Decoupled Meta-Learning with Structured Latents

Russell Mendonca [1] , Sergey Levine [1], Chelsea Finn [2]
[1] University of California, Berkeley [2] Stanford University
russellm@berkeley.edu, svlevine@eecs.berkeley.edu, cbfinn@cs.stanford.edu

1 Introduction

A key aspect of human and animal intelligence is leveraging a rich variety of past experience in order to adapt quickly to acquire new skills. Meta reinforcement learning (RL) has shown promising results in enabling agents to rapidly adapt to new tasks by using prior experience and can, in principle, learn across broad distribution of tasks. However, in practice, most current methods can only meta-train on very limited task distributions. This is often due to the exploration and optimization challenges that arise when trying to quickly adapt to perform very diverse behaviors, challenges that have also plagued multi-task RL methods [8, 11, 13].

Current meta-RL methods are trained to infer the task and adapt or compute the policy for the task from sampled data (using techniques such as policy gradients [4] or amortized inference [3, 9]), where the adapted policy is optimized for high return. However, such task adaptation approaches often do not clearly differentiate between different tasks early on in the meta-training process. As a result, the aforementioned multi-task optimization challenges are exacerbated, as the policy is not even informed of the task. Furthermore, diverse task distributions of interest often consist of some tasks for which the reward is relatively sparse. Meta-training on these tasks requires good exploration strategies, which are difficult to discover. Finally, since most current methods optimize for post-adaptation performance on the entire set of tasks, the harder tasks which have sparser reward are generally ignored by current methods while only the easier meta-training tasks are successfully acquired.

One strategy for handling diverse task distributions is to decouple the meta-learning problem across tasks. This involves learning local policies for each of the training tasks using off-policy RL, which are then used to guide the meta-policy via supervised distillation. While this approach has been shown to enable meta-training on harder tasks, such as those with sparse reward or image observations [6], it has only been demonstrated on relatively narrow task distributions, such as varying goal locations or angles of a block or a door. In this work, we leverage the decoupled training structure to transfer knowledge across families of tasks. However, even with the decoupled structure, we still need multi-modal exploration strategies in order to discover rewards for tasks from different families, which is necessary for adaptation. Inspired by recent work on amortized task inference [9], we propose to use a mixture model to represent the prior over tasks in order to capture different exploration modes. Each family of tasks corresponds to a different mixture component, to which the contexts of relevant tasks are constrained, instead of using a single global Gaussian prior for all tasks.

The main contribution of our work is a meta-RL method that can meta-train over diverse task distributions, consisting of multiple distinct task families. Our algorithm, which we call Decoupled Meta-Learning with Structured Latents (DAMSEL) achieves this by leveraging local trained experts and using multi-modal priors for task latent variables corresponding to distinct task families. In our experimental evaluation, we find that DAMSEL successfully enables a simulated sawyer robot to meta-train on a challenging task distribution that includes pushing objects, opening doors, and opening drawers to varying positions, while significantly outperforming prior state-of-the-art meta-learning methods.

3rd Workshop on Meta-Learning at NeurIPS 2019, Vancouver, Canada.

2 Related Work

We build on the meta-learning framework [14, 1], where the objective is to adapt quickly using prior experience. Prior meta-RL algorithms include context-based methods [9], recurrent network learners [3, 15, 7], policy gradient methods which adapt quickly from a learned initialization [4, 5, 10], and methods that quickly adapt models which are then used with planners [2, 12]. Recent work in off-policy meta-RL [9, 6] has resulted in greatly improved sample efficiency during meta-training as compared to prior methods. Our approach draws on prior works that decouple the meta-objective across tasks [6] and methods that use latent task variables [9]. However, unlike these prior works, we use a multi-model prior that is decoupled across task families to enable effective meta-training with diverse task distributions, leading to significant empirical gains over these prior methods.

3 Decoupled Meta Learning with Structured Latents

3.1 Problem Statement

Given a distribution $\mathcal{T}_i \sim p(\mathcal{T})$ over tasks, where each task $\mathcal{T}_i$ is a different Markov decision process, the meta-RL problem is to efficiently learn a policy that can quickly adapt to maximize the expected return for a new task drawn from $p(\mathcal{T})$. This adaptation relies on a dataset of tuples $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}')$ sampled from the new task, which we refer to as a context $\mathbf{c}$.

We define a task family to be a collection of tasks that require similar exploration strategies to detect significant reward, which is necessary for producing effective contexts for adaptation. For each meta-training task, we assume access to which task family it belongs to. We do not assume knowledge of the task family identity when presented with a new task at meta-test time.

3.2 Algorithm

We first learn optimal policies for each of the meta-training tasks, using off-policy RL to minimize number of samples required. Rollouts from these trained experts are then saved to a dataset to be used for meta-learning.

We build on PEARL, which uses amortized inference to learn a latent from the provided context for a given task $\mathcal{T}_i$. When sampling data from the environment, the context is updated with each new sampled trajectory. We use an inference network q parameterized by ϕ to produce a latent $\mathbf{z_i}$ from a context $\mathbf{c_i}$, which is then used to condition a policy (π, parameterized by θ).The policy is trained via a supervised loss using data from the expert dataset. Specifically, the supervised loss for task $\mathcal{T}_i$, with expert traces $\mathcal{D}_i^{\text{val}}$ is given by $\mathcal{L}_i^{\text{BC}}(\theta_\pi, \mathcal{D}_i^{\text{val}}, \mathbf{z}_i) = -\sum_{(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{D}} (\mathbf{a}_t^{pred} - \mathbf{a}_t)^2$, where $\mathbf{a}_t^{pred}$ is the action produced by the policy θ_π for state $\mathbf{s}_t$ when conditioned on $\mathbf{z_i}$

To encourage multi-modal exploration, we use a mixture model to represent the prior over tasks, with each task family corresponding to a different mixture component. Since we use imitation learning to train the policy on expert data and guide exploration, we do not need probabilistic latents as in PEARL [9]. The constraint loss for a training task with latent $\mathbf{z_i}$ and mixture component of the corresponding task family $\mathbf{f}$ is given by $\mathcal{L}_i^{\text{constraint}} = \|\mathbf{z_i} - \mathbf{f}\|_2$. Further, to ensure that the latent encodes relevant task infor-

Algorithm 1 DAMSEL : Meta-training

Require: Set of meta-training tasks $\{\mathcal{T}_i\}$
Require: Task-family $\mathcal{F}_i$ for each task $\mathcal{T}_i$.
Use RL to acquire π_i^* for each task $\mathcal{T}_i$
Store roll-outs from each π_i^* in dataset $\mathcal{D}_i^*$
Initialize replay buffers $\mathcal{B}_i$ for each training task
Randomly initialize θ , ϕ
while not done **do**
  for each $\mathcal{T}_i$ in mini-batch $\{\mathcal{T}_j\}_{j=1...L}$ **do**
    Initialize $\mathbf{z}_i$ to the mixture component corresponding to task-family $\mathcal{F}_i$
    Initialize context $\mathbf{c}_i = \{\}$
    for $k = 1, ...K$ **do**
      Sample latent $\mathbf{z}_i = q_\phi(\mathbf{c}_i)$
      Collect data from policy $\pi_\theta(\mathbf{a}|\mathbf{s}, \mathbf{z_i})$ and add to buffer $\mathcal{B}_i$
      Sampled data from the buffer $\{(\mathbf{s}_j, \mathbf{a}_j, \mathbf{s}'_j, \mathbf{r}_j)\}_{j:1..N} \sim \mathcal{B}_i$, and append to $\mathbf{c_i}$
    end for
  end for
  for step in training steps **do**
    for each $\mathcal{T}_i$ in mini-batch $\{\mathcal{T}_j\}_{j=1...M}$ **do**
      Sample context $\mathbf{c}_i \sim \mathcal{B}_i$ and obtain latent $\mathbf{z}_i = q_\phi(\mathbf{c}_i)$
      Sample expert trajectories $\mathcal{D}_i^{\text{val}} \sim \mathcal{D}_i^*$
    end for
    $\mathcal{L}_i^{\text{BC}} = \mathcal{L}_i^{\text{BC}}(\theta, \mathcal{D}_i^{\text{val}}, \mathbf{z}_i)$
    $\phi \leftarrow \phi - \nabla_\phi \sum_i (\mathcal{L}_i^{\text{BC}} + \mathcal{L}_i^{\text{model}} + \mathcal{L}_i^{\text{constraint}})$
    $\theta \leftarrow \theta - \nabla_\theta \sum_i \mathcal{L}_i^{\text{BC}}$
  end for
end while

mation, we train a model $\mathcal{M}$ that takes in $(\mathbf{s}, \mathbf{a})$ data from a context $c = (\mathbf{s}, \mathbf{a}, \mathbf{s}', \mathbf{r})$, and conditioned on the latent seeks to regenerate $(\mathbf{s}', \mathbf{r})$. We optimize the model using the following loss : $\mathcal{L}_i^{\text{model}} = -\left\|(\mathbf{s}', \mathbf{r}) - \mathcal{M}(\mathbf{s}, \mathbf{a}, \mathbf{z_i})\right\|_2$

At test time, we need to adapt to a new validation task, without access to its task family. We roll out the policy conditioned on each of the task family priors, and set the task family based on which prior yields highest return. The task latent for the validation task is initialized to this prior, and then subsequently updated as more data is sampled from the environment.

Algorithm 2 DAMSEL : Test-time adaptation

Require: Validation task $\mathcal{T}$
1: **for** each mixture-model component $\mathbf{f}$ **do**
2: Collect data using $\pi_\theta(\mathbf{a}|\mathbf{s}, \mathbf{f})$, and record average return.
3: **end for**
4: Initialize $\mathbf{z}_i$ to mixture component $\mathbf{f}^*$ which gave best average return.
5: Initialize context $\mathbf{c}$ = {}
6: **for** $k = 1, ...K$ **do**
7: Obtain $\mathbf{z} = q_\phi(\mathbf{c})$
8: Roll out policy $\pi_\theta(\mathbf{a}|\mathbf{s}, \mathbf{z})$ to collect data , and append this to context
9: **end for**

4 Experimental Evaluation

We aim to answer the following questions: (1) How well does DAMSEL compare to prior state of the art meta-learning approaches for meta-training on diverse task distributions and adapting to new validation tasks? (2) For our method to perform well, how important is (a) decoupling the meta objective and (b) structuring the latent space?

For evaluation, we use a challenging multi-family task distribution involving a simulated 7-DOF sawyer robot having to push objects, open doors and open drawers to different positions. For each of the tasks, the target location is not observed and needs to be inferred. The robot is controlled via 3D position control, and observations include the 3D position of the end effector and positions of the block, door and drawer.

- Pushing : Each task comprises moving a block from a fixed starting position to a goal location sampled from a 20 cm $\times$ 10 cm region.
- Door opening: The task distribution involves opening the door to a target angle chosen from 0 to 45 degrees.
- Drawer opening: Each task requires the robot to open the drawer to a target position chosen from 0 to 15 cm.

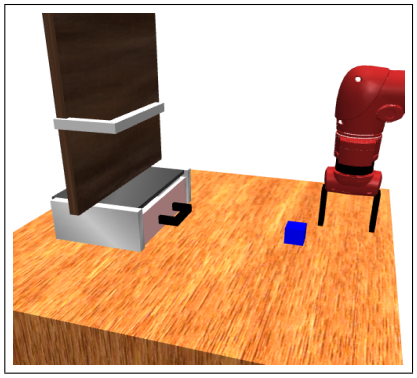

Figure 1: Environment visualization

We compare our method (DAMSEL) to prior methods including PEARL and an ablation without the structured latent space (similar to GMPS). After meta-training on a set of 60 tasks, including 20 tasks of each family type , we evaluate performance on 10 pushing, 10 door, and 10 drawer tasks. Fig 2 shows plots of final distance to target position versus number of trajectories sampled (a) with performance averaged across tasks (1st column) and also (b) with performance plotted for each individual validation task (2nd to 4th columns).

Defining success to be within 5cm of the target for pushing, within 0.1 radians for door opening and within 0.02 cm for drawer opening, we see that DAMSEL is successful on almost all tasks. On the other hand, PEARL makes no progress on any of the door and drawer tasks. This is because reward detection for door and drawer opening requires exploration strategies that are harder to learn than for pushing, and hence PEARL ignores the harder tasks while meta-training, and only meta-trains on the pushing tasks. Consequently, it is unable to perform validation tasks from the harder families. Even on the pushing tasks on which PEARL makes some progress (50% success), DAMSEL has a much higher success rate (90 % success).

The performance improvement between PEARL and the GMPS variant highlights the advantage of decoupling the optimization problem, as the provided supervision from the learned experts enables GMPS to make progress on all families. However, GMPS doesn't solve tasks as accurately as DAMSEL. From Fig. 3, we see that for GMPS on the drawer or drawer families, the inference network produces the same latents for multiple tasks. On the other hand, some pushing task latents

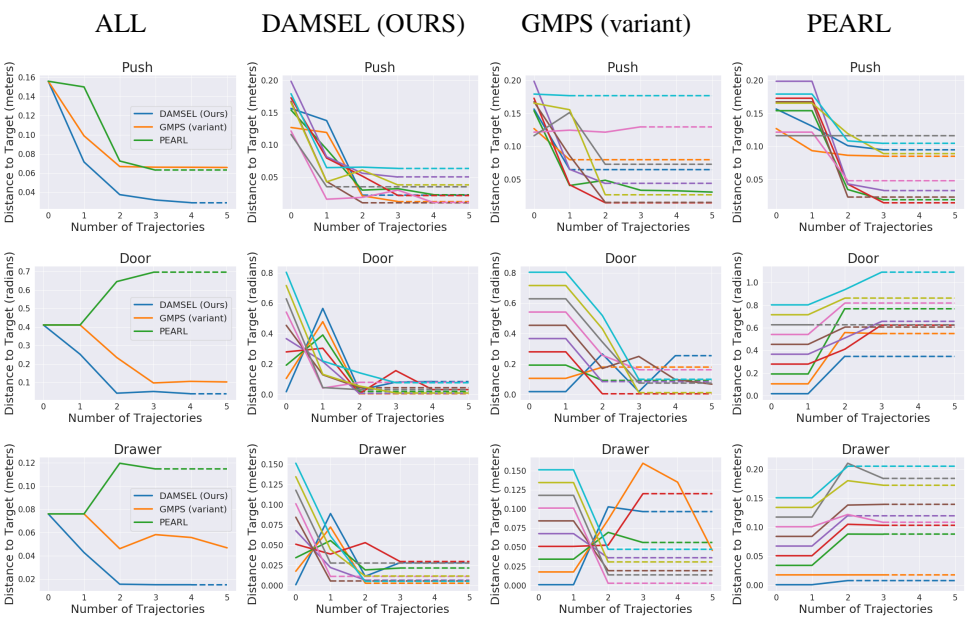

Figure 2: Comparison for performance (distance to target) on validation tasks. (a) Column 1: Performance averaged across all training tasks for each task family. (b) Performance plotted for each validation task individually for DAMSEL (Ours) (column 2), GMPS (variant) (column 3) and PEARL (column 4).**Top:** Block Pushing **Middle:** Door Opening **Bottom:** Drawer Opening

are closer to the door or drawer families rather than to other task latents corresponding to pushing. By imposing a mixture -model to model the prior over the three task families, DAMSEL learns a much more structured task latent space (as seen in Fig. 3), resulting in much superior accuracy in solving the validation tasks.

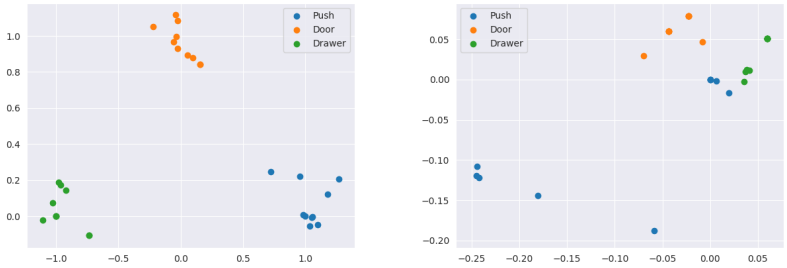

Figure 3: Task context visualization **Left:** DAMSEL (Ours). **Right:** a variation of GMPS

5 Discussion and Future Work

We have presented an algorithm that can meta-learn on diverse task distributions involving multiple task families. The key aspects of the approach include decoupling the meta-learning problem across tasks and imposing structure on the task latent space. We believe this work represents a first step towards realizing the full potential of meta-learning on broad distributions of skills. Future work could explore better ways of structuring the latents, such as learning the task family priors instead of assigning them arbitrarily. Further, an important direction for future exploration is studying how smooth, yet structured global priors can be acquired, as they may enable generalize to entirely new task families at test time, if meta-trained on a sufficiently diverse set of task families.

References

[1] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*.

[2] Ignasi Clavera, Anusha Nagabandi, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt: Meta-learning for model-based control. *arXiv preprint arXiv:1803.11347*, 2018.

[3] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

[5] Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pages 5307–5316, 2018.

[6] Russell Mendonca, Abhishek Gupta, Rosen Kralev, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Guided meta-policy search. *Advances in Neural Information Processing Systems*, 2019.

[7] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[8] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2016.

[9] Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. *International Conference on Machine Learning (ICML)*, 2019.

[10] Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. *International Conference on Learning Representations (ICLR)*, 2019.

[11] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *International Conference on Learning Representations (ICLR)*, 2016.

[12] Steindór Sæmundsson, Katja Hofmann, and Marc Peter Deisenroth. Meta reinforcement learning with latent variable gaussian processes. *CoRR*, abs/1803.07551, 2018.

[13] Tom Schaul, Diana Borsa, Joseph Modayil, and Razvan Pascanu. Ray interference: a source of plateaus in deep reinforcement learning. *arXiv preprint arXiv:1904.11455*, 2019.

[14] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[15] Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Rémi Munos, Charles Blundell, Dharshan Kumaran, and Matthew Botvinick. Learning to reinforcement learn. *CoRR*, abs/1611.05763, 2016.