

## A Implementation Details

### A.1 Task Description

**MuJoCo Locomotion.** MuJoCo locomotion encompasses several standard locomotion tasks commonly utilized in RL research, such as Hopper, Halfcheetah, and Walker2d. In each task, the RL agent controls a robot to achieve forward movement. D4RL [9] benchmark provides four qualities of datasets for each task: random-v2, medium-v2, medium-replay-v2, medium-expert-v2.

**Maze2D** The Maze2D domain is a navigation task requiring a 2D agent to reach a fixed goal location. The tasks are designed to provide a simple test of the ability of offline RL algorithms to stitch together previously collected subtrajectories to find the shortest path to the evaluation goal. The variations of this environment can be initialized with different maze configurations and increasing levels of complexity. Three maze layouts are provided: umaze, medium, and large. The task in the environment is for a 2-DoF ball that is force-actuated in the cartesian directions  $x$  and  $y$ , to reach a target goal in a closed maze.

**AntMaze Navigation.** Our tests on AntMaze navigation benchmark consist of 4 datasets, namely umaze-diverse-v2, medium-play-v2, medium-diverse-v2, and large-play-v2 from D4RL [9]. The objective is for an ant to learn how to walk and navigate from the starting point to the destination in a maze environment, with only sparse rewards provided. This task poses a challenge for online RL algorithms to explore high-quality data effectively without access to offline datasets or additional domain knowledge.

### A.2 Implementations and Hyperparameters in AD2S

Our Cal-QL implementation is based on previous work [35, 55], and primarily followed their recommended RL algorithm settings. The code can be found at <https://github.com/tinkoff-ai/CORL> and <https://github.com/liuxhym/EDIS>, which are released under an Apache license. The hyperparameters used in our AD2S’s other module are detailed in the Table A.2.

For all diffusion-based baselines, we use a 6-layer residual MLP as the denoising network. The residual denoising MLP not only provides high-fidelity data generation, but also enables a friendly computational cost in the online fine-tuning stage compared to other popular denoising networks such as U-net [26] or transformer [13]. During online fine-tuning, the diffusion synthesizer is retrained on offline and online samples for every 10,000 environment steps. We also use 5,000 steps at the start of online fine-tuning to warm up the online replay buffer in AD2S and PGR. For the diffusion sampling process, we follow previous works [39, 57, 35], using the stochastic SDE sampler of Karras et al. [22] with the same hyperparameter used in EDIS [35].

**Computation Resources** We train AD2S integrated with the base algorithm on an NVIDIA GeForce RTX 3090 GPU and a 32-core CPU.

## B Additional Experiments

To evaluate AD2S in sparse-reward and complex environments, we conduct experiments on the AntMaze navigation benchmark using 4 distinct datasets. Empirical results (Table 4) demonstrate that AD2S consistently outperforms baseline methods in settings with sparse rewards and complex action spaces. However, we observe that all methods exhibit unstable online fine-tuning, which we attribute to the inherent challenges of the AntMaze benchmark.

## C Additional Ablation Study

### C.1 Sensitivity Analysis.

**Distance-based alignment ratio.** We conduct experiments on walker2d task with 4 dataset qualities to give the sensitivity analysis on  $p_{DR}$  in AD2S. We choose  $p_{DR}$  from [0.1, 0.3, 0.5, 0.7] and the results are presented in Table 5. As demonstrated in our results, AD2S suffers from performance

Hyperparameter	Setting
Network Type (Denoising)	Residual MLP
Denoising Network Depth	6 layers
Denoising Steps	128 steps
Denoising Network Learning Rate	$3 \times 10^{-4}$
Denoising Network Hidden Dimension	1024 units
Denoising Network Batch Size	256
Denoising Network Activation Function	ReLU
Denoising Network Optimizer	Adam
Condition Dropout Rate	0.25
Learning Rate Schedule (Denoising Network)	Cosine Annealing
Training Epochs (Denoising Network)	50,000 epochs
Training Interval Environment Step (Denoising Network)	Every 10,000 steps
Replay Buffer Warm Up Step	5,000 steps
Density Ratio Network $w_\psi$ Hidden Dimension	256 units
Density Ratio Network $w_\psi$ Activation Function	ReLU
Dynamics Prediction Network $g$ Hidden Dimension	256 units
Dynamics Prediction Network $g$ Activation Function	Swish
$w_\psi$ & $g$ Learning Rate	$3 \times 10^{-4}$
$w_\psi$ & $g$ Optimizer	Adam
Amplified Ratio $\alpha$	1.2
Partial Noising Scale $\mu$	0.5 in MuJoCo locomotion 0.8 in Maze2D 0.25 in AntMaze
Density-based Prioritized Sampling Ratio $p_{DR}$	0.5 in MuJoCo locomotion 0.8 in Maze2D 0.3 in AntMaze
Curiosity-based Prioritized Sample Ratio $p_{Curi}$	0.5 in MuJoCo locomotion 0.8 in Maze2D 0.8 in AntMaze
Advantage Weight for Density Ratio $\beta_{DR}$	10
Advantage Weight for Curiosity $\beta_{curi}$	1

Table 3: Hyperparameters of AD2S for offline-to-online RL.

Dataset	Cal-QL	EDIS	AD2S
antmaze-umaze-diverse-v2	93.4 $\pm$ 4.6	95.9 $\pm$ 2.8	<b>96.8<math>\pm</math> 3.0</b>
antmaze-medium-play-v2	86.8 $\pm$ 1.6	93.9 $\pm$ 2.7	<b>94.4<math>\pm</math> 5.2</b>
antmaze-medium-diverse-v2	81.4 $\pm$ 3.9	<b>89.3<math>\pm</math> 4.8</b>	<u>85.0<math>\pm</math>10.0</u>
antmaze-large-play-v2	42.5 $\pm$ 5.2	66.1 $\pm$ 8.2	<b>72.5<math>\pm</math>11.8</b>

Table 4: D4RL normalized scores over five seeds on the antmaze tasks with the highest scores highlighted in **bold**. We also underline the AD2S’s score when it is close to the best score in each task. We conduct experiments to demonstrate the performance of AD2S on sparse rewards and complex environments.

degradation at minimal  $p_{DR}$  values. The constrained alignment ratio narrows the range of reusable data, thereby compromising the diversity of the synthesized distribution. Notably, while larger  $p_{DR}$  values preserve AD2S’s superior performance on high-quality data, they simultaneously introduce substantial performance fluctuations across different random initializations.

**Curiosity prioritization ratio.** We also investigate the choice of  $p_{Curi}$  for walker2d task with 4 data qualities in Table 6 and choose 4 levels  $p_{Curi}$  from [0.1, 0.3, 0.5, 0.7]. We observe that larger ratios improve performance for low-quality datasets, while smaller ratios perform better with high-quality datasets. This occurs because, in low-quality settings, the agent struggles to extract useful knowledge from the current data, necessitating policy improvement from all available data points. Conversely,

Dataset	$p_{\text{DR}} = 0.1$	$p_{\text{DR}} = 0.3$	$p_{\text{DR}} = 0.5$	$p_{\text{DR}} = 0.7$
walker2d-random-v2	36.8 $\pm$ 26.9	91.8 $\pm$ 2.5	76.0 $\pm$ 12.4	62.4 $\pm$ 23.6
walker2d-medium-v2	86.1 $\pm$ 1.9	118.6 $\pm$ 2.8	116.2 $\pm$ 3.5	117.5 $\pm$ 1.7
walekr2d-medium-replay-v2	120.3 $\pm$ 3.4	121.3 $\pm$ 3.7	105.2 $\pm$ 6.7	119.2 $\pm$ 6.7
walker2d-medium-expert-v2	110.7 $\pm$ 1.4	123.6 $\pm$ 1.7	121.1 $\pm$ 0.9	119.8 $\pm$ 3.0

Table 5: D4RL normalized scores over five seeds on the walker2d task with 4 data qualities. Here we investigate the sensitivity of the distance alignment ratio  $p_{\text{DR}}$  in AD2S.

Dataset	$p_{\text{Curi}} = 0.1$	$p_{\text{Curi}} = 0.3$	$p_{\text{Curi}} = 0.5$	$p_{\text{Curi}} = 0.7$
walker2d-random-v2	78.9 $\pm$ 7.6	77.4 $\pm$ 1.6	76.0 $\pm$ 12.4	81.6 $\pm$ 9.8
walker2d-medium-v2	117.7 $\pm$ 4.9	119.4 $\pm$ 3.0	116.2 $\pm$ 3.5	117.8 $\pm$ 4.9
walekr2d-medium-replay-v2	119.7 $\pm$ 2.5	119.2 $\pm$ 5.4	105.2 $\pm$ 6.7	114.8 $\pm$ 2.5
walker2d-medium-expert-v2	124.2 $\pm$ 1.2	124.7 $\pm$ 4.1	121.1 $\pm$ 0.9	123.4 $\pm$ 1.8

Table 6: D4RL normalized scores over five seeds on the walker2d task with 4 data qualities. Here we investigate the sensitivity of the curiosity alignment ratio  $p_{\text{Curi}}$  in AD2S.

when learning from high-quality data, the agent can efficiently optimize its policy from the existing samples, allowing it to increase curiosity-driven exploration.

## C.2 Synthetic Data Analysis

To verify the validity of the synthetic samples generated by AD2S, we use the ground-truth simulator to measure the dynamics distance (i.e., MSE error) between AD2S or other diffusion-based baselines with the real next state. We also measure the Oracle rewards defined by the simulator to verify the fidelity of the synthetic data. Empirical results are presented in Figures 5 and 6. These measurements provide an insight into the efficacy of our proposed method. As we can see, PGR can generate data with a large L2-distance but lacks the attribute of being highly rewarding. SynthER and EDIS can cover a ranger rewarding distribution than PGR, but rarely generate data in under-explored regions.

While AD2S incurs a higher dynamic distance due to its curiosity-driven data prioritization, it synthesizes data is further than all baseline methods in the distance metric. This capability helps mitigate the inherent pessimism in offline-pretrained Q-functions and improves online exploration. Furthermore, our results demonstrate that AD2S not only generates highly novel data but also shifts synthetic data with small L2 distances toward higher-rewarding regions in two out of three tasks. Notably, unlike PGR, which solely generates under-explored data, AD2S successfully produces data within high-rewarding distributions, as evidenced in Walker2d and Hopper environments (see Figure 6), and all methods fail to shift reward distributions in the HalfCheetah task.

## D Limitations and future works.

In the AD2S framework, we do not consider modeling the trajectory level information for data generation, which could help the sequential modeling method like DT [3, 67] for online fine-tuning. Trajectory augmentation may also be beneficial for fine-tuning the policy in sparse environments due to its high-fidelity transition modeling. Additionally, we find that the unconditioned diffusion-based data generation obtains promising performance on high-quality datasets but encounters degradation on low-quality datasets. Future work could explore the potential of combining our proposed data alignment and unconditioned data generation.

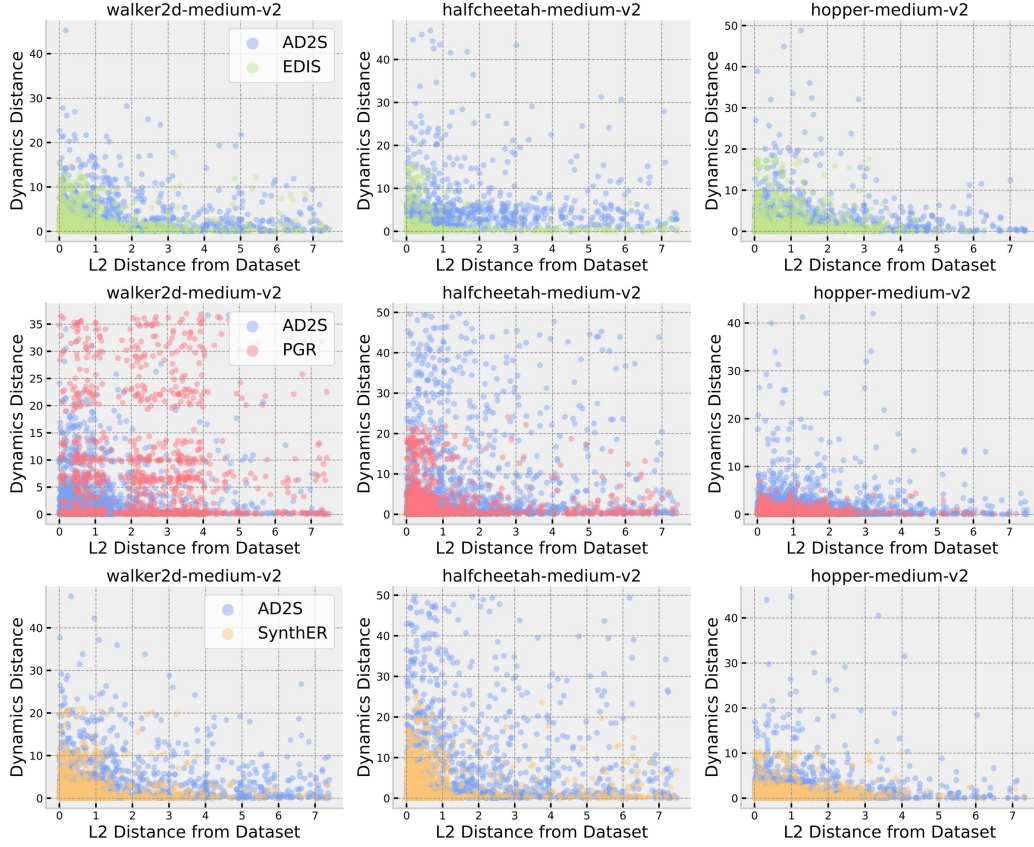


Figure 5: We plot L2 distance from online collected data, and dynamic distance under AD2S or diffusion-based baselines. **Top:** EDIS, **Middle:** PGR, and **Bottom:** SynthER. Compared to curiosity-based PGR, AD2S generates a wider marginal distribution over the distance from the real data. This indicates that our method can adaptively generate higher novelty data than the existing SOTA method.

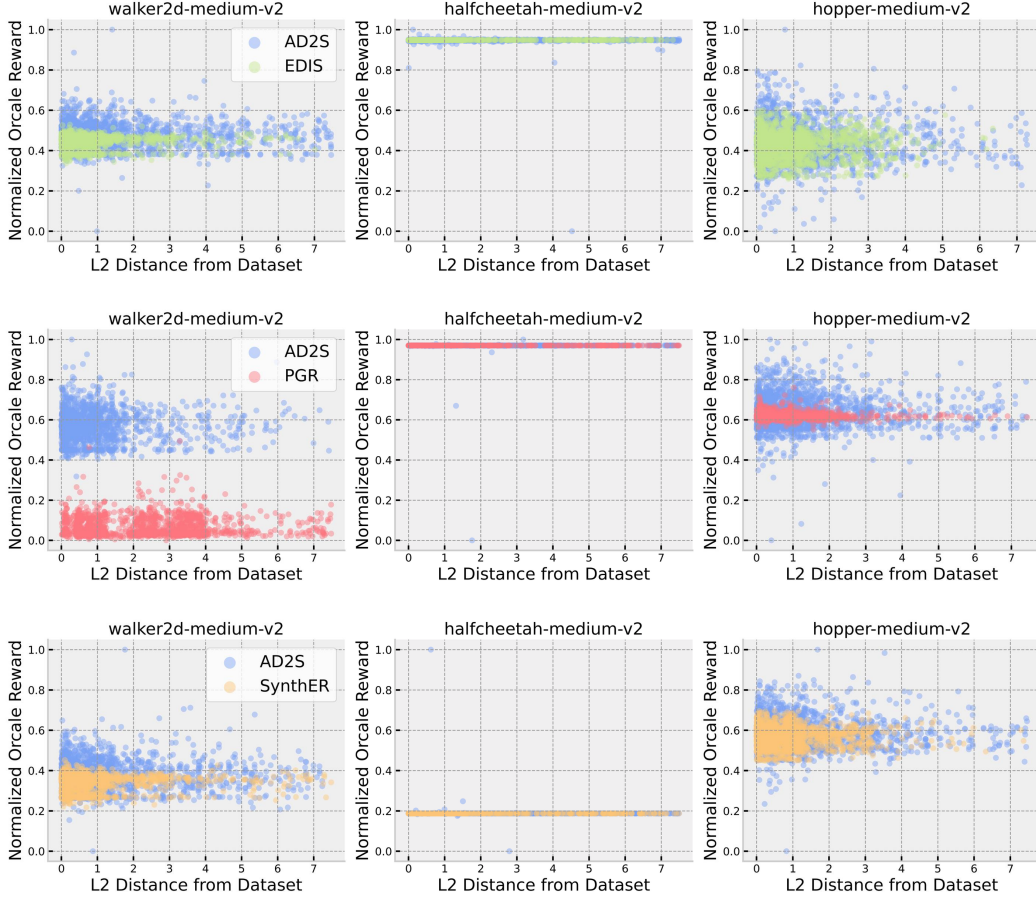


Figure 6: We plot L2 distance from online collected data, and oracle rewards under AD2S or diffusion-based baselines. **Top:** EDIS, **Middle:** PGR, and **Bottom:** SynthER. AD2S not only generates data with high novelty but also pushes the synthetic data with small L2 distance towards higher-rewarding regions in 2 of 3 tasks. This suggests that using the relative advantage to weight the data alignment and the guided data generation ensures high-quality synthetic data, thereby enabling effective online fine-tuning.