Supplement to Meta Reinforcement Learning with Task Embedding and Shared Policy

1 Task Setup

We consider four simulated environments: wheeled locomotion, ant locomotion, 2-link reacher, and 4-link reacher. For each environment, we set the horizon to $H = 200$. The reward function is comprised of the distance to the target location and energy/control costs:

$$r(s, a) = -\|s_{\text{curr}} - x_{\text{target}}\|_2 - 0.001 * \|a\|_2^2, \tag{1}$$

where s_{curr} is the current location of the agent (resp. the end-effector) for locomotion (resp. reacher) tasks and x_{target} is the target location.

At each timestep, regarding to the location information, the agent only observes the current absolute location.

When sampling tasks, we sample D and D' within a circle $\rho < 1.1$ and sample D'' within an annulus $1.1 < \rho < 1.5$.

2 Experimental Details

We implement our method TESP and baselines based on RLlib [Liang et al., 2018] and tune hyperparameters with Tune [Liaw et al., 2018]. In all experiments, at each meta-update during training, we randomly sample 20 tasks from D and sample $100,000$ timesteps in total. When post-processing episodes, we use a linear baseline [Duan et al., 2016] and generalized advantage estimation (GAE) [Schulman et al., 2015] to reduce variances, and we set $\gamma = 0.99$ and $\lambda = 0.97$. During fast-update, the learning rate of VPG is set to 0.001 for MAML, and the adaptive per-parameter learning rates of VPG are initialized with 0.001 for other methods (i.e., Meta-SGD, SV, and TESP). During meta-update, the PPO clipping hyperparameter is set to $\epsilon = 0.15$ and the KL penalty is set to 0. The learning rate of Adam optimizer of PPO is set to 0.0003. In addition, we set the size of the episode buffer to $M = 16$ for TESP.

For all methods, the policy network is a 2-layer MLP with the same number of units per layer and tanh nonlinearities, which outputs the mean and variance of a Gaussian action distribution. In addition, the task encoder of TESP is modeled as an RNN with GRU cell followed by a fully-connected layer. The detailed specifications of network architecture are summarized in Table 1.

3 Generalization to Novel Transition Probabilities

In prior experiments, tasks differ only in their target locations (i.e., part of reward functions). To further examine the generalization ability of our proposed approach, we extend Ant locomotion tasks by introducing noised state transitions to simulate different transition probabilities. Specifically, at each timestep of each rollout at test time, we add a noise signal sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ to the next state transition, which causes novel tasks D' and D'' that not only have different reward functions but also have different transition probabilities. Note that training tasks D still differ only in their reward functions without noised dynamics.

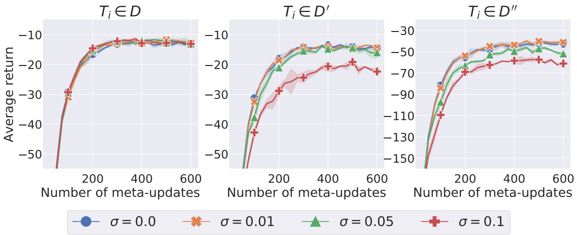

Figure 1: The performance of TESP on Ant locomotion tasks with respect to different σ.

Table 1: The detailed specifications of network architecture for different tasks, where d denotes the dimension of task embeddings and η denotes the coefficient of the regularization term. Here the coefficient is redefined with $\eta \leftarrow \frac{1}{d}\eta$.

(a) Wheeled locomotion.

	MLP units	GRU units	d	η
MAML	$[256, 256]$	N/A	N/A	N/A
Meta-SGD	$[256, 256]$	N/A	N/A	N/A
AdaptSV	$[256, 256]$	N/A	8	0.01
TESP	$[256, 256]$	256	8	0.01

(b) Ant locomotion.

	MLP units	GRU units	d	η
MAML	$[512, 512]$	N/A	N/A	N/A
Meta-SGD	$[512, 512]$	N/A	N/A	N/A
AdaptSV	$[512, 512]$	N/A	32	0.01
TESP	$[512, 512]$	256	32	0.01

(c) 2-link reacher.

	MLP units	GRU units	d	η
MAML	$[256, 256]$	N/A	N/A	N/A
Meta-SGD	$[256, 256]$	N/A	N/A	N/A
AdaptSV	$[256, 256]$	N/A	8	0.001
TESP	$[256, 256]$	256	8	0.005

(d) 4-link reacher.

	MLP units	GRU units	d	η
MAML	$[512, 512]$	N/A	N/A	N/A
Meta-SGD	$[512, 512]$	N/A	N/A	N/A
AdaptSV	$[512, 512]$	N/A	16	0.005
TESP	$[512, 512]$	256	16	0.005

Figure 1 shows the performance of TESP with respect to different σ. We observe that our proposed TESP is generalizable and robust to noised state prediction (i.e., novel dynamics).

References

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *ICML*, 2016.

Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *ICML*, 2018.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.