EFFICIENT OFF-POLICY META-REINFORCEMENT LEARNING VIA PROBABILISTIC CONTEXT VARIABLES

Kate Rakelly*, Aurick Zhou*, Deirdre Quillen, Chelsea Finn, Sergey Levine
Department of Electrical Engineering and Computer Science
UC Berkeley
Berkeley CA, 94709
{rakelly, azhou, dquillen, cbfinn, svlevine}@eecs.berkeley.edu

1 INTRODUCTION

Conventional RL methods learn a separate policy per task, each often requiring millions of interactions with the environment. Learning large repertoires of behaviors with such methods quickly becomes prohibitive. Fortunately, many of the problems we would like our autonomous agents to solve share common structure. For example screwing a cap on a bottle and turning a doorknob both involve grasping an object in the hand and rotating the wrist. Exploiting this structure to learn new tasks more quickly remains an open and pressing topic. Meta-learning methods learn this structure from experience by making use of large quantities of experience collected across a distribution of tasks. Once learned, these methods can adapt quickly to new tasks given a small amount of experience.

While meta-learned policies adapt to new tasks with only a few trials, during training they require massive amounts of data drawn from a large set of distinct tasks, exacerbating the problem of sample efficiency that plagues RL algorithms. Most current meta-RL methods require on-policy data during both meta-training and adaptation Finn et al. (2017); Wang et al. (2016); Duan et al. (2016); Mishra et al. (2018); Rothfuss et al. (2018); Houthooft et al. (2018), rendering them exceedingly inefficient during meta-training. However, making use of off-policy data for meta-RL poses new challenges. Meta-learning typically operates on the principle that the test tasks should be drawn from the same distribution as the training tasks – for example, to meta-learn to classify images of new animals from a few examples, the algorithm should be meta-trained on animal species from across the animal kingdom Vinyals et al. (2016). This makes it inherently difficult to meta-train a policy to adapt from off-policy data, which is systematically different from the data the policy would see when it explores (on-policy) in a new task at meta-test time.

To achieve both adaptation and meta-training data efficiency, we propose an approach that integrates online inference of probabilistic context variables with existing off-policy RL algorithms. To achieve rapid adaptation, meta-RL requires reasoning about distributions: when exposed to a new task for the first time, the optimal meta-learned policy must carry out a stochastic exploration procedure to visit potentially rewarding states, as well as adapt to the task at hand Gupta et al. (2018). During meta-training, we learn a probabilistic encoder that accumulates the necessary statistics from past experience that enable the policy to perform the task. At meta-test time, when the agent is faced with an unseen task, our method adapts by sampling context variables ("task hypotheses"), acting according to that task, and then updating its belief about the task by updating the posterior over the context variables.

The primary contribution of our work is an off-policy meta-RL algorithm Probabilistic Embeddings for Actor-critic RL (PEARL) that achieves excellent sample efficiency during meta-training, enables fast adaptation by accumulating experience online, and performs structured exploration by reasoning about uncertainty over tasks. In our experimental evaluation, we demonstrate state-of-the-art results with 20-100X improvement in meta-training sample efficiency and substantial increases in asymptotic performance over prior state-of-the-art on six continuous control meta-learning environments. We further examine how our model conducts structured exploration to adapt rapidly to new tasks in a 2-D navigation environment with sparse rewards.

2 RELATED WORK

Our work builds on the meta-learning framework Schmidhuber (1987); Bengio et al. (1990); Thrun & Pratt (1998) in the context of reinforcement learning. Recurrent Duan et al. (2016); Wang et al. (2016)

and recursive Mishra et al. (2018) meta-RL methods adapt to new tasks by aggregating experience into a latent representation on which the policy is conditioned. We model latent task variables as probabilistic and use a simpler aggregation function. Prior work has explored training recurrent Q-functions with off-policy Q-learning methods Heess et al. (2015); Hausknecht & Stone (2015). We find the straightforward application of these methods to meta-RL difficult, and explore how to effectively make use of off-policy data during meta-training. Gradient-based meta-RL methods focus on on-policy learning, using policy gradients Finn et al. (2017); Stadie et al. (2018); Rothfuss et al. (2018); Xu et al. (2018a), meta-learned loss functions Sung et al. (2017); Houthooft et al. (2018), or hyperparameters Xu et al. (2018b). We instead focus on meta-learning from off-policy data, which is non-trivial to do with these prior methods. Outside of RL, meta-learning methods for few-shot supervised learning problems have explored a wide variety of approaches and architectures Santoro et al. (2016); Vinyals et al. (2016); Ravi & Larochelle (2017); Oreshkin et al. (2018). Our permutation-invariant embedding function is inspired by the embedding function of prototypical networks Snell et al. (2017).

Prior work has applied probabilistic models to meta-learning. For supervised learning, Rusu et al. (2019); Gordon et al. (2019); Finn et al. (2018) adapt model predictions using probabilistic latent task variables inferred via amortized approximate inference. In RL, Hausman et al. (2018) also conditions the policy on inferred task variables, but the aim is to compose skills via the learned embedding space, while we focus on adapting to new tasks. While we infer task variables and explore via posterior sampling, Gupta et al. (2018) adapts via gradient descent and explores via sampling from the prior.

Our approach can be viewed as a meta-learned variant of posterior sampling Strens (2000); Osband et al. (2013); the probabilistic context posterior over possible tasks enables temporally extended exploration by acting optimally according to the task. Adaptation at test time in meta-RL can be viewed as a special case of RL in a POMDP Kaelbling et al. (1998) by including the task as the unobserved part of the state. We use a variational approach similar to Igl et al. (2018) to estimate belief over the task. While they focus on general POMDPs and make use of on-policy methods, we leverage the assumptions of the meta-learning problem to simplify inference, use posterior sampling for exploration in a new task, and demonstrate how to integrate our approach with off-policy learning.

3 METHOD

Similar to previous meta-RL formulations, we assume a distribution of tasks $p(\mathcal{T})$, where each task is a Markov decision process (MDP). Formally, a task $\mathcal{T} = \{p(\mathbf{s}_0), p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t), r(\mathbf{s}_t, \mathbf{a}_t)\}$ consists of an initial state distribution $p(\mathbf{s}_0)$, transition distribution $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, reward function $r(\mathbf{s}_t, \mathbf{a}_t)$. We assume that the transition and reward functions are unknown, but can be sampled by taking actions in the environment. Given a set of training tasks sampled from $p(\mathcal{T})$, the meta-training process learns a policy that adapts to the task at hand by conditioning on the history of past transitions, which we refer to as *context* C. Let $C_n^{\mathcal{T}} = (\mathbf{s}_n, \mathbf{a}_n, r_n, \mathbf{s}'_n)$ be one transition in the task $\mathcal{T}$ so that $C_{1:N}^{\mathcal{T}}$ comprises the experience collected so far in the current task (we will often write simple C when considering one task and an arbitrary number of timesteps). At test-time, the policy must adapt to a new set of tasks from $p(\mathcal{T})$.

3.1 PROBABILISTIC LATENT CONTEXT

We capture knowledge about how the current task should be performed in a latent probabilistic context variable Z, on which we condition the policy as $\pi_\theta(\mathbf{a}|\mathbf{s}, \mathbf{z})$. Meta-training consists of leveraging data from a variety of training tasks to learn to infer Z from a recent history of experience in the new task, as well as optimizing the policy to solve the task given samples from the posterior over Z. In this section we describe the structure of the meta-trained inference mechanism. We address how meta-training can be performed with off-policy RL algorithms in Section 3.2.

To enable adaptation, the latent context Z must encode salient information about the task. We adopt an amortized variational inference approach Kingma & Welling (2014); Rezende et al. (2014); Alemi et al. (2016) to learn to infer Z. We train an *inference network* $q_\phi(\mathbf{z}|c)$ that estimates the posterior $p(\mathbf{z}|c)$. While there are several choices for the objective to optimize $q_\phi(\mathbf{z}|c)$ including learning predictive models of rewards and dynamics or maximizing returns through the policy, we choose to

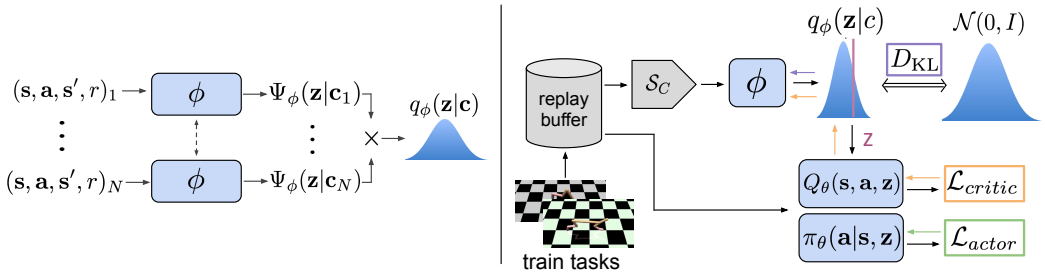

Figure 1: (left) The amortized inference network predicts the posterior over the latent context variables $q_\phi(\mathbf{z}|c)$ as a permutation-invariant function of prior experience. Samples from this posterior condition the policy. (right) The actor and critic are meta-learned jointly with the inference network, which is optimized with gradients from the critic as well as from an information bottleneck on Z. Sampling context ($\mathcal{S}_C$) from a pool of recently collected, and thus more on-policy, data is critical for off-policy meta-learning.

optimize it to predict the task state-action value function. The resulting training objective is:

$$\mathbb{E}_{\mathcal{T}}[\mathbb{E}_{Z\sim q(\mathbf{z}|c^{\mathcal{T}})}[R(\mathcal{T}, Z) + \beta D_{\text{KL}}(q(\mathbf{z}|c^{\mathcal{T}})||p(\mathbf{z}))] \tag{1}$$

The KL divergence term can also be interpreted as resulting from a variational approximation to an information bottleneck Alemi et al. (2016) that constrains the mutual information between Z and C. where $p(\cdot)$ is our variational approximation to the marginal distribution of Z and is assumed to be a unit Gaussian, and $R(\mathcal{T}, Z)$ is the Bellman error for a state-action value function conditioned on Z. While the parameters of q_ϕ are optimized during meta-training, at meta-test time the latent context for a new task is simply inferred from gathered experience.

In designing the form of the inference network $q_\phi(\mathbf{z}|c)$, we would like it to be expressive enough to capture minimal sufficient statistics of task-relevant information, without modeling irrelevant dependencies. We note that an encoding of a fully observed MDP should be permutation invariant: if we would like to infer what the task is, identify the MDP model, or train a value function, it is enough to have access to a collection of transitions $\{\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i\}$, without regard for the order in which these transitions were observed. We therefore choose a permutation-invariant representation $q_\phi(\mathbf{z}|c_{1:N})$ factorized as

$$q(\mathbf{z}|c_{1:N}) \propto \Pi_{n=1}^{N} \Psi(\mathbf{z}|c_n) \tag{2}$$

To keep the method tractable, we use Gaussian factors $\Psi(\mathbf{z}|c_n) = \mathcal{N}(f_\phi^\mu(c_n), f_\phi^\sigma(c_n))$, which result in a Gaussian posterior, see Figure 1 (left).

For fast adaptation at meta-test time, it is critical for the agent to be able to explore and determine the task efficiently. In prior work, posterior sampling for RL Strens (2000); Osband et al. (2013) begins with a prior distribution over MDPs, computes a posterior distribution conditioned on the experience it has seen so far, and executes the optimal policy for an MDP sampled from the posterior for the duration of an episode as an efficient method for exploration. In particular, acting optimally according to a random MDP allows for temporally extended exploration, meaning that the agent can act to test hypotheses even when the results of actions are not immediately informative of the task. In the single-task deep RL setting, the benefits of posterior sampling were explored in Osband et al. (2016), which maintains an approximate posterior over value functions via bootstraps. In contrast, PEARL directly infers a posterior over the latent context Z, which encodes the value function as it is optimized with gradients from the critic. Meta-training learns a prior over Z that captures the distribution over tasks. At meta-test time, we sample z's (initially from the prior and then the updated posterior) and hold them constant across an episode, thus exploring in a temporally extended and diverse manner which becomes more optimal for the current task as our belief narrows.

3.2 Off-Policy Meta-Reinforcement Learning

While our probabilistic context model is straightforward to combine with on-policy policy gradient methods, a primary goal of our work is to enable efficient off-policy meta-reinforcement learning. By contrast, prior work largely makes use of stable but relatively inefficient on-policy algorithms Duan et al. (2016); Finn et al. (2017); Gupta et al. (2018); Mishra et al. (2018). However, designing off-policy meta-RL algorithms is non-trivial partly because modern meta-learning is predicated on the assumption that the distribution of data used for adaptation will match across meta-training and meta-test. In RL, this implies that since at meta-test time on-policy data will be used to adapt,

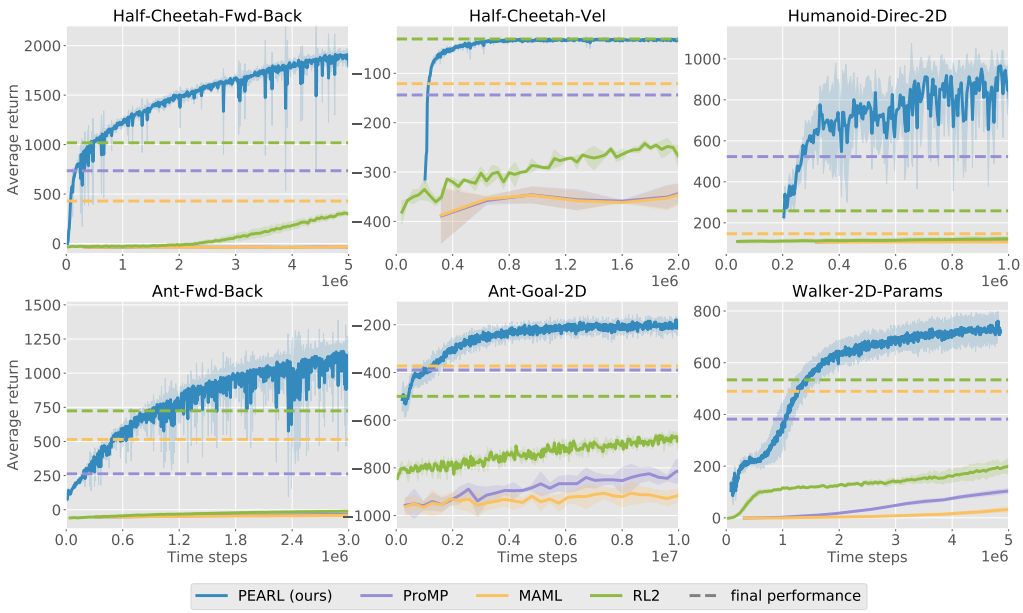

Figure 2: Test-task performance vs. samples collected during *meta-training* on continuous control domains. Our approach PEARL outperforms previous meta-RL methods both in terms of asymptotic performance and meta-training sample efficiency across six benchmark tasks. Dashed lines correspond to the maximum return achieved by each baseline after 1e8 steps. By leveraging off-policy data during meta-training, PEARL is $20-100$x more sample efficient than the baselines, and achieves consistently equal or better final performance compared to the best performing prior method in each environment.

on-policy data should be used during meta-training as well. Furthermore, meta-RL requires the policy to reason about *distributions* to learn effective stochastic exploration strategies. This problem inherently cannot be solved by off-policy RL methods that minimize temporal-difference error, as they do not have the ability to directly optimize for distributions of states visited. In contrast, policy gradient methods have direct control over the actions taken by the policy. In practice, we were unable to optimize a straightforward combination of meta-learning and value-based RL.

Our main insight in designing an off-policy meta-RL method with the probabilistic model in Section 3.1 is that the data used to train the probabilistic encoder need not be the same as the data used to train the policy. The policy can treat the context $\mathbf{z}$ as part of the state in an off-policy RL loop, while the stochasticity of the exploration process is provided by the uncertainty in the encoder $q_\phi(\mathbf{z}|c)$. Given a replay buffer containing all the data collected during training, we define separate samplers $\mathcal{S}_c$ and $\mathcal{S}_{RL}$ to sample the context and RL mini-batch respectively. The actor and critic are always trained with off-policy data sampled from the entire replay buffer B. We define a sampler $\mathcal{S}_C$ to sample context batches for training the encoder. Allowing $\mathcal{S}_c$ to sample from the entire buffer presents too extreme a distribution mismatch with on-policy test data. However, the context does not need to be strictly on-policy; we find that an in between strategy of sampling from a pool of recently collected data retains on-policy performance with better efficiency. We summarize our training procedure in Figure 1 (right).

3.3 Implementation

We build our algorithm on top of the soft actor-critic algorithm (SAC) Haarnoja et al. (2018), an off-policy actor-critic method based on the maximum entropy RL objective which augments the traditional sum of discounted returns with the entropy of the policy.

SAC exhibits good sample efficiency and stability, and further has a probabilistic interpretation which integrates well with probabilistic latent contexts. We optimize the parameters of the inference network $q(\mathbf{z}|c)$ jointly with the parameters of the actor $\pi_\theta(\mathbf{a}|\mathbf{s},\mathbf{z})$ and critic $Q_\theta(\mathbf{s},\mathbf{a},\mathbf{z})$, using the reparameterization trick to compute gradients for parameters of $q_\phi(\mathbf{z}|c)$ through sampled $\mathbf{z}$'s. We train the inference network using gradients from the Bellman update for the critic, given by the

following loss function

$$\mathcal{L}_{critic} = \mathbb{E}_{\substack{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\sim\mathcal{B} \\ \mathbf{z}\sim q_\phi(\mathbf{z}|c)}}(Q_\theta(\mathbf{s},\mathbf{a},\mathbf{z}) - (r + \bar{V}(\mathbf{s}',\bar{\mathbf{z}})))^2 \tag{3}$$

where $\bar{V}$ is a target network and $\bar{\mathbf{z}}$ indicates that gradients are not computed through it. The context data sampler S_c samples uniformly from data collected in the previous 1000 training steps, while the actor and critic are trained with samples from the entire replay buffer.

4 Experiments

Sample Efficiency and Performance We evaluate PEARL on six continuous control environments focused around robotic locomotion, simulated via the MuJoCo simulator Todorov et al. (2012). These locomotion task distributions require adaptation across dynamics (Walker-2D-Params) or across reward functions (the rest of the domains), and were previously introduced by Finn et al. (2017) and Rothfuss et al. (2018). We compare to existing policy gradient meta-RL methods ProMP Rothfuss et al. (2018) and MAML-TRPO Finn et al. (2017) using publicly available code. We also re-implement the recurrence-based policy gradient RL^2 Duan et al. (2016) with PPO Schulman et al. (2017). We attempted to adapt recurrent DDPG Heess et al. (2015) to our setting, however we were unable to optimize it. We hypothesize that this is due either to training with batches of correlated sequences, or to the distribution mismatch in adaptation data, as discussed in Section 3.2.

To evaluate on the meta-testing tasks, we perform online adaptation at the trajectory level, where the first trajectory is collected with context variable $\mathbf{z}$ sampled from the prior $r(\mathbf{z})$. Subsequent trajectories are collected with $\mathbf{z} \sim q(\mathbf{z}|c)$. In these domains, final test-time rollouts are collected after collecting a context of two trajectories. PEARL significantly outperforms prior meta-RL methods across all domains, Figure 2, in terms of both asymptotic performance and sample efficiency. Our approach uses 20-100x fewer samples during meta-training than previous policy gradient approaches while often also improving final asymptotic performance.

Posterior Sampling For Exploration In this section we demonstrate that posterior sampling in our model enables effective exploration strategies in sparse reward MDPs. Intuitively, by sampling from the prior context distribution $r(\mathbf{z})$, the agent samples a hypothesis based on the training tasks it has seen before. As the agent acts in the environment, the context posterior $p(\mathbf{z}|c)$ is updated, allowing it to reason over multiple hypotheses to determine the task. We demonstrate this behavior with a 2-D navigation task in which a point robot must navigate to different locations on a semi-circle. A shaped reward is given only when the agent is within a certain radius of the goal (we experiment with radius 0.2 and 0.8). We sample training and testing sets of tasks, each consisting of 100 randomly sampled goals. While our aim is to adapt to new tasks with sparse rewards, meta-training with sparse rewards is extremely difficult as it amounts to solving many sparse reward tasks from scratch. For simplicity we therefore assume access to the dense reward during meta-training, as in Gupta et al. (2018), but this burden could also be mitigated with task-agnostic exploration strategies.

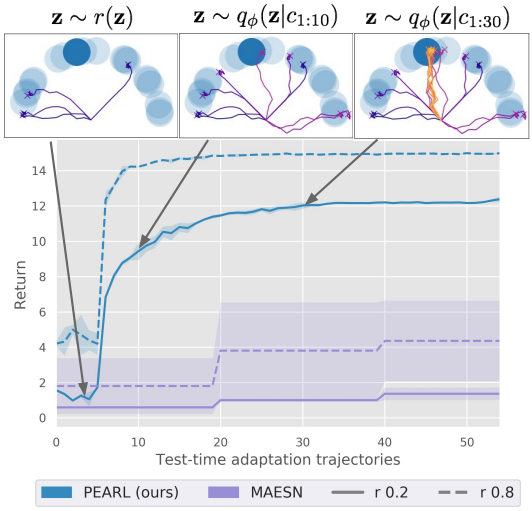

Figure 3: Sparse 2D navigation test-time adaptation. PEARL is able to start adapting to the task after collecting on average only 5 trajectories. We compare to MAESN (Gupta et al. (2018)).

In this setting, we compare to MAESN (Gupta et al. (2018)) and demonstrate we are able to adapt to the new sparse goal in fewer trajectories, while also requiring far fewer samples for meta-training to solve the task, Figure 3. Even with fewer samples, PEARL also outperforms MAESN in terms of final performance.

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Université de Montréal, 1990.

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.

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

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 9537–9548, 2018.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019.

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*, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

Nicolas Heess, Jonathan J Hunt, Timothy P Lillicrap, and David Silver. Memory-based control with recurrent neural networks. *arXiv preprint arXiv:1512.04455*, 2015.

Rein Houthooft, Richard Y Chen, Phillip Isola, Bradly C Stadie, Filip Wolski, Jonathan Ho, and Pieter Abbeel. Evolved policy gradients. In *Neural Information Processing Systems (NIPS)*, 2018.

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, 2018.

Leslie Pack Kaelbling, Michael Littman, and Anthony Cassandra. Planning and acting in partially observable stochastic domains. volume 101, pp. 99–134, 1998.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Boris N Oreshkin, Alexandre Lacoste, and Pau Rodriguez. Tadam: Task dependent adaptive metric for improved few-shot learning. 2018.

Ian Osband, Benjamin Van Roy, and Daniel Russo. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, 2013.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, 2016.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. In *International Conference on Learning Representations*, 2018.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, 2016.

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

John Schulman, Filip Wolski, Prafulla Dhariwal Dhariwal, Alec Radford Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Bradly C Stadie, Ge Yang, Rein Houthooft, Xi Chen, Yan Duan, Yuhuai Wu, Pieter Abbeel, and Ilya Sutskever. Some considerations on learning to explore via meta-reinforcement learning. *arXiv preprint arXiv:1803.01118*, 2018.

Malcom Strens. A bayesian framework for reinforcement learning. In *International Conference on Machine Learning*, 2000.

Flood Sung, Li Zhang, Tao Xiang, Timothy Hospedales, and Yongxin Yang. Learning to learn: Meta-critic networks for sample efficient learning. *arXiv preprint arXiv:1706.09529*, 2017.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 1998.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033, 2012.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, 2016.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Tianbing Xu, Qiang Liu, Liang Zhao, and Jian Peng. Learning to explore via meta-policy gradient. In *International Conference on Machine Learning*, pp. 5459–5468, 2018a.

Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. *arXiv:1805.09801*, 2018b.