Proceedings of Machine Learning Research vol 242:1174–1186, 2024

On Task-Relevant Loss Functions in Meta-Reinforcement Learning

Jaeuk Shin SJU5379@SNU.AC.KR
Department of Electrical and Computer Engineering, ASRI, Seoul National University

Giho Kim CHICIOUE512@SNU.AC.KR
Department of Electrical and Computer Engineering, ASRI, Seoul National University

Howon Lee HO1DOL2@SNU.AC.KR
Interdisciplinary Program in Artificial Intelligence, ASRI, Seoul National University

Joonho Han SNOWHAN1021@SNU.AC.KR
Department of Electrical and Computer Engineering, ASRI, Seoul National University

Insoon Yang INSOONYANG@SNU.AC.KR
Department of Electrical and Computer Engineering, ASRI, Seoul National University

Abstract

Designing a competent meta-reinforcement learning (meta-RL) algorithm in terms of data usage remains a central challenge to be tackled for its successful real-world applications. In this paper, we propose a sample-efficient meta-RL algorithm that learns a model of the system or environment at hand in a task-directed manner. As opposed to the standard model-based approaches to meta-RL, our method exploits the value information in order to rapidly capture the decision-critical part of the environment. The key component of our method is the loss function for learning both the task inference module and the system model. This systematically couples the model discrepancy and the value estimate, thereby enabling our proposed algorithm to learn the policy and task inference module with a significantly smaller amount of data compared to the existing meta-RL algorithms. The proposed method is evaluated in high-dimensional robotic control, empirically verifying its effectiveness in extracting information indispensable for solving the tasks from observations in a sample-efficient manner.

Keywords: Reinforcement learning, meta-reinforcement learning.

1. Introduction

Meta-reinforcement learning (meta-RL) has become widely recognized and accepted in the domain of decision-making, particularly for its capability to systematically adapt to environmental changes. The main strategy of meta-RL involves devising a mechanism for identifying the task-specific features from data and utilizing them to generate actions tailored to the task. Even though the field has elicited considerable research interest in the development of a task inference module that rapidly adapts to novel tasks, a critical challenge persists: the reduction of the severe sample complexity associated with meta-training, i.e., learning such a module along with the policy from data. This inefficiency is notably revealed in popular *model-free* meta-RL algorithms, including gradient-based methods (Finn et al., 2017; Gupta et al., 2018) and off-policy context-based methods (Rakelly et al., 2019; Bing et al., 2023), thereby impeding their application to complex real-world decision-making scenarios.

The standard RL community has produced a series of studies investigating the statistical advantages of model-based RL methods over model-free RL methods in either theoretical (Tu and Recht, 2019; Sun et al., 2019) or experimental (Pong et al., 2018) contexts. The core message from these works is that the model-based methods often enjoy better sample efficiency compared to the model-free methods. This insight has triggered the rapid development of *model-based meta-RL* methods to alleviate data dependency (Nagabandi et al., 2018; Sæmundsson et al., 2018; Galashov et al., 2019; Wang and Van Hoof, 2022). The methods focus on learning the full dynamics of the environment, regardless of which goals are specified for the agent. In Nagabandi et al. (2018), a neural network model is adapted online through gradient-based updates and deployed in a model predictive control (MPC) controller. However, using the gradient estimates is often problematic, especially when only a limited amount set of samples are used to construct the estimator. Accordingly, such a method requires a large amount of data for task inference, which is undesirable if rapid environmental adaptation is required.

In Sæmundsson et al. (2018), the Gaussian process (GP) is used to model the unknown system dynamics. Additionally, a context-based approach is adopted by introducing latent variables that represent task features, enabling online model inference. However, learning and inferring the model are done independently of the quality of the resulting policies. In Shin et al. (2022), MPC controllers with local GP models are employed in an event-based manner for generating high-quality action samples to derive an efficienct meta-RL algorithm, but again the models are learned without the consideration of the value information.

In model-free methods, there have been numerous attempts to define the notion of *task-relevant* information and to develop the methods that efficiently capture and exploit such information. The intuition behind these approaches is that the perfect description of the tasks may not be essential for optimal decision-making. For instance, PEARL (Rakelly et al., 2019) trains its task inference module to discover the Q-function with a small temporal difference (TD) error so that the task inference directly corresponds to identifying the optimal policy from the data. In Fu et al. (2021), the noisy nature of TD errors is highlighted as a main issue of performing such task inference, and contrastive learning-based task recognition method is proposed. Liu et al. (2021) formulates a mutual information minimization problem that may neglect any task-irrelevant information for detecting the optimal value function of each task. However, to the best of our knowledge, efficiently extracting task-relevant information has been sparsely studied in a model-based context. On the other hand, the effect of model mismatch on the policy performance has received attention in single-task model-based reinforcement learning (Lambert et al., 2020). To address the problem, *value-aware model learning* explicitly incorporates the value function information into model learning objectives, similar to our approach (Farahmand et al., 2017; Abachi, 2020; Voelcker et al., 2021). However, such methods do not take into account task variability.

To faithfully learn the decision-oriented models and accelerate the meta-learning procedure, we analyze the suboptimality of the model-based policy that emphasizes the quantification of the divergence in value prediction. This leads us to introduce *task-**r**elevant **m**eta-**r**einforcement **l**earning (TRMRL)*, a novel model-based method that systematically exploits reward information to learn task-adaptive models. Our method focuses on inferring the environmental model that precisely estimates the nominal values. This stems from the principle that the model does not have to exhaustively capture the full dynamics of the system; rather, only its capability of correctly predicting state values does matter for obtaining a near-optimal policy. Such an effort to recognize task-centric information promotes sample efficiency, thereby advancing the real-world applicability of meta-RL.

Our contributions are summarized as follows:

- We propose a theoretically principled loss function for learning task-directed models of unknown system components, namely the system dynamics and the reward function. The loss function is carefully designed through the analysis of the policy's suboptimality bound that involves the value estimation capability of the learned model.

- Using the loss function, we design a novel model-based meta-RL method that selectively learns the task-relevant models of an environment. The method uses a notably smaller amount of data than existing meta-RL methods to deliver comparable performance. This is achieved by training both the task inference module and the model of the task components based on the proposed loss function, enabling the task inference module to produce a policy adequate for a new task.
- Our meta-RL method is experimentally evaluated in a complex robotic control problem, where the physical properties of the robot and the environment vary across the tasks. The results of our experiments show that our method outperforms other state-of-the-art off-policy meta-RL methods in terms of sample efficiency and average return.

Throughout the paper, we let $\Delta(\mathcal{X})$ denote the space of all probability measures defined over a set $\mathcal{X}$. The notation δ_x is used to denote the Dirac measure on $x \in \mathcal{X}$. For $P \in \Delta(\mathcal{X})$ and $f \in L^1(\mathcal{X}, P)$, $\mathbb{E}_{x \sim P}[f(x)]$ denotes the expectation of f with respect to P.

2. Background and Motivation

Recent developments of data-driven sequential decision-making methods necessitate a principled method toward the efficient control of complex real-world systems, most of which involve systems characterized by environmental variability. In practice, such changes are hardly anticipated by decision-makers *a priori*, which forces the decision-making agents to deduce these changes from experimental data and adapt their policies accordingly. Consequently, modeling these problems as single-task problems is unreliable, primarily because using a policy calibrated for a single system model may lead to catastrophic outcomes when it faces environmental changes (Nagabandi et al., 2018). To systematically address the *multi-task* challenge of real-world scenarios, meta-RL has been actively studied (Wang et al., 2016; Duan et al., 2016; Finn et al., 2017; Beck et al., 2023; Beukman et al., 2024). Mathematically, meta-RL considers a collection $\mathcal{M}$ of *Markov decision processes (MDPs)* (Puterman, 2014), or a collection of *tasks*, equivalently, and a *task distribution* $\mathscr{T}$ over $\mathcal{M}$. Each task consists of data $M = \langle \mathcal{S}, \mathcal{A}, T^M, T_0, R^M, \gamma \rangle \in \mathcal{M}$ where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $T^M : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a stochastic transition kernel that describes the dynamics of the system, $T_0 \in \Delta(\mathcal{S})$ is an initial state distribution, $R^M : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, and $\gamma \in [0, 1)$ is a discount factor. In each episode, a task $M = \langle T^M, R^M \rangle$ is randomly drawn from $\mathscr{T}$, where a *policy* π determines an action $a_k \sim \pi(\cdot|h_k)$ to execute based on the history $h_k := (s_0, a_0, r_0, s_1, \ldots, s_{k-1}, a_{k-1}, r_{k-1}, s_k)$ of the states, actions, and rewards up to k. [1] Then, the goal of meta-RL is to solve the following policy optimization problem:

$$\sup_{\pi} \mathop{\mathbb{E}}_{M \sim \mathscr{T}} \mathbb{E}^{T_0, T^M, \pi} \left[\sum_{k=0}^{\infty} \gamma^k R^M(s_k, a_k) \right], \tag{1}$$

1. Formally, a policy $\pi : H \to \Delta(\mathcal{A})$ maps each history to a distribution over the action space, where H is defined as a set of possible histories:

$$H := \bigcup_{k \geq 0} H_k, \quad H_k := (\mathcal{S} \times \mathcal{A} \times \mathbb{R})^k \times \mathcal{S}.$$

where the inner expectation is taken over the process $s_0 \sim T_0$, $a_k \sim \pi(\cdot|h_k)$, $s_{k+1} \sim T^M(\cdot|s_k, a_k)$, $k = 0, 1, \ldots$. Therefore, the policy needs to act adaptively to an unknown MDP M as identifying essential information about M from the history. The problem (1) is formalized as a standard MDP called *Bayes-adaptive MDP (BAMDP)* (Duff, 2002) by introducing the *belief state* that represents the probabilistic estimate of the task components $\langle T^M, R^M \rangle$. Formally, a belief state b_k given a history h_k is defined as a posterior distribution $b(\cdot|h_k)$ over $\mathcal{M}$ determined via Bayes' theorem:

$$b_k(M) := b(M|h_k) \propto \mathscr{T}(M) \prod_{i=0}^{k-1} T^M(s_{i+1}|s_i, a_i) \mathbb{I}_{R^M(s_i,a_i)}(r_i), \tag{2}$$

where $\mathbb{I}_{R^M(s_i,a_i)}$ is the indicator function for $R^M(s_i, a_i)$, and a policy $\pi(s_k, b_k)$ takes s_k and b_k as its inputs in place of the entire history h_k to yield an action a_k. Unfortunately, BAMDPs are generally intractable, as both *(i)* performing the update (2) and *(ii)* computing the near-optimal value functions are computationally demanding when the task distribution is high-dimensional. Instead, the posterior distribution in (2) is often approximated as a distribution over a low-dimensional *latent space* where each latent vector corresponds to a task. Such a method is called *context-based meta-RL* (Rakelly et al., 2019; Zintgraf et al., 2019; Zhang et al., 2020; Wang et al., 2023). In context-based meta-RL, each MDP M is represented as a *context variable* $\mathbf{z} \in Z := \mathbb{R}^d$ by a task inference module called a *context encoder* $\mathbf{enc}$. The encoder consumes data $\mathcal{D} = \{(s_i, a_i, r_i, s_{i+1})\}$ from M and generates a latent distribution $\mathcal{P}(d\mathbf{z}) = \mathbf{enc}(\mathcal{D}) \in \Delta(Z)$. This can be understood as a proxy of the posterior distribution (2) inferred from $\mathcal{D}$ but in the latent space Z instead of $\mathcal{M}$. Then, a context variable $\mathbf{z}$ is sampled from $\mathcal{P}(d\mathbf{z})$ and used as an input of a policy $\pi_{\mathbf{z}}(s)$ or a model $\langle T_{\mathbf{z}}(s'|s, a), R_{\mathbf{z}}(s, a) \rangle$, enabling us to perform task-adaptive actions. The encoder may be jointly trained with a policy in a way that the resulting policy $\pi_{\mathbf{z}}(s)$ is near-optimal for M (Rakelly et al., 2019) or the likelihood of sample trajectories is maximized under $T_{\mathbf{z}}(s'|s, a)$ and $R_{\mathbf{z}}(s, a)$ via approximate inference (Zintgraf et al., 2019; Wang et al., 2023). These methods are more data-efficient than other meta-RL methods that rely on gradient-based searches for policy adaptation (Finn et al., 2017; Gupta et al., 2018) since the task inference can be done with a relatively small amount of data and computational resources. Nevertheless, they suffer from sample inefficiency when it comes to meta-training the policies and the task inference modules, mainly because they are model-free algorithms. On the other hand, there have been attempts to leverage techniques from model-based reinforcement learning to resolve the problem (Perez et al., 2018; Nagabandi et al., 2018; Sæmundsson et al., 2018; Belkhale et al., 2021), but the central question of how to exploit the reward information for learning the models has not yet been appropriately addressed.

3. Task-Relevant Loss Functions

In this section, we tackle the suggested problem of sample inefficiency by integrating the value function information into the model learning process via a specifically designed *task-relevant loss function*. We begin with a theoretical motivation for the use of the task-relevant loss function. The loss function will then be adapted in the meta-RL framework to jointly train the model and the encoder. Our loss function is inspired by the following theorem, which can be viewed as a generalization of Theorem 7 of Pires and Szepesvári (2016):

Theorem 1 *Suppose that $\{M_{\mathbf{z}} = \langle T_{\mathbf{z}}, R_{\mathbf{z}} \rangle\}_{\mathbf{z} \in Z}$ is a collection of tasks parameterized by $\mathbf{z} \in Z$, and $\{V_{\mathbf{z}} : \mathcal{S} \to \mathbb{R}\}_{\mathbf{z} \in Z}$ is a collection of value functions satisfying $\|V^\star_{M_{\mathbf{z}}} - V_{\mathbf{z}}\|_\infty \le \varepsilon$ for each $M_{\mathbf{z}}$,* *where $V^\star_{M_{\mathbf{z}}}$ denotes the optimal value function for $M_{\mathbf{z}}$. Let V^π_M be the value function of π for M, and $(TV)(s,a) \coloneqq \mathbb{E}_{s'\sim T(\cdot|s,a)}\left[V(s')\right]$. If $\pi_{\mathbf{z}}$ is greedily constructed from $M_{\mathbf{z}}$ and $V_{\mathbf{z}}$, i.e.,*

$$\pi_{\mathbf{z}}(s) \in \arg\max_a \Big(R_{\mathbf{z}}(s,a) + \gamma \mathbb{E}_{s'\sim T_{\mathbf{z}}(\cdot|s,a)}\left[V_{\mathbf{z}}(s')\right] \Big), \quad s \in \mathcal{S},$$

then for any task $M = \langle T, R\rangle \in \mathcal{M}$ and $\mathbf{z} \in Z$, the following inequality holds:

$$\left\|V^\star_M - V^{\pi_{\mathbf{z}}}_M\right\|_\infty \leq \underbrace{\frac{2}{1-\gamma}\left\|(R - R_{\mathbf{z}}) + \gamma\left(T - T_{\mathbf{z}}\right)V_{\mathbf{z}}\right\|_\infty}_{\textit{task inference error}} + \underbrace{\frac{(4+2\gamma(1-\gamma))\varepsilon}{(1-\gamma)^2}}_{\textit{planning error}}. \tag{3}$$

The proof of Theorem 1 is postponed to Section 3.2. The error bound (3) uncovers an intricate relationship between the quality of the learned model and the suboptimality of the policy: Quantifying a model mismatch without considering the value might be conservative. Intuitively, Theorem 1 suggests that two conditions suffice for adaptively seeking an efficient policy: *(i)* The encoder generates $\mathbf{z}$ whose associated model $\langle T_{\mathbf{z}}, R_{\mathbf{z}}\rangle$ aligns accurately with the value across states and actions rather than precisely replicating the transitions, and *(ii)* $V_{\mathbf{z}}$ and $\pi_{\mathbf{z}}$ are optimized with respect to the model. Indeed, guessing the value of $R + \gamma T V_{\mathbf{z}}$ hinges on matching the expectation of $V_{\mathbf{z}}$ over $T(\cdot|s,a)$, which is markedly less demanding than precisely estimating the entire transition probability distribution. Accordingly, (3) motivates a scheme that learns both the model and the inference module to reduce the task inference error, while refining the value function and the policy on the model to mitigate the planning error. This enables us to rapidly obtain the model sufficient for generating a high-performance policy even if it is not capable of exactly matching the state transitions. Furthermore, the task inference error in (3) does not depend on the optimal value function, thus the quantity $(R - R_{\mathbf{z}}) + \gamma\left(T - T_{\mathbf{z}}\right)V_{\mathbf{z}}$ can be explicitly constructed in practice.

3.1. Meta-RL algorithm

To design a sample-efficient meta-RL algorithm, we exploit Theorem 1 to propose a novel method called *task-relevant meta-reinforcement learning (TRMRL)*, a model-based meta-RL algorithm that effectively extracts task-relevant information from data. As a model-based method, TRMRL maintains the models $T_{\theta,\mathbf{z}}(s'|s,a)$ and $R_{\theta,\mathbf{z}}(s,a)$ of the MDP dynamics and the reward function, respectively, where θ represents learnable parameters, and $\mathbf{z}$ is the context variable inferred by the encoder $\mathbf{enc}$ from transitions $\mathcal{D}$. Given the models $\langle T_\theta, R_\theta\rangle$, a common existing approach of updating them involves minimizing the least square loss (or maximizing the log-likelihood counterpart) of another dataset $\mathcal{D}'$ from M. For instance, the loss function for updating $\langle T_\theta, R_\theta\rangle$ and $\mathbf{enc}_\phi$ might be expressed as

$$\mathcal{L}_T(\theta,\phi) \coloneqq \frac{1}{|\mathcal{D}'|}\sum_{(s,a,s')\in\mathcal{D}'} \log \mathbb{E}[T_{\theta,\mathbf{z}}(s'|s,a)], \quad \mathcal{L}_R(\theta,\phi) \coloneqq \frac{1}{|\mathcal{D}'|}\sum_{(s,a,r)\in\mathcal{D}'} (r - \mathbb{E}[R_{\theta,\mathbf{z}}(s,a)])^2 \tag{4}$$

where the expectations are taken over $\mathbf{z} \sim \mathbf{enc}_\phi(\mathcal{D})$. A limitation of using (4) is that it is completely independent of the structure of the underlying decision-making problem. To address this issue, TRMRL takes the task inference error of (3) as the loss function for both the encoder $\mathbf{enc}_\phi$ and the model $\langle T_\theta, R_\theta\rangle$. It is logical since Theorem 1 provides a clear answer to the role of the encoder for

Algorithm 1 Task-relevant meta-reinforcement learning (TRMRL)

1: **Input**: task batch size $N_{\text{task}} > 0$, training task distribution $\mathscr{T}_{\text{train}}$, N_{episode}, prior latent distribution $\mathcal{P}_0 = \mathcal{N}(0, I)$, replay buffer $\mathcal{B}_i$ for each training task M_i
2: **for** $k = 1, 2, \ldots$ **do**
3:   `// data collection`
4:   **for** $i = 1, \ldots, N_{\text{task}}$ **do**
5:     Sample a training task $M_i \sim \mathscr{T}_{\text{train}}$ and randomly draw a latent vector $\mathbf{z}_0 \sim \mathcal{P}_0(d\mathbf{z})$;
6:     **for** episode $j = 0, \ldots, N_{\text{episode}} - 1$ **do**
7:       Collect a trajectory τ^j by executing $\pi_{\psi',\mathbf{z}}(a|s_j)$;
8:       Add τ^j to the replay buffer $\mathcal{B}_i$;
9:       Infer $\mathcal{P}_{j+1}(d\mathbf{z}) \coloneqq \mathbf{enc}_\phi(\tau^0, \ldots, \tau^j)$ and sample $\mathbf{z}_{j+1} \sim \mathcal{P}_{j+1}(d\mathbf{z})$;
10:     **end for**
11:   **end for**
12:   `// training`
13:   Randomly sample $\mathbf{z} \sim \mathcal{P}_0$ & randomly generate state-action pairs $B = \{(s^\ell, a^\ell)\}$;
14:   Learn $\pi_{\psi',\mathbf{z}}$ and $V_{\psi,\mathbf{z}}$ for the model $M_{\theta,\mathbf{z}}$ using B;
15:   Construct task inference loss (5) by sampling $\mathcal{D}$ & $\mathcal{D}'$ from $\mathcal{B}_i$'s;
16:   Perform the gradient descent updates to learn $\mathbf{enc}_\phi$ and $\langle T_\theta, R_\theta \rangle$:

$$\theta \longleftarrow \theta - \alpha_\theta \nabla_\theta \mathcal{L}(\theta, \phi), \qquad \phi \longleftarrow \phi - \alpha_\phi \nabla_\phi \mathcal{L}(\theta, \phi).$$

17: **end for**

obtaining a task-adaptive policy: The encoder can be trained so that, given $\mathcal{D}$, it selects $\mathbf{z}$ which minimizes the task inference error. This motivates using the following type of loss functions

$$\boxed{\mathcal{L}(\theta, \phi) \coloneqq \mathop{\mathbb{E}}_{\substack{M \sim \mathscr{T}_{\text{train}} \\ \mathcal{D}, \mathcal{D}' \sim M}} \left\{ \mathop{\mathbb{E}}_{\substack{\mathbf{z} \sim \mathbf{enc}_\phi(\mathcal{D}) \\ (s,a,r,s') \sim \mathcal{D}'}} \left(r - R_{\theta,\mathbf{z}}(s,a) + \gamma V_{\psi,\mathbf{z}}(s') - \gamma \mathop{\mathbb{E}}_{T_{\theta,\mathbf{z}}(s'|s,a)} V_{\psi,\mathbf{z}}(s') \right)^2 \right\}} \tag{5}$$

where $\mathscr{T}_{\text{train}}$ denotes the training task distribution, V_ψ represents the approximate value function with learnable parameters ψ, $\mathcal{D}$ is for task inference, and $\mathcal{D}'$ is for evaluating $(R - R_{\mathbf{z}}) + \gamma\,(T - T_{\mathbf{z}})\,V_{\mathbf{z}}$ over a subset of $\mathcal{S} \times \mathcal{A}$. Note that the terms r and $V_{\psi,\mathbf{z}}(s')$ in (5) serve as approximations for $R(s,a)$ and $\mathbb{E}_{T(s'|s,a)}[V_{\psi,\mathbf{z}}(s')]$, respectively. A similar observation has been exploited to design a loss function for model learning (Farahmand, 2018) in the standard RL setting, where only the model of the system dynamics is uncertain.

The outline of TRMRL is given in Algorithm 1. To collect data, a set of training tasks is sampled, and the latent variable $\mathbf{z}_0$ is drawn from the prior distribution $\mathcal{P}_0(d\mathbf{z})$. The prior distribution may be thought of as representing the entire task distribution, as no information about a given task is assigned. After each trajectory τ^j is collected by executing the policy $\pi_{\psi',\mathbf{z}_j}(a|s)$, the posterior distribution is updated to $\mathcal{P}_{j+1}(d\mathbf{z})$ by the encoder $\mathbf{enc}_\phi$ and a new latent vector $\mathbf{z}_{j+1}$ is assigned to the policy. When training, the policy $\pi_{\mathbf{z}}$ and the value function $V_{\mathbf{z}}$ for each task $M_{\mathbf{z}}$ are learned through planning (lines 13–14). In particular, training the policy and the value functions may be done using synthetic data generated from the model $T_{\theta,\mathbf{z}}$ and $R_{\theta,\mathbf{z}}$, thereby taking advantage of the sample efficiency of model-based methods. In practice, we apply a Dyna-style method (Sutton, 1991; Munos and Szepesvári, 2008) [2] to learn $V_{\psi,\mathbf{z}}(s)$ and $\pi_{\psi',\mathbf{z}}(s)$ of the entropy-regularized MDPs (Haarnoja et al., 2018; Geist et al., 2019). The reason for planning on the entropy-regularized MDPs is that the learned policies are capable of exploring the state and action spaces and therefore collecting data sufficiently diverse for task identification. Finally, the model $\langle T_\theta, R_\theta\rangle$ and the encoder $\mathbf{enc}_\phi$ are trained using the task-relevant loss function (5) (lines 15–16), which makes it possible to preemptively learn value-critical parts of the system dynamics. To summarize, TRMRL is an end-to-end method that reflects the inequality (3): It alternates between planning over the learned models and using data to jointly learn the encoder and the models.

3.2. Proof of Theorem 1

We now present the proof of Theorem 1, which is the main theoretical result.

Proof Throughout the proof, we suppress the subscript ∞ of $\|\cdot\|_\infty$ for clarity. To begin with, we introduce the following notation for the Bellman operators: $(F_M^\pi V)(s) := \int_{a\in\mathcal{A}} \pi(da|s)\big(R(s,a) + \gamma TV(s,a)\big)$ and $(F_M V)(s) := \sup_a \big(R(s,a) + \gamma TV(s,a)\big)$.

Applying the triangle inequality, the suboptimality gap can be bounded by the sum of two types of errors as follows:

$$\|V_M^\star - V_M^{\pi_\mathbf{z}}\| \le \underbrace{\|V_{M_\mathbf{z}}^{\pi_\mathbf{z}} - V_M^{\pi_\mathbf{z}}\| + \|V_M^\star - V_{M_\mathbf{z}}^\star\|}_{\text{task mismatch error}} + \underbrace{\|V_{M_\mathbf{z}}^\star - V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\|}_{\text{planning error}}.$$

First, the planning error is bounded using (Bertsekas and Tsitsiklis, 1996, Proposition 6.1) as follows:

$$\|V_{M_\mathbf{z}}^\star - V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\| \le \frac{2\gamma}{1-\gamma}\underbrace{\|V_{M_\mathbf{z}}^\star - V_\mathbf{z}\|}_{\leqslant\varepsilon} \le \frac{2\gamma\varepsilon}{1-\gamma}, \tag{A.1}$$

where we use the assumption that the approximate value function $V_\mathbf{z}$ differs from the optimal one $V_{M_\mathbf{z}}^\star$ by at most ε in terms of ℓ^∞-norm. Furthermore, the first term of the task mismatch error, which measures the variation of the policy's value across the tasks, is bounded as follows:

$$\begin{aligned}
\|V_M^{\pi_\mathbf{z}} - V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\| &= \|F_M^{\pi_\mathbf{z}}(V_M^{\pi_\mathbf{z}}) - F_{M_\mathbf{z}}^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}})\| \\
&\le \|F_M^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}}) - F_{M_\mathbf{z}}^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}})\| + \|F_M^{\pi_\mathbf{z}}(V_M^{\pi_\mathbf{z}}) - F_M^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}})\| \\
&\le \|F_M^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}}) - F_{M_\mathbf{z}}^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}})\| + \gamma\|V_M^{\pi_\mathbf{z}} - V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\|, \\
\Longrightarrow \quad \|V_M^{\pi_\mathbf{z}} - V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\| &\le \frac{1}{1-\gamma}\underbrace{\|F_M^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}}) - F_{M_\mathbf{z}}^{\pi_\mathbf{z}}(V_{M_\mathbf{z}}^{\pi_\mathbf{z}})\|}_{:=(A)}
\end{aligned}$$

since $V_M^{\pi_\mathbf{z}}$ and $V_{M_\mathbf{z}}^{\pi_\mathbf{z}}$ are the fixed points of $F_M^{\pi_\mathbf{z}}$ and $F_{M_\mathbf{z}}^{\pi_\mathbf{z}}$, respectively, and $F_M^{\pi_\mathbf{z}}$ is a γ-contraction. Note that

$$\begin{aligned}
(A) &= \sup_s \left|\int_\mathcal{A} \big(R(s,a) - R(s,a,\mathbf{z}) + \gamma(T - T_\mathbf{z})V_{M_\mathbf{z}}^{\pi_\mathbf{z}}(s,a)\big)\,\pi(da|s,\mathbf{z})\right| \\
&\le \sup_s \int_\mathcal{A} \left|R(s,a) - R(s,a,\mathbf{z}) + \gamma(T - T_\mathbf{z})V_{M_\mathbf{z}}^{\pi_\mathbf{z}}(s,a)\right|\pi(da|s,\mathbf{z}) \\
&\le \sup_{s,a} \left|R(s,a) - R(s,a,\mathbf{z}) + \gamma(T - T_\mathbf{z})V_{M_\mathbf{z}}^{\pi_\mathbf{z}}(s,a)\right| \\
&= \left\|(R - R_\mathbf{z}) + \gamma\,(T - T_\mathbf{z})\,V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\right\|,
\end{aligned}$$

2. There are possible alternatives, such as using model predictive control (Lowrey et al., 2018).

where we use $\int_{\mathcal{A}} \pi(da|s,\mathbf{z}) = 1$ for all $s \in \mathcal{S}$. Thus, we have

$$\|V_M^{\pi_\mathbf{z}} - V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\| \leq \frac{1}{1-\gamma} \left\|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\right\|.$$

Similarly, we bound the second term of the task mismatch error concerning the dependency of the optimal value function with respect to the choice of a task as follows:

$$\begin{aligned}
\|V_M^\star - V_{M_\mathbf{z}}^\star\| &= \|F_M(V_M^\star) - F_{M_\mathbf{z}}(V_{M_\mathbf{z}}^\star)\| \\
&\leq \|F_M(V_M^\star) - F_M(V_{M_\mathbf{z}}^\star)\| + \|F_M(V_{M_\mathbf{z}}^\star) - F_{M_\mathbf{z}}(V_{M_\mathbf{z}}^\star)\| \\
&\leq \gamma\|V_M^\star - V_{M_\mathbf{z}}^\star\| + \|F_M(V_{M_\mathbf{z}}^\star) - F_{M_\mathbf{z}}(V_{M_\mathbf{z}}^\star)\| \\
\implies \quad \|V_M^\star - V_{M_\mathbf{z}}^\star\| &\leq \frac{1}{1-\gamma} \underbrace{\|F_M(V_{M_\mathbf{z}}^\star) - F_{M_\mathbf{z}}(V_{M_\mathbf{z}}^\star)\|}_{:=(B)}.
\end{aligned}$$

Then, we deduce

$$\begin{aligned}
(B) &= \sup_s \left| \sup_a \left(R(s,a) + \gamma T V_{M_\mathbf{z}}^\star(s,a)\right) - \sup_a \left(R(s,a,\mathbf{z}) + \gamma T_\mathbf{z} V_{M_\mathbf{z}}^\star(s,a)\right) \right| \\
&\leq \sup_s \left| \sup_a \left(R(s,a) - R(s,a,\mathbf{z}) + \gamma(T - T_\mathbf{z}) V_{M_\mathbf{z}}^\star(s,a)\right) \right| \\
&\leq \sup_{s,a} \left| R(s,a) - R(s,a,\mathbf{z}) + \gamma(T - T_\mathbf{z}) V_{M_\mathbf{z}}^\star(s,a) \right| \\
&= \left\|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_{M_\mathbf{z}}^\star\right\|,
\end{aligned}$$

which implies

$$\|V_M^\star - V_{M_\mathbf{z}}^\star\| \leq \frac{1}{1-\gamma} \left\|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_{M_\mathbf{z}}^\star\right\|.$$

Since $\|V_{M_\mathbf{z}}^\star - V_\mathbf{z}\| \leq \varepsilon$, we have

$$\begin{aligned}
\left\|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_{M_\mathbf{z}}^\star\right\| &\leq \|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_\mathbf{z}\| \\
&\quad + \left\|\left(T - T_\mathbf{z}\right)\left(V_{M_\mathbf{z}}^\star - V_\mathbf{z}\right)\right\| \\
&\leq \|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_\mathbf{z}\| + 2\varepsilon \\
\implies \quad \|V_M^\star - V_{M_\mathbf{z}}^\star\| &\leq \frac{1}{1-\gamma} \|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_\mathbf{z}\| + \frac{2\varepsilon}{1-\gamma}.
\end{aligned} \tag{A.2}$$

Furthermore, an analogous argument leads to

$$\begin{aligned}
\left\|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\right\| &\leq \|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_\mathbf{z}\| + 2\frac{1+\gamma}{1-\gamma}\varepsilon \\
\implies \quad \|V_M^{\pi_\mathbf{z}} - V_{M_\mathbf{z}}^{\pi_\mathbf{z}}\| &\leq \frac{1}{1-\gamma} \|(R - R_\mathbf{z}) + \gamma\left(T - T_\mathbf{z}\right) V_\mathbf{z}\| + \frac{2(1+\gamma)\varepsilon}{(1-\gamma)^2}
\end{aligned} \tag{A.3}$$

because $\|V_{M_\mathbf{z}}^{\pi_\mathbf{z}} - V_\mathbf{z}\|_\infty \leq \frac{1+\gamma}{1-\gamma}\varepsilon$, which can be shown by using a technique similar to the one used in (Bertsekas and Tsitsiklis, 1996, Proposition 6.1). Finally, combining (A.1), (A.2), and (A.3), we obtain the desired result. ∎

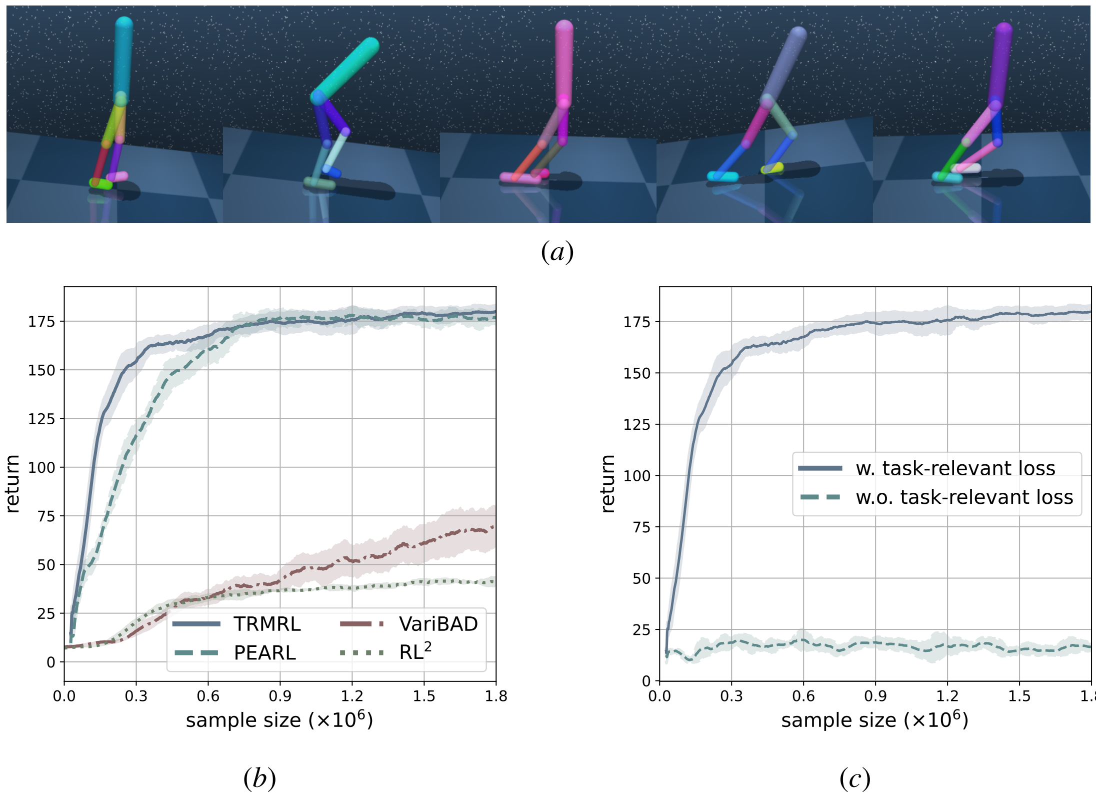

Figure 1: *(a)* Illustration of the bipedal walker control problem with varying physical parameters. *(b)* Comparison between the proposed method and the baseline methods. *(c)* Demonstration of the effectiveness of the task-relevant loss. All methods are evaluated for 200 steps in each test task, and their returns are plotted in the y-axis. As the reward function of each task is designed to take values in $[0, 1]$, an upper bound of the return per episode is 200.

4. Empirical Evaluation

In this section, we empirically validate the effectiveness of using the task-relevant loss function and test the proposed methods through meta-RL problems. The source code of our TRMRL implementation is available online: `https://github.com/CORE-SNU/TRMRL`. We evaluate TRMRL on a robotic control problem, where the physical properties of the system vary across tasks. Specifically, we consider the bipedal walker modeled in DEEPMIND CONTROL SUITE (Tunyasuvunakool et al., 2020) and simulated by MUJOCO (Todorov et al., 2012). The robot has the state space $\mathcal{S} \subseteq \mathbb{R}^{24}$ and the action space $\mathcal{A} = [-1, 1]^6$, and its objective is to walk along a sloped terrain at a steady speed of 1*m/s*. However, the physical parameters governing the dynamics of the robot, such as its *link densities, foot length, joint damping coefficients, link friction coefficients*, and the *slope angle of the terrain*, differ for each task as shown in Figure 1*(a)*. Each physical parameter, except for the slope angle, is randomly perturbed from its default value by multiplying it with the random variable X, where $\log_{1.3}(X)$ follows the uniform distribution $\mathbf{UNIFORM}[-3, 3]$. Meanwhile, the slope angle is drawn from $\mathbf{UNIFORM}[-0.2, 0.2]$. Overall, 120 training tasks are generated, and RL agents are evaluated on 5 test tasks that are different from the training tasks. To train the feed-forward neural networks $\langle T_{\theta,\mathbf{z}}, R_{\theta,\mathbf{z}} \rangle$, $V_{\psi,\mathbf{z}}$, $\pi_{\psi'\mathbf{z}}$, and $\mathbf{enc}_\phi$, we use Adam optimizer (Kingma and Ba, 2015) with learning rates $\alpha_\theta = 10^{-3}$ and $\alpha_\psi = \alpha_{\psi'} = \alpha_\phi = 3 \times 10^{-4}$.

Three state-of-the-art meta-RL algorithms, RL^2 (Duan et al., 2016), PEARL (Rakelly et al., 2019), and VariBAD (Zintgraf et al., 2019), are used for performance comparison. Figure 1(*b*) compares the performance of the proposed methods and the baseline algorithms. As a model-based method, TRMRL enjoys sample efficiency, exhibiting a faster learning speed than PEARL, in addition to outperforming the on-policy methods VariBAD and RL^2. Specifically, the quality of TRMRL policies dramatically improves until 400,000 samples are collected, which amounts to less than 4 hours of operating the robot at 40 *Hz*. In addition to its learning efficiency, TRMRL is comparable to PEARL concerning the quality of the learned controllers. Finally, Figure 1(*c*) illustrates the advantage of using the task-relevant loss by comparing our method and its counterpart that uses the least square loss functions (4). The latter version does not improve the quality of the learned policies at all. In particular, the result illustrates the difficulty of learning the reward models and the dynamics models separately via the least square loss functions (4) when both models are conditioned on the same latent variable. The inconsistency arises because the context encoder has to be trained on two losses, one for the reward model and one for the dynamics model, and these losses have different scales. This subtle issue complicates the design of model-based algorithms using (4) in the meta-RL context, making the learning process more challenging. TRMRL simply gets around this issue by exploiting the proposed task-relevant loss function (5).

5. Conclusion

To alleviate the issue of sample inefficiency in meta-RL, we proposed TRMRL, a model-based method that uses a carefully designed task-relevant loss function for both the task inference module and the system or environment model. This was inspired by the policy suboptimality bound that indicates the significance of measuring the value function discrepancy for learning the environmental model. The efficiency of the strategy was demonstrated by empirically evaluating it in the bipedal walker control problem under large environmental changes. Among many future research directions, we plan to extend our method to directly integrate raw sensor data and apply it to real-world robotic systems.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea funded by MSIT (2020R1C1C1009766), and the Information and Communications Technology Planning and Evaluation (IITP) grant funded by MSIT(2022-0-00124, 2022-0-00480).

References

Romina Abachi. *Policy-aware model learning for policy gradient methods*. University of Toronto (Canada), 2020.

Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

Suneel Belkhale, Rachel Li, Gregory Kahn, Rowan McAllister, Roberto Calandra, and Sergey Levine. Model-based meta-reinforcement learning for flight with suspended payloads. *IEEE Robotics and Automation Letters*, 6(2):1471–1478, 2021.

Dimitri Bertsekas and John N Tsitsiklis. *Neuro-dynamic Programming*. Athena Scientific, 1996.

Michael Beukman, Devon Jarvis, Richard Klein, Steven James, and Benjamin Rosman. Dynamics generalisation in reinforcement learning via adaptive context-aware policies. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhenshan Bing, Lukas Knak, Long Cheng, Fabrice O Morin, Kai Huang, and Alois Knoll. Meta-reinforcement learning in nonstationary and nonparametric environments. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

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.

Michael O'Gordon Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002.

Amir-massoud Farahmand. Iterative value-aware model learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

Haotian Fu, Hongyao Tang, Jianye Hao, Chen Chen, Xidong Feng, Dong Li, and Wulong Liu. Towards effective context for meta-reinforcement learning: an approach based on contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7457–7465, 2021.

Alexandre Galashov, Jonathan Schwarz, Hyunjik Kim, Marta Garnelo, David Saxton, Pushmeet Kohli, SM Eslami, and Yee Whye Teh. Meta-learning surrogate models for sequential decision making. *arXiv preprint arXiv:1903.11907*, 2019.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *Advances in Neural Information Processing Systems*, 31, 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*, pages 1861–1870. PMLR, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Learning for Dynamics and Control*, pages 761–770, 2020.

Evan Z Liu, Aditi Raghunathan, Percy Liang, and Chelsea Finn. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *International Conference on Machine Learning*, pages 6925–6935. PMLR, 2021.

Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. *arXiv preprint arXiv:1811.01848*, 2018.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2018.

Christian F Perez, Felipe Petroski Such, and Theofanis Karaletsos. Efficient transfer learning and online adaptation with latent variable models for continuous control. *arXiv preprint arXiv:1812.03399*, 2018.

Bernardo Ávila Pires and Csaba Szepesvári. Policy error bounds for model-based reinforcement learning with factored linear models. In *Conference on Learning Theory*, pages 121–151. PMLR, 2016.

Vitchyr Pong, Shixiang Gu, Murtaza Dalal, and Sergey Levine. Temporal difference models: Model-free deep RL for model-based control. In *International Conference on Learning Representations*, 2018.

Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, pages 5331–5340. PMLR, 2019.

S Sæmundsson, K Hofmann, and MP Deisenroth. Meta reinforcement learning with latent variable Gaussian processes. In *34th Conference on Uncertainty in Artificial Intelligence*, volume 34, pages 642–652. AUAI, 2018.

Jaeuk Shin, Astghik Hakobyan, Mingyu Park, Yeoneung Kim, Gihun Kim, and Insoon Yang. Infusing model predictive control into meta-reinforcement learning for mobile robots in dynamic environments. *IEEE Robotics and Automation Letters*, 7(4):10065–10072, 2022.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933. PMLR, 2019.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

Stephen Tu and Benjamin Recht. The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint. In *Conference on Learning Theory*, pages 3036–3083. PMLR, 2019.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.org/10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

Claas A Voelcker, Victor Liao, Animesh Garg, and Amir-massoud Farahmand. Value gradient weighted model-based reinforcement learning. In *International Conference on Learning Representations*, 2021.

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.

Mingyang Wang, Zhenshan Bing, Xiangtong Yao, Shuai Wang, Huang Kai, Hang Su, Chenguang Yang, and Alois Knoll. Meta-reinforcement learning based on self-supervised task representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10157–10165, 2023.

Qi Wang and Herke Van Hoof. Model-based meta reinforcement learning using graph structured surrogate models and amortized policy search. In *International Conference on Machine Learning*, pages 23055–23077. PMLR, 2022.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2020.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A very good method for bayes-adaptive deep RL via meta-learning. In *International Conference on Learning Representations*, 2019.