Supplementary Material

1 Formulations for Algorithm

This section serves as an introduction for the underlying formulation of the algorithm. Since we use a meta-learning algorithm to train the programmatic policy, the training procedure consists of an outer loop and an inner loop. We use Reptile as the meta-learning framework to update the parameter in outer loop. In inner loop, we use Proximal Policy Optimization (PPO) as the basic reinforcement learning (RL) algorithm to learn "fast" parameter.

In our method, we use PPO-Clip, which utilizes a clipping mechanism in the objective function to remove incentives that may prompt a new policy towards diverging greatly from an existing policy. The PPO-Clip algorithm does not incorporate a KL-divergence term in its objective function. The PPO-Clip objective is:

$$\begin{aligned}\theta_{k+1} &= \arg\max_{\theta} J(\theta) \\ &= \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|\,T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\mathcal{P}_\theta\left(a_t \mid s_t\right)}{\mathcal{P}_{\theta_k}\left(a_t \mid s_t\right)} A^{\mathcal{P}_{\theta_k}}\left(s_t, a_t\right), \quad g\left(\epsilon, A^{\mathcal{P}_{\theta_k}}\left(s_t, a_t\right)\right)\right)\end{aligned}$$

, where

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1+\epsilon)A & A < 0 \end{cases} .$$

To illustrate, $\mathcal{D}_k$ represents a set of trajectories by running the policy $\mathcal{P}_{\theta_k}$ in the environment. The $A^{\mathcal{P}_{\theta_k}}$ is the advantage estimation based on the value function and the hyperparameter ϵ corresponds to how far away the new policy can go from the old while still profiting the objective. The θ is the parameter for the programmatic policy $\mathcal{P}_\theta$, which indeed are two parameters $\theta = (\mathcal{W}, \varphi)$. The $\mathcal{W}$ represents the policy architecture as well as the φ is the parameter for policy actions. These two parameters are jointly optimized by PPO. The train process is a bilevel iterative optimization process. At each iteration k of training, we carry out two steps. During the first step, we optimize the φ while preserving the frozen architecture parameter weights $\mathcal{W}$:

$$\varphi_{k+1} = \arg\max_{\varphi} J(\mathcal{W}_k, \varphi)$$

Next, during the second step, we optimize the $\mathcal{W}$ by fixing φ:

$$\mathcal{W}_{k+1} = \arg\max_{\mathcal{W}} J(\mathcal{W}, \varphi_k)$$

The bilevel training steps alternate throughout the training iterations until the reward converges. Then we choose the optimal $\mathcal{W}$ to freezing the architecture of the programmatic policy. Finally, we train the parameters of the selected architecture, continuously until the parameter values learned by means of RL converge to $\widehat{\theta}$.

In the outer loop, the "fast" parameter is used for updating the "slow" parameter. Multiple gradient descents are applied on each task in order to obtain the corresponding parameter value $\widehat{\theta}_i$. Afterwards, the difference vector between the parameters of each task and the main task is calculated as the update direction:

$$\theta_{i+1} \leftarrow \beta(\theta_i - \widehat{\theta}_i)$$

, where β is the meta learning rate. By repeating this process iteratively, the global initialized parameters are finally obtained. Intuitively, by using the gradient of a single task's parameter as the rough direction of gradient descent and decreasing the total loss function of the training tasks roughly but steadily, one can often obtain a good initialization parameter.

2 Implementation Details

In this section, we present the details of implementation, which cover the input and reward structures of each environment, architecture design and training hyperparameters.

2.1 Environment Details

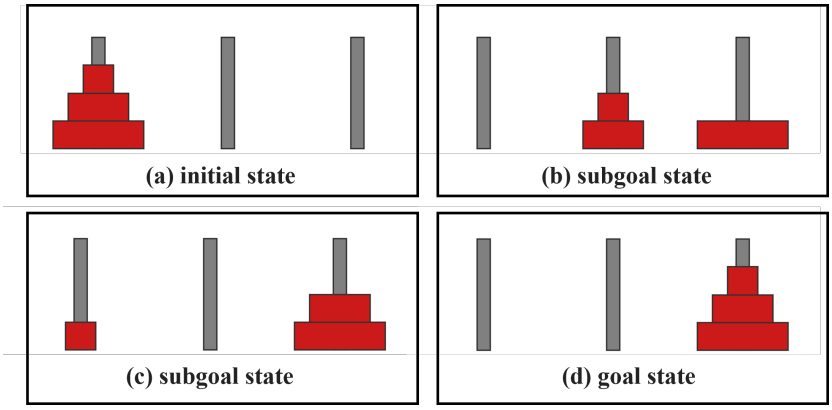


Figure 1: Details for sub-goals in hanoi

Hanoi In the Tower of Hanoi environment, there are three adjacent pillars labeled A, B, C, and several different-sized discs stacked in a pyramid shape from bottom to top on pillar A. The objective is to move all the discs one by one to pillar B, but at no time

can a larger disc be placed on top of a smaller one on the same pillar. Each action is to move a disc from the current pillar to another pillar.

In the Tower of Hanoi environment, the variation is the number and size of discs. For different environment variations, the goal of the task is to move all the discs to another pillar. Based on the number of discs on each pillar, as well as the location of the discs, based on their size, we encode the input of this environment into a vector with a dimension of 1×9. Regardless of the number of discs, we can only set 6 actions, i.e., A to B, A to C, B to A, B to C, C to A, and C to B.

The Tower of Hanoi environment is a discrete environment with low input and output dimensions. However, completing such a task requires the agent to sequentially complete a series of actions, and a small number of mistakes in the action sequence can result in very low rewards for the whole task, or even the inability to complete it. Thus, we set some intermediate rewards for hanoi environment to assist the agent in learning the policy more easily. As shown in Figure 1, for a Tower of Hanoi environment with three discs, we set (b) and (c) as sub-goals, the environment returns a small reward when the agent accomplishes this for the first time. When the agent moves a disc once, the reward value is -1. When the agent performs an invalid action, such as placing a large disc on top of a small disc, the environment will judge that the action has failed, keep the environment state unchanged, and give a penalty of reward value -2 to the agent. When the agent completes the task, i.e. placing the discs in the state shown in (d), the environment gives a large reward value.

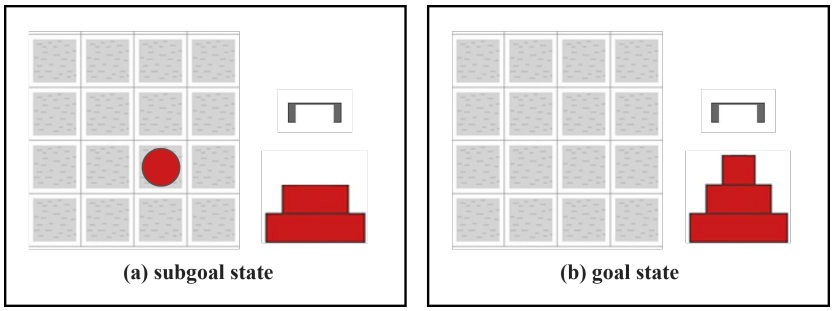


Figure 2: Details for sub-goals in stacking

Stacking In the Stacking environment, there are several differently-sized plates on the table, and a gripper that can move above the table. The goal is to collect them one by one using the gripper, putting them together in descending order of size.

In the Stacking environment, the variations of the environment are the number, size, and location of the plates. To simplify, we use a grid to represent the table. The gripper can move on the grid in up, down, left, and right directions, and can grab a plate. In this environment, we encode the table status, the stacked plates and the location of the gripper as the environment state, with a dimension of 1×35. The gripper can perform actions of moving up, down, left, and right and selecting a plate to stack and the action is a 5-dimension vector.

The Stacking environment is similar to the Tower of Hanoi environment, as the agent needs to control the gripper to select plates in a specific order to complete the task. To reduce the sparseness of rewards, we also set intermediate rewards. For example, as shown in (a) of Figure 2, when the agent first completes stacking two plates, it receives a small reward. Each action taken by the gripper results in a reward value of -1. When the gripper performs an incorrect action, such as moving out of the table or trying to stack plates of wrong size, the environment will judge that the action has failed. Then the environment state keeps unchanged, and give a penalty of -2 to the agent. When the agent completes the task, i.e. stacking the plates in the state shown in (b), the environment gives a positive final reward.

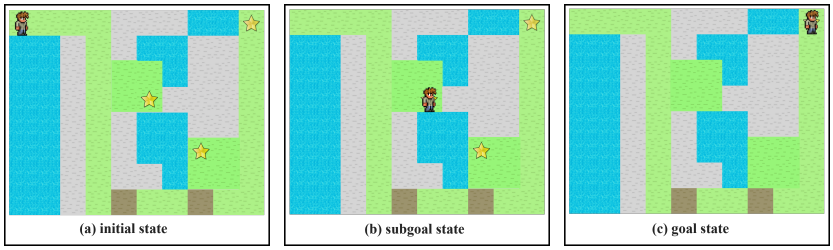


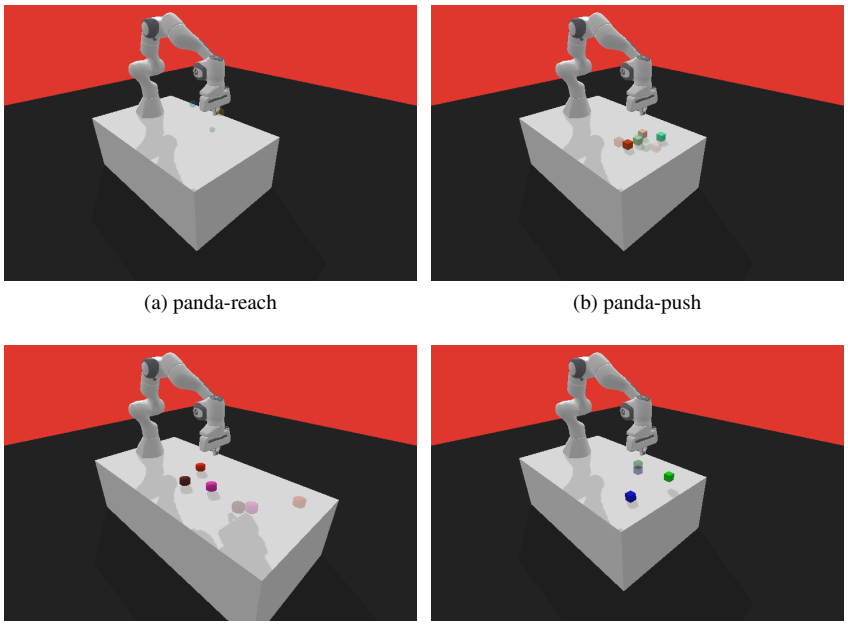
Figure 3: Details for sub-goals in hiking

Hiking In the Hiking environment, the characters on the map represent the agent and the stars on the map are the targets that the agents need to collect. The character can move on the ground, with green and gray blocks indicating walkable ground and blue water blocks representing obstacles that the character cannot pass through. The goal in this environment is to make the character move step by step and collect all the stars on the map.

The variations in this environment lies in the the number and position of the stars. The environment is also discretized into a grid. The entire state of the map is encoded as the state of the environment, with the dimension of 6×7. The character, ground, and water on the map are all marked with different symbols. The character's actions are up, down, left, and right, so the dimension of the action vector for the environment is 1×4.

In the Hiking environment, we also set up intermediate rewards, as shown in (b) of Figure 3, where the environment gives the agent a intermediate reward when the character moves to a star and the star disappears from the map. Every time the character take an action in this environment, the environment returns a reward of -1. When the character moves out of the map boundary or into the water, the environment judges the action as a failure and keeps the entire state unchanged, while giving a punishment of reward value -2 to the agent. The final state is shown as in (c).

PandaReach,PandaPush,PandaSlide,PandaStack PandaReach, PandaPush, PandaSlide and PandaStack are designed for robot manipulation, as in Figure 4. A Panda robot, equiped with a robot arm, is placed on a control console with a series of manipulable



(a) panda-reach (b) panda-push

(c) panda-slide (d) panda-stack

Figure 4: Benchmarks for continuous tasks

objects. The robot is required to manipulate the end-effector to finish the specified tasks. For example, in the environment of PandaPush, the robot needs to take actions to move the blocks to the target positions with the corresponding color. Once all the blocks are placed in the target positions, the task is completed.

The variations in this environment are derived from the number and position of the manipulable objects. Since the four environments have similar scenario, we consider the observation states uniformly. Through the integration of the position and velocity of the end-effector with the position, rotation, velocity, angluar velocity of the manipulable objects, we encode the observation state as a 1×32 vector. The action of the robot is a 4-dimension vector, which contains a 3D coordinate for the end-effector and a finger control.

We set up intermediate rewards to improve the training. When one of the objects is placed to the target position, a positive return is provided. Each move of the robot will obtain a penalty return from the environments.

2.2 Hyperparameters

Except the programmatic policy, all the actor networks and value networks are constructed with a 3-layer linear networks. The input layer is used to process the input of environment state. The hidden size of the middle layer is 64. The dimension of the output layer is determined by the action space of the environment. For continuous tasks,

we use Gaussian distribution to construct the output layer. Following hyperparameters are used to train PPO algorithm.

- max train steps $2e6$.
- replay buffer of size 2048.
- mini-batch size 64.
- discount factor 0.99.
- Adam optimizer; actor learning rate $3e-4$; critic learning rate $3e-4$.
- clip parameter ϵ 0.2.
- Entropy coefficient 0.01.
- Max gradient norm 0.5.
- KL-Divergence limit 0.03.

Following hyperparameters are used to train SAC algorithm.

- max train steps $2e6$.
- replay buffer of size 512.
- mini-batch size 64.
- discount factor 0.99.
- tau 0.005.
- Adam optimizer; actor learning rate $2e-4$; critic learning rate $2e-4$.
- clip parameter ϵ 0.2.
- entropy coefficient 0.01.
- max gradient norm 0.5.
- KL-divergence limit 0.03.

Following hyperparameters are used to train ReptilePPO algorithm.

- max train steps 5e6.
- meta iterations 250.
- inner iterations 1.
- max train steps 2e6.
- replay buffer of size 2048.

- mini-batch size 64.
- discount factor 0.99.
- Adam optimizer; actor learning rate 2e-2; critic learning rate 2e-2.
- Adam optimizer; actor learning rate 1e-5; critic learning rate 1e-5.
- clip parameter ϵ 0.2.
- entropy coefficient 0.01.
- max gradient norm 0.5.
- KL-divergence limit 0.03.

The components of the programmatic policy are all linear cells. Following hyperparameters are used to train ReptilePRL algorithm as well as PRL.

- depth 6.
- max train steps 5e6.
- meta iterations 250.
- inner iterations 1.
- max train steps 5e6.
- replay buffer of size 2048.
- batch size
- mini-batch size 64.
- discount factor 0.99.
- Adam optimizer; actor learning rate 2e-2; critic learning rate 2e-2.
- Adam optimizer; actor learning rate 1e-4; critic learning rate 1e-4.
- clip parameter ϵ 0.2.
- entropy coefficient 0.01.
- max gradient norm 0.5.
- KL-divergence limit 0.03.

In our method, compared with ReptilePRL algorithm, some of the components are GRU layers. And in the updating process, the state of the RNN will be preserved across different episodes. Besides, due to the use of RNN, we have slightly changed the structure of the replay buffer. Following hyperparameters are used to train our method.

- depth 6.

- max train steps 5e6.
- meta iterations 250.
- inner iterations 1.
- max train steps 2e6.
- replay buffer of size 500.
- batch size 16
- mini-batch size 2.
- Discount factor 0.99.
- Adam optimizer; actor learning rate 2e-2; critic learning rate 2e-2.
- Adam optimizer; actor learning rate 4e-4; critic learning rate 4e-4.
- clip parameter ϵ 0.2.
- entropy coefficient 0.01.
- max gradient norm 0.5.
- KL-divergence limit 0.03.

3 Comprehensive Experimental Results

This section aims to present the comprehensive experimental results. In table 1 and table 2, we evaluated the PPO, SAC, HIRO, PRL, ReptilePPO, ReptilePRL and Our approach in seven benchmarks—Hanoi, Hiking, Stacking for task planning and PandaReach, PandaPush, PandaSlide and PandaStack for robot manipulation. The number of steps for the agents is utilized as the performance metric. A smaller number of action steps indicates better performance. The performance is measured by mean number of steps an agent takes to achieve goals, along with standard deviations. The best score is marked with bold font.

To conclude, our method basically showcases the best performance across the benchmarks. In some tasks, the best scores are achieved by ReptilePPO or ReptilePRL. It means that the policies trained by a meta-learning algorithm are capable of generalizing to different tasks in one task domain. In addition, when evaluated in the simple situations, such as Hiking1, Stacking1 or PandaReach1, the policies trained by SAC, HIRO or PRL are evaluated as the best scores. However, our method performs better in more complex environments according to the table. This indicates that our method has the ability to synthesize policies that show strong generalization in complex tasks. Even when confronted with unseen tasks from the same task domain, the policy learned by our method achieves high scores. This is most likely because the policy can capture reusable subroutines and states patterns.

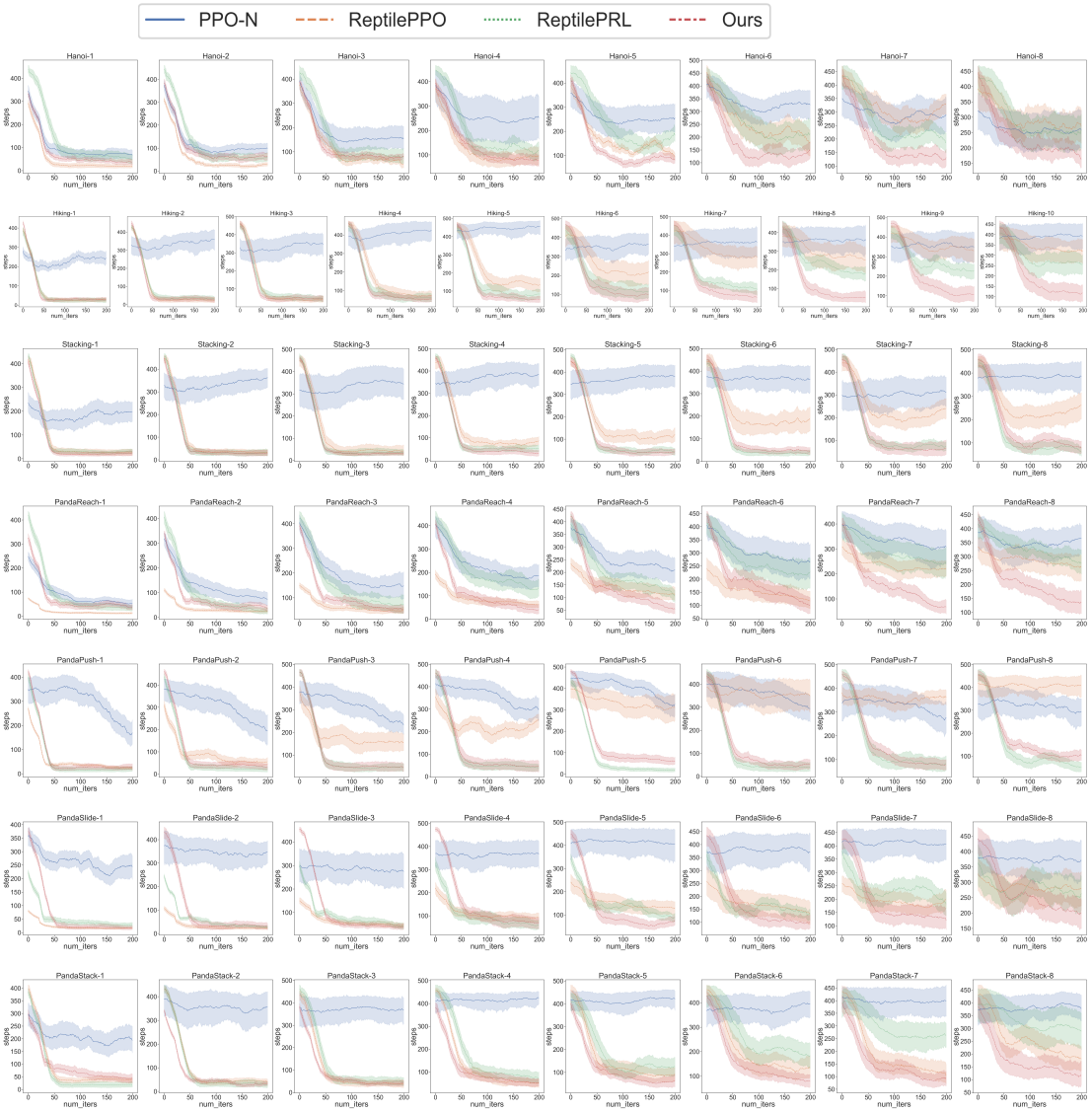


Figure 5: Comparison of the training curve for **PPO-N**, **ReptilePPO**, **ReptilePRL** and **Ours** in the seven benchmarks. Results are averaged over 10 random seeds.

The comprehensive results of the ablation study are illustrated as Figure 5. The vertical axis represents the number of steps required for the agent to achieve its objective, while the horizontal axis denotes the number of iterations involving agent-environment interaction episodes.

The policies learned by PPO-N struggle to achieving convergence in all benchmarks which indicates the necessity of a meta-learning algorithm. Overall, ReptilePRL

and ReptilePPO perform well under different benchmarks respectively, according to Figure 5. Overall, our approach performs better than ReptilePRL and ReptilePPO in most benchmarks. The more complex the environment, the better our methods perform. In unseen tasks, the ability to learn and reuse subroutines becomes very important. Our proposed idea of the combination of a meta-learning framework with a designed programmatic policy and the utilization of RNN blocks in policy architecture, is very useful.

test envs	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL
	train envs: Hanoi-1				train envs: Hanoi-2				train envs: Hanoi-3			
Hanoi-1	212	176	284	223	53	82	8	**7**	27	37	19	217
Hanoi-2	331	271	394	379	102	137	**24**	27	100	119	53	384
Hanoi-3	444	329	481	448	165	137	81	110	187	241	95	423
Hanoi-4	466	413	491	475	228	227	153	222	247	267	153	483
Hanoi-5	491	424	499	487	271	284	231	294	290	257	223	493
Hanoi-6	493	402	500	500	345	288	319	355	342	299	251	499
Hanoi-7	499	427	499	497	379	311	383	412	386	387	311	500
Hanoi-8	500	370	499	500	412	345	381	434	399	343	366	500

test envs	PPO	SAC	HIRO	PRL	ReptilePPO	ReptilePRL	Ours
	train envs: Hanoi-4				train envs: Hanoi-1234		
Hanoi-1	16	20	180	53	15	26	25
Hanoi-2	58	73	315	102	**24**	40	35
Hanoi-3	116	139	374	165	52	66	**51**
Hanoi-4	217	211	395	228	98	102	**41**
Hanoi-5	252	260	434	271	150	128	**98**
Hanoi-6	308	342	466	345	174	160	**113**
Hanoi-7	349	360	467	379	250	258	**151**
Hanoi-8	419	345	478	412	256	283	**187**

test envs	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL
	train envs: Hiking-1				train envs: Hiking-2				train envs: Hiking-3			
Hiking-1	21.07 ± 13.92	25.33 ± 15.34	21.49 ± 14.06	21.52 ± 12.92	$\mathbf{12.29 \pm 7.85}$	14.03 ± 8.28	24.24 ± 14.61	72.43 ± 30.98	34.39 ± 20.58	34.74 ± 20.56	34.90 ± 21.36	13.92 ± 10.65
Hiking-2	224.35 ± 77.96	240.23 ± 78.74	206.04 ± 73.67	295.30 ± 82.92	38.26 ± 18.38	$\mathbf{21.41 \pm 15.20}$	52.78 ± 26.40	139.54 ± 53.61	121.17 ± 54.23	138.73 ± 53.38	105.56 ± 51.98	34.83 ± 24.54
Hiking-3	380.65 ± 81.55	311.43 ± 83.87	306.02 ± 86.98	357.43 ± 74.99	137.72 ± 49.15	137.28 ± 55.24	132.88 ± 58.79	197.81 ± 73.00	210.65 ± 61.66	220.81 ± 80.06	202.00 ± 57.02	117.88 ± 49.32
Hiking-4	374.77 ± 86.81	381.91 ± 85.01	319.57 ± 88.84	401.71 ± 79.83	227.94 ± 80.10	214.57 ± 76.21	227.00 ± 75.32	257.18 ± 83.75	252.40 ± 86.02	273.58 ± 89.51	213.98 ± 87.42	198.51 ± 77.97
Hiking-5	373.78 ± 85.43	338.88 ± 96.64	322.52 ± 94.46	402.69 ± 79.18	280.89 ± 89.26	275.07 ± 93.61	249.22 ± 79.19	306.33 ± 91.74	328.44 ± 66.49	268.17 ± 89.64	323.16 ± 61.49	267.31 ± 69.90
Hiking-6	367.13 ± 91.95	409.93 ± 81.68	409.23 ± 66.29	380.54 ± 85.24	360.78 ± 84.82	393.33 ± 70.38	282.02 ± 90.82	285.38 ± 80.83	321.71 ± 90.61	301.36 ± 96.33	319.04 ± 88.74	316.29 ± 90.69
Hiking-7	364.89 ± 92.28	360.70 ± 87.07	342.16 ± 93.41	362.11 ± 93.44	322.85 ± 94.08	342.52 ± 92.80	323.37 ± 96.22	328.89 ± 90.11	305.85 ± 91.81	370.23 ± 73.04	290.12 ± 91.85	372.27 ± 77.96
Hiking-8	353.80 ± 97.65	385.44 ± 79.11	350.94 ± 89.63	379.63 ± 88.51	404.62 ± 78.48	350.28 ± 93.00	347.76 ± 94.19	350.99 ± 95.77	312.27 ± 89.26	368.32 ± 88.97	342.81 ± 92.32	348.86 ± 91.02
Hiking-9	347.96 ± 97.67	382.71 ± 84.60	388.65 ± 87.41	379.67 ± 86.09	372.46 ± 85.51	364.27 ± 87.87	406.73 ± 76.55	385.59 ± 87.32	309.51 ± 96.15	339.97 ± 92.74	328.09 ± 92.19	293.05 ± 93.05
Hiking-10	355.69 ± 95.04	360.78 ± 95.82	376.14 ± 84.47	376.78 ± 90.42	304.82 ± 103.10	385.12 ± 77.07	352.51 ± 94.55	419.59 ± 69.93	373.66 ± 63.78	414.39 ± 73.58	378.64 ± 62.93	385.74 ± 61.30

test envs	PPO	SAC	HIRO	PRL	ReptilePPO	ReptilePRL	Ours
	train envs: Hiking-4				train envs: Hiking-12345		
Hiking-1	50.28 ± 24.61	43.22 ± 23.07	91.03 ± 34.63	22.65 ± 12.91	17.48 ± 11.56	23.05 ± 15.74	21.43 ± 15.19
Hiking-2	98.18 ± 51.29	119.42 ± 55.93	93.88 ± 34.48	52.75 ± 24.98	31.99 ± 20.57	25.66 ± 17.45	31.14 ± 20.12
Hiking-3	190.83 ± 73.43	152.45 ± 72.71	113.21 ± 52.79	136.21 ± 56.51	$\mathbf{21.95 \pm 13.79}$	47.21 ± 18.84	32.18 ± 16.36
Hiking-4	253.99 ± 78.08	197.51 ± 85.07	209.76 ± 67.47	186.93 ± 79.02	107.11 ± 48.85	53.05 ± 29.52	$\mathbf{33.42 \pm 22.08}$
Hiking-5	249.21 ± 88.59	307.84 ± 82.31	242.99 ± 89.89	245.57 ± 82.69	129.46 ± 46.49	51.85 ± 24.45	$\mathbf{32.50 \pm 16.10}$
Hiking-6	295.35 ± 94.93	304.81 ± 88.85	286.80 ± 75.52	242.85 ± 86.73	190.41 ± 82.29	122.60 ± 57.10	$\mathbf{49.54 \pm 29.70}$
Hiking-7	328.49 ± 86.19	300.20 ± 91.93	351.31 ± 79.94	314.04 ± 92.06	263.69 ± 88.03	183.38 ± 66.13	$\mathbf{85.75 \pm 44.21}$
Hiking-8	307.88 ± 93.16	334.19 ± 75.55	286.90 ± 92.37	310.95 ± 97.28	310.40 ± 86.14	195.27 ± 70.20	$\mathbf{83.27 \pm 41.18}$
Hiking-9	406.20 ± 68.53	382.85 ± 74.61	362.57 ± 90.48	445.42 ± 55.05	278.41 ± 93.47	214.42 ± 82.90	$\mathbf{74.61 \pm 49.33}$
Hiking-10	391.34 ± 83.86	380.98 ± 75.06	435.03 ± 54.62	408.69 ± 69.59	325.13 ± 65.63	281.65 ± 51.50	$\mathbf{67.86 \pm 31.07}$

test envs	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL
	train envs: Stacking-1				train envs: Stacking-2				train envs: Stacking-3			
Stacking-1	90.41 ± 41.76	87.51 ± 44.45	19.51 ± 11.43	166.28 ± 73.29	71.22 ± 32.48	74.28 ± 28.20	27.21 ± 18.82	26.19 ± 14.26	25.77 ± 17.88	32.08 ± 17.80	14.20 ± 10.46	19.37 ± 12.85
Stacking-2	240.60 ± 72.85	169.45 ± 68.58	47.80 ± 25.53	266.71 ± 94.60	126.27 ± 45.95	101.49 ± 44.17	48.74 ± 28.97	79.51 ± 41.77	46.87 ± 25.50	48.18 ± 26.84	32.19 ± 18.09	104.42 ± 40.26
Stacking-3	289.80 ± 84.97	248.66 ± 72.74	135.28 ± 53.78	425.83 ± 57.79	162.87 ± 52.82	144.99 ± 50.58	53.29 ± 35.30	126.83 ± 62.01	55.97 ± 26.02	80.80 ± 38.70	29.29 ± 15.58	114.96 ± 37.90
Stacking-4	345.10 ± 76.79	264.50 ± 93.31	231.37 ± 80.40	425.63 ± 68.32	225.72 ± 72.30	157.20 ± 66.69	136.09 ± 62.68	273.78 ± 86.69	87.34 ± 46.62	99.84 ± 47.91	64.03 ± 40.23	151.54 ± 71.11
Stacking-5	268.01 ± 95.69	279.98 ± 95.12	289.93 ± 92.26	364.23 ± 91.97	226.09 ± 83.10	262.06 ± 82.81	130.11 ± 62.32	333.28 ± 86.11	147.54 ± 49.16	168.47 ± 73.32	136.06 ± 45.22	226.06 ± 64.15
Stacking-6	326.05 ± 88.54	363.28 ± 78.66	337.54 ± 83.76	372.62 ± 90.63	267.98 ± 86.31	292.27 ± 90.25	197.81 ± 76.28	369.17 ± 83.43	201.48 ± 70.71	192.93 ± 74.46	170.47 ± 84.27	275.05 ± 91.59
Stacking-7	338.64 ± 88.71	307.31 ± 93.23	311.89 ± 98.22	354.78 ± 94.50	282.60 ± 91.27	293.55 ± 88.88	172.09 ± 77.38	333.48 ± 88.99	218.81 ± 83.58	226.31 ± 90.85	294.14 ± 85.79	290.35 ± 91.14
Stacking-8	318.77 ± 97.15	338.77 ± 93.58	368.02 ± 78.64	409.05 ± 76.63	329.78 ± 79.07	300.53 ± 96.64	334.58 ± 86.22	434.79 ± 50.15	267.47 ± 66.48	322.58 ± 88.01	317.98 ± 71.15	341.84 ± 71.79

test envs	PPO	SAC	HIRO	PRL	ReptilePPO	ReptilePRL	Ours
	train envs: Stacking-4				train envs: Stacking-1234		
Stacking-1	13.41 ± 6.25	$\mathbf{12.48 \pm 4.97}$	72.08 ± 37.10	26.41 ± 14.20	25.16 ± 16.05	15.79 ± 8.81	33.42 ± 22.08
Stacking-2	27.35 ± 12.62	32.31 ± 11.59	71.22 ± 41.71	29.12 ± 17.36	39.25 ± 23.56	29.87 ± 18.09	$\mathbf{20.64 \pm 16.32}$
Stacking-3	42.87 ± 18.41	58.60 ± 22.54	76.74 ± 45.34	86.70 ± 39.82	43.35 ± 19.50	23.40 ± 12.52	$\mathbf{22.29 \pm 10.96}$
Stacking-4	88.30 ± 32.15	74.59 ± 32.30	124.57 ± 61.31	141.40 ± 55.65	79.05 ± 42.99	$\mathbf{32.29 \pm 21.33}$	41.24 ± 25.39
Stacking-5	113.27 ± 43.54	127.87 ± 50.16	174.11 ± 78.26	187.73 ± 71.74	113.71 ± 33.91	$\mathbf{41.02 \pm 17.55}$	41.36 ± 18.70
Stacking-6	172.32 ± 54.59	183.84 ± 58.45	179.47 ± 82.31	162.43 ± 72.50	156.32 ± 67.52	63.60 ± 31.85	$\mathbf{53.00 \pm 30.06}$
Stacking-7	248.55 ± 64.18	180.77 ± 72.30	233.20 ± 86.79	274.93 ± 90.94	195.19 ± 78.27	$\mathbf{69.78 \pm 43.41}$	72.11 ± 40.49
Stacking-8	248.77 ± 76.45	276.84 ± 62.65	270.77 ± 90.77	240.78 ± 91.64	235.10 ± 63.27	72.77 ± 31.93	$\mathbf{61.44 \pm 28.07}$

Table 1: Performance comparison for evaluating policies in Hanoi, Hiking and Stacking, averaged over 5 random seeds.

test envs	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL
	train envs: PandaReach-1				train envs: PandaReach-2				train envs: PandaReach-3			
PandaReach-1	27.65 ± 15.34	26.02 ± 14.83	259.49 ± 69.33	59.21 ± 27.11	19.11 ± 10.50	**15.89 ± 10.10**	249.72 ± 76.67	48.72 ± 24.93	86.49 ± 51.67	80.27 ± 50.75	51.55 ± 24.74	29.31 ± 21.73
PandaReach-2	215.13 ± 76.61	243.50 ± 83.84	292.22 ± 86.25	151.35 ± 50.16	36.11 ± 18.80	33.70 ± 19.72	292.30 ± 82.68	48.87 ± 35.71	173.13 ± 87.27	139.91 ± 80.19	101.63 ± 42.69	69.56 ± 34.68
PandaReach-3	327.78 ± 85.91	318.04 ± 95.67	378.69 ± 77.13	221.06 ± 76.44	126.36 ± 50.36	132.29 ± 48.02	379.95 ± 74.64	116.90 ± 61.86	192.15 ± 66.30	242.49 ± 84.71	119.99 ± 38.16	103.89 ± 53.00
PandaReach-4	385.07 ± 90.15	359.42 ± 94.28	378.57 ± 79.84	211.68 ± 74.22	239.50 ± 76.91	236.62 ± 85.99	383.72 ± 86.40	155.11 ± 78.13	245.36 ± 88.47	230.88 ± 93.69	190.72 ± 78.33	202.67 ± 79.66
PandaReach-5	413.22 ± 77.28	381.32 ± 86.85	428.32 ± 67.35	335.10 ± 77.39	319.11 ± 87.23	321.82 ± 90.63	333.75 ± 98.73	230.23 ± 86.41	283.06 ± 74.24	222.15 ± 90.92	220.81 ± 61.86	195.31 ± 42.78
PandaReach-6	385.58 ± 89.73	405.11 ± 81.32	427.36 ± 65.08	374.57 ± 78.69	338.80 ± 83.93	374.19 ± 84.62	375.80 ± 84.43	302.89 ± 84.12	296.75 ± 88.41	290.60 ± 93.04	295.19 ± 86.74	268.13 ± 94.04
PandaReach-7	383.77 ± 90.65	456.76 ± 54.10	355.95 ± 91.07	367.29 ± 82.87	307.00 ± 94.53	362.29 ± 89.86	375.56 ± 84.34	270.14 ± 85.77	302.57 ± 98.42	362.70 ± 77.21	348.26 ± 83.02	276.22 ± 82.23
PandaReach-8	437.66 ± 55.39	336.20 ± 93.74	440.01 ± 68.43	351.70 ± 92.55	321.13 ± 96.69	346.57 ± 88.34	354.89 ± 94.26	223.86 ± 81.56	384.47 ± 65.41	313.26 ± 95.68	359.77 ± 62.23	440.76 ± 50.25

test envs	PPO	SAC	HIRO	PRL	ReptilePPO	ReptilePRL	Ours
	train envs: PandaReach-4				train envs: PandaReach-1234		
PandaReach-1	24.80 ± 13.82	27.86 ± 15.00	21.89 ± 15.55	168.77 ± 84.59	19.23 ± 6.60	27.54 ± 16.95	39.47 ± 20.23
PandaReach-2	42.46 ± 23.49	53.09 ± 23.55	30.21 ± 20.01	196.56 ± 91.42	**30.92 ± 14.35**	42.75 ± 24.31	36.66 ± 21.26
PandaReach-3	128.58 ± 46.00	127.28 ± 54.19	67.92 ± 34.47	283.42 ± 91.50	47.77 ± 12.48	70.57 ± 30.16	**40.35 ± 16.13**
PandaReach-4	211.02 ± 76.84	182.33 ± 80.03	97.12 ± 44.09	234.85 ± 93.90	$80.81 \textbackslash textbackslash\{\}pm25.04	136.00 ± 58.69	**57.72 ± 35.93**
PandaReach-5	281.16 ± 86.89	278.76 ± 82.84	147.22 ± 58.62	332.58 ± 88.90	96.40 ± 27.72	174.34 ± 50.19	**53.10 ± 25.06**
PandaReach-6	288.48 ± 89.67	313.39 ± 87.60	222.29 ± 78.94	332.22 ± 85.15	175.77 ± 62.65	220.98 ± 84.79	**84.58 ± 47.08**
PandaReach-7	331.08 ± 89.40	315.66 ± 89.36	210.20 ± 77.66	343.70 ± 82.83	233.20 ± 61.01	258.01 ± 86.91	**98.50 ± 51.62**
PandaReach-8	406.25 ± 69.44	376.30 ± 75.28	224.63 ± 89.15	336.01 ± 93.85	373 ± 52.69	347 ± 62.67	**125.16 ± 44.26**

test envs	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL
	train envs: PandaPush-1				train envs: PandaPush-2				train envs: PandaPush-3			
PandaReach-1	27.65 ± 15.34	26.02 ± 14.83	259.49 ± 69.33	59.21 ± 27.11	19.11 ± 10.50	**15.89 ± 10.10**	249.72 ± 76.67	48.72 ± 24.93	86.49 ± 51.67	80.27 ± 50.75	51.55 ± 24.74	29.31 ± 21.73
PandaReach-2	215.13 ± 76.61	243.50 ± 83.84	292.22 ± 86.25	151.35 ± 50.16	36.11 ± 18.80	33.70 ± 19.72	292.30 ± 82.68	48.87 ± 35.71	173.13 ± 87.27	139.91 ± 80.19	101.63 ± 42.69	69.56 ± 34.68
PandaReach-3	327.78 ± 85.91	318.04 ± 95.67	378.69 ± 77.13	221.06 ± 76.44	126.36 ± 50.36	132.29 ± 48.02	379.95 ± 74.64	116.90 ± 61.86	192.15 ± 66.30	242.49 ± 84.71	119.99 ± 38.16	103.89 ± 53.00
PandaReach-4	385.07 ± 90.15	359.42 ± 94.28	378.57 ± 79.84	211.68 ± 74.22	239.50 ± 76.91	236.62 ± 85.99	383.72 ± 86.40	155.11 ± 78.13	245.36 ± 88.47	230.88 ± 93.69	190.72 ± 78.33	202.67 ± 79.66
PandaReach-5	413.22 ± 77.28	381.32 ± 86.85	428.32 ± 67.35	335.10 ± 77.39	319.11 ± 87.23	321.82 ± 90.63	333.75 ± 98.73	230.23 ± 86.41	283.06 ± 74.24	222.15 ± 90.92	220.81 ± 61.86	195.31 ± 42.78
PandaReach-6	385.58 ± 89.73	405.11 ± 81.32	427.36 ± 65.08	374.57 ± 78.69	338.80 ± 83.93	374.19 ± 84.62	375.80 ± 84.43	302.89 ± 84.12	296.75 ± 88.41	290.60 ± 93.04	295.19 ± 86.74	268.13 ± 94.04
PandaReach-7	383.77 ± 90.65	456.76 ± 54.10	355.95 ± 91.07	367.29 ± 82.87	307.00 ± 94.53	362.29 ± 89.86	375.56 ± 84.34	270.14 ± 85.77	302.57 ± 98.42	362.70 ± 77.21	348.26 ± 83.02	276.22 ± 82.23
PandaReach-8	437.66 ± 55.39	336.20 ± 93.74	440.01 ± 68.43	351.70 ± 92.55	321.13 ± 96.69	346.57 ± 88.34	354.89 ± 94.26	223.86 ± 81.56	384.47 ± 65.41	313.26 ± 95.68	359.77 ± 62.23	440.76 ± 50.25

test envs	PPO	SAC	HIRO	PRL	ReptilePPO	ReptilePRL	Ours
	train envs: PandaPush-4				train envs: PandaPush-1234		
PandaReach-1	24.80 ± 13.82	27.86 ± 15.00	21.89 ± 15.55	168.77 ± 84.59	19.23 ± 6.60	27.54 ± 16.95	39.47 ± 20.23
PandaReach-2	42.46 ± 23.49	53.09 ± 23.55	30.21 ± 20.01	196.56 ± 91.42	**30.92 ± 14.35**	42.75 ± 24.31	36.66 ± 21.26
PandaReach-3	128.58 ± 46.00	127.28 ± 54.19	67.92 ± 34.47	283.42 ± 91.50	47.77 ± 12.48	70.57 ± 30.16	**40.35 ± 16.13**
PandaReach-4	211.02 ± 76.84	182.33 ± 80.03	97.12 ± 44.09	234.85 ± 93.90	$80.81 \textbackslash textbackslash\{\}pm25.04	136.00 ± 58.69	**57.72 ± 35.93**
PandaReach-5	281.16 ± 86.89	278.76 ± 82.84	147.22 ± 58.62	332.58 ± 88.90	96.40 ± 27.72	174.34 ± 50.19	**53.10 ± 25.06**
PandaReach-6	288.48 ± 89.67	313.39 ± 87.60	222.29 ± 78.94	332.22 ± 85.15	175.77 ± 62.65	220.98 ± 84.79	**84.58 ± 47.08**
PandaReach-7	331.08 ± 89.40	315.66 ± 89.36	210.20 ± 77.66	343.70 ± 82.83	233.20 ± 61.01	258.01 ± 86.91	**98.50 ± 51.62**
PandaReach-8	406.25 ± 69.44	376.30 ± 75.28	224.63 ± 89.15	336.01 ± 93.85	373 ± 52.69	347 ± 62.67	**125.16 ± 44.26**

test envs	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL
	train envs: PandaSlide-1				train envs: PandaSlide-2				train envs: PandaSlide-3			
PandaSlide-1	28.20 ± 13.28	18.91 ± 12.69	288.62 ± 75.97	37.01 ± 23.85	20.06 ± 10.04	29.98 ± 13.32	30.27 ± 19.23	66.27 ± 30.59	47.43 ± 30.00	36.90 ± 24.89	266.58 ± 76.84	27.26 ± 15.33
PandaSlide-2	58.26 ± 33.56	80.95 ± 36.60	339.02 ± 87.09	41.16 ± 25.15	37.50 ± 19.75	30.16 ± 17.92	114.75 ± 44.84	41.56 ± 23.67	52.02 ± 34.59	45.99 ± 33.52	308.12 ± 90.43	51.65 ± 28.61
PandaSlide-3	135.45 ± 60.88	142.03 ± 63.27	391.03 ± 81.02	57.32 ± 32.64	72.67 ± 37.02	70.56 ± 42.01	87.50 ± 45.06	32.25 ± 20.38	61.78 ± 30.15	92.05 ± 46.24	372.86 ± 73.99	68.66 ± 28.10
PandaSlide-4	200.21 ± 76.46	234.09 ± 77.04	416.52 ± 75.48	65.33 ± 33.32	127.72 ± 60.49	99.07 ± 52.51	217.21 ± 76.46	57.42 ± 14.06	114.93 ± 61.91	126.66 ± 61.35	392.21 ± 81.89	152.54 ± 59.26
PandaSlide-5	251.27 ± 89.95	285.72 ± 88.29	416.38 ± 76.80	127.04 ± 48.75	135.46 ± 65.42	188.21 ± 70.04	260.15 ± 84.39	57.41 ± 32.67	113.08 ± 53.65	156.47 ± 69.96	440.74 ± 52.91	145.43 ± 42.65
PandaSlide-6	301.79 ± 80.85	272.88 ± 86.94	391.17 ± 85.51	148.62 ± 63.47	140.73 ± 64.74	173.66 ± 71.34	228.19 ± 89.04	154.79 ± 28.47	199.08 ± 84.09	173.23 ± 74.28	363.80 ± 88.10	230.17 ± 82.54
PandaSlide-7	332.14 ± 91.35	316.47 ± 84.93	411.70 ± 74.88	215.84 ± 63.65	265.48 ± 85.95	197.65 ± 82.44	344.19 ± 77.89	167.88 ± 38.03	246.11 ± 84.88	190.72 ± 81.49	376.65 ± 85.56	256.05 ± 92.51
PandaSlide-8	350.20 ± 90.29	333.64 ± 88.65	379.52 ± 86.51	274.39 ± 82.01	255.69 ± 86.92	217.93 ± 86.01	338.82 ± 88.18	278.07 ± 38.64	257.03 ± 77.26	249.52 ± 89.30	449.24 ± 45.62	310.81 ± 58.87

test envs	PPO	SAC	HIRO	PRL	ReptilePPO	ReptilePRL	Ours
	train envs: PandaSlide-4				train envs: PandaSlide-1234		
PandaSlide-1	79.38 ± 38.19	89.66 ± 38.84	238.44 ± 77.46	21.58 ± 13.06	**15.60 ± 6.41**	27.82 ± 11.96	33.62 ± 17.78
PandaSlide-2	167.80 ± 68.47	212.48 ± 66.70	298.56 ± 89.42	63.16 ± 27.63	**25.80 ± 11.73**	32.13 ± 16.64	37.82 ± 23.57
PandaSlide-3	235.42 ± 79.34	229.04 ± 77.45	392.48 ± 77.42	140.13 ± 55.30	38.57 ± 11.99	39.46 ± 18.98	**34.69 ± 17.38**
PandaSlide-4	244.66 ± 95.79	309.07 ± 90.99	419.79 ± 67.22	200.98 ± 68.31	84.06 ± 29.41	**56.51 ± 30.67**	65.43 ± 33.46
PandaSlide-5	316.99 ± 85.03	320.15 ± 90.46	381.97 ± 86.44	275.11 ± 86.10	106.71 ± 26.51	74.42 ± 29.25	**50.64 ± 25.41**
PandaSlide-6	334.51 ± 96.76	292.15 ± 95.14	375.90 ± 90.22	298.33 ± 93.62	170.75 ± 54.21	189.16 ± 69.72	**103.82 ± 48.27**
PandaSlide-7	360.66 ± 90.63	369.12 ± 84.88	352.90 ± 93.66	368.90 ± 79.74	209.05 ± 81.18	241.69 ± 82.35	**123.84 ± 56.72**
PandaSlide-8	372.82 ± 84.24	352.18 ± 88.04	382.22 ± 88.55	331.24 ± 96.59	295.95 ± 60.06	266.47 ± 63.84	**196.67 ± 47.36**

test envs	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL	PPO	SAC	HIRO	PRL
	train envs: PandaStack-1				train envs: PandaStack-2				train envs: PandaStack-3			
PandaStack-1	18.51 ± 11.72	26.57 ± 13.89	68.65 ± 31.08	41.23 ± 21.30	47.31 ± 23.41	42.00 ± 19.18	55.11 ± 29.41	**12.38 ± 6.07**	64.19 ± 28.30	89.33 ± 30.85	27.19 ± 14.25	30.93 ± 19.27
PandaStack-2	33.07 ± 21.01	31.38 ± 20.28	122.88 ± 40.84	88.35 ± 44.02	94.36 ± 41.88	95.68 ± 38.91	126.14 ± 60.94	29.33 ± 10.48	100.90 ± 43.61	99.57 ± 44.02	45.05 ± 26.04	88.69 ± 40.87
PandaStack-3	77.03 ± 38.81	55.60 ± 32.02	166.99 ± 61.99	108.19 ± 60.36	146.68 ± 50.63	126.09 ± 53.61	231.46 ± 78.05	53.35 ± 20.45	146.32 ± 40.27	130.46 ± 55.15	117.07 ± 40.49	103.30 ± 51.75
PandaStack-4	99.52 ± 47.02	110.94 ± 50.15	215.32 ± 75.46	168.25 ± 75.44	177.36 ± 62.66	203.82 ± 74.24	268.39 ± 85.20	80.81 ± 29.79	196.21 ± 60.72	170.98 ± 55.88	193.45 ± 70.11	198.52 ± 81.79
PandaStack-5	118.75 ± 53.30	166.97 ± 75.06	302.34 ± 78.14	215.14 ± 81.03	261.17 ± 77.12	211.75 ± 76.90	231.34 ± 83.10	96.25 ± 42.24	236.55 ± 55.26	287.26 ± 76.10	229.11 ± 58.39	221.52 ± 69.39
PandaStack-6	188.44 ± 81.65	166.18 ± 77.84	354.07 ± 79.21	260.77 ± 93.72	327.35 ± 83.68	272.55 ± 79.04	281.51 ± 92.19	166.67 ± 61.63	282.34 ± 80.97	299.72 ± 79.16	303.17 ± 84.28	291.84 ± 93.64
PandaStack-7	204.04 ± 83.74	168.45 ± 73.69	362.35 ± 79.76	307.89 ± 92.03	311.71 ± 77.51	288.35 ± 87.40	318.28 ± 93.16	245.36 ± 62.64	377.54 ± 73.04	323.14 ± 90.35	345.98 ± 79.15	277.16 ± 92.98
PandaStack-8	285.36 ± 88.24	315.12 ± 80.11	348.83 ± 80.83	273.26 ± 91.78	376.39 ± 74.23	351.22 ± 91.58	318.83 ± 93.49	263.14 ± 71.82	352.95 ± 64.16	295.67 ± 86.04	377.76 ± 71.03	339.82 ± 64.12

test envs	PPO	SAC	HIRO	PRL	ReptilePPO	ReptilePRL	Ours
	train envs: PandaStack-4				train envs: PandaStack-1234		
PandaStack-1	224.64 ± 71.94	215.21 ± 69.50	250.58 ± 72.11	63.55 ± 43.35	25.66 ± 16.79	17.44 ± 12.46	39.22 ± 21.87
PandaStack-2	273.83 ± 92.81	368.94 ± 66.97	268.97 ± 92.94	109.66 ± 66.74	31.58 ± 20.78	**28.24 ± 16.85**	33.09 ± 20.77
PandaStack-3	366.77 ± 82.50	336.32 ± 86.70	365.44 ± 88.94	164.50 ± 72.75	30.75 ± 15.29	34.12 ± 17.39	**27.53 ± 13.76**
PandaStack-4	402.04 ± 74.27	412.57 ± 71.67	360.46 ± 88.96	232.70 ± 93.64	58.88 ± 34.41	98.83 ± 48.30	**54.05 ± 26.61**
PandaStack-5	417.08 ± 70.71	337.15 ± 92.44	398.27 ± 82.91	298.83 ± 94.19	62.60 ± 26.42	137.47 ± 51.21	**45.46 ± 20.12**
PandaStack-6	373.60 ± 90.82	410.29 ± 75.32	388.48 ± 75.17	264.19 ± 89.90	106.74 ± 49.96	152.70 ± 71.12	**95.46 ± 45.20**
PandaStack-7	377.04 ± 88.24	411.49 ± 74.72	344.39 ± 98.02	286.38 ± 91.96	123.16 ± 62.19	261.53 ± 82.34	**85.38 ± 38.75**
PandaStack-8	411.66 ± 81.44	377.94 ± 88.50	412.10 ± 69.61	305.32 ± 90.27	184.81 ± 47.94	279.55 ± 74.26	**132.67 ± 42.02**

Table 2: Performance comparison for evaluating policies in PandaReach, PandaPush, PandaSlide and PandaStackm, averaged over 5 random seeds.