MetaCURE: Meta Reinforcement Learning with Empowerment-Driven Exploration

Jin Zhang [*1] Jianhao Wang [*1] Hao Hu [1] Tong Chen [1] Yingfeng Chen [2] Changjie Fan [2] Chongjie Zhang [1]

Abstract

Meta reinforcement learning (meta-RL) extracts knowledge from previous tasks and achieves fast adaptation to new tasks. Despite recent progress, efficient exploration in meta-RL remains a key challenge in sparse-reward tasks, as it requires quickly finding informative task-relevant experiences in both meta-training and adaptation. To address this challenge, we explicitly model an exploration policy learning problem for meta-RL, which is separated from exploitation policy learning, and introduce a novel empowerment-driven exploration objective, which aims to maximize information gain for task identification. We derive a corresponding intrinsic reward and develop a new off-policy meta-RL framework, which efficiently learns separate context-aware exploration and exploitation policies by sharing the knowledge of task inference. Experimental evaluation shows that our meta-RL method significantly outperforms state-of-the-art baselines on various sparse-reward MuJoCo locomotion tasks and more complex sparse-reward Meta-World tasks.

1. Introduction

Human intelligence is able to transfer knowledge across tasks and acquire new skills within limited experiences. Current reinforcement learning (RL) agents often require a far more amount of experiences to achieve human-level performance (Hessel et al., 2018; Vinyals et al., 2019; Hafner et al., 2019). To enable sample-efficient learning, meta reinforcement learning (meta-RL) methods have been proposed, which automatically extract prior knowledge from previous tasks and achieve fast adaptation in new tasks (Schmidhuber, 1995; Finn et al., 2017). Despite fast progress, meta-RL with sparse rewards remains challenging, as task-relevant information is scarce in such settings, and efficient exploration is required to quickly find the most informative experiences in both meta-training and fast adaptation.

The problem of learning effective exploration strategies has been extensively studied for meta-RL with dense rewards, such as E-MAML (Stadie et al., 2018), ProMP (Rothfuss et al., 2019) and VariBAD (Zintgraf et al., 2019). Recently, several works have investigated learning to explore in sparse-reward tasks. For example, MAESN (Gupta et al., 2018) learns temporally-extended exploration behaviors by injecting structured noises into the exploration policy. PEARL (Rakelly et al., 2019) explores by posterior sampling (Thompson, 1933; Osband et al., 2013), which is optimal in asymptotic performance (Leike et al., 2016). However, both methods assume access to dense rewards during meta-training.

To address the challenge of meta-RL with sparse rewards, in this paper, we explicitly model the problem of learning to explore, and separate it from exploitation policy learning. This separation allows the learned exploration policy to purely focus on collecting the most informative experiences for enabling efficient meta-training and adaptation. As the common task-inference component in context-based meta-RL algorithms extracts task information from experiences (Rakelly et al., 2019; Zintgraf et al., 2019; Humplik et al., 2019), the exploration policy should collect experiences that contain rich task-relevant information and support efficient task inference. By leveraging this insight, we design a novel empowerment-driven exploration objective that aims to maximize the mutual information between exploration experiences and task identification. We then derive an insightful intrinsic reward from this objective, which is related to model prediction.

To incorporate our efficient exploration method, we develop a new off-policy meta-RL framework, called Meta-RL with effiCient Uncertainty Reduction Exploration (MetaCURE). MetaCURE performs probabilistic task inference, and learns separate context-aware exploration and exploitation policies. As both context-aware policies depend on task information extracted from the context, MetaCURE shares a common component of task inference for their learning, greatly im-

[*]Equal contribution [1]Institute for Interdisciplinary Information Sciences, Tsinghua University, China [2]Fuxi AI Lab, NetEase, China. Correspondence to: Jin Zhang <jinzhan20@mails.tsinghua.edu.cn>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

proving learning efficiency. In the adaptation phase, the exploration policy is intrinsically motivated to perform sequential exploratory behaviors across episodes, while the exploitation policy maximizes expected extrinsic return in the last episode of adaptation phase. MetaCURE is extensively evaluated on various sparse-reward MuJoCo locomotion tasks as well as sparse-reward Meta-World tasks. Empirical results show that it outperforms baseline algorithms by a large margin. To illustrate the advantages of our algorithm, we also visualize how it explores during adaptation and compare it with baseline algorithms.

2. Background

The field of meta-RL deals with a distribution of tasks $p(\kappa)$, with each task κ modelled as a Markov Decision Process (MDP), which consists of a state space, an action space, a transition function and a reward function (Sutton & Barto, 2018). In common meta-RL settings (Duan et al., 2016; Finn et al., 2017; Zintgraf et al., 2019), tasks differ in the transition and/or reward function, so we can describe a task κ with a tuple $\langle p_0^\kappa(s_0), p^\kappa(s'|s,a), r^\kappa(s,a)\rangle$, whose components denote the initial state distribution, the transition probability and the reward function, respectively. Off-policy meta-RL (Rakelly et al., 2019) assumes access to a batch of meta-training tasks $\{\kappa_m\}_{m=1,2,...,M}$, with M the total number of meta-training tasks. With a slight abuse of notations, we further denote κ as the task identification, and $\mathcal{K}$ indicates the random variable representing κ.

We denote context $c_n = (s_n, a_n, r_n, s_{n+1})$ as an experience collected at timestep n, and $c_{:t} = \langle c_0, c_1, ..., c_{t-1}\rangle$[1] indicates all experiences collected during t timesteps. Note that t may be larger than the episode length, and when it is the case, $c_{:t}$ represents experiences collected across episodes. $C_{:t}$ denotes a random variable representing $c_{:t}$.

A common objective for meta-RL is to optimize final performance after few-shot adaptation (Finn et al., 2017; Gupta et al., 2018; Stadie et al., 2018; Rothfuss et al., 2019; Gurumurthy et al., 2020). During adaptation, an agent first utilizes some exploration policy π_e to explore for a few episodes, and then updates an exploitation policy π to maximize the expected return. Such a meta-RL objective can be formulated as:

$$\max_{\pi,\pi_e} \mathbb{E}_{\kappa\sim p(\mathcal{K})}[R(\kappa, \pi(c^\kappa_{\pi_e}))], \tag{1}$$

where $c^\kappa_{\pi_e}$ is a set of experiences collected in task κ by policy π_e, and $R(\kappa, \pi(c^\kappa_{\pi_e}))$ is the last episode's expected return with policy π in task κ. The policy π is adapted with $c^\kappa_{\pi_e}$ for optimizing final performance. Both π_e and π can be *context-aware* (Lee et al., 2020), that is, they take context $c_{:i}$ into account while making decisions at timestep i.

3. Empowerment-Driven Exploration

In this section, we present a novel information-theoretic objective for optimizing exploration in both meta-training and adaptation. In Section 3.1, we derive an insightful intrinsic reward from this exploration objective, which measures experiences' information gain on task identification. In Section 3.2, we illustrate several implications of this intrinsic reward by a didactic example.

3.1. Exploration By Maximizing Information Gain

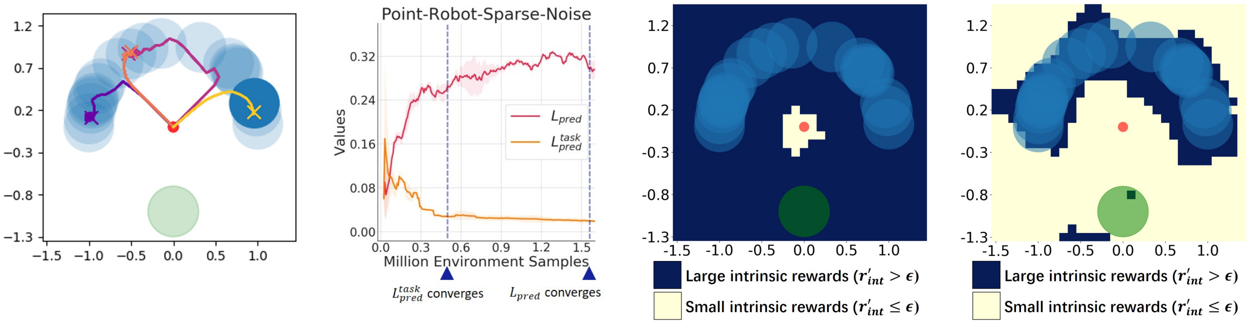
To enable efficient meta-RL with sparse rewards, our exploration strategy aims to collect experiences that maximize information gain about the identification of the current task. Consider an exploration policy π_e and experiences $C_{:H}$ collected by executing π_e for H timesteps in task $\mathcal{K}$. Our exploration objective $\mathcal{J}^{\pi_e}$ is formulated as maximizing the mutual information between exploration experiences $C_{:H}$ and task identification $\mathcal{K}$:

$$\begin{aligned}
&\mathcal{J}^{\pi_e}(C_{:H}, \mathcal{K}) \\
&= I^{\pi_e}(C_{:H}; \mathcal{K}) \\
&= \mathbb{E}_{(c_{:H},\kappa)\sim(C_{:H},\mathcal{K})}\left[\log \frac{p^{\pi_e}(c_{:H}|\kappa)}{p^{\pi_e}(c_{:H})}\right] && (2) \\
&= \mathbb{E}_{(c_{:H},\kappa)\sim(C_{:H},\mathcal{K})} \\
&\qquad \left[\sum_{t=0}^{H-1} \log \frac{p^{\pi_e}(a_t|c_{:t},\kappa)p(r_t, s_{t+1}|\kappa, c_{:t}, a_t)}{p^{\pi_e}(a_t|c_{:t})p(r_t, s_{t+1}|c_{:t}, a_t)}\right] && (3) \\
&= \mathbb{E}_{(c_{:H},\kappa)\sim(C_{:H},\mathcal{K})}\left[\sum_{t=0}^{H-1} \log \frac{p(r_t, s_{t+1}|\kappa, s_t, a_t)}{p(r_t, s_{t+1}|c_{:t}, a_t)}\right]. && (4)
\end{aligned}$$

In Eq. (2), the mutual information is expressed as the expectation over random variables $C_{:H}$ and $\mathcal{K}$, which follow the probability distribution $p^{\pi_e}(c_{:H}, \kappa) = p(\kappa)p^{\pi_e}(c_{:H}|\kappa)$. We then decompose the expectation to each timestep t by the chain rule, as shown in Eq. (3). Eq. (4) simplifies the expression by taking the following properties: 1) as an exploration policy needs to generalize to new tasks whose identifications are unknown, κ is not used for π_e's input, and $p^{\pi_e}(a_t|c_{:t},\kappa) = p^{\pi_e}(a_t|c_{:t})$; and 2) according to the Markov property, $p(r_t, s_{t+1}|\kappa, c_{:t}, a_t) = p(r_t, s_{t+1}|\kappa, s_t, a_t)$ for task κ.

On the right-hand side of Eq. (4), the numerator $p(r_t, s_{t+1}|\kappa, s_t, a_t)$ indicates the predictability of rewards and transitions given the task identification κ, while the denominator $p(r_t, s_{t+1}|c_{:t}, a_t)$ indicates predictability given current context $c_{:t}$. The logarithm of these two terms' division measures the amount of task information that the task identification contains more than current context. Note that

[1] For the clarity of following derivations, we define $c_{:0} = \langle c_{-1}\rangle = \left\langle(\vec{0}, \vec{0}, \vec{0}, s_0)\right\rangle$.



(a) Environment illustration.

(b) Learning curves of L_{pred} and L^{task}_{pred}.

(c) Intrinsic rewards at early stages of meta-training.

(d) Intrinsic rewards at late stages of meta-training.

Figure 1. (a) Illustration of the environment. The dark blue circle indicates the goal for the current task, while the light blue circles indicate goals for other meta-testing tasks. The green circle represents noisy regions, and the red circle is agent's initial position. We also illustrate efficient exploration behaviors motivated by our intrinsic reward. The dark purple line is the first episode's trajectory, while the light yellow line represents the last episode's trajectory. (b) Learning curves of the two model prediction errors. L^{task}_{pred} converges much faster (at 0.5 million samples) than L_{pred} (at 1.5 million samples). (c) At early stages of meta-training, our intrinsic reward approximates agent's curiosity about the environment. ϵ>0 is a constant to filter out neural network approximation error. Regions around the origin are fully explored, and the intrinsic reward is close to zero. (d) At late stages of meta-training, our intrinsic reward encourages exploration to maximize task information gain.

the expected log probability can be interpreted as negative cross-entropy prediction loss, so our exploration intrinsic reward is defined as:

$$r'_{\text{int}}(c_{:t+1}, \kappa) \\ = \underbrace{-\log p(r_t, s_{t+1}|c_{:t}, a_t)}_{L_{pred}(c_{:t+1})} + \underbrace{\log p(r_t, s_{t+1}|\kappa, s_t, a_t)}_{-L^{task}_{pred}(\kappa, c_t)}. \tag{5}$$

This intrinsic reward can be interpreted as the difference of two model prediction errors L_{pred} and L^{task}_{pred}, which can be estimated by training two model predictors, respectively: the *Meta-Predictor* makes predictions based on the current context, while the *Task-Predictor* makes predictions based on the task identification. To estimate the log probabilities tractably, we follow the common approach of utilizing L2 distances as an approximation of the negative log probability (Chung et al., 2015; Babaeizadeh et al., 2018):

$$L_{pred}(c_{:t+1}) \approx \left(r_t - \hat{r}^{pred}_t(c_{:t}, a_t)\right)^2 \\ + \left\|s_{t+1} - \hat{s}^{pred}_{t+1}(c_{:t}, a_t)\right\|_2^2 \\ L^{task}_{pred}(\kappa, c_t) \approx \left(r_t - \tilde{r}^{pred}_t(\kappa, s_t, a_t)\right)^2 \\ + \left\|s_{t+1} - \tilde{s}^{pred}_{t+1}(\kappa, s_t, a_t)\right\|_2^2, \tag{6}$$

where $\hat{r}^{pred}_t$ and $\hat{s}^{pred}_{t+1}$ are reward and transition predicted by the Meta-Predictor, while $\tilde{r}^{pred}_t$ and $\tilde{s}^{pred}_{t+1}$ are predicted by the Task-Predictor. Proofs of this section are deferred to Appendix A.

3.2. Didactic Example

In this section, we demonstrate the underlying implications of our intrinsic reward r'_{int} with detailed analyses on a didactic example.

We propose a 2-D navigation task set called Point-Robot-Sparse-Noise, as shown in Figure 1(a). The agent's observation is a 3-D vector composed of its current position (x, y) and a noise term u. u is a Gaussian noise if the agent is located in the green circle, and is zero otherwise. Goals are uniformly distributed on a semicircle of radius 1 and only sparse reward is provided. To understand how our intrinsic reward works, we illustrate its values during the learning process, as shown in Figure 1. This illustration demonstrates the following properties of our intrinsic reward:

Approximate curiosity-driven exploration at early stages of meta-training: in order to learn an efficient exploration policy, a meta-RL agent needs to effectively explore the tasks during meta-training. We found that at early stages of meta-training, our intrinsic reward demonstrates approximate curiosity-driven exploration (Schmidhuber, 1991). This is because that the second model prediction error L^{task}_{pred} converges much faster than L_{pred}, as shown in Figure 1(b), as it utilizes extra information by being aware of the task identity. Thus, r'_{int} approximately equals to L_{pred}, which can be viewed as a curiosity-driven intrinsic reward (Burda et al., 2018a;b) measured by model prediction errors (Pathak et al., 2017). As shown in Figure 1(c), at early stages of meta-training, r'_{int} obtains large values in unexplored regions, encouraging visitation to these regions.

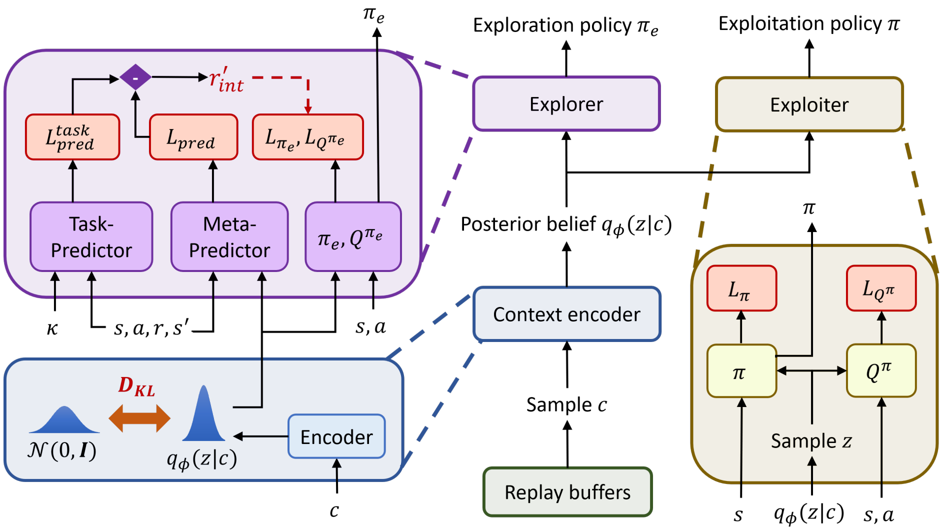


Figure 2. MetaCURE's meta-training pipeline. L_{π_e}, $L_{Q^{\pi_e}}$, L_π and L_{Q^π} are corresponding SAC loss functions for the exploration and exploitation policies.

Empowerment-driven exploration at late stages of meta-training: in both meta-training and adaptation, the exploration policy needs to collect experiences that contain rich task information. At late stages of training, the intrinsic reward encourages collection of experiences that are informative for inferring the current task, as it is derived from the objective of maximizing information gain about task identification (Eq. (4)). As shown in Figure 1(d), r'_{int} is large in regions where goals are possibly located, which shows that our intrinsic reward has implicitly learned the task distribution and encourages efficient exploration to acquire task information.

Robustness to irrelevant noises: Tasks may obtain uninformative noises that distract the agent from efficient exploration. At late stages of meta-training, r'_{int} not only encourages the agent to focus on uncertainties that help inference of task identification, but also ignores irrelevant noises. This is achieved by subtracting the second term L_{pred}^{task}. L_{pred}^{task} measures uncertainty given the true task identification, which is not helpful for inferring current task, e.g., the well-known noisy TV problem (Burda et al., 2018b). In our example, at late stages of meta-training (Figure 1(d)), in the noisy green circle, the mean of L_{pred} is 1.306, and the mean of L_{pred}^{task} is 1.312. In contrast, the mean of r'_{int} is only -0.006. Thus, the agent no longer explores the noisy regions.

4. MetaCURE Framework

This section presents MetaCURE, a new off-policy context-based meta-RL framework that learns separate exploration and exploitation policies. It also performs probabilistic task inference, which captures uncertainty over the task. As shown in Figure 2, MetaCURE is composed of four main components: (i) a context encoder $q_\phi(z|c)$ that extracts task information from context c and infers the posterior belief over the task embedding z, (ii) an *Explorer* that learns a context-aware exploration policy π_e, as well as two model predictors used to estimate our intrinsic reward, (iii) an *Exploiter* that learns a context-aware exploitation policy π, and (iv) replay buffers storing training data. Our exploration objective proposed in Section 3 is different from the common exploitation objective of maximizing expected extrinsic returns (Duan et al., 2016; Zintgraf et al., 2019), so we learn separate exploration and exploitation policies. As for the context encoder and the Exploiter, we adopt a variational inference structure similar to PEARL (Rakelly et al., 2019). During meta-training, data is collected by iteratively inferring the posterior task belief with contexts and performing both exploration and exploitation policies. During adaptation, only the exploration policy is used to collect informative experiences for task inference, and the exploitation policy utilizes this posterior belief to maximize final performance.

The context encoder q_ϕ uses variational inference methods (Kingma & Welling, 2013; Alemi et al., 2016) to model the exploitation policy π's state-action value function Q^π. The evidence lower bound for training the context encoder is:

$$\mathbb{E}_{z\sim q_\phi(z|c),s,a}[\log p(Q^\pi(s,a,z)) \\ - \beta D_{KL}(q_\phi(z|c)||p(z))], \quad (7)$$

where β is a hyper-parameter, s, a and c are sampled from the replay buffers. In practice, to tractably estimate the log probability of Q^π, we follow PEARL's method of uti-

Algorithm 1 MetaCURE: Meta-training Phase

```
Require: A set of meta-training tasks $\{\kappa_m\}_{m=1,2,...,M}$ drawn from $p(\mathcal{K})$
Initialize replay buffers $\mathcal{B}^{\kappa_m}$ for each task $\kappa_m$
Initialize exploration policy $\pi_e$, exploitation policy $\pi$, context encoder $q_\phi$, Task-Predictor and Meta-Predictor
while not done do
    for each task $\kappa_m$ do                    ▷ Data collection
        Collect exploration and exploitation experiences by running Algorithm 2 on $\kappa_m$
        Add collected experiences to $\mathcal{B}^{\kappa_m}$
    end for
    for steps in training steps do                 ▷ Training
        for each $\kappa_m$ do
            Sample context batch $b_{enc}^{\kappa_m}$ from $\mathcal{B}^{\kappa_m}$
            Sample policy training batch $b^{\kappa_m}$ from $\mathcal{B}^{\kappa_m}$
            Train Task-Predictor and Meta-Predictor by minimizing $L_{pred}^{task}$ and $L_{pred}$ in Eq. (6), respectively
            Train context encoder $q_\phi$ by maximizing Eq. (7)
            Compute $\pi_e$'s reward $r_{\text{exploration}}$ using Eq. (8)
            Train $\pi$ and $\pi_e$ by minimizing corresponding SAC losses in Eq. (9)
        end for
    end for
end while
```

lizing negative TD errors as an approximation of the log probability.

The Explorer consists of an exploration policy π_e that effectively explores during both meta-training and adaptation, its corresponding state-action value function Q^{π_e}, as well as two model predictors for estimating our intrinsic reward. The reward signal for π_e is defined as follows:

$$r_{\text{exploration}}(c_{:t+1}, \kappa) = r'_{\text{int}}(c_{:t+1}, \kappa) + \lambda r_t, \qquad (8)$$

where κ is the task identification, r'_{int} is our intrinsic reward defined in Eq. (5), $\lambda>0$ is a hyper-parameter, and r_t is the extrinsic reward. r'_{int} is estimated by training two model predictors and subtracting their prediction errors: the *Task-Predictor* makes predictions based on the task identification κ, while the *Meta-Predictor* makes predictions based on the context c. Both predictors are trained by minimizing their prediction errors in order to estimate L_{pred}^{task} and L_{pred}, respectively. Note that the exploration policy learns biased exploration behaviors by incorporating the extrinsic reward, as it effectively supports task inference, and we find empirically that considering extrinsic rewards leads to superior performance. Both the exploration policy π_e and the Meta-Predictor need to extract task information from the context, which can be represented as agent's posterior task belief $q_\phi(z|c)$. Thus MetaCURE takes $q_\phi(z|c)$ instead of context c as the input of both π_e and the Meta-Predictor. This knowledge reuse greatly improves learning efficiency.

Algorithm 2 MetaCURE: Adaptation Phase

```
Require: Meta-test task drawn from $p(\mathcal{K})$, number of adaptation episodes $E$
Initialize context $c = \{\}$
for episodes=1,...,$E-1$ do                ▷ Exploration phase
    for steps=1,...,$T$ do
        Take action according to $\pi_e(a|s, q_\phi(z|c))$
        Add collected experience $(s, a, r, s')$ to $c$
    end for
end for
Sample $z \sim q_\phi(z|c)$
for steps=1,2,...,$T$ do                   ▷ Exploitation phase
    Take action according to $\pi(a|s, z)$
end for
```

The Exploiter consists of an exploitation policy $\pi(a|s, z)$ that utilizes exploration experiences to perform exploitation behaviors, as well as its corresponding state-action value function Q^π. We design π to take state s and the task embedding z sampled from the posterior task belief $q_\phi(z|c)$ as input, and optimizes it with the extrinsic reward r.

The replay buffers $\{\mathcal{B}^{\kappa_m}\}_{m=1,2,...,M}$ are used for storing data from the meta-training tasks $\{\kappa_m\}_{m=1,2,...,M}$. As off-policy learning enables training on data collected by other policies, and our intrinsic reward can be computed with off-policy experience batches, π_e and π share the same replay buffers. This allows useful experiences to be shared between policies, greatly improving sample efficiency.

Both π_e and π are off-policy trained with SAC (Haarnoja et al., 2018). As shown in Figure 2, the loss functions for training the policies are as follows:

$$\begin{aligned}
L_\pi &= \mathbb{E}_{s,c,z}\left[D_{KL}(\pi(a|s,\overline{z})||\frac{exp(Q^\pi(s,a,\overline{z}))}{\mathcal{Z}_\pi(s,\overline{z})})\right] \\
L_{\pi_e} &= \mathbb{E}_{s,c} \\
&\left[D_{KL}\left(\pi_e(a|s,\overline{q_\phi}(z|c))||\frac{exp(Q^{\pi_e}(s,a,\overline{q_\phi}(z|c)))}{\mathcal{Z}_{\pi_e}(s,\overline{q_\phi}(z|c))}\right)\right] \\
L_{Q^\pi} &= \mathbb{E}_{(s,a,r,s'),c,z}\left[Q^\pi(s,a,z) - (r + \gamma\overline{V_\pi}(s',\overline{z}))\right]^2 \\
L_{Q^{\pi_e}} &= \mathbb{E}_{(s,a,r,s'),c}\left[Q^{\pi_e}(s,a,\overline{q_\phi}(z|c))\right. \\
&\left. -(r_{exploration} + \gamma\overline{V_{\pi_e}}(s',\overline{q_\phi}(z|c')))\right]^2,
\end{aligned} \qquad (9)$$

where $\overline{z}$ and $\overline{q_\phi}$ indicate that gradients do not flow through them, $\mathcal{Z}_\pi$ and $\mathcal{Z}_{\pi_e}$ are normalization functions that do not affect gradients, $\overline{V_\pi}$ and $\overline{V_{\pi_e}}$ are target value functions, and $c' = c \cup \{(s, a, r, s')\}$ is the updated context. All the expectations over s and (s, a, r, s') are averaged over the replay buffers. c is randomly sampled from exploration experiences collected by π_e, and all the expectations over z are averaged over the posterior task belief $q_\phi(z|c)$. Pseudo-codes for meta-training and adaptation are avail-

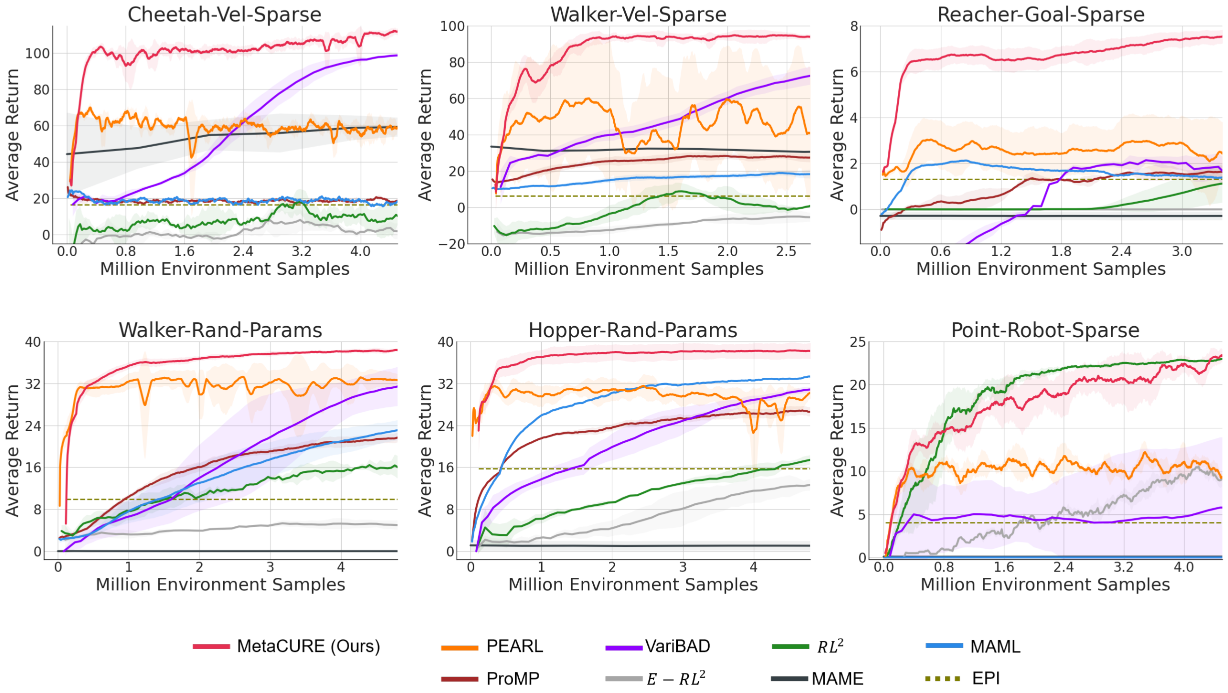


Figure 3. Evaluation of MetaCURE and several meta-RL baselines on various sparse-reward continuous control task sets. We plot the algorithms' meta-testing performance as a function of the number of experiences collected during meta-training. MetaCURE achieves substantially better performance than baseline algorithms.

able in Algorithm 1 and Algorithm 2, respectively. Additional implementation details are deferred to Appendix B. Our implementation codes are available at `https://github.com/NagisaZj/MetaCURE-Public`.

5. Experiments

In this section, we aim at answering the following questions: 1) Can MetaCURE achieve excellent adaptation performance in sparse-reward tasks that require efficient exploration in both meta-training and adaptation (Section 5.1 & 5.2)? 2) Can the exploration policy collect informative experiences efficiently (Section 5.3)? 3) Are the intrinsic reward as well as the separation of exploration and exploitation critical for MetaCURE's performance (Section 5.4)?

5.1. Adaptation Performance on Continuous Control

Environment Setup: algorithms are evaluated on six continuous control task sets with sparse rewards, in which exploration is vital for superior performance. Tasks vary in either the reward function (goal location in Point-Robot-Sparse and Reacher-Goal-Sparse, target velocity in Cheetah-Vel-Sparse and Walker-Vel-Sparse) or the transition function (Walker-Params-Sparse and Hopper-Rand-Params). These tasks (except for Point-Robot-Sparse) are simulated via MuJoCo (Todorov et al., 2012) and are benchmarks commonly used by current meta-learning algorithms (Mishra et al., 2018; Finn et al., 2017; Rothfuss et al., 2019; Rakelly et al., 2019). Unlike previous evaluation settings, we limit the length of adaptation phase to investigate the efficiency of exploration. Also, dense reward is not provided in meta-training, which is different from the setting of PEARL. Detailed parameters and reward function settings are deferred to Appendix C.

Algorithm setup: MetaCURE is compared against several representative meta-RL algorithms, including PEARL (Rakelly et al., 2019), VariBAD (Zintgraf et al., 2019), RL^2 (Duan et al., 2016), MAML (Finn et al., 2017), ProMP (Rothfuss et al., 2019), E-RL^2 (Stadie et al., 2018) and MAME (Gurumurthy et al., 2020). We also compare with a variant of EPI (Zhou et al., 2018), which considers both dynamics predictions and reward predictions[2]. We use open-source codes provided by the original papers, and performance is averaged over six random seeds.

Results and analyses: algorithms' performance is evaluated by the last adaptation episode's return averaged over all meta-testing tasks. As shown in Figure 3, MetaCURE

[2] EPI is not trained end-to-end, and we plot its final performance in dash line.

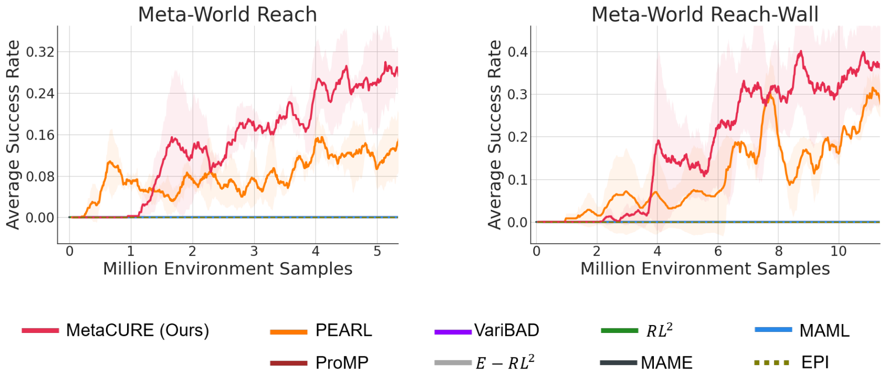


Figure 4. Evaluation of MetaCURE and meta-RL baselines on the challenging sparse-reward Meta-World task sets.

significantly outperforms baseline algorithms. PEARL fails to achieve satisfactory performance, as it utilizes posterior sampling (Thompson, 1933; Osband et al., 2013) for exploration, which is only optimal in asymptotic performance and may fail to effectively explore within short adaptation. In contrast, MetaCURE learns a separate policy for efficient exploration and achieves superior performance. RL^2 performs well in Point-Robot-Sparse, a task set with simple dynamics, but fails in more complex tasks. This is possibly because that RL^2 fails to effectively handle task uncertainty in complex sparse-reward tasks. MetaCURE achieves better performance in complex tasks by utilizing probabilistic task inference to model task uncertainty. The rest of the baselines perform poorly with sparse rewards. As for sample efficiency, MetaCURE and PEARL outperform other methods by realizing off-policy training. Note that MetaCURE and PEARL achieve similar sample efficiency, while MetaCURE learns two policies and PEARL only learns one. This superior sample efficiency is acquired by sharing the task inference component and training data.

5.2. Adaptation Performance on Meta-World

Meta-World (Yu et al., 2020) is a recently proposed challenging evaluation benchmark for meta-RL, including a variety of robot arm control tasks. We evaluate MetaCURE as well as baselines on two Meta-World task sets: Reach and Reach-Wall. To investigate the efficiency of exploration, we make the rewards sparse, providing non-zero rewards only when the agent succeeds in the task. This setting is extremely hard, as task information is very scarce. Following the original paper (Yu et al., 2020), we evaluate algorithms by their final success rates, and results are shown in Figure 4. MetaCURE achieves significantly higher success rates than baselines by achieving efficient exploration. Among the baselines, PEARL is the only algorithm that manages to solve the tasks, but its success rate is significantly lower than MetaCURE.

5.3. Adaptation Visualization

To prove that MetaCURE learns efficient exploration and exploitation strategies, we visualize MetaCURE's adaptation phase in Point-Robot-Sparse and Walker-Vel-Sparse, as shown in Figure 5 and 6, respectively. We compare against PEARL, which explores via posterior sampling.

In Point-Robot-Sparse (Figure 5), the agent needs to identify the task by exploring regions where the goal may exist, and then carry out exploitation behaviors. While MetaCURE efficiently explores to identify goal location and then exploits, PEARL only explores a small region every episode, as its policy only performs exploitation behaviors and can not explore effectively.

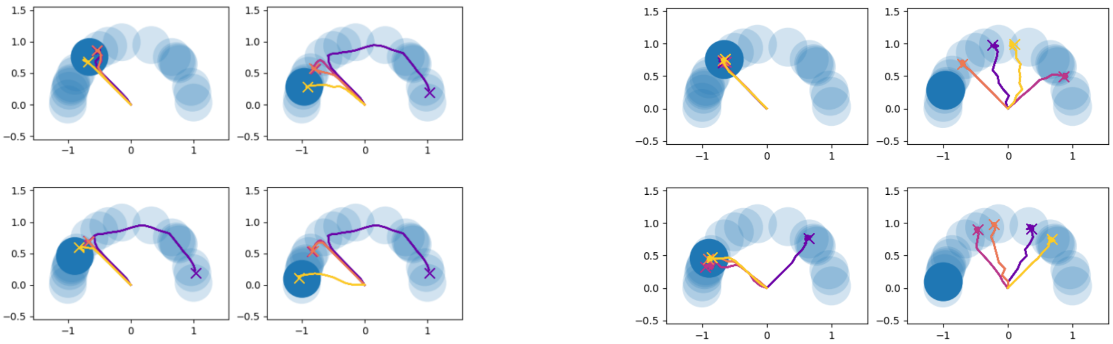
In Walker-Vel-Sparse (Figure 6), the agent needs to identify the goal velocity. MetaCURE covers possible velocities in the first episode with the exploration policy, and then reaches the goal velocity with the exploitation policy in the second episode. In contrast, PEARL only covers a small range of velocities in an episode. Additional visualization results are deferred to Appendix D.

5.4. Ablation Study

This section evaluates the essentiality of MetaCURE's components, including the intrinsic reward, separation of exploitation and exploration policies, and exploration policy using extrinsic rewards.

Intrinsic reward: to demonstrate the effectiveness of our intrinsic reward, we test a variant of MetaCURE that ablates the intrinsic reward. As shown in Figure 7, this variant suffers from a massive decrease in performance. This result shows that our intrinsic reward is critical for efficient exploration.

Exploitation policy: to investigate the effectiveness of learning separate exploration and exploitation policies, we



(a) MetaCURE

(b) PEARL

Figure 5. Adaptation visualization of (a) MetaCURE and (b) PEARL on Point-Robot-Sparse. The agent is given four episodes to perform adaptation. Purple lines indicate the first adaptation episode's trajectories, while light yellow lines indicate the last adaptation episode's trajectories. Dark blue circles represent rewarding regions for the current task, while light blue circles represent rewarding regions for other meta-testing tasks. MetaCURE achieves more efficient exploration.

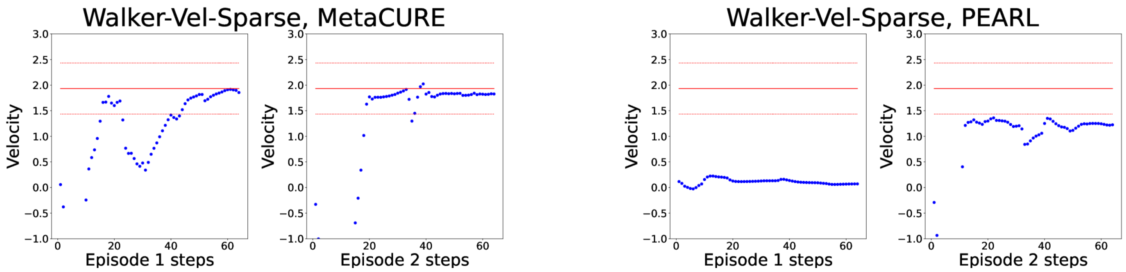


Figure 6. Visualization of MetaCURE and PEARL on Walker-Vel-Sparse. The agent is given two episodes to perform adaptation. The solid red line is the target velocity, and the region bounded by red dash lines represents velocities that get informative rewards. While PEARL tries to keep a certain velocity for an entire episode, MetaCURE first efficiently explores the goal velocity before performing exploitation behaviors.

test a variant of MetaCURE which does not obtain an exploitation policy, and utilizes rewards and dynamics prediction as the decoder. As shown in Figure 8, on Point-Robot-Sparse, this variant significantly underperforms the original MetaCURE. The exploitation policy is critical for superior performance, as it serves two important purposes: learning unbiased exploitation behaviors, and providing a decoding objective for training the context encoder.

Exploration policy using extrinsic rewards: as shown in Figure 9, we find that on Cheetah-Vel-Sparse, by adding extrinsic rewards to the exploration policy's reward function, MetaCURE achieves superior performance compared to the variant that only maximizes intrinsic rewards. This is because that extrinsic reward signals contain rich task information, and can effectively guide exploration.

Additional ablation studies on hyper-parameters, knowledge and experience sharing as well as a baseline with our intrinsic reward are deferred to Appendix E.

6. Related Work

Exploration in meta-RL: in contrast to meta supervised learning, in meta-RL (Schmidhuber, 1995; Finn et al., 2017) the agent is not given a task-specific dataset to adapt to, and it must explore the environment to collect useful information. This exploration part is vital for both meta-training and adaptation (Schmidhuber, 1997; Finn & Levine, 2019).

The problem of exploration policy learning in gradient-based meta-RL is mainly addressed by computing gradients to the pre-update policy's state distribution (Stadie et al., 2018; Rothfuss et al., 2019). MAESN (Gupta et al., 2018) introduces temporally-extended exploration with latent variables. These methods generally require dense rewards in meta-training. MAME (Gurumurthy et al., 2020) augments MAML (Finn et al., 2017) with a separate exploration policy, but its exploration policy learning shares the same objective with exploitation policy learning and is not effective for general sparse-reward tasks. A branch of context-based methods automatically learns to trade-off exploration and

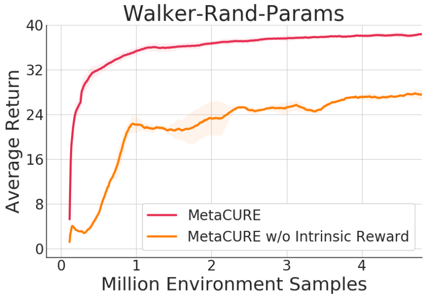


Figure 7. Ablation study on MetaCURE's intrinsic reward.

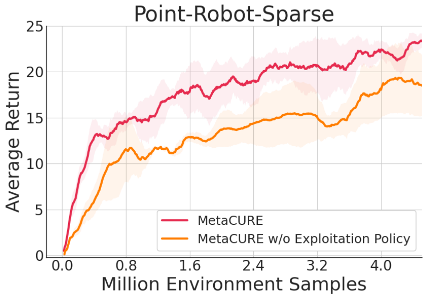


Figure 8. Ablation study on MetaCURE's exploitation policy.

exploitation by maximizing average adaptation performance (Duan et al., 2016; Zintgraf et al., 2019), and E-RL^2 (Stadie et al., 2018) directly optimizes for the final performance. PEARL (Rakelly et al., 2019) utilizes posterior sampling (Thompson, 1933; Osband et al., 2013) for exploration. EPI (Zhou et al., 2018) considers the setting of tasks with different dynamics, and introduces intrinsic rewards based on prediction improvement in dynamics. In contrast, we propose an empowerment-driven exploration objective aiming to maximize information gain about the current task, and derive a corresponding intrinsic reward to achieve efficient exploration in both meta-training and adaptation.

Exploration with information-theoretic intrinsic rewards: encouraging the agent to gain task information is a promising way to facilitate exploration (Storck et al., 1995). VIME (Houthooft et al., 2016) measures the mutual information between trajectories and the transition function, while EMI (Kim et al., 2019) measures the mutual information between state and action with the next state in consideration, as well as the mutual information between state and next state with the action in consideration, both in the latent space. Sun et al. (2011) discusses exploration with information gain, but is restricted to planning problems and requires an oracle estimating the posterior. These works focus on traditional RL. Our intrinsic reward maximizes the information gain for task identification during meta-RL's training and adaptation, which is different from previous works.

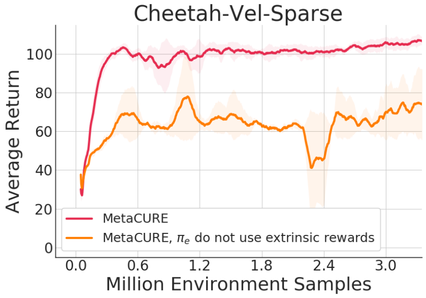


Figure 9. Ablation study on MetaCURE's exploration policy using extrinsic rewards.

Prediction loss as intrinsic reward: prediction losses can serve as a kind of curiosity-driven intrinsic reward to encourage exploration (Schmidhuber, 1991), and is widely used in traditional RL. Oh et al. (2015) directly predicts the image observation, and Stadie et al. (2015) utilizes prediction loss in the latent space, in order to focus on useful features extracted from observations. To avoid trivial solutions in learning the latent space, Pathak et al. (2017) introduces an inverse model to guide the learning of latent space, only predicting things that the agent can control. Burda et al. (2018a) utilizes random neural networks as projections onto the latent space and achieved superior performance on Atari games. Our work focuses on the exploration problem in meta-RL, and utilizes differences of model prediction errors as a means to measure information gain.

7. Conclusion

In this paper, to enable efficient meta-RL with sparse rewards, we explicitly model the problem of exploration policy learning, and propose a novel empowerment-driven exploration objective, which aims at maximizing agent's information gain about the current task. We derive a corresponding intrinsic reward from our exploration objective, and a didactic example shows that our intrinsic reward facilitates efficient exploration in both meta-training and adaptation. A new off-policy meta-RL algorithm called MetaCURE is also proposed, which incorporates our intrinsic reward and learns separate exploration and exploitation policies. MetaCURE achieves superior performance on various sparse-reward MuJoCo locomotion task sets as well as more difficult sparse-reward Meta-World tasks.

Acknowledgements

This work is supported in part by Science and Technology Innovation 2030 – "New Generation Artificial Intelligence" Major Project (No. 2018AAA0100904), and a grant from the Institute of Guo Qiang,Tsinghua University.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. Stochastic variational video prediction. In *International Conference on Learning Representations*, 2018.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2018a.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018b.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. A recurrent latent variable model for sequential data. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pp. 2980–2988, 2015.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl ^{2}: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Finn, C. and Levine, S. Meta-learning: from few-shot learning to rapid reinforcement learning. https://sites.google.com/view/icml19metalearning, 2019.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pp. 5302–5311, 2018.

Gurumurthy, S., Kumar, S., and Sycara, K. Mame: Model-agnostic meta-exploration. In *Conference on Robot Learning*, pp. 910–922. PMLR, 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1856–1865, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

Humplik, J., Galashov, A., Hasenclever, L., Ortega, P. A., Teh, Y. W., and Heess, N. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

Kim, H., Kim, J., Jeong, Y., Levine, S., and Song, H. O. Emi: Exploration with mutual information. In *International Conference on Machine Learning*, pp. 3360–3369, 2019.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Lee, K., Seo, Y., Lee, S., Lee, H., and Shin, J. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 5757–5766. PMLR, 2020.

Leike, J., Lattimore, T., Orseau, L., and Hutter, M. Thompson sampling is asymptotically optimal in general environments. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 417–426, 2016.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Oh, J., Guo, X., Lee, H., Lewis, R. L., and Singh, S. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pp. 2863–2871, 2015.

Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, pp. 5331–5340, 2019.

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., Abbeel, P., Shingarey, D., Kaul, L., Asfour, T., Athanasios, C. D., Zhou, Y., et al. Promp: Proximal meta-policy search. In *International Conference on Learning Representations*, volume 3, pp. 4007–4014, 2019.

Schmidhuber, J. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

Schmidhuber, J. On learning how to learn learning strategies. 1995.

Schmidhuber, J. What's interesting? 1997.

Stadie, B. C., Levine, S., and Abbeel, P. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Stadie, B. C., Yang, G., Houthooft, R., Chen, X., Duan, Y., Wu, Y., Abbeel, P., and Sutskever, I. Some considerations on learning to explore via meta-reinforcement learning. *ArXiv*, abs/1803.01118, 2018.

Storck, J., Hochreiter, S., and Schmidhuber, J. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, volume 2, pp. 159–164. Citeseer, 1995.

Sun, Y., Gomez, F., and Schmidhuber, J. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *International Conference on Artificial General Intelligence*, pp. 41–51. Springer, 2011.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pp. 1–5, 2019.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020.

Zhou, W., Pinto, L., and Gupta, A. Environment probing interaction policies. In *International Conference on Learning Representations*, 2018.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representations*, 2019.