A Background

Off-policy RL with Soft Actor-Critic. The Soft Actor-Critic (SAC) [1] is a leading off-policy RL algorithm. Like other off-policy RL methods, such as DQN [2] or DDPG [3], SAC optimizes a Q function but does so based on the maximum entropy framework for RL [4]. In addition to maximizing the reward function, SAC also maximizes the policy entropy which leads to improved exploration and helps prevent overfitting. As an actor-critic method, SAC optimizes both the actor's policy by maximizing a value function as well as a critic with a Bellman loss. The actor's parameters are updated to maximize the Q function and policy entropy which is encapsulated by the following equation:

$$\mathcal{L}^{\texttt{SAC}}_{\texttt{actor}} = \mathbb{E}_{s_t \sim \mathcal{B}, a_t \sim \pi_{\theta_1}} \Big[\alpha \log \pi_{\theta_1}(a_t|s_t) - Q_{\theta_2}(s_t, a_t) \Big]. \tag{1}$$

Here, (s_t, a_t) are state-action pairs, $\mathcal{B}$ is a replay buffer, θ_1 is the actor's parameters, θ_2 are the critic's parameters, and α is a scalar value that control the entropy strength. The policy π_{θ_1} is parametrized by a multi-variate Gaussian with a diagonal covariance matrix and outputs the means and standard deviations that are then used to sample actions from the Gaussian distribution. To update the critic's parameters, SAC optimizes a soft Q function by minimizing the soft Bellman loss:

$$\mathcal{L}^{\texttt{SAC}}_{\texttt{critic}} = \mathbb{E}_{\tau_t} \Big[\big(Q_{\theta_2}(\mathbf{s}_t, a_t) - R_t - \gamma \big[Q_{\bar{\theta_2}}(s_t, a_t) - \alpha \log \pi_{\theta_1}(a_t|s_t) \big] \big)^2 \Big], \tag{2}$$

where $\tau_t = (s_t, a_t, s_{t+1}, R_t)$ is a single timestep transition, $\bar{\theta}$ denotes the Polyak averaging of the critic's parameters, and α is a temperature parameter.

B Implementation Details

B.1 Regularize SAC by Prior

$$\mathcal{L}^{\texttt{SAC}}_{\texttt{actor}} = \mathbb{E}_{z_t \sim \mathcal{B}, a_t \sim \pi_{\theta_1}} \Big[\alpha D_{KL}(\pi_{\theta_1}(a_t|z_t), p_a(z_t|s_t)) - Q_{\theta_2}(z_t t, a_t) \Big]. \tag{3}$$

$$\mathcal{L}^{\texttt{SAC}}_{\texttt{critic}} = \mathbb{E}_{\tau_t} \Big[\big(Q_{\theta_2}(\mathbf{s}_t, a_t) - R_t - \gamma \big[Q_{\bar{\theta_2}}(z_t, a_t) - \alpha D_{KL}(\pi_{\theta_1}(a_t|z_t), p_a(z_t|s_t)) \big] \big)^2 \Big], \tag{4}$$

where $p_a(z_t|s_t)$ is the prior distribution learned from offline dataset

B.2 Hyperparamters

Because we built off of SPiRL [5], we used the same set of hyperparamters for skill extraction and online RL training. The reward model learning from human preference has the same hyperparamters as in PEBBLE. [6].

Hyperparameters for Skill Extraction	**Value**
Skill Horizon	10
Ensemble Size	3
Hidden Units	200
Non-linearity	ReLU
Optimizer	Adam
Learning Rate	0.001
Weight Decay	0.0001
(β_1, β_2)	$(.9, .999)$

Hyperparameters for Skill Execution	**Value**
Query Batch Size	128
Query Frequency	$100,000$
Segment Size	5
Sampling Scheme	Entropy Exploit

C Effect of Segment Size

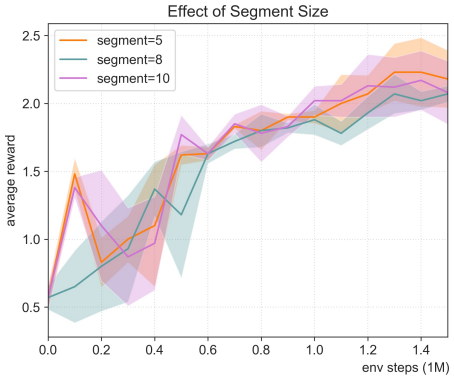

Figure 1: The plot compares SkiP with different segment size over the Kettle-Burner-Cab environment. Lines and shaded area represent mean and standard error over three seeds, respectively.

As shown in Fig 1, unlike PEBBLE [6], we did not find segment size to affect our method's performance.

D Robustness Ablation

What if the offline dataset is of mixed-quality demonstrations? To test our method on more realistic mixed-quality demonstration, we added an ablation that simulated imperfect demonstrations by adding Gaussian noise to some of the action sequences in the offline dataset. The scale of the Gaussian noise is 10% of maximum action magnitude. We ablate the downstream performance on the three-task Kettle-Burner-Cabinet environment against the percentage of noise present in the offline dataset. We found that the weighting during skill extraction is still mostly effective even with noisy dataset, as shown in 1

Table 1: Percentage of noise in offline dataset versus downstream performance

PERCENTAGE OF NOISY DATA	RETURN AT 1.5M STEPS	RETURN AT 800K STEPS
0%	2.8 ± 0.2	2.5 ± 0.2
5%	2.7 ± 0.4	2.3 ± 0.6
10%	2.3 ± 0.6	1.7 ± 0.7
20%	2.4 ± 0.6	1.8 ± 1.0

What if the human makes mistakes? We ablate the downstream performance on the three-task Kettle-Burner-Cabinet environment against the percentage of random human preference labels, as shown in 2.

Table 2: Percentage of random human preference vs downstream performance

PERCENTAGE OF RANDOM PREFERENCE	RETURN AT 1.5M STEPS	RETURN AT 800K STEPS
0%	2.8 ± 0.2	2.5 ± 0.2
4%	1.9 ± 0.6	1.6 ± 0.4
10%	1.5 ± 0.4	1.3 ± 0.8
20%	1.2 ± 0.7	1.0 ± 0.5

Table 3: Skill horizon vs downstream performance. We found that while too small of a skill horizon length did hurt performance, a longer skill horizon does not affect the performance by much

SKILL HORIZON LENGTH	RETURN AT 1.5M STEPS	RETURN AT 800K STEPS
5	1.0 ± 0.1	1.0 ± 0
10	2.8 ± 0.2	2.5 ± 0.2
20	2.6 ± 0.2	2.5 ± 0.5
40	2.8 ± 0.1	1.8 ± 1.6

References

[1] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

[4] B. D. Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 2010.

[5] K. Pertsch, Y. Lee, and J. J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning (CoRL)*, 2020.

[6] K. Lee, L. Smith, and P. Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021.