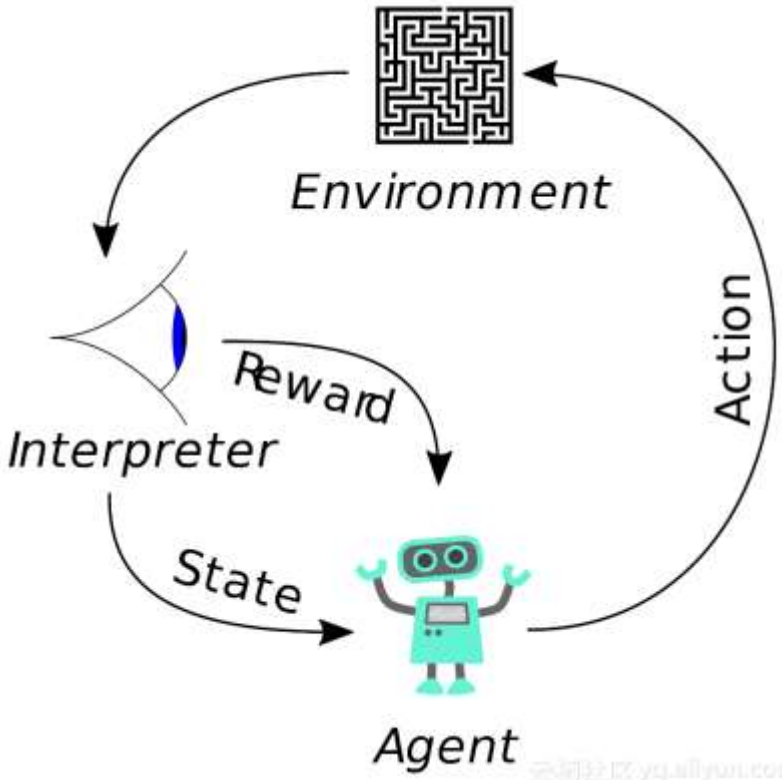
Residual Skill Policies: Learning an Adaptable Skill-based Action Space for Reinforcement Learning for Robotics

Krishan Rana[1], **Ming Xu**[1], **Brendan Tidd**[2], **Michael Milford**[1], **Niko Sünderhauf**[1]
[1]QUT Centre for Robotics, Queensland University of Technology
[2]Data61 Robotics and Autonomous Systems Group, CSIRO
ranak@qut.edu.au

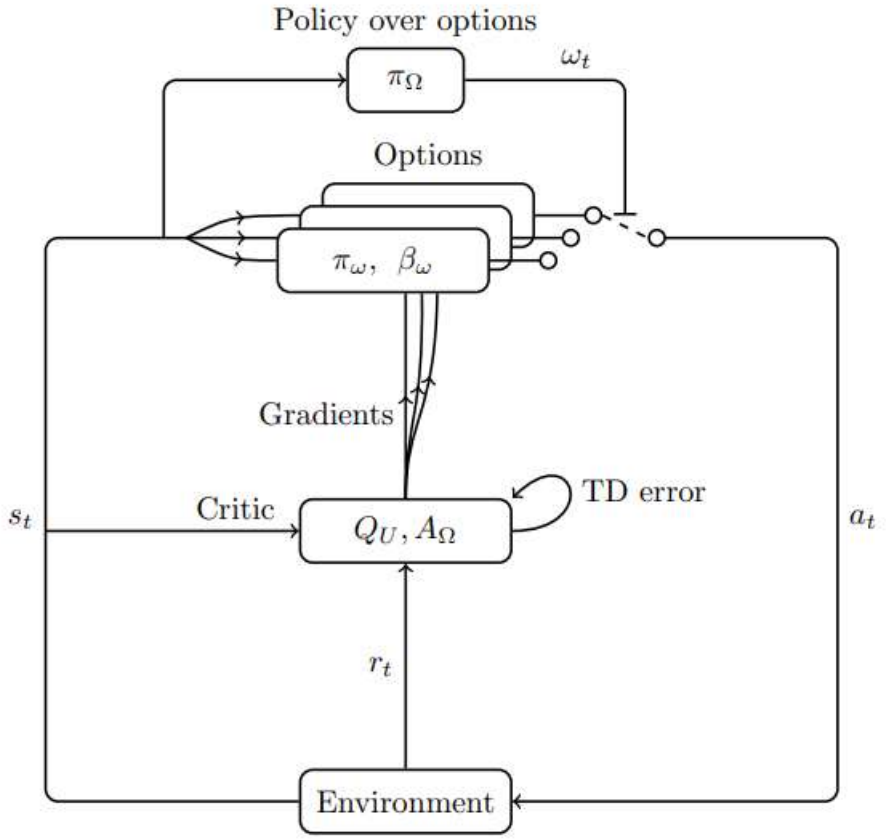
CORL 2022

Preliminaries



Atom Action Control

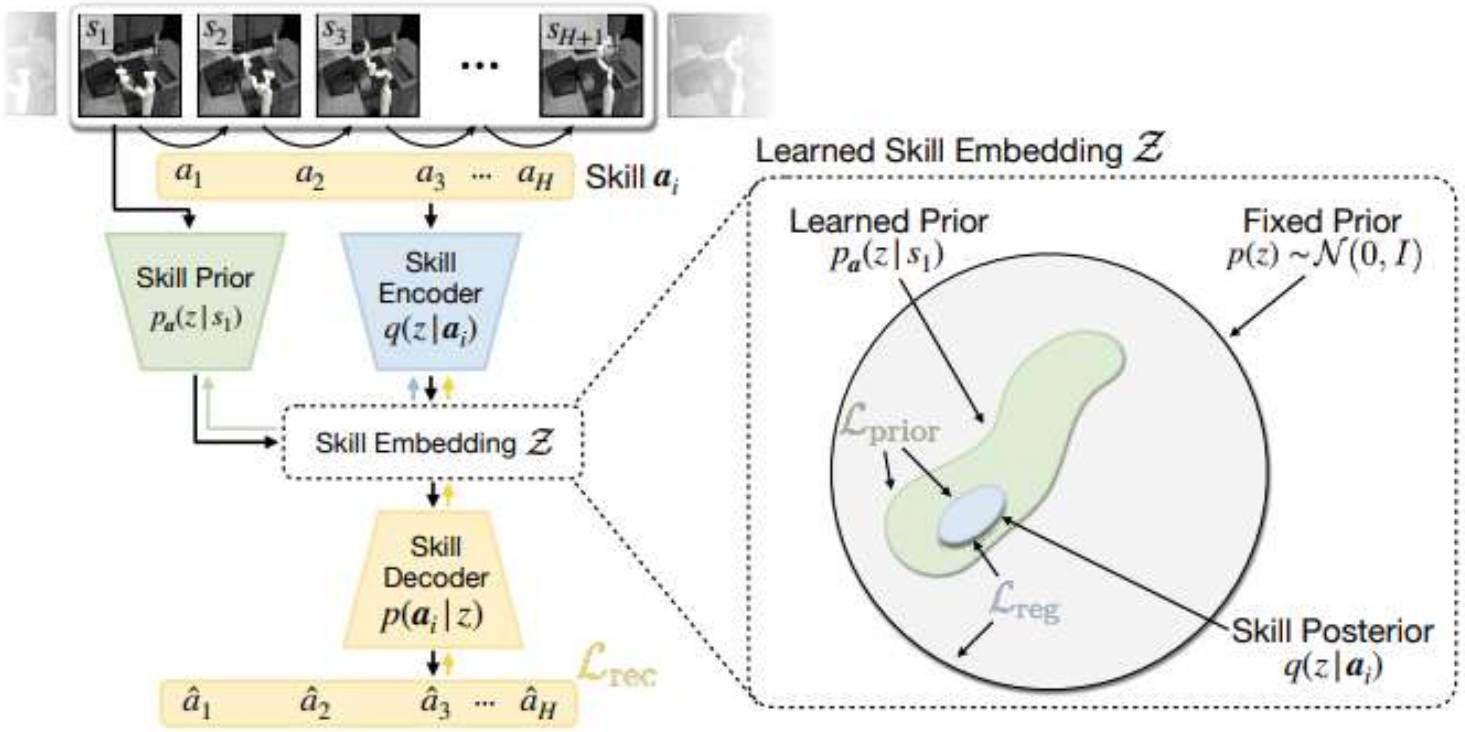
Buffer:(s,a,r,s')

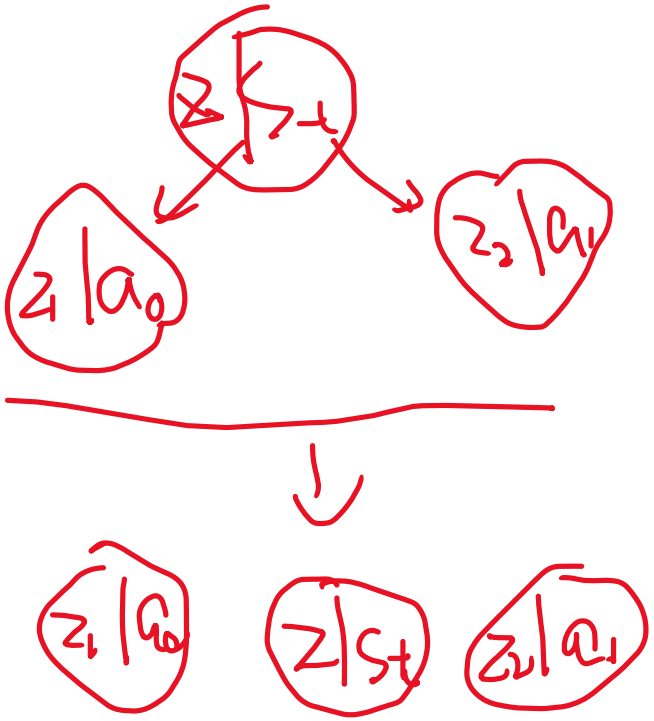


Skill-based Control

Buffer:(s,z,r,s')

Preliminaries-SPIRL





$$\log p(\boldsymbol{a}_i) \geq \mathbb{E}_q \Big[\underbrace{\log p(\boldsymbol{a}_i|z)}_{\text{reconstruction}} - \beta \big(\underbrace{\log q(z|\boldsymbol{a}_i) - \log p(z)}_{\text{regularization}} \big) \Big].$$

$$\mathbb{E}_{(s,\boldsymbol{a}_i)\sim\mathcal{D}} D_{\mathrm{KL}}\big(q(z|\boldsymbol{a}_i), p_{\boldsymbol{a}}(z|s_t)\big).$$

Skill-conditioned BC

Distribution alignment

Motivation

Troubles in unsupervised skill discovery:

1. Unsupervised Skill discovery is time-consuming and low-qualitified.

2. Hard to achieve the coverage of the state space.

3. Data-Driven semi-supervised Skill Discovery is growing up.

Troubles in previous Skill-based RL:

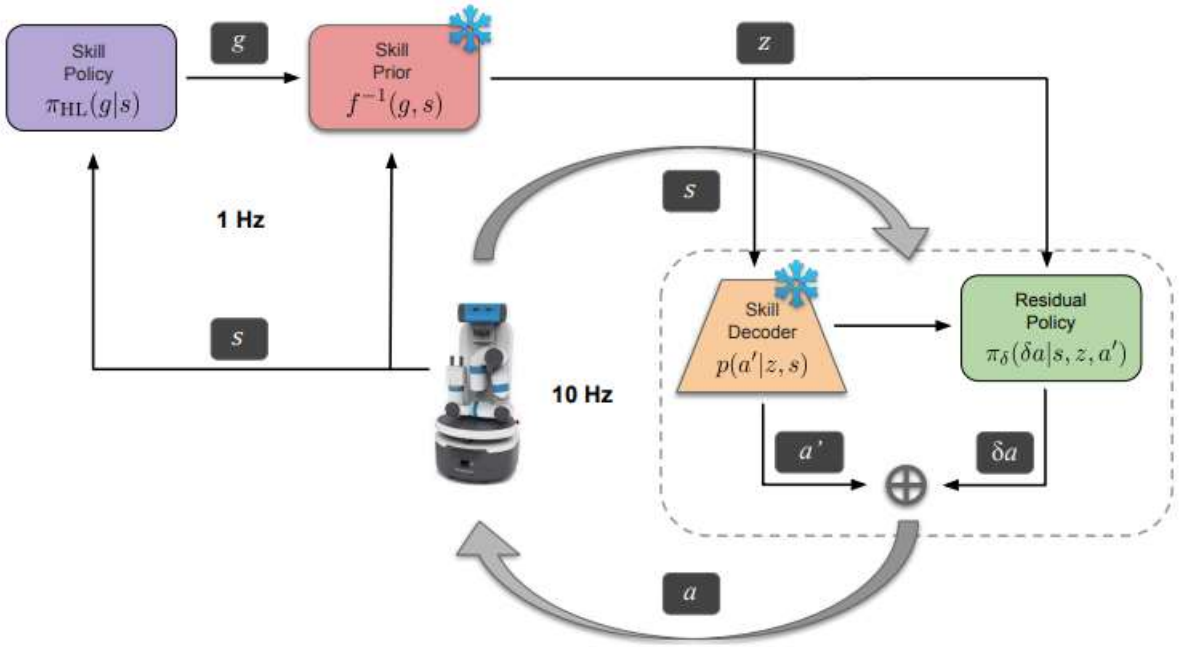
1. Spirl explores the whole skill space to find suitable skill, which is time-consuming.

   The reason for introducing the R-NVP model.

2. The skill space extracted from the dataset is not capable to finish different downstream tasks.

   The reason for introducing the residual-policy module.

Framework



Details:

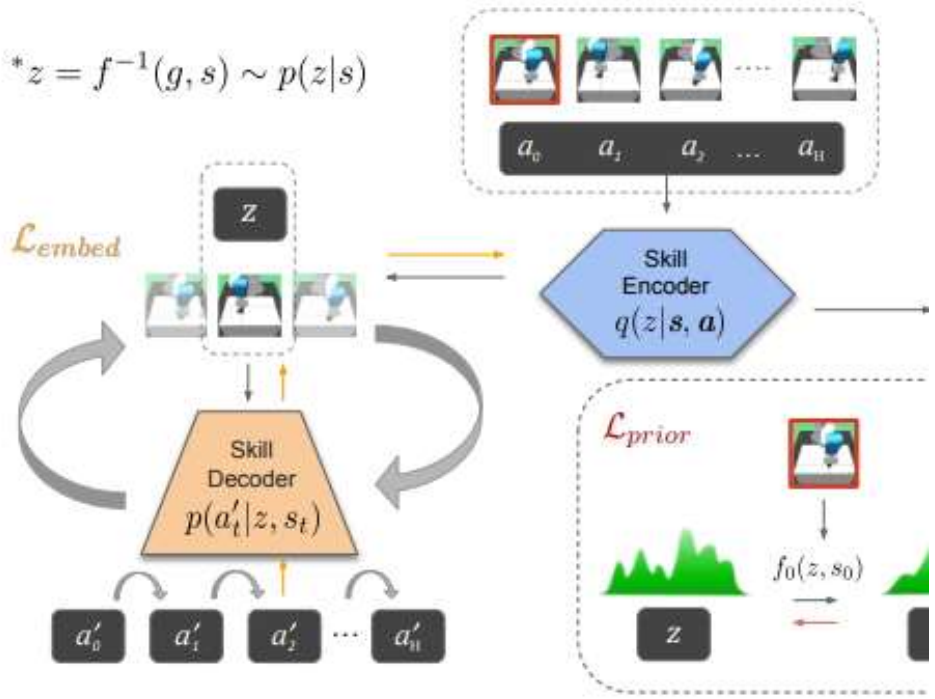
1. How to encode trajectories into a skill z and decode a skill z and current state to the action?

2. How to fix the skill exploration space of current state s?

3. How to make skills adaptive to different downstream tasks?

Algorithm-Skill Extraction

1. How to encode trajectories into a skill z and decode a skill z and current state to the action?



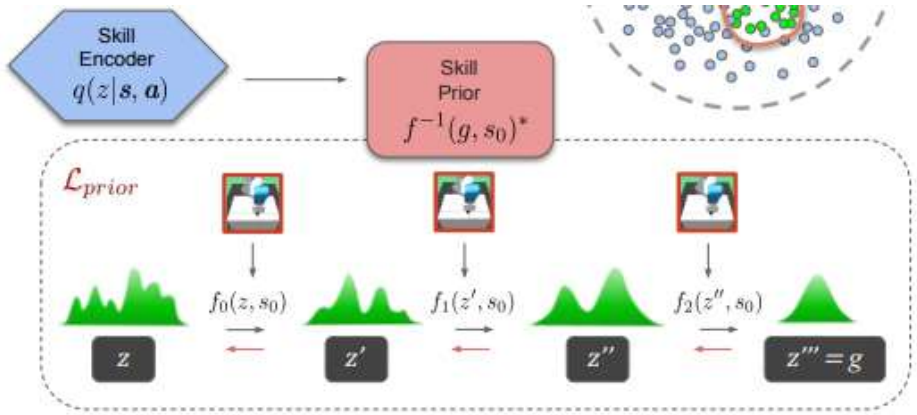
$$\mathcal{L}_{embed} = \mathbb{E}_{q_\phi(z|\boldsymbol{s},\boldsymbol{a})}\left[\sum_{t=0}^{H-1} \log p_\theta\left(a_t \mid z, s_t\right)\right] - \beta D_{KL}(q_\phi\left(z \mid \boldsymbol{s}, \boldsymbol{a}\right) \| p(z)),$$

```python
def forward(self, x):
    states = x['obs']
    actions = x['actions']

    # Encoding
    x_cat = torch.cat((states, actions), 2)
    x = self.run_inference(x_cat)
    q = AttrDict(mu=x[:,0,:],
                 log_var=x[:,1,:])

    z = self.reparameterize(q.mu, q.log_var)
    z_tiled = z.repeat(1,self.seq_len).view(actions.shape[0],self.seq_len,self.n_z)

    # Decoding
    # Closed loop decoding
    decode_inputs = torch.cat((states, z_tiled), 2)
    reconstruction = self.run_decode_batch(decode_inputs, self.decoder)

    return AttrDict(reconstruction=reconstruction, q=q, z=z)
```

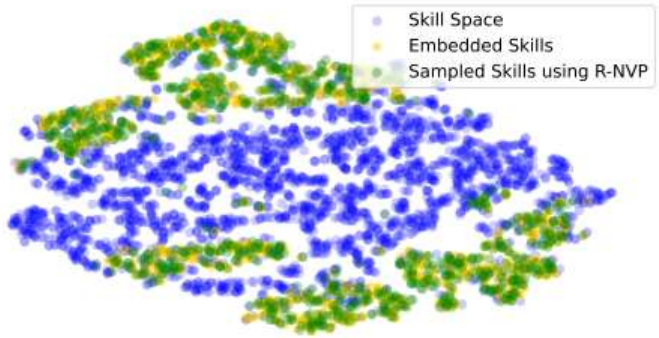
```python
def vae_loss(self, inputs, output, beta=0.00000001):
    bc_loss = self.bc_criterion(output.reconstruction, inputs["actions"])
    kld_loss = (-0.5 * torch.sum(1 + output.q.log_var - output.q.mu.pow(2) - output.q.log_var.exp())) * b
    return bc_loss, kld_loss
```

Algorithm-Skill Exploration

2. How to fix the skill exploration space of current state s?



$$p_{\text{prior}}(a|s) = p_z\left(f_\phi^{-1}(a;s)\right)\left|\det\left(\partial f_\phi^{-1}(a;s)/\partial a\right)\right|$$

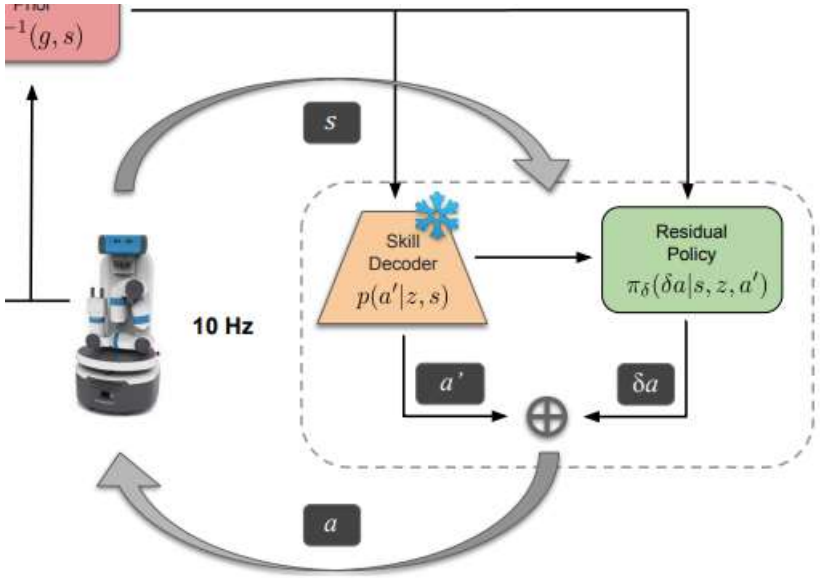



```python
class stacked_NVP(nn.Module):
    def __init__(self, d, k, n_hidden, state_size, n, device):
        super().__init__()
        self.bijectors = nn.ModuleList([
            R_NVP(d, k, state_size, n_hidden=n_hidden, device=device).to(device) for _ in range(n)
        ])
        self.flips = [True if i%2 else False for i in range(n)]

    def forward(self, x):
        log_jacobs = []

        for bijector, f in zip(self.bijectors, self.flips):
            x, log_pz, lj = bijector(x, flip=f)
            log_jacobs.append(lj)

        return x, log_pz, sum(log_jacobs)

    def inverse(self, z):
        for bijector, f in zip(reversed(self.bijectors), reversed(self.flips)):
            z = bijector.inverse(z, flip=f)
        return z
```

```python
z, log_pz, log_jacob = self.sp_nvp(sp_input)
sp_loss = (-log_pz - log_jacob).mean()
sp_loss.backward()
self.sp_optimizer.step()
```

Algorithm-Residual Policy

3. How to make skills adaptive to different downstream tasks?




```
residual_agent = PPO(ac_kwargs=dict(hidden_sizes=[args.residual_hid]*args.residual_l),
                     gamma=args.residual_gamma,
                     seed=args.seed,
                     steps_per_epoch=(args.skill_steps_per_epoch*seq_len),
                     epochs=args.epochs,
                     clip_ratio=args.residual_clip_ratio,
                     pi_lr=args.residual_pi_lr,
                     vf_lr=args.residual_vf_lr,
                     train_pi_iters=args.residual_train_pi_iters,
                     train_v_iters=args.residual_train_v_iters,
                     lam=args.residu (function) residual_target_kl: Any
                     target_kl=args.residual_target_kl,
                     obs_dim=n_obs + n_features + env.action_space.shape[0],
                     act_dim=env.action_space.shape[0],
                     act_limit=0.5)
```

Experiment

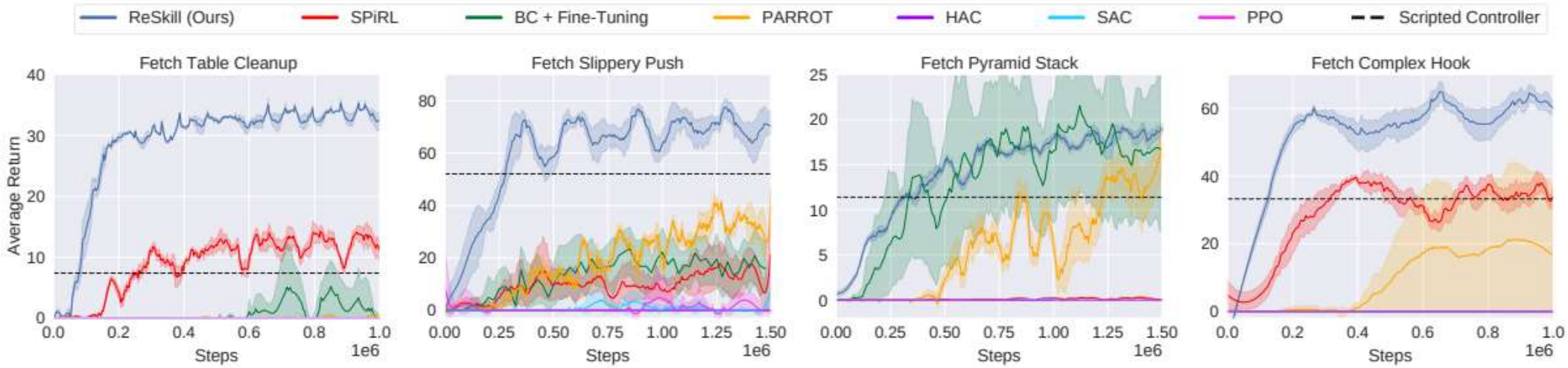


Figure 5: **Training Performance.** Average training performance across 5 random seeds for the different tasks. ReSkill outperforms all the baselines in both sample efficiency and convergence to the highest-performing final policy.

Experiment

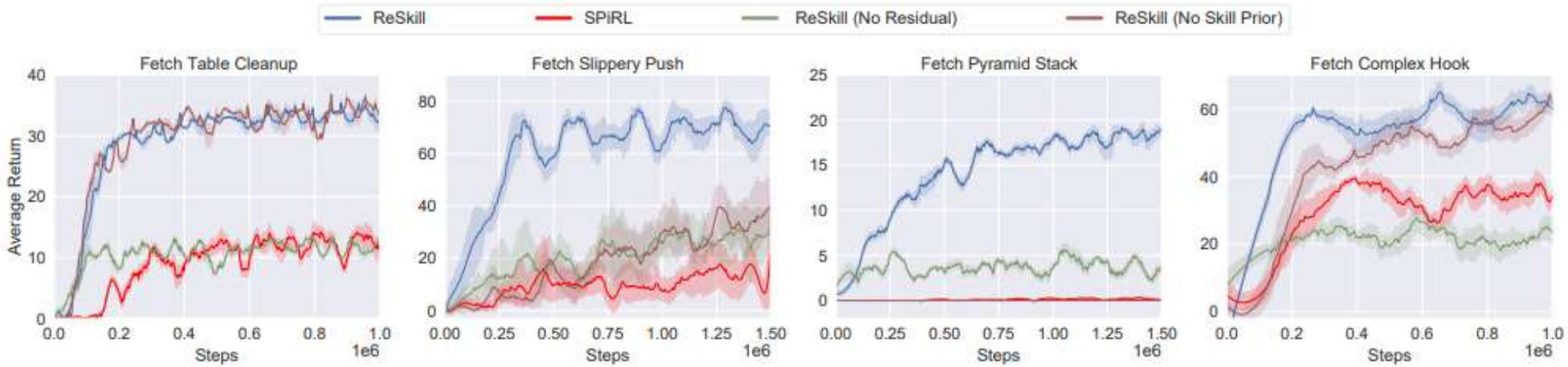


Figure 6: **Ablation Study.** Analysing the impact of the residual and skill prior on the average training performance across the 4 tasks. The skill prior plays an important role in accelerating learning with the more difficult tasks, while the residual is important for attaining higher rewards by adapting the skills to the variations in the training environment.

Experiment

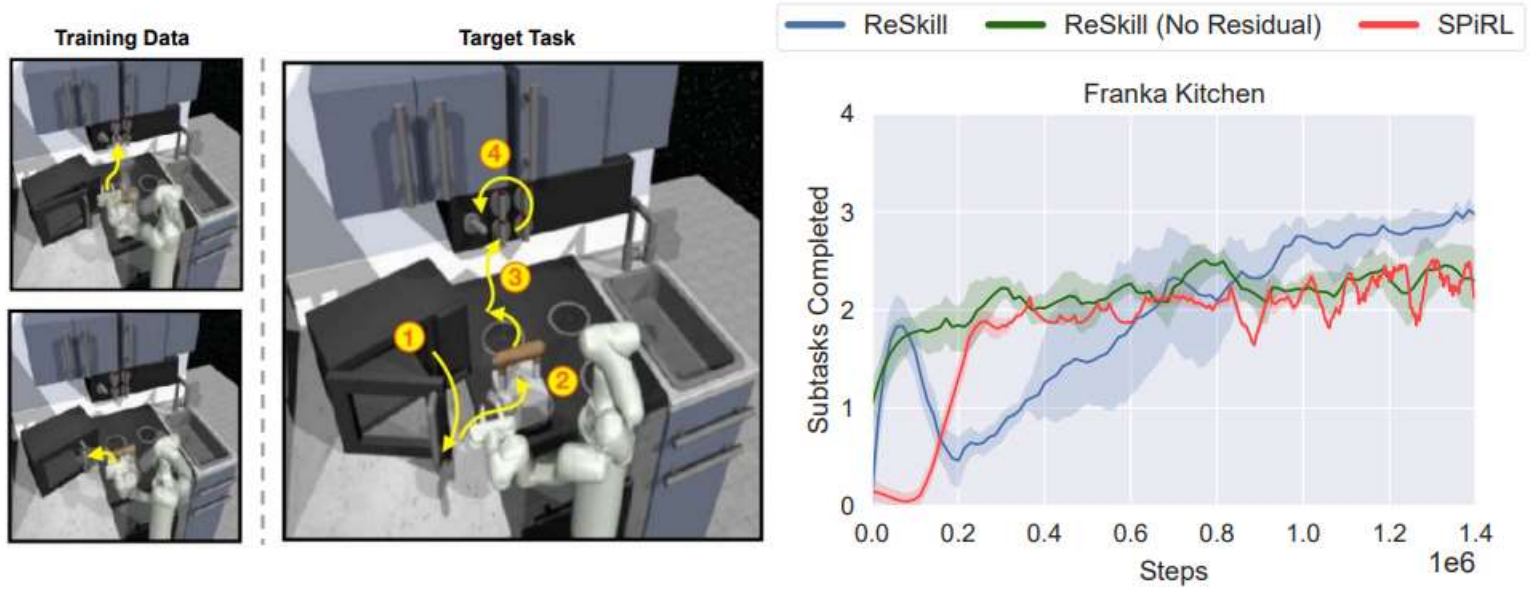


Figure 8: **Evaluation on Franka Kitchen Environment.** (**Left**) Franka Kitchen environment proposed by Gupta *et al.* [36] which requires the agent to manipulate a kitchen setup to reach a target configuration. (**Right**) Note how ReSkill exhibits faster convergence to a higher reward than SPiRL.

Some ideas

Skill z学的够好了可以推导出通过decoder解码状态s和skill z能够得到较准确的动作a

逆否命题

当前状态s不存在一个skill z能够还原出较准确的动作a可以推出当前轨迹并没有被完全学习过。

所以可以在此基础上加入主动学习以减少学习量。

Some ideas

			maze2d-umaze	maze2d-medium-25	maze2d-medium-24	maze2d-medium-23	maze2d-medium-22	maze2d-medium-21		
0	Method	Uncertainty	738(729-750)	618.6(604-663)	619.8(611-637)	671.4(657-703)	653(628-663)	630.6(600-661)		红色-最大值
		Random	732.4(727-750)	49.4(42-73)	587.8(559-672)	646(594-709)	660.8(619-693)	654(609-708)		蓝色-双最大
		Total	723(708-750)	638(613-667)	221.8(203-244)	606.8(576-642)	609.4(580-648)	635.4(600-688)		紫色-均值最大
1	Method	Uncertainty		639(600-683)	692.8(660-727)	544.2(486-679)	674.2(652-708)	565.8(555-600)		
		Random		633.2(611-676)	567(420-687)	338.6(300-488)	612.4(591-646)	577.2(552-599)		
		Total		255.2(247-277)	113.8(96-151)	599(584-613)	650.8(609-722)	614.6(600-630)		

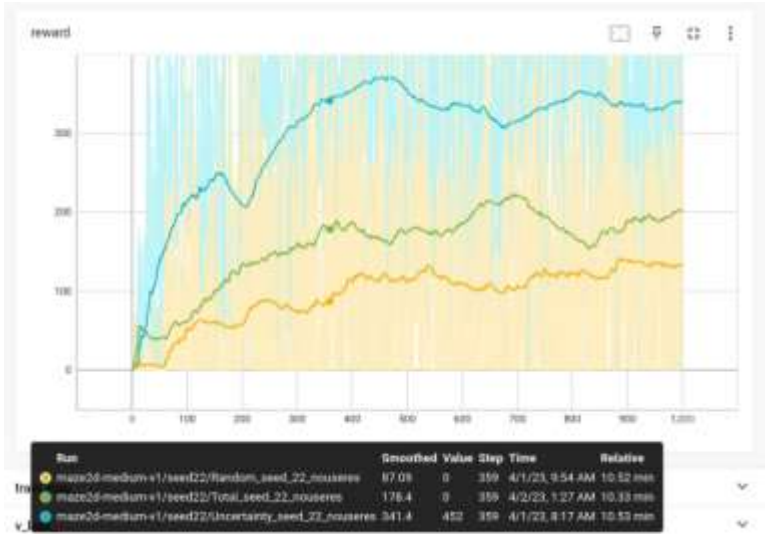
			maze2d-large-21	maze2d-large-22	maze2d-large-23	maze2d-large-24	maze2d-large-25
0	Method	Uncertainty	402.4(362-470)	117(85-156)	314.8(258-492)	473.2(421-555)	264(208-300)
		Random	209.6(160-309)	417.8(373-550)	219.8(207-244)	41.2(39-44)	268.4(170-447)
		Total	59.4(52-74)	27.8(21-36)	230(184-293)	150.6(71-223)	247.2(215-300)
1	Method	Uncertainty					
		Random					
		Total					

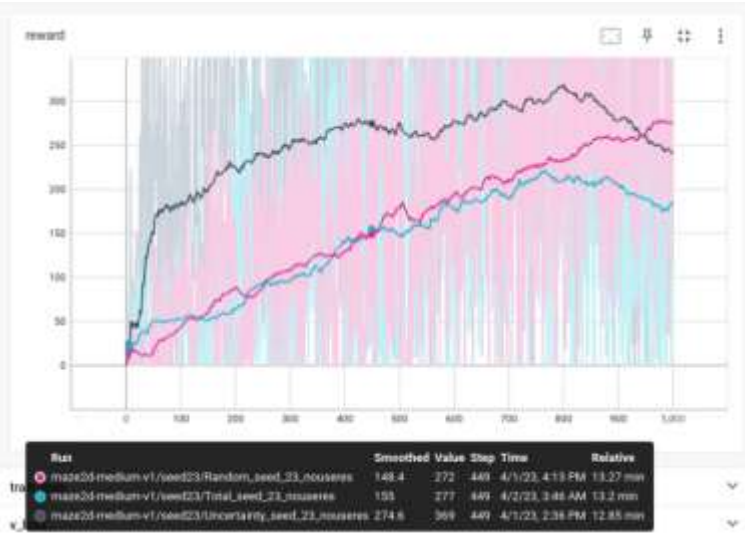
			hopper-random-21	hopper-expert-medium-2	halfcheetah-random-21	walker2d-expert-medium-21	walker2d-random-21	halfcheetah-medium-expert-21
0	Method	Uncertainty	1552(1522-1607)	2644(2632-2683)	226(225-228)	2795(2764-2834)	1619.6(1604-1651)	3,263.4(3175-3458)
		Random	1378(1360-1394)	2485(2407-2664)	114.4(109-127)	2428.8(2362-2499)	1360.8(1346-1378)	3501(3461-3579)
		Total	1543(1523-1580)	2657(2644-2671)	303(301-309)	2600(2536-2702)	1394.4(1382-1438)	3457(3345-3613)
1	Method	Uncertainty						
		Random						
		Total						

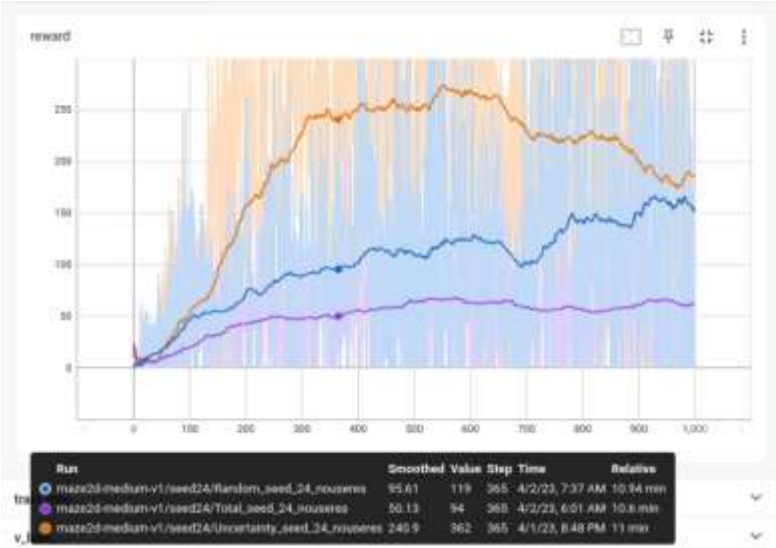
			kitchen-partial-21	kitchen-mixed-21	kitchen-complete-21
0	Method	Uncertainty	5(5-5)	5(5-5)	2.4(2-3)
		Random	4.2(4-5)	4.6(4-5)	6(6-6)
		Total	6(6-6)	3.4(3-4)	4.6(4-5)
1	Method	Uncertainty			
		Random			
		Total			

Some ideas


reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed21/Random_seed_21_nouseres 27 27 0 4/1/23, 3:26 AM 0
maze2d-medium-v1/seed21/Total_seed_21_nouseres 66 66 0 4/1/23, 10:59 PM 0
maze2d-medium-v1/seed21/Uncertainty_seed_21_nouseres 0 0 0 4/1/23, 1:48 AM 0

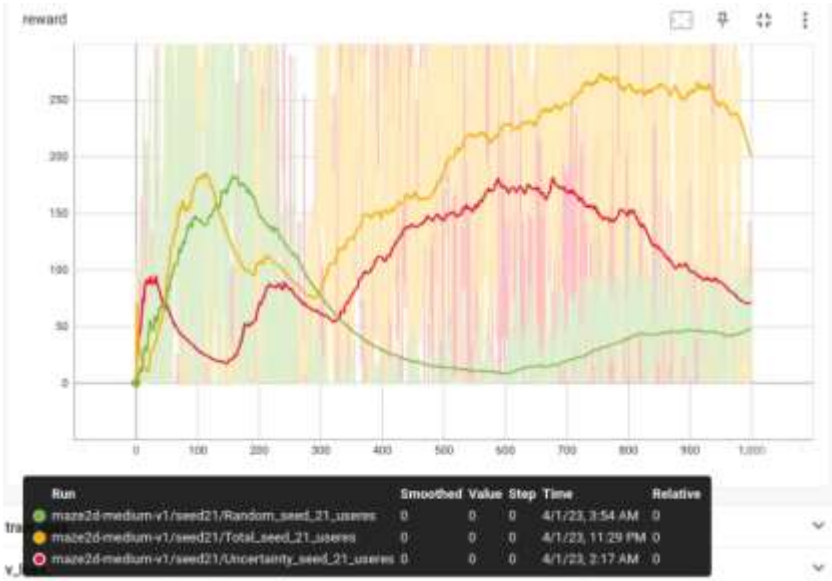

reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed22/Random_seed_22_nouseres 87.09 0 359 4/1/23, 9:54 AM 10.52 min
maze2d-medium-v1/seed22/Total_seed_22_nouseres 178.4 0 359 4/2/23, 1:27 AM 10.33 min
maze2d-medium-v1/seed22/Uncertainty_seed_22_nouseres 341.4 452 359 4/1/23, 8:17 AM 10.53 min


reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed23/Random_seed_23_nouseres 148.4 272 449 4/1/23, 4:13 PM 13.27 min
maze2d-medium-v1/seed23/Total_seed_23_nouseres 155 277 449 4/2/23, 3:46 AM 13.2 min
maze2d-medium-v1/seed23/Uncertainty_seed_23_nouseres 274.6 369 449 4/1/23, 2:36 PM 12.85 min

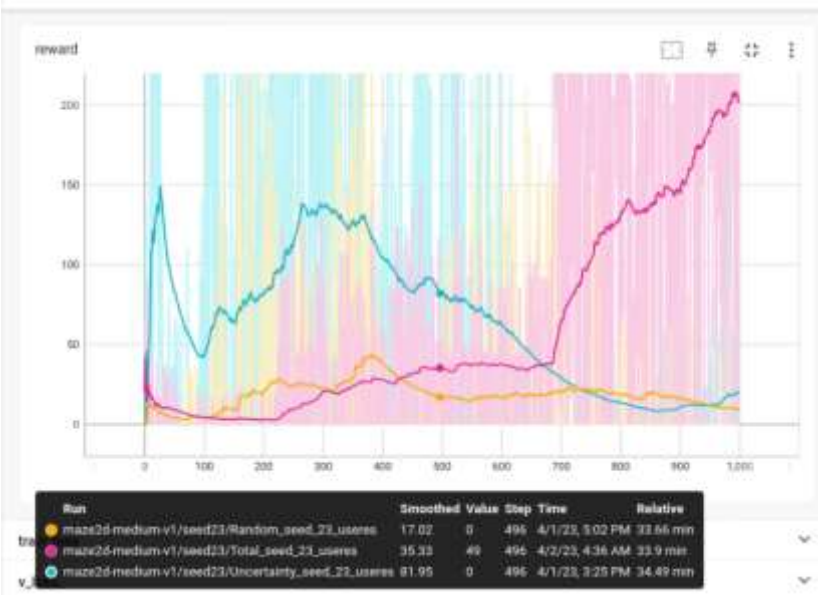

reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed24/Random_seed_24_nouseres 95.61 119 365 4/2/23, 7:37 AM 10.94 min
maze2d-medium-v1/seed24/Total_seed_24_nouseres 50.13 94 365 4/2/23, 6:01 AM 10.8 min
maze2d-medium-v1/seed24/Uncertainty_seed_24_nouseres 240.9 362 365 4/1/23, 8:48 PM 11 min

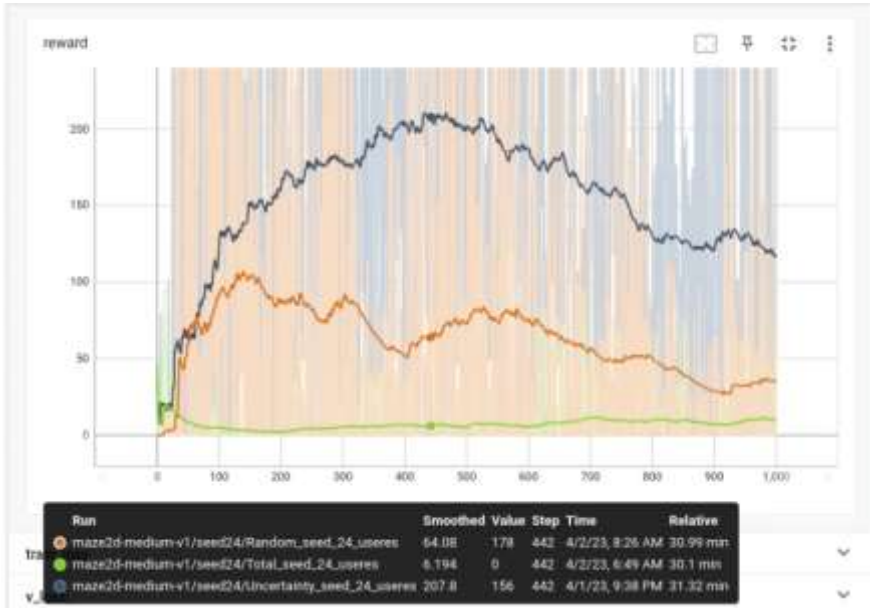

reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed25/Random_seed_25_nouseres 2.188 0 267 4/2/23, 1:50 PM 7.657 min
maze2d-medium-v1/seed25/Total_seed_25_nouseres 204.1 497 267 4/2/23, 10:36 AM 7.815 min
maze2d-medium-v1/seed25/Uncertainty_seed_25_nouseres 223.8 0 267 4/2/23, 12:14 PM 8.033 min

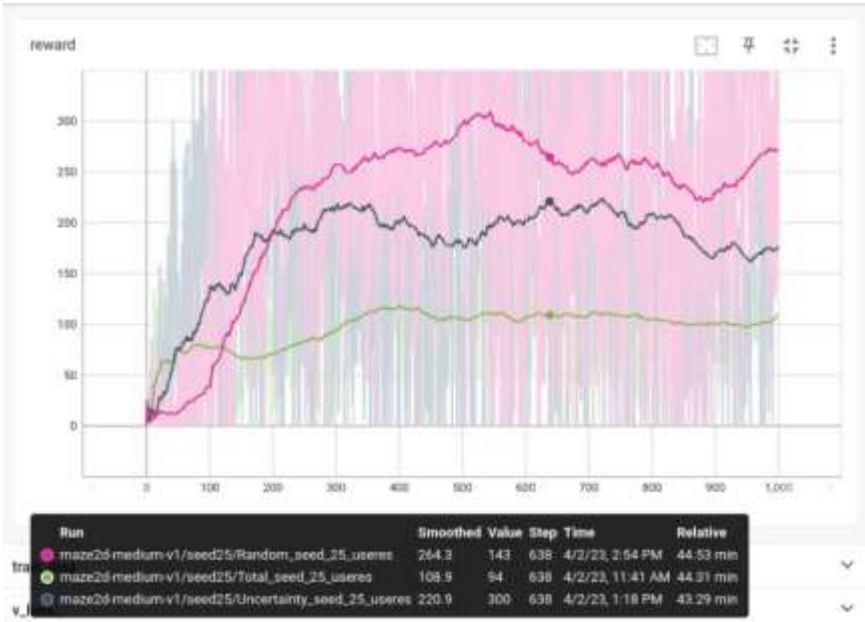
Some ideas


reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed21/Random_seed_21_useres 0 0 0 4/1/23, 3:54 AM 0
maze2d-medium-v1/seed21/Total_seed_21_useres 0 0 0 4/1/23, 11:29 PM 0
maze2d-medium-v1/seed21/Uncertainty_seed_21_useres 0 0 0 4/1/23, 2:17 AM 0


reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed22/Random_seed_22_useres 298.9 449 444 4/1/23, 10:42 AM 30.62 min
maze2d-medium-v1/seed22/Total_seed_22_useres 195.3 0 444 4/2/23, 2:15 AM 30.4 min
maze2d-medium-v1/seed22/Uncertainty_seed_22_useres 128.2 176 444 4/1/23, 9:05 AM 30.45 min


reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed23/Random_seed_23_useres 17.02 0 496 4/1/23, 5:02 PM 33.66 min
maze2d-medium-v1/seed23/Total_seed_23_useres 35.33 49 496 4/2/23, 4:36 AM 33.9 min
maze2d-medium-v1/seed23/Uncertainty_seed_23_useres 81.95 0 496 4/1/23, 3:25 PM 34.49 min


reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed24/Random_seed_24_useres 64.08 178 442 4/2/23, 8:26 AM 30.99 min
maze2d-medium-v1/seed24/Total_seed_24_useres 6.194 0 442 4/2/23, 6:49 AM 30.1 min
maze2d-medium-v1/seed24/Uncertainty_seed_24_useres 207.8 156 442 4/1/23, 9:38 PM 31.32 min


reward
Run Smoothed Value Step Time Relative
maze2d-medium-v1/seed25/Random_seed_25_useres 264.3 143 638 4/2/23, 2:54 PM 44.53 min
maze2d-medium-v1/seed25/Total_seed_25_useres 108.9 94 638 4/2/23, 11:41 AM 44.31 min
maze2d-medium-v1/seed25/Uncertainty_seed_25_useres 220.9 300 638 4/2/23, 1:18 PM 43.29 min

Some ideas

生成模型的长尾问题：流模型相当于将一个分布映射到另一个简单分布上，那么在状态s上存在不同的skill，一个skill z1学习的次数相较于skill z2更多，那么应该会导致逆向过程中skill z1被采样的概率更高，而对于下游任务不同的情况时，我们不希望发生这种带有因为数据集本身而产生的一种策略偏向问题（就是这个skill的采样的过程已经出现了向某一个skill倾斜的情况）。
所以相同环境的不同下游任务这种设定下，我们更希望当前状态skill z空间的采样概率是公平的，从而适应于不同的下游任务。目前想到就是加入上下采样的办法或者是reweight。

Skill Expand：AdaptDiffuser提出了通过不同的外在驱动驱动扩散模型生成理想的轨迹，在思考可否使用在skill expansion上，有限数据集学到的skill终归是不能完全遍历到下游任务的，那么是否可以通过无监督的方法让扩散模型生成一些新的skill，然后加入到目前的skill库中。

解耦skill的运行时长：这个在goal-conditioned中被图建模已经很好地解决了，但是在skill部分理论上也可以通过建图实现，《VMP》论文提出了整段轨迹建图的解耦，目前还在思考有没有更为简单的方法来解耦skill。

Thanks