Appendices

A Environments Tested

Following are the environments we evaluated in Sec. 5:

D4RL Maze2d (Fu et al., 2020)**.** The maze2d task is a navigation task that requires a 2D agent to reach a fixed goal location in the maze. This task jusitifies the ability of offline RL algorithms to stitch previously collected subtrajectories to get the shortest path to the goal location. There are three layouts in this task, including umaze, medium and large. The dataset of this environment is generated by selecting waypoints randomly and using a planner which could generate subtrajectories among the waypoints.

D4RL AntMaze (Fu et al., 2020)**.** The Antmaze task is a navigation task that replaces the 2D ball from Maze2D with a 8-Dof Ant quadraped robot. This task combines the challenges of controlling the robot and navigting the robot to the goal location. There are three different layouts in this environment, including umaze, medium, and large. The environment also contains3 three flavors of datasets, including fixed, diverse, and play, wich differs in the chosen of the start and goal locations.

D4RL Locomotion (Fu et al., 2020)**.** The Locomotion environment contains three different types of tasks (walker2d, hopper, and halfcheetah), including 12 different offline data with varying levels of expertise (random, medium, medium-replay, and medium-expert). The medium datasets are generated by a policy trained with a early-stopping SAC (Haarnoja et al., 2018). The random datasets are generated by a random initilized policy. The medium-replay datasets consist of samples in the replay buffer during the training until the policy reaches the medium performance. The medium-expert dataset contains part of the expert demonstrations and part of the suboptimal trajectories.

D4RL Kitchen (Fu et al., 2020)**.** The Kitchen task involves a simulated environment where a 9-DoF robot manipulates various objects, such as sliding a cabinet door, switching an overhead light, and opening a microwave. Initially introduced by (Gupta et al., 2019), this task requires the robot to complete a sequence of multiple subtasks, each rewarded with a sparse, binary reward upon successful completion. The offline dataset provided includes only portions of the complete sequence, necessitating that the agent learn to assemble these sub-trajectories effectively.

Meta-World (Yu et al., 2019). Meta-World is an extensive platform created to assess and enhance algorithms in both reinforcement learning and multi-task learning. With 50 unique robotic manipulation tasks, it provides a varied and demanding setting for evaluating how well algorithms can generalize and rapidly learn new skills.

B Hyperparameters

We list all the hyperparameters here, which are applied to all the environments. In addition, we will release our code upon acceptance.

Hyperparameter	Value
Batch Size	16
Training Steps	10^6
Optimizer	Adam
Learning Rate	2×10^{-4}
Trajectory Length	10
Distance Threshold	1.5
Diffusion Steps	128
Number of Generations	5×10^6

Table 8: Hyperparameter settings used in our experiments.

Table 9: Results of CQL and Decision Transformer on the D4RL Maze, Antmaze, and Kitchen environments. **The numbers denote the performance increase by the data augmentation method compared to the original result.** RTDiff consistently improves the performance of offline reinforcement learning algorithms in all these environments.

Environment	Data Type	CQL (Kumar et al., 2020)			DT (Chen et al., 2021)		
		RTDiff	SynthER	ATraDiff	RTDiff	SynthER	ATraDiff
maze2d	umaze	**12.3**$^{\pm 3.5}$	6.3$^{\pm 4.1}$	7.1$^{\pm 4.0}$	**17.2**$^{\pm 4.7}$	8.3$^{\pm 4.3}$	9.0$^{\pm 4.1}$
	medium	**8.3**$^{\pm 2.7}$	5.8$^{\pm 3.2}$	6.2$^{\pm 3.1}$	**9.8**$^{\pm 2.5}$	6.3$^{\pm 2.5}$	5.9$^{\pm 2.6}$
	large	**11.3**$^{\pm 3.3}$	7.4$^{\pm 2.8}$	7.8$^{\pm 2.9}$	**12.7**$^{\pm 4.5}$	7.8$^{\pm 3.6}$	7.5$^{\pm 3.7}$
antmaze-umaze	fixed	**5.2**$^{\pm 3.3}$	4.9$^{\pm 3.7}$	4.8$^{\pm 3.6}$	**5.7**$^{\pm 3.5}$	5.4$^{\pm 3.8}$	5.5$^{\pm 3.7}$
	diverse	**4.3**$^{\pm 2.7}$	4.3$^{\pm 3.1}$	4.3$^{\pm 3.0}$	**4.2**$^{\pm 3.1}$	3.9$^{\pm 2.8}$	4.0$^{\pm 2.9}$
antmaze-medium	play	**7.9**$^{\pm 4.2}$	7.5$^{\pm 3.6}$	7.4$^{\pm 3.5}$	8.3$^{\pm 3.2}$	**8.4**$^{\pm 2.7}$	8.2$^{\pm 2.8}$
	diverse	**9.2**$^{\pm 3.8}$	8.5$^{\pm 3.6}$	8.8$^{\pm 3.7}$	**8.9**$^{\pm 2.4}$	8.3$^{\pm 3.6}$	8.5$^{\pm 3.4}$
antmaze-large	play	**6.5**$^{\pm 3.5}$	5.4$^{\pm 2.8}$	5.6$^{\pm 3.0}$	**5.4**$^{\pm 2.5}$	4.8$^{\pm 2.0}$	4.6$^{\pm 2.1}$
	diverse	**6.3**$^{\pm 3.4}$	5.7$^{\pm 2.5}$	5.9$^{\pm 2.7}$	**5.8**$^{\pm 5.5}$	4.7$^{\pm 6.2}$	5.0$^{\pm 6.0}$
kitchen	complete	**6.6**$^{\pm 7.4}$	3.4$^{\pm 8.3}$	4.0$^{\pm 8.0}$	**5.3**$^{\pm 7.2}$	3.6$^{\pm 6.5}$	4.2$^{\pm 6.7}$
	partial	**13.6**$^{\pm 6.3}$	8.3$^{\pm 7.2}$	9.0$^{\pm 7.0}$	**14.2**$^{\pm 7.8}$	6.4$^{\pm 6.8}$	7.0$^{\pm 6.9}$
	mixed	**11.3**$^{\pm 8.5}$	6.2$^{\pm 9.1}$	6.0$^{\pm 9.0}$	**10.3**$^{\pm 7.5}$	7.2$^{\pm 7.7}$	7.5$^{\pm 7.6}$

Table 10: Results of CQL and DT on the D4RL Locomotion environment. **The numbers denote the performance increase by the data augmentation method compared to the original result.** RTDiff improves the performance of these reinforcement learning methods in different tasks.

Environment	Data Type	CQL (Kumar et al., 2020)			DT (Chen et al., 2021)		
		RTDiff	SynthER	ATraDiff	RTDiff	SynthER	ATraDiff
walker2d	mixed	**5.2**$^{\pm 2.3}$	4.9$^{\pm 4.3}$	5.1$^{\pm 3.8}$	**2.2**$^{\pm 1.3}$	2.4$^{\pm 2.4}$	2.2$^{\pm 2.0}$
	medium	**2.6**$^{\pm 4.7}$	2.3$^{\pm 3.7}$	2.5$^{\pm 4.1}$	**2.3**$^{\pm 2.1}$	2.1$^{\pm 2.8}$	2.2$^{\pm 2.5}$
	medexp	**0.1**$^{\pm 0.4}$	0.0$^{\pm 0.4}$	0.1$^{\pm 0.4}$	**0.6**$^{\pm 0.8}$	0.4$^{\pm 0.7}$	0.5$^{\pm 0.7}$
hopper	mixed	16.4$^{\pm 1.7}$	**18.4**$^{\pm 2.4}$	17.6$^{\pm 2.1}$	11.2$^{\pm 5.3}$	**13.6**$^{\pm 4.7}$	13.2$^{\pm 4.5}$
	medium	**6.3**$^{\pm 6.0}$	5.8$^{\pm 4.8}$	6.1$^{\pm 5.4}$	**4.3**$^{\pm 1.5}$	3.5$^{\pm 2.3}$	4.0$^{\pm 2.0}$
	medexp	**5.3**$^{\pm 4.4}$	3.6$^{\pm 5.2}$	4.9$^{\pm 4.8}$	**1.6**$^{\pm 1.2}$	1.3$^{\pm 2.2}$	1.5$^{\pm 1.9}$
halfcheetah	mixed	**2.4**$^{\pm 0.8}$	1.9$^{\pm 0.5}$	2.3$^{\pm 0.6}$	**2.4**$^{\pm 0.8}$	1.9$^{\pm 0.5}$	2.3$^{\pm 0.6}$
	medium	**0.9**$^{\pm 0.3}$	0.6$^{\pm 0.4}$	0.8$^{\pm 0.4}$	**0.9**$^{\pm 0.3}$	0.6$^{\pm 0.4}$	0.8$^{\pm 0.4}$
	medexp	**1.3**$^{\pm 0.8}$	0.0$^{\pm 0.6}$	1.0$^{\pm 0.7}$	**1.3**$^{\pm 0.8}$	0.0$^{\pm 0.6}$	1.0$^{\pm 0.7}$

C More experimental results

In this section, we show more experimental results to support the conclusion of our paper.

C.1 Results with different basic RL algorithms

To illustrate that our RTDiff indeed improves the performance of general offline RL methods, here we include more experiments involving Decision Transformer (Chen et al., 2021) and CQL (Kumar et al., 2020), which are representative sequence modeling baseline and model-free baseline. The results shown in Tables 9 and 10 illustrate that our method consistently improves the performance of different offline RL methods.

C.2 Original performance report

The performance increase reported in Section 5.1 is measured by the difference between the normalized score with data augmentation and the original normalized score without any data augmentation methods. The original results are shown in Table 11.

C.3 More ablation studies

Threshold of the OOD detector. We select the value of this threshold with the following method, using D4RL Locomotion environment as the representative environment: We use grid search to

Table 11: Original normalized return of the methods we used in our paper on the D4RL Locomotion environment.

Environment	Data Type	CQL	TD3+BC	DT	IQL
walker2d	mixed	73.1 ± 13.2	85.6 ± 4.0	81.8 ± 6.9	82.2 ± 3.0
	medium	80.8 ± 3.3	82.7 ± 4.8	65.1 ± 1.6	80.9 ± 3.2
	medexp	109.6 ± 0.4	110.0 ± 0.4	110.4 ± 0.3	111.7 ± 0.9
hopper	mixed	95.1 ± 5.3	64.4 ± 21.5	59.9 ± 2.7	97.4 ± 6.4
	medium	59.1 ± 3.8	60.4 ± 3.5	67.6 ± 2.5	67.5 ± 3.8
	medexp	95.1 ± 5.3	101.2 ± 9.1	107.1 ± 1.0	107.4 ± 7.8
halfcheetah	mixed	45.0 ± 0.3	44.8 ± 0.6	38.9 ± 0.5	44.5 ± 0.2
	medium	47.0 ± 0.2	48.1 ± 0.2	42.2 ± 0.3	48.3 ± 0.2
	medexp	95.6 ± 0.4	90.8 ± 6.0	91.6 ± 1.0	94.7 ± 0.5

find the best choice of the hyperparameter, and then do a cross-validation of the representative environment to ensure its robustness. After selecting the threshold, we directly apply this threshold to all the environments we used, without any further tuning. To demonstrate the robustness of our threshold, we conduct a further ablation study on the environment maze2d. The results shown in Table 12 illustrate that this threshold $dis_M = 1.5$ is reasonable across different environments.

Table 12: Performance of maze2d environments under different thresholds. $dis_M = 1.5$ achieves the overall best performance compared with other threshold choices.

Threshold	1.0	1.3	1.5	2.0
maze2d-umaze	7.2	12.5	12.3	4.1
maze2d-medium	5.7	7.9	8.3	2.8
maze2d-large	7.1	9.6	11.3	3.7