APPENDIX

Table of Contents

A IMPLEMENTATION DETAILS ON OUR METHOD

In this section, we describe the additional implementation details on our proposed method. The details on model architecture is presented in Section A.1, followed by the training detailed described in Section A.2.

A.1 MODEL ARCHITECTURE

We describe the details on our model architecture in this section.

A.1.1 SKILL PRIOR

We followed architecture and learning procedure of Pertsch et al. (2020) for learning a low-level skill policy and a skill prior. Please refer to Pertsch et al. (2020) for more details on the architectures for learning skills and skill priors from offline datasets.

A.1.2 TASK ENCODER

Following Rakelly et al. (2019), our task encoder is a permutation invarient neural network. Specifically, we adopt Set Transformer (Lee et al., 2019) that consists of layers $[2 \times \text{ISAB}_{32} \rightarrow \text{PMA}_1 \rightarrow 3 \times \text{MLP}]$ for expressive and efficient set encoding. All the hidden layers are 128-dimensional and all attention layers have 4 attention heads. The encoder takes a set of high-level transitions as input, where each transition is a vector concatenation of high-level transition tuple. The output of the encoder is $(\mu_{z_T}, \sigma_{z_T})$ which are the parameters of Gaussian task posterior $p(z_T|\tau) = \mathcal{N}(z_T; \mu_{z_T}, \sigma_{z_T})$. We varied task vector dimension $\dim(z_s)$ depends on task distribution complexity. $\dim(z_s) = 10$ for Kitchen Manipulation, $\dim(z_s) = 6$ for Maze Navigation with 40 meta-training tasks, and $\dim(z_s) = 5$, otherwise.

A.1.3 POLICY

We parameterize our policy with neural network. We employed 4-layer MLPs with 256 hidden units for Maze Navigation, and 6-layer MLPs with 128 hidden unit for Kitchen Manipulation experiment. Instead of direct parameterization of policy, the network output is added to skill-prior to make learning more stable. Specifically, the policy network takes concatenation of (s, z_T) as input, and then outputs (μ_s, σ_s), which is used as the parameters of a Gaussian distribution. From this predicted Gaussian distribution $\mathcal{N}(z_s|\mu_p + \mu_s, \sigma_s)$, we sample to obtain the input to control the low-level skill policy.

Here μ_p is, the mean of skill-prior for given state, $\mathbb{E}[p(z_s|s)]$.

A.1.4 CRITIC

The critic network takes concatenation of s, z_T, and skill z_s as input and outputs an estimation of task-conditioned Q-value $Q(s, z_s, z_t)$. We employ double Q networks (Van Hasselt et al., 2016) to mitigate Q-value overestimation. The architecture of critic follows the policy.

A.2 TRAINING DETAILS

For all the network updates, we used Adam optimizer (Kingma & Ba, 2015) with a learning rate of $3e-4$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We describe the training details of the skill-based meta-training phase in Section A.2.1 and the target task learning phase Section A.2.2.

A.2.1 SKILL-BASED META-TRAINING

Our meta-training procedure is similar to the procedure adopted in (Rakelly et al., 2019). We update the networks with the average gradients of 20 randomly sampled tasks. Each batch of gradients is computed from 1024 and 256 transitions for Maze Navigation and Kitchen Manipulation experiment, respectively. We train our models for 10000, 18000, and 16000 episodes for the Maze Navigation experiments with 10, 20, 40 meta-training tasks, respectively, and 3450 episodes for Kitchen Manipulation.

As stated in Section 4.2, we apply different regularization coefficients depending on the size of the conditioning transitions. In Maze Navigation experiment, we set target KL divergence to 0.1 for batch that is conditioned on size 4 transitions and 0.4 for batch conditioned on size 8192 transitions. In Kitchen Manipulation experiment, we set target KL divergence to 0.4 for bath conditioned with a size 1024 transitions while KL coefficient for batch conditioned on size 2 transitions is fixed to 0.3.

A.2.2 TARGET TASK LEARNING

We initialize the Q function and the auto-tuning value α with the values that learned in the skill-based meta-training phase. The policy is initialized after observing and encoding 20 episodes obtained from the task unconditioned policy rollouts. For the target task learning phase, the target KL δ is 1 for Maze Navigation, and 2 for Kitchen Manipulation experiments. To compute a gradient step, 256 high-level transitions are sampled from a replay buffer with size 20000. Note that we used same setup for baselines that uses SPiRL fine-tuning (SPiRL and MTRL).

B IMPLEMENTATION DETAILS ON BASELINES

In this section, we describe the additional implementation details on producing the results of the baselines.

B.1 SAC

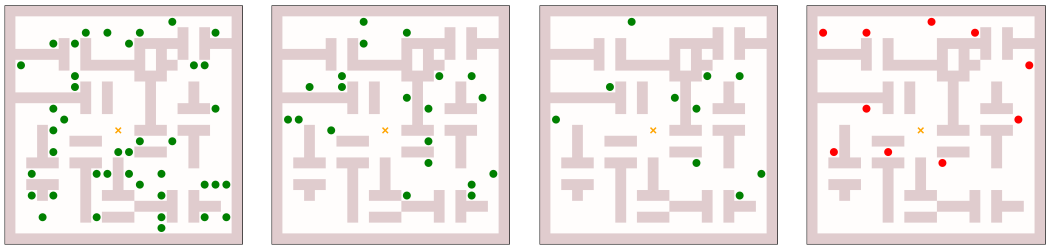
The SAC (Haarnoja et al., 2018) baseline learns to solve a target task from scratch without leveraging the offline dataset nor the meta-training tasks.

We initialize α to 0.1 and set the target entropy to $\mathcal{H} = -\dim(\mathcal{A})$. To compute a gradient step, 256 environment transitions are sampled from a replay buffer.

B.2 PEARL AND PEARL-FT

PEARL (Rakelly et al., 2019) learns from the meta-training tasks but does not use the offline dataset. Therefore, we directly train PEARL models on the meta-training tasks without the phase of learning from offline datasets. We use gradients averaged from 20 randomly sampled tasks where each task gradient is computed by batch sampled from a per-task buffer. The target entropy is set to $\mathcal{H} = -\dim(\mathcal{A}) = -2$ and α is initialized to 0.1.

While the method proposed in Rakelly et al. (2019) does not fine-tune on target/meta-testing tasks, we extend PEARL to be fine-tuned on target tasks for a fair comparison, called PEARL-ft. Since PEARL does not use learned skills or a skill prior, the target task learning of PEARL is simply running SAC



(a) Meta-training 40 Tasks (b) Meta-training 20 Tasks (c) Meta-training 10 Tasks (d) Target Tasks

Figure 7: **Maze Meta-training and Target Task Distributions.** The green dots represent the goal locations of meta-training tasks and the red dots represent the goal locations of target tasks. The yellow cross represent the initial location of the agent.

with task-encoded initialization. Similar to the target task learning of our method, we initialize the Q function and entropy coefficient α to the valued learned during the meta-training phase. Also, we initialize the policy to the task conditioned policy after observing 20 episodes of experience from the task unconditioned policy rollouts. The hyperparameters used for fine-tuning are the same as SAC.

B.3 SPiRL

Similar to our method, we initialize the high-level policy to skill-prior while fixing low-level policy for target task learning for SPiRL. α is initialized to 0.01 and we use the same hyperparameters for the SPiRL models as our method.

B.4 Multi-task RL (MTRL)

Inspired by Distral (Teh et al., 2017), our multi-task RL baseline is designed to first learns a set of individual policies, where each of them is specialized in one task; then, a shared/multi-task policy is learned by distilling the individual polices. Since it is inefficient to learn an individual policy from scratch, we learn each individual policy using SPiRL with learned skills and a skill prior. Then, we distill the individual policies using the following objective :

$$\max_{\pi_0} \mathbb{E}_T \left[\sum_{t=0}^{H-1} r_T(s_t, z_s) - \alpha D_{\text{KL}}\big(\pi_0(z_{s,t}|s_t), p(z_s|s_t)\big) \right], \tag{6}$$

where the KL constraint is now with skill-prior instead of task expert policy. We use the same setup for α as our method, where α is auto-tuned to satisfy a target KL, $\delta = 0.1$ for Maze Navigation and $\delta = 0.2$ for Kitchen Manipulation.

While the target task learning phase for MTRL is similar ours, except that MTRL is not initialized with a meta-trained Q function and learned α.

C Meta-training Tasks and Target Tasks.

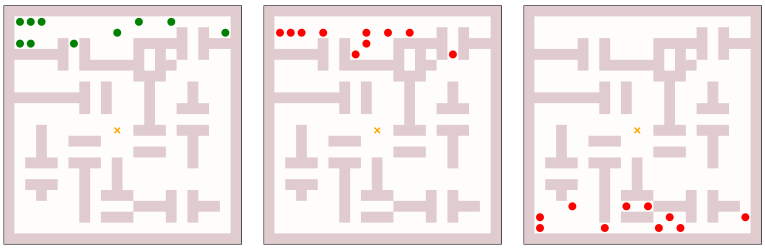
In this section, we present the meta-training tasks and target tasks used in the maze navigation domain and the kitchen manipulation domain.

C.1 Maze Navigation

The meta-training tasks and target tasks are visualized in Figure 7 and Figure 8.

C.2 Kitchen Manipulation

The meta-training tasks are:



(a) $\mathcal{T}_{\text{TRAIN-TOP}}$ (b) $\mathcal{T}_{\text{TARGET-TOP}}$ (c) $\mathcal{T}_{\text{TARGET-BOTTOM}}$

Figure 8: **Maze Meta-training and Target Task Distributions for Meta-training Task Distribution Analysis.** The green dots represent the goal locations of meta-training tasks and the red dots represent the goal locations of target tasks. The yellow cross represent the initial location of the agent.

- microwave→kettle→bottom burner→slide cabinet
- microwave→bottom burner→top burner→slide cabinet
- microwave→top burner→light switch→hinge cabinet
- kettle→bottom burner→light switch→hinge cabinet
- microwave→bottom burner→hinge cabinet→top burner
- kettle→top burner→light switch→slide cabinet
- microwave→kettle→slide cabinet→bottom burner
- kettle→light switch→slide cabinet→bottom burner
- microwave→kettle→bottom burner→top burner
- microwave→kettle→slide cabinet→hinge cabinet
- microwave→bottom burner→slide cabinet→top burner
- kettle→bottom burner→light switch→top burner
- microwave→kettle→top burner→light switch
- microwave→kettle→light switch→hinge cabinet
- microwave→bottom burner→light switch→slide cabinet
- kettle→bottom burner→top burner→light switch
- microwave→light switch→slide cabinet→hinge cabinet
- microwave→bottom burner→top burner→hinge cabinet
- kettle→bottom burner→slide cabinet→hinge cabinet
- bottom burner→top burner→slide cabinet→light switch
- microwave→kettle→light switch→slide cabinet
- kettle→bottom burner→top burner→hinge cabinet
- bottom burner→top burner→light switch→slide cabinet

The target tasks are:

- microwave→bottom burner→light switch→top burner
- microwave→bottom burner→top burner→light switch
- kettle→bottom burner→light switch→slide cabinet
- microwave→kettle→top burner→hinge cabinet
- kettle→bottom burner→slide cabinet→top burner
- kettle→light switch→slide cabinet→hinge cabinet
- kettle→bottom burner→top burner→slide cabinet

- microwave→bottom burner→slide cabinet→hinge cabinet
- bottom burner→top burner→slide cabinet→hinge cabinet
- microwave→kettle→bottom burner→hinge cabinet