feedforward models are typically believed to suffer from a limited modeling horizon due to accumulated modeling errors caused by deficient model capacities (Janner et al. 2019; Amos et al. 2021; Zhan, Zhu, and Xu 2022). To combat this, recent innovations in sequential modeling techniques shift towards achieving precise long-horizon modeling through the use of high-capacity sequence model architectures such as Transformer (Chen et al. 2021; Janner, Li, and Levine 2021; Jiang et al. 2023; Wang et al. 2022; Konan, Seraj, and Gombolay 2023; Hu et al. 2023b; Wu, Wang, and Hamaya 2024; Villafior et al. 2022) and diffusion models (Janner et al. 2022; Ajay et al. 2022; Niu et al. 2024). However, Fig. 1 manifests that the adoption of expressive models imposes a large amount of computational load and complexity. In this study, RSP adopts a recursive coarse-grained paradigm for sub-goal prediction solely with shallow 2-layer MLPs, achieving exceptional results while bypassing training inefficiency and high inference latency issues led by expressive models.

Preliminaries

In this paper, we consider the standard Markov Decision Process (MDP), denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, \rho, r, \gamma)$, which is characterized by states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition probabilities $T(s_{t+1}|a_t, s_t)$, initial state distribution $\rho(s_0)$, reward function r and the discount factor $\gamma \in (0, 1)$. In each episode, the agent is given a reward function $r(s_t, a_t)$ and embodies with a policy $\pi(a | s)$ to interact with the environment. Let $\tau := (s_0, a_0, s_1, a_1, \dots)$ denote an infinite length trajectory. We employ $\pi(\tau)$ to represent the probability of sampling trajectory τ from the policy $\pi(a | s)$. The primary goal is to optimize the expected cumulative discounted rewards incurred along a trajectory, where the objective and the corresponding Q-functions are expressed as:

$$\begin{aligned} \max_{\pi} \mathbb{E}_{\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \\ Q^{\pi}(s, a) := \mathbb{E}_{\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| \begin{matrix} s_0 = s \\ a_0 = a \end{matrix} \right]. \end{aligned} \quad (1)$$

We are interested in the offline setting in this work and the objective of offline RL is in line with traditional RL, but with infeasible additional online interactions. In the offline setting, a fixed dataset is given by $\mathcal{D} := \{\tau_1, \tau_2, \tau_3, \dots\}$, which comprises multiple trajectories gathered from unknown behavior policies, denoted with $\beta(a|s)$ in this work.

RL via Goal-Conditioned Supervised Learning

Algorithm 1: General Procedure of GCSL

- | | |
|--|----------|
| 1: Input: Offline dataset \mathcal{D} | |
| 2: $\text{RELABEL}(\tau), \forall \tau \sim \mathcal{D}$ | ▷ Step 1 |
| 3: Sample transitions $(s, a, e) \in \mathcal{D}$ | |
| 4: Update $\pi(a s, e)$ with Eqn. (2) | ▷ Step 2 |
| 5: return $\pi(a s, e)$ | |

Reformulating RL as GCSL, previous works (Schmidhuber 2019; Emmons et al. 2022) typically employ hindsight relabeling to perform conditional imitation learning. The core idea underlying this approach is that any trajectory can be considered a successful demonstration of reaching certain goals, i.e. future states or accumulating rewards within the same trajectory. GCSL demonstrates significant potential in the offline setting, largely due to the availability of that goal information (Kumar, Peng, and Levine 2019; Chen et al. 2021; Emmons et al. 2022; Feng et al. 2022).

The general procedure of GCSL is simple and straightforward, as shown in 1, consisting of two main steps. In the first step, GCSL relabels the dataset by adding an additional goal information e to each state-action pair and forms a state-action-goal tuple, i.e., (s, a, e) . The goal can be the target state, cumulative return, or language description (Lynch et al. 2020; Ghosh et al. 2021; Kumar, Peng, and Levine 2019; Chen et al. 2021; Feng et al. 2022), if applicable. For the state s_t , the goal e_t is randomly sampled from the future steps from the current step t along a valid trajectory in the dataset. The key idea behind this resampling is to consider

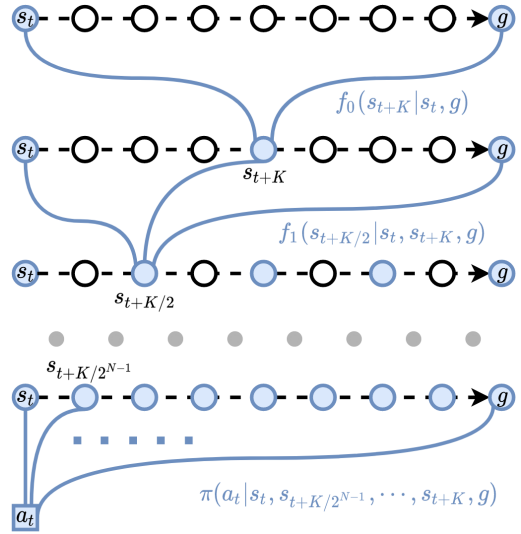


Figure 2: The illustration of Recursive Skip-Step Planning

any future goal e_t is reachable by taking action a_t and state s_t . In the second step, GCSL learns a goal-conditioned policy within the relabeled dataset via maximum log-likelihood:

$$\arg \max_{\pi} \mathbb{E}_{(s,a,e) \sim \mathcal{D}} [\log \pi_{\phi}(a | s, e)]. \quad (2)$$

Recursive Skip-Step Planning

Why Non-recursive Planning Might Struggle?

Fine-grained planning using single-step dynamics $f(s_{t+1}|s_t, g)$ often faces significant challenges due to the rapid accumulation of step-wise errors in sequential modeling, led by such a limited planning horizon. Although expressive models like Transformers (Chen et al. 2021; Janner, Li, and Levine 2021) and Diffusers (Janner et al. 2022; Ajay et al. 2022) aim to mitigate these issues by modeling long-term dependencies, their complex architectures significantly increase computational load. This approach ultimately struggles to balance prediction accuracy with efficiency, remaining constrained within the limitations of fine-grained sequential modeling.

To overcome this, coarse-grained planning uses skip-step dynamics, predicting intermediate sub-goals via $f(s_{t+k}|s_t, g)$ with an extended planning horizon k . However, setting an appropriate skip-step horizon k is non-trivial: too short a horizon fails to alleviate error accumulation as in fine-grained planning, while too long a horizon weakens the learning signal, compromising policy performance in long sequential tasks. Thus, a delicate balance must be achieved between the prediction stability of the sub-goal s_{t+k} and the control precision of the policy $\pi(a_t|s_t, s_{t+k}, g)$. Non-recursive planning may struggle to select sub-goal information that provides both adequate long-term guidance from the final goal and sufficient supervision for current action.

Coarse-grained Planning in a Recursive Way

The dilemma for non-recursive planning suggests using a higher-level dynamics model to predict distant sub-goals

$f'(s_{t+k'}|s_t, g)$ and conditioning intermediate predictions on the current-level dynamics $f(s_{t+k}|s_t, s_{t+k'}, g)$. This hierarchical approach stabilizes sub-goal predictions \hat{s}_{t+k} in the same way that sub-goals enhance action extraction \hat{a}_t . By recursively introducing next-level sub-goal prediction with more distant horizons, sub-goal approximation error is reduced at each level. As these sub-goals approach the final target, their error also inherently diminishes. Planning in such a recursive manner thus effectively mitigates error accumulation in long-horizon tasks.

In the RSP framework, each current state s_t in the dataset is first relabeled with a fixed-horizon future target state s_{t+k} as the sub-goal, resulting in an augmented dataset $\mathcal{D} = \{(s_t, a_t, s_{t+k}, g)\}$. This allows us to revise the optimization objective of goal-conditioned policies in Equation 2 to focus on goal-conditioned sub-goal prediction:

$$\hat{f} = \arg \max_f \mathbb{E}_{(s_t, s_{t+k}, g) \sim \mathcal{D}} [\log f(s_{t+k}|s_t, g)]. \quad (3)$$

As explained in the previous section, planning with a learned coarse-grained dynamics model \hat{f} is insufficient for balancing the minimization of action and sub-goal prediction errors. To further reduce the sub-goal prediction error, we extend it to a recursive skip-step prediction strategy, as depicted in Fig. 2, predicting high-level sub-goals to stabilize the lower-level sub-goal predictions in sequential modeling. To standardize the recursive process, we set $K^{(n)} := t + K/2^n, n \in [0, \dots, N-1]$ where N is the recursion depth, K is the skip step of the highest-level sub-goal, and $s_{K^{(n)}}$ are sub-goal ground truths. The highest-level sub-goal skips K steps while the lowest-level sub-goal skips $k = K/2^{N-1}$ steps from the current step. For N -level recursion, we need to relabel the dataset $\mathcal{D} = \{(s_t, a_t, g)\}$ with additional N skip-step sub-goal ground truths:

$$\mathcal{D} \leftarrow \left\{ \left(a_t, \kappa_t^{(N)} := (s_t, s_{K^{(N-1)}}, \dots, s_{K^{(0)}}), g \right) \right\} \quad (4)$$

where $\kappa_t^{(0)} = (s_t, g); \forall n \in [1, \dots, N], \kappa_t^{(n)} = s_{K^{(n-1)}} \cup \kappa_t^{(n-1)}$ contains all the information required by the n -th level dynamics model to fit the sub-goal ground truth $s_{K^{(n-1)}}$. For brevity, we omit t and refer to $\kappa^{(n)}$ in the following descriptions. With all the definitions above, the highest(first)-level coarse-grained dynamics model could be denoted by $\hat{f}_0(s_{K^{(0)}}|s_t, g)$. Subsequently, we can learn more coarse-grained dynamics models to push the limits of the benefits of coarse-grained planning, as depicted in Fig. 2. In terms of the n -th dynamics model, the recursion solution naturally extends to the optimization objective in Eq. 5:

$$\max_{f_n} \mathbb{E}_{\kappa^{(n)} \sim \mathcal{D}} [\log f_n(s_{K^{(n-1)}}|\kappa^{(n-1)})], \forall n \in [1, \dots, N]. \quad (5)$$

where the current level sub-goal generation is conditioned on all the sub-goals predicted with higher-level dynamics for stability, which proves a significant design in our experiments.

During the evaluation process, we can formulate the recursive sub-goal planning as:

$$s_{K^{(n-1)}} = \arg \max_{s_{K^{(n-1)}}} \hat{f}_n(s_{K^{(n-1)}}|\widehat{\kappa^{(n-1)}}), \forall n \in [1, \dots, N] \quad (6)$$

Algorithm 2: Recursive Skip-Step Planning (RSP)

```

1: Input: Dataset  $\mathcal{D}$ , Recursion depth  $N$ , Fixed planning step  $K$ 
2: Relabel Dataset with Eq. 4 ▷ Step 1
3: // Training ▷ Step 2
4: for every training step do
5:   Sample transitions  $(s_t, a_t, s_{K^{(0)}}, \dots, s_{K^{(N-1)}}, g) \in \mathcal{D}$ 
6:   for  $n = 1, \dots, N$  do
7:     Update the  $n$ -th dynamics model  $\hat{f}_n$  by Eq. 5
8:   end for
9:   Update  $\pi_\phi$  by Eq. 7
10: end for
11: // Evaluation ▷ Step 3
12: Get initial state  $s$ , set done as False
13: while not done do
14:   Get action  $a$  from Eq. 8
15:   Roll out  $a$  and get  $(s', r, \text{done})$ 
16:   Set  $s = s'$ 
17: end while

```

where $\widehat{\kappa^{(0)}} = \kappa^{(0)} = (s_t, g); \forall n \in [1, \dots, N], \widehat{\kappa^{(n)}} = s_{K^{(n-1)}} \cup \widehat{\kappa^{(n-1)}}$. Eventually, we get the predicted sub-goals and leverage $\widehat{\kappa^{(N)}}$ for policy extraction.

Policy Extraction

Different from other planning methods, the coarse-grained planning approach employed by RSP does not explicitly integrate the corresponding policy into the learning process, necessitating a separate policy extraction step. To maintain the simplicity and efficiency of RSP, we extract the policy by maximizing the log-likelihood within a supervised learning framework. We introduce a goal-conditioned policy, distinct in that it incorporates all the sub-goals planned from prior coarse-grained dynamics models:

$$\max_{\phi} \mathbb{E}_{(a_t, \kappa^{(N)}) \sim \mathcal{D}} [\log \pi_\phi(a_t | \kappa^{(N)})]. \quad (7)$$

At the evaluation stage, given the current state s_t , the action is determined by both the coarse-grained dynamics model and the goal-conditioned policy with:

$$\hat{a}_t = \arg \max_{a_t} \hat{\pi}_\phi(a_t|\widehat{\kappa^{(N)}}) \quad (8)$$

where $\widehat{\kappa^{(N)}} = (s_t, s_{K^{(N-1)}}, \dots, s_{K^{(0)}}), g$ is obtained from Eq. 6 recursively.

The final procedure in Algorithm 2 consists of three steps. Initially, it relabels the dataset \mathcal{D} , augmenting each transition with the future skip-step sub-goal and final goal ground truths. Subsequently, RSP learns N coarse-grained dynamics models from Eq. 5 and a goal-conditioned policy from Eq. 7 that determines the action based on the sub-goals predicted with all the dynamics models. Finally, RSP plans the sub-goal sequence recursively with Eq. 6 and executes action from sub-goals with Eq. 8.

Discussions

In this section, we provide an in-depth explanation of why recursive skip-step planning might outperform from the perspective of model rollout prediction errors. We conduct experiments on different (lowest-level) skip-step horizons ($k =$

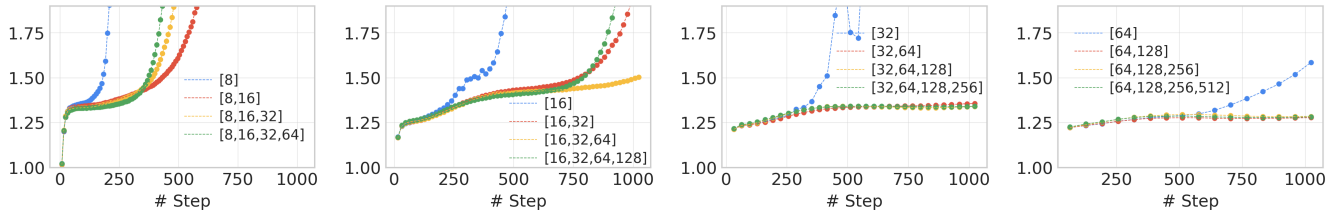


Figure 3: Root mean squared error (RMSE) between rollout sub-goal predictions and ground truths on Antmaze-Ultra.

8, 16, 32, 64) and recursion depths ($N = 1, 2, 3, 4$) in the long-horizon Antmaze-Ultra-Diverse task, where trajectories can extend up to 3000 steps. For example, [16,32,64] in Fig. 3 denotes $N = 3$ and $k = 16$. We plot the root mean squared errors (RMSE) between the model rollout predictions and the ground truths of skip-step state trajectories, averaged over 10 seeds. Specifically, we obtain the next skip-step prediction \hat{s}_{t+k} through the recursive approach of RSP; we iteratively query the next skip-step prediction \hat{s}_{t+nk} from the last prediction $\hat{s}_{t+(n-1)k}$ and finally obtain rollout sub-goal predictions $s_t, \dots, \hat{s}_{t+(n-1)k}, \hat{s}_{t+nk}, \dots, \hat{s}_{t+1024}$; then calculate RMSE between the predictions and ground truths.

In Fig. 3, as k increases from 8 to 64, the accuracy of the model rollout prediction improves, aligning more closely with the ground truth over longer-horizon rollouts. This suggests that coarse-grained prediction outperforms fine-grained prediction in mitigating accumulated compounding errors as the skip-step horizon k increases, thereby enhancing long-horizon planning capabilities. Besides, as the recursion depth N increases, we observe that the RMSE of model rollout predictions achieves a lower asymptotic value and remains closer to the ground truth over longer-horizon rollouts, demonstrating the effectiveness of recursive planning.

Choices of recursion depth and skip-step horizon. At first glance, Fig. 3 indicates that increasing recursion depth and skip-step horizon improves the planning performance of RSP. However, these plots assume that an oracle policy model perfectly follow the skip-step state predictions. In practice, deviations from optimal state trajectories can easily occur due to suboptimal policies, especially when the policy model $\pi(a_t|s_t, s_{t+k}, \dots, g)$ is conditioned on a distant skip-step state prediction s_{t+k} , which offers minimal guidance for current actions. Therefore, selecting recursion depth and horizon requires to strike the delicate balance between the expressiveness of policy models and the long-horizon capability of dynamics models. Given that we use a 2-layer MLP for policy learning for simplicity in experiments, we opt for $k = 32$ and $N = 1$ in recursive skip-step dynamics models across all tasks in order to provide sufficient long-horizon planning ability while easing the burden for policy learning. Practitioners using more advanced policy modeling and learning methods may consider enlarging the horizon and deepening the recursion process accordingly.

Experiments

In this section, we conduct comprehensive evaluations to showcase the effectiveness of RSP. Specifically, we com-

pare RSP against other competitive baselines on the standard offline RL benchmark, D4RL (Fu et al. 2020). The results indicate that despite its simplicity, RSP can obtain on-par or superior performance compared to others using expressive models, especially on the long-horizon Antmaze and multi-stage Kitchen tasks (Section *D4RL Benchmark Results*). Moreover, RSP exhibits fast training and inference time, with training for only around 180 seconds and an inference latency of under 1ms (Section *Training Cost and Inference Latency*). At last, we provide extensive ablation studies on some critical components of RSP including the planning horizon and recursion depth (Section *Ablation Studies*).

D4RL Benchmark Results

Evaluation setups We evaluate the performance of various methods using a diverse set of challenging tasks from D4RL benchmark (Fu et al. 2020), including Antmaze navigation, Kitchen Manipulation, Adroit hand manipulation, and Mujoco locomotion tasks. The Antmaze navigation tasks require intricate long-horizon planning to guide an ant navigate from the starting point to the goal location, which are pretty challenging for traditional offline RL to solve. To test the planning capabilities on extremely long-horizon tasks (> 2000 steps), we consider a more extensive and challenging Ultra Antmaze environment. Kitchen tasks involve a 9-DoF Franka robot interacting with a kitchen scene to complete a combination of household sub-tasks, which requires multi-skill long-horizon composition ability. In contrast, the Adroit hand manipulation tasks involve controlling a dexterous robot hand to complete tasks like opening a door or rotating a pen, with planning horizons typically ranging from 100 to 200 steps, which are relatively simple. Finally, the Mujoco locomotion tasks are believed as the simplest tasks among all these tasks. They only focus on controlling different robots to move forward as fast as possible, which are relatively simple since the control pattern is repetitive. To assess performance, we normalize the final scores by following the benchmark standards (Fu et al. 2020), where 0 represents random policy performance and 100 means expert-level performance. We use 10 random seeds and 100 episodes per seed for every task in our evaluation, with mean and standard deviation reported. All the results are obtained on a server with 512G RAM, $4 \times$ A800 PCIe 80GiB (GPU) and $2 \times$ AMD EPYC 7T83 64-Core Processor (CPU).

Baselines We compare RSP against the following state-of-the-art (SOTA) baselines. *BC* is the simplest imitation

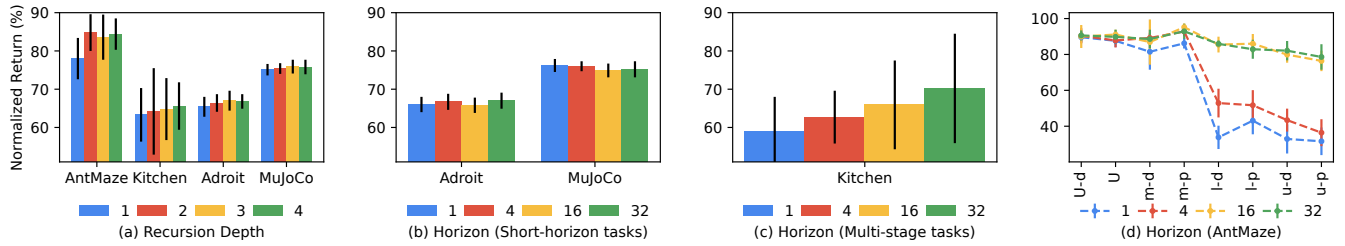


Figure 4: Ablations on choices of recursion depth and horizon.

learning approach that imitates the offline dataset; *RvS* (Emmons et al. 2022) is an SOTA conditioned imitation learning method that casts offline RL as supervised learning problems; *CQL* (Kumar et al. 2020) is an offline RL method that penalizes Q-values for OOD regions; *IQL* (Kostrikov, Nair, and Levine 2022) is an SOTA in-sample learning offline RL method; *DT* (Chen et al. 2021) and *TT* (Janner, Li, and Levine 2021) are two methods that utilize transformer (Vaswani et al. 2017) architecture, regarding offline RL problems as sequential modeling problems; *TAP* (Zhang et al. 2022) is also a transformer-based method that plans in a latent space encoded by VQ-VAE (Van Den Oord, Vinyals et al. 2017); *Diffuser* (Janner et al. 2022) builds on top of diffusion model, directly generating the entire fixed-length trajectory and executing the first step.

Main results We present evaluation results on D4RL benchmark in Table 1¹. The results demonstrate that RSP consistently outperforms or obtains on-par performance compared to other baselines. In particular, RSP achieves unparalleled success in complex *Antmaze* navigation and *Kitchen* manipulation tasks that necessitate long-horizon and/or multi-stage planning. For instance, all baselines fail miserably on the extremely challenging *Antmaze-Ultra* tasks, while RSP can obtain similar success rates as it achieves on the *Antmaze-Large* tasks. This indicates that the increased difficulty resulting from longer planning horizons has a marginal impact for RSP, demonstrating the superior long-horizon planning capabilities of RSP. Apart from the long-horizon tasks, RSP also demonstrates consistently good performance on *Adroit* hand manipulation and *Mujoco* locomotion tasks. On these tasks, Table 1 shows that even with much simpler model architecture, RSP can surprisingly obtain on-par performance results compared to RL as sequential modeling methods that utilize expressive models such as *TT*, *TAP*, and *Diffuser*, as well as TD-based offline RL baselines such as *CQL* and *IQL*.

¹Most baselines didn’t report results of the extremely challenging *Antmaze-Ultra* task so we reproduce BC, DT and RvS and report results of IQL, TT, TAP from TAP paper (Zhang et al. 2022). We collect IQL results on other *Antmaze-v2* tasks from DOGE (Li et al. 2023) since IQL paper reports scores on v0. We also report self-reproduced results on *Adroit-Expert* task due to lack of reported scores. Other results are from the original papers.

Training Cost and Inference Latency

To further showcase the advantages of RSP, we monitor the average training time (1M steps) and inference latency (1 step) of different methods on *Antmaze* tasks and illustrate the connections between training time, inference latency and the performance for different methods in Fig. 1. All measurements were taken with models trained and inferred on a single GPU in isolation, ensuring no interference from other GPU activities on the server. We can clearly observe that by leveraging expressive models for sequential modeling, one can sometimes obtain minor improvements in policy performance againsts TD-based offline RL approaches. However, these marginal gains come at the expense of exponentially increased training time and inference latency. This inevitably imposes significant computational burdens, restricting the applicability of expressive models in many real-world scenarios. In contrast, RSP with horizon $k = 8$ and recursion depth $N = 1, 4$ obtains the best performance and meanwhile enjoys fast training and minimal inference latency. In particular, RSP ($N = 1$) completes the training in approximately 180 seconds and exhibits an inference latency of under 1ms, highlighting its exceptional computational efficiency. This sheds light on its potential real-world applicability, given its impressive performance, simplicity, and efficiency.

Ablation Studies

In this section, we conduct extensive ablation studies on the critical components of RSP including the recursion depth N and the skip-step horizon of the lowest-level sub-goal for policy extraction k (referred to as “horizon” hereafter for simplicity). The results are presented in Fig. 4.

We first fix the horizon $k = 8$ and conduct ablation studies on different recursion depths, i.e. $N = 1, 2, 3, 4$. The results in Fig. 4(a) indicate that a deeper recursion process consistently improves performance. However, performance in some tasks tends to plateau once $N > 2$. This is because increasing recursion depth can introduce cumulative errors from each prediction level even though each gets more accurate, which may counterbalance the benefits of RSP for low-level policy learning and result in no further performance improvement. Then we can conclude the following implementation insights: (1) choose minimal recursion depth that achieves saturated performance: $N = 2$ is sufficient for most benchmark tasks or those with similar configurations. There is no need to risk RSP performance by unnecessarily increasing depth; (2) advance policy learning as mentioned in Section *Discussions*: Strong policy models may exhibit robustness to slight inac-

Dataset	Environment	BC	RvS	CQL	IQL	HIQL	Diffuser	DT	TT	TAP	RSP (Ours)
Umaze	AntMaze	65.0	65.4 (+G)	74.0	88.2	—	—	59.2	100.0 (+Q)	—	88.2 ±5.7 (+G)
Umaze-Diverse	AntMaze	55.0	60.9 (+G)	84.0	66.7	—	—	53.2	—	—	92.9 ±3.5 (+G)
Medium-Play	AntMaze	0.0	58.1 (+G)	61.2	70.4	84.1	—	0.0	93.3 (+Q)	78.0 (+G)	91.4 ±8.8 (+G)
Medium-Diverse	AntMaze	0.0	67.3 (+G)	53.7	74.6	86.8	—	0.0	100.0 (+Q)	85.0 (+G)	92.7 ±3.4 (+G)
Large-Play	AntMaze	0.0	32.4 (+G)	15.8	43.5	86.1	—	0.0	66.7(+Q)	74.0 (+G)	83.0 ±3.7 (+G)
Large-Diverse	AntMaze	0.0	36.9 (+G)	14.9	45.6	88.2	—	0.0	60.0(+Q)	82.0 (+G)	86.0 ±3.8 (+G)
Ultra-Play	AntMaze	0.0	33.2 (+G)	—	8.3	39.2	—	0.0	20.0(+Q)	22.0 (+G)	80.3 ±4.2 (+G)
Ultra-Diverse	AntMaze	0.0	31.6 (+G)	—	15.6	52.9	—	0.0	33.3(+Q)	26.0 (+G)	80.5 ±4.1 (+G)
Average		15.0	48.2 (+G)	50.6	51.6	—	—	14.0	67.6(+Q)	61.2 (+G)	86.9 ±4.7 (+G)
Partial	Kitchen	38.0	51.4 (+G)	20.8	46.3	65.0	—	—	—	—	76.5 ±8.1 (+G)
Mixed	Kitchen	51.5	60.3 (+G)	24.2	51.0	67.7	—	—	—	—	64.0 ±20.4 (+G)
Average		44.8	55.9 (+G)	22.5	48.7	66.4	—	—	—	—	70.2 ±14.3 (+G)
Medium-Expert	HalfCheetah	55.2	92.2	91.6	86.7	—	88.9	86.8	95.0	91.8	92.5 ±0.5
Medium-Expert	Hopper	52.5	101.7	105.4	91.5	—	103.3	107.6	110.0	105.5	109.6 ±0.7
Medium-Expert	Walker2d	107.5	106.0	108.8	109.6	—	106.9	108.1	101.9	107.4	106.7 ±1.0
Medium	HalfCheetah	42.6	41.6	44.0	47.4	—	42.8	42.6	46.9	45.0	42.9 ±0.2
Medium	Hopper	52.9	60.2	58.5	66.3	—	74.3	67.6	61.1	63.4	63.4 ±2.5
Medium	Walker2d	75.3	71.7	72.5	78.3	—	79.6	74.0	79.0	64.9	76.6 ±1.6
Medium-Replay	HalfCheetah	36.6	38.0	45.5	44.2	—	37.7	36.6	41.9	40.8	40.4 ±0.4
Medium-Replay	Hopper	18.1	73.5	95.0	94.7	—	93.6	82.7	91.5	87.3	88.2 ±5.6
Medium-Replay	Walker2d	26.0	60.6	77.2	73.9	—	70.6	66.6	82.6	66.8	66.9 ±2.9
Average		51.9	71.7	77.6	77.0	—	77.5	74.7	78.9	74.8	76.4 ±1.7
Cloned	Pen	56.9	—	39.2	37.3	—	—	—	11.4	57.4	67.9 ±6.9
Cloned	Hammer	0.8	—	2.1	2.1	—	—	—	0.5	1.2	3.6 ±3.1
Cloned	Door	-0.1	—	0.4	1.6	—	—	—	-0.1	11.7	0.8 ±0.6
Cloned	Relocate	-0.1	—	-0.1	-0.2	—	—	—	-0.1	-0.2	0.0 ±0.0
Expert	Pen	85.1	—	107.0	133.1	—	—	—	72.0	127.4	120.9 ±4.6
Expert	Hammer	125.6	—	86.7	119.5	—	—	—	15.5	127.6	129.7 ±0.9
Expert	Door	34.9	—	101.5	105.7	—	—	—	94.1	104.8	105.0 ±0.3
Expert	Relocate	101.3	—	95.0	106.1	—	—	—	10.3	105.8	108.1 ±0.4
Average		50.6	—	54.0	63.1	—	—	—	25.4	67.0	67.0 ±2.1

Table 1: Comparison of performance in AntMaze navigation, Kitchen manipulation, Mujoco locomotion tasks, and Adroit hand manipulation tasks from the D4RL dataset. AntMaze tasks require long-horizon planning to navigate from the starting point to the goal. Adroit tasks involve high-dimensional action space control of a 24-DoF robot hand. Baselines in teal utilize expressive models including Transformer or diffusion models. Emphasis is placed on scores within 5% of the maximum for locomotion tasks following, while maximum scores are highlighted for other tasks. +G denotes goal-conditioned methods. +Q denotes using Q-function as a search heuristic following (Janner, Li, and Levine 2021). We use “—” if the original papers do not report scores on the specific tasks. Note that only a few baseline methods report their performance on the challenging Adroit and Kitchen tasks, so we only compare with baselines having reported scores.

curacies that accumulate in sub-goal predictions, enabling safe extension of recursion depth and fully harnessing the potential of RSP. Other design considerations in policy formulation, such as how to appropriately condition actions on predicted sub-goals, could further enhance performance.

Next, we fix the recursion depth $N = 1$ and ablate on the choices of $k = 1, 4, 16, 32$. A larger horizon corresponds to more coarse-grained planning. We observe that the horizon has minimal impact on short-horizon tasks, such as Adroit and MuJoCo, as shown in Fig. 4(b). However, reducing the horizon significantly leads to performance degradation in long-horizon or multi-stage tasks, such as Antmaze and Kitchen, as seen in Fig. 4(c) and (d).

Conclusion

In this study, we propose a novel recursive skip-step planning framework for offline RL, which leverages a set of coarse-

grained dynamics models to perform recursive sub-goal prediction, and use a goal-conditioned policy to extract planned actions based on these sub-goals. Our method can smartly bypass the long-horizon compounding error issue in existing sequence modeling-based offline RL methods through hierarchical recursive planning, while still maintaining the advantage of learning in a completely supervised manner. Notably, even using lightweight MLP networks, our method can provide comparable or even better performance as compared to sequence modeling methods that use heavy architectures like Transformers or diffusion models, while providing much better training and inference efficiency. Our work highlights the need to rethink the existing design principles for offline RL algorithms: instead of relying on increasingly heavier models, maybe it’s time to introduce more elegant and lightweight modeling schemes to tackle existing challenges in offline RL.

Acknowledgements

This work is supported by National Key Research and Development Program of China under Grant (2022YFB2502904), National Natural Science Foundation of China under Grant No. 62333015, and Beijing Natural Science Foundation L231014. This work is also supported by Beijing Academy of Artificial Intelligence (BAAI).

References

- Ada, S. E.; Oztog, E.; and Ugur, E. 2024. Diffusion policies for out-of-distribution generalization in offline reinforcement learning. *IEEE Robotics and Automation Letters*.
- Ajay, A.; Du, Y.; Gupta, A.; Tenenbaum, J.; Jaakkola, T.; and Agrawal, P. 2022. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*.
- Amos, B.; Stanton, S.; Yarats, D.; and Wilson, A. G. 2021. On the model-based stochastic value gradient for continuous reinforcement learning. In *Learning for Dynamics and Control*, 6–20. PMLR.
- An, G.; Moon, S.; Kim, J.-H.; and Song, H. O. 2021. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in Neural Information Processing Systems*.
- Bai, C.; Wang, L.; Yang, Z.; Deng, Z.-H.; Garg, A.; Liu, P.; and Wang, Z. 2021. Pessimistic Bootstrapping for Uncertainty-Driven Offline Reinforcement Learning. In *International Conference on Learning Representations*.
- Chen, H.; Lu, C.; Ying, C.; Su, H.; and Zhu, J. 2023. Offline reinforcement learning via high-fidelity generative behavior modeling. *International Conference on Learning Representations*.
- 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. *Advances in neural information processing systems*, 34: 15084–15097.
- Cheng, P.; Zhan, X.; Zhang, W.; Lin, Y.; Wang, H.; Jiang, L.; et al. 2024. Look beneath the surface: Exploiting fundamental symmetry for sample-efficient offline rl. *Advances in Neural Information Processing Systems*, 36.
- Emmons, S.; Eysenbach, B.; Kostrikov, I.; and Levine, S. 2022. RvS: What is Essential for Offline RL via Supervised Learning? In *International Conference on Learning Representations*.
- Feng, X.; Jiang, L.; Yu, X.; Xu, H.; Sun, X.; Wang, J.; Zhan, X.; and Chan, W. K. V. 2022. Curriculum Goal-Conditioned Imitation for Offline Reinforcement Learning. *IEEE Transactions on Games*.
- Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *ArXiv preprint*.
- Fujimoto, S.; and Gu, S. S. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34: 20132–20145.
- Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In *International Conference on Machine Learning*, 2052–2062.
- Garg, D.; Hejna, J.; Geist, M.; and Ermon, S. 2023. Extreme Q-Learning: MaxEnt RL without Entropy. In *The Eleventh International Conference on Learning Representations*.
- 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*.
- Hansen-Estruch, P.; Kostrikov, I.; Janner, M.; Kuba, J. G.; and Levine, S. 2023. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*.
- Hu, J.; Sun, Y.; Huang, S.; Guo, S.; Chen, H.; Shen, L.; Sun, L.; Chang, Y.; and Tao, D. 2023a. Instructed Diffuser with Temporal Condition Guidance for Offline Reinforcement Learning. *arXiv preprint arXiv:2306.04875*.
- Hu, S.; Shen, L.; Zhang, Y.; and Tao, D. 2023b. Graph Decision Transformer. *arXiv preprint arXiv:2303.03747*.
- Janner, M.; Du, Y.; Tenenbaum, J.; and Levine, S. 2022. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning*, 9902–9915. PMLR.
- Janner, M.; Fu, J.; Zhang, M.; and Levine, S. 2019. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32.
- Janner, M.; Li, Q.; and Levine, S. 2021. Offline Reinforcement Learning as One Big Sequence Modeling Problem. *Advances in Neural Information Processing Systems*.
- Jiang, Z.; Zhang, T.; Janner, M.; Li, Y.; Rocktäschel, T.; Grefenstette, E.; and Tian, Y. 2023. Efficient Planning in a Compact Latent Action Space. In *The Eleventh International Conference on Learning Representations*.
- Konan, S. G.; Seraj, E.; and Gombolay, M. 2023. Contrastive decision transformers. In *Conference on Robot Learning*, 2159–2169. PMLR.
- Kostrikov, I.; Fergus, R.; Tompson, J.; and Nachum, O. 2021. Offline Reinforcement Learning with Fisher Divergence Critic Regularization. In *International Conference on Machine Learning*, 5774–5783.
- Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.
- Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. In *Advances in Neural Information Processing Systems*, 11761–11771.
- Kumar, A.; Peng, X. B.; and Levine, S. 2019. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*.
- Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*.
- Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv preprint*.

- Li, J.; Zhan, X.; Xu, H.; Zhu, X.; Liu, J.; and Zhang, Y.-Q. 2023. When data geometry meets deep function: Generalizing offline reinforcement learning. In *International Conference on Learning Representations*.
- Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.
- Lu, C.; Chen, H.; Chen, J.; Su, H.; Li, C.; and Zhu, J. 2023. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, 22825–22855. PMLR.
- Lynch, C.; Khansari, M.; Xiao, T.; Kumar, V.; Tompson, J.; Levine, S.; and Sermanet, P. 2020. Learning latent plans from play. In *Conference on robot learning*.
- Lyu, J.; Ma, X.; Li, X.; and Lu, Z. 2022. Mildly conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 1711–1724.
- Mazouze, B.; Talbott, W.; Bautista, M. A.; Hjelm, D.; Toshev, A.; and Susskind, J. 2023. Value function estimation using conditional diffusion models for control. *arXiv preprint arXiv:2306.07290*.
- Niu, H.; Chen, Q.; Liu, T.; Li, J.; Zhou, G.; ZHANG, Y.; HU, J.; and Zhan, X. 2024. xTED: Cross-Domain Adaptation via Diffusion-Based Trajectory Editing. In *NeurIPS 2024 Workshop on Open-World Agents*.
- Niu, H.; Sharma, S.; Qiu, Y.; Li, M.; Zhou, G.; Hu, J.; and Zhan, X. 2022. When to trust your simulator: dynamics-aware hybrid offline-and-online reinforcement learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 36599–36612.
- Schmidhuber, J. 2019. Reinforcement Learning Upside Down: Don't Predict Rewards—Just Map Them to Actions. *arXiv preprint arXiv:1912.02875*.
- Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Villaflor, A. R.; Huang, Z.; Pande, S.; Dolan, J. M.; and Schneider, J. 2022. Addressing optimism bias in sequence modeling for reinforcement learning. In *international conference on machine learning*, 22270–22283. PMLR.
- Wang, H.; Lin, Y.; Han, S.; and Lv, K. 2023. Offline Reinforcement Learning with Diffusion-Based Behavior Cloning Term. In *International Conference on Knowledge Science, Engineering and Management*, 267–278. Springer.
- Wang, K.; Zhao, H.; Luo, X.; Ren, K.; Zhang, W.; and Li, D. 2022. Bootstrapped transformer for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 34748–34761.
- Wang, X.; Xu, H.; Zheng, Y.; and Zhan, X. 2024. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. *Advances in Neural Information Processing Systems*, 36.
- Wang, Z.; Hunt, J. J.; and Zhou, M. 2023. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.
- Wu, Y.; Tucker, G.; and Nachum, O. 2019. Behavior Regularized Offline Reinforcement Learning. *arXiv preprint arXiv:1911.11361*.
- Wu, Y.-H.; Wang, X.; and Hamaya, M. 2024. Elastic decision transformer. *Advances in Neural Information Processing Systems*, 36.
- Xiao, C.; Wang, H.; Pan, Y.; White, A.; and White, M. 2023. The In-Sample Softmax for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.
- Xu, H.; Jiang, L.; Li, J.; Yang, Z.; Wang, Z.; Chan, V. W. K.; and Zhan, X. 2023. Offline RL with No OOD Actions: In-Sample Learning via Implicit Value Regularization. In *The Eleventh International Conference on Learning Representations*.
- Xu, H.; Jiang, L.; Li, J.; and Zhan, X. 2022. A Policy-Guided Imitation Approach for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*.
- Zhan, X.; Xu, H.; Zhang, Y.; Zhu, X.; Yin, H.; and Zheng, Y. 2022. Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4680–4688.
- Zhan, X.; Zhu, X.; and Xu, H. 2022. Model-based offline planning with trajectory pruning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.
- Zhang, T.; Janner, M.; Li, Y.; Rocktäschel, T.; Grefenstette, E.; Tian, Y.; et al. 2022. Efficient Planning in a Compact Latent Action Space. In *International Conference on Learning Representations*.
- Zheng, Y.; Li, J.; Yu, D.; Yang, Y.; Li, S. E.; Zhan, X.; and Liu, J. 2024. Safe Offline Reinforcement Learning with Feasibility-Guided Diffusion Model. In *International Conference on Learning Representations*.