

A ADDITIONAL EXPERIMENTS

To better understand the role of the offline dataset as a prior in EXPO, we study EXPO in the setting of fine-tuning a pre-trained policy without the offline dataset used for pre-training. Instead of retaining the offline dataset, we use the pre-trained policy to collect data to warm-start the training. We present the results on Lift and Can in Figure 8 and make a comparison to Cal-QL pre-training followed by SAC fine-tuning baseline. For this ablation, we collect the same number of warm-start rollouts as contained in the offline dataset used for pre-training. We find that even without retaining the offline data, EXPO was able to learn to solve the tasks with high sample efficiency similar to retaining the dataset. This is compared to Cal-QL pre-training followed by SAC finetuning, which was not able to solve the task with this setup. This suggests the pre-train policy alone can act as a strong prior for EXPO to fine-tune and improve from, and in the context of pre-trained policies, EXPO can be used for effective, sample efficient fine-tuning even without the offline dataset used to pre-train the base policy.

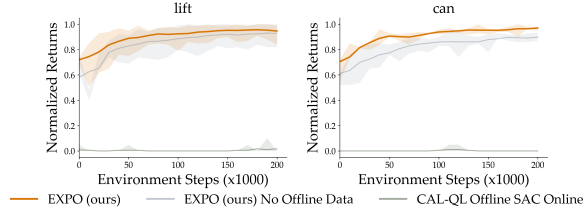


Figure 8: **Ablation on not keeping the offline dataset for fine-tuning.** We find that EXPO can learn effectively even without retaining the offline dataset after pre-training.

B EXPERIMENT DETAILS

Hyperparameters. Hyperparameters we used for EXPO can be found in Table 1. Each training run presented is with three seeds and error bars indicating max and min. For offline-to-online training, we present the number of pretraining steps for each suite. We do not pretrain in the online setting. We use the same residual block structure for the base policy as IDQL (Hansen-Estruch et al., 2023).

Hyperparameter	Robomimic	Adroit	Antmaze	Mimicgen
Optimizer		Adam		
Batch Size		256		
Learning Rate		3e-4		
Discount Factor		0.99		
Target Network Update τ		0.005		
Q -Ensemble Size		10		
N Action Samples		8		
UTD Ratio		20		
Num Min Q		2		
T		10		
Beta Schedule		Variance Preserving		
Base Policy MLP Hidden Dim		256		
Base Policy Num Residual Blocks		3		
Edit Policy MLP Hidden Dim		256		
Edit Policy MLP Hidden Layers		3		
Pretraining Steps	200k	20k	500k	200k
Edit Policy Dropout	None	0.1	None	None
Edit Policy β Online	0.05	0.7	0.05	0.05
Edit Policy β Offline-to-Online	0.1	0.7	0.05	0.05

Table 1: **Hyperparameters for EXPO.**

For our experiments, we find that EXPO generally works well across a fix set of hyperparameters and we only tune the edit policy β from $[0.05, 0.1, 0.3, 0.7]$. In terms of practical hyperparameter recommendations, we recommend a smaller value of β (e.g., 0.05 or 0.1) to start for tasks with a good offline dataset, and a larger value of β (e.g., 0.5, 0.7) to start for tasks where it is more important to explore to find the optimal strategy. While we do not extensively tune the number of action samples N , we note that a higher number of N might work better for higher dimensional action spaces.

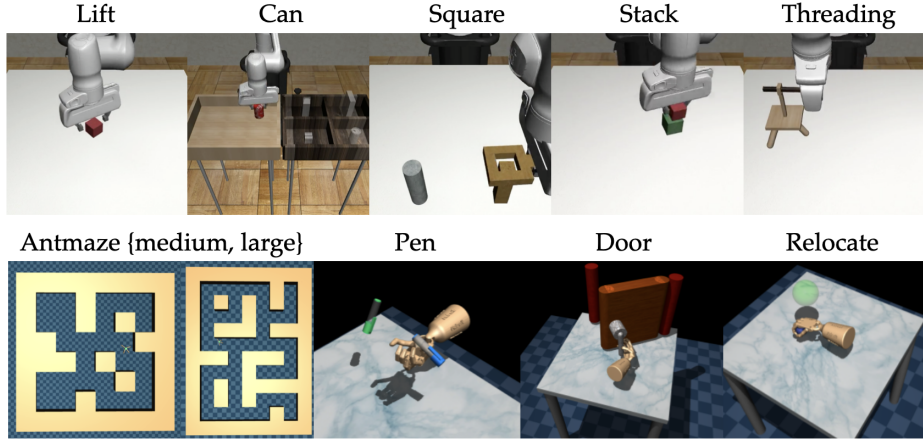


Figure 9: Visualizations of 12 sparse-reward environments we evaluate on. Note that Antmaze medium and Antmaze large both have two dataset variants.

Dataset. We list the details of the dataset used to pretrain (offline-to-online) and initialize (online) for the Robomimic and Mimicgen environments in Table 2. We subsample 10 trajectories for Lift and use the MH dataset for Can to make the tasks harder. The Adroit and Antmaze environments use the default D4RL provided datasets.

Hyperparameter	Num Data	Composition
MimicGen Stack	200	10 human and 190 generated by MimicGen
MimicGen Threading	50	10 human and 40 generated by MimicGen
Robomimic Lift	10	PH
Robomimic Square	200	PH
Robomimic Can	300	MH

Table 2: Dataset details for Robomimic and MicmicGen environments.

Evaluation. Evaluation is performed every 5k steps with 100 episodes for the Adroit and Antmaze environments and every 10k steps with 50 episodes for Robomimic and MimicGen environments. For the Adroit environments, normalized return is calculated as the percentage of the total timesteps the task is considered solved. This is the same metric as used in RLPD (Ball et al., 2023). All tasks use a sparse binary reward indicating whether the task has been completed successfully or not.

C BASELINES

IDQL (Hansen-Estruch et al., 2023). IDQL similarly features training an expressive diffusion policy via imitation learning and sampling multiple actions and selecting the one that maximizes the Q -value. However, the crucial differences are: (1) IDQL only uses the implicit policy for online exploration and use implicit Q -learning loss function for the TD backup (Kostrikov et al., 2021), (2) IDQL selects actions from action candidates directly sampled from the imitation learning policy.

RLPD (Ball et al., 2023). RLPD is a highly sample efficient algorithm that leverages prior data and oversamples from it for learning. RLPD uses a simpler Gaussian policy and has been shown to be better in performance compared to many offline-to-online methods even without pretraining. For both evaluation settings, we run RLPD without offline pre-training.

DAC (Fang et al., 2024). DAC is an offline RL method that uses an expressive diffusion policy. DAC includes action gradient of the Q -function as part of the diffusion loss to guide its denoising process towards generating more optimal actions. We adapt this method to the offline-to-online RL setting by first pre-training it with the offline RL and the continue to fine-tune it online with the same objective.

Cal-QL (Nakamoto et al., 2023) (Offline-to-Online only). Cal-QL is a standard offline-to-online RL baseline that does not use an expressive policy. Instead, Cal-QL calibrates the Q -function with Monte-Carlo returns as a way to balance pessimism of offline RL and optimism of online fine-tuning and prevent policy unlearning from offline to online training.

QSM (Psenka et al., 2023) (Online only). QSM is an online RL method that trains diffusion policies by matching the diffusion loss to action gradients. QSM aims to avoid instability of value propagation to the expressive policy by incorporating losses to guide the denoising process.