MetaCARD: Meta-Reinforcement Learning with Task Uncertainty Feedback via Decoupled Context-Aware Reward and Dynamics Components

Min Wang[1], Xin Li[1]*, Leiji Zhang[1], Mingzhong Wang[2]

[1] Beijing Institute of Technology
[2] University of the Sunshine Coast
{minwangcs, xinli, ljzhang}@bit.edu.cn, mwang@usc.edu.au

Abstract

Meta-Reinforcement Learning (Meta-RL) aims to reveal shared characteristics in dynamics and reward functions across diverse training tasks. This objective is achieved by meta-learning a policy that is conditioned on task representations with encoded trajectory data or context, thus allowing rapid adaptation to new tasks from a known task distribution. However, since the trajectory data generated by the policy may be biased, the task inference module tends to form spurious correlations between trajectory data and specific tasks, thereby leading to poor adaptation to new tasks. To address this issue, we propose the Meta-RL with task unCertAinty feedback through decoupled context-aware Reward and Dynamics components (MetaCARD). MetaCARD distinctly decouples the dynamics and rewards when inferring tasks and learning the policy, and integrates task uncertainty feedback from policy evaluation into the task inference module. This design effectively reduces uncertainty in tasks with changes in dynamics or/and reward functions, thereby enabling accurate task identification and adaptation. The experiment results on both Meta-World and classical MuJoCo benchmarks show that MetaCARD significantly outperforms prevailing Meta-RL baselines, demonstrating its remarkable adaptation ability in sophisticated environments that involve changes in both reward functions and dynamics.

Introduction

Real-world scenarios often feature multiple tasks sharing similar foundational structures. For instance, both closing curtains and opening drawers share a pulling action. Meta-Reinforcement Learning (Meta-RL) (Duan et al. 2016; Beck et al. 2023) is formulated to discern this shared structure, which can be viewed as dynamics and reward functions among various training tasks. The ultimate goal is to meta-learn a flexible policy that can quickly adapt to new tasks.

Context-based Meta-RL approaches are regarded as promising for achieving such a goal. These approaches leverage a context encoder to infer task-specific information from past experiences and train a policy conditioned on this derived latent context to ensure smooth adaptation to new tasks. A critical challenge in context-based Meta-RL is referred to as Markov Decision Process (MDP) ambiguity (Li et al. 2019), in which the context encoder mistakenly establishes a spurious correlation between task representations and history data. Consequently, this leads to a latent context-conditioned policy that struggles to distinguish between distinct tasks.

To alleviate this problem, recent advanced methods primarily center around improving the representation of latent context. For example, Probabilistic Embeddings for Actor-critic meta-RL (PEARL) (Rakelly et al. 2019) adopts an amortized variational inference method to learn a probabilistic latent representation of prior experiences. Leveraging from the recent advance of representation learning, (Fu et al. 2020; Wang et al. 2021; Yuan and Lu 2022) employs contrastive learning to train a compact context encoder so as to capture the distribution of tasks. However, a major limitation of these methods is their entangled latent contexts, making it difficult for the agent to distinguish between changes in dynamics or/and reward functions, thus hampering its adaptability in complex environments.

In contrast, (Lee et al. 2020) proposes to bifurcate context encoding and dynamics prediction to ensure that the context encoder effectively captures local dynamics and predicts both forward and backward dynamics. Variational Bayes-Adaptive Deep RL (variBAD) (Zintgraf et al. 2019) trains the posterior distribution of task embeddings through a decoder to predict the subsequent state and reward separately. (Han and Wu 2022) first proposes to leverage Successor Features (SFs) to enhance the identification ability of the context encoder in meta-RL. These methods improve the agent's perception of changing dynamics or rewards when inferring tasks with a context encoder. However, in practice, they only rely on generating additional trajectory data by the policy to self-correct inaccurate task inference. Nevertheless, the collected trajectory data may be biased, and as a result, the context encoder's learning process may only correlate with the low-quality trajectory data to minimize the loss function, leading to mistakenly identifying the specific task. Besides, the aforementioned methods also neglect the fact that decoupling dynamics and rewards in the policy space could directly convey the information about accurately capturing various MDP structures back to the context encoder. This oversight often culminates in misaligned adaptation amid current changes in dynamics and rewards. As a result, their

*Corresponding author.

context encoder continues to be incapable of better fitting the task distribution.

To address the MDP ambiguity problem, we propose the Meta-RL with task unCertAinty feedback through decoupled context-aware Reward and Dynamics components (MetaCARD), which aims to reduce uncertainty from tasks delineated by changes in dynamics and rewards while imparting this uncertainty feedback from policy evaluation to the task inference module. MetaCARD first derives the context-aware dynamics and reward components by linearly decomposing a reward function. The dynamics component is parameterized by both the state and the latent context, while the reward component is solely parameterized by the latent context. The Q function also considers the disentangled components constrained by the latent context. This not only serves as a more accurate proxy for policy evaluation but also contributes to effective policy improvement. Meanwhile, the updates to these two components provide task uncertainty feedback and apply constraints to the context encoder, thus establishing a direct and strong connection between task inference and policy evaluation. In this way, the efficacy of the policy can directly manifest within the learning procedure of the context encoder, compelling the context encoder to more accurately fit the distribution of tasks. Moreover, achieving accurate task inference becomes instrumental in acquiring improved policies. In this way, our model can proficiently capture the essential characteristics of different types of environmental shifts. Consequently, MetaCARD demonstrates an enhanced capability for flexible and efficient adaptability to diverse tasks.

The key contributions of our work can be summarized as:

1. We introduce decomposed context-aware representations for dynamics and reward weights. This design enables the policy, during its learning phase, to capture the unique features of both the decoupled reward functions and dynamics. Moreover, the trajectories generated by the policy are more representative of the inherent trends of environmental changes.
2. We incorporate the feedback on task structures from policy evaluation to assist the context encoder in promptly reducing the uncertainty about task inference and better fitting the true task distribution, thereby improving the inference accuracy and adaptation efficiency.
3. We perform comprehensive experiments on challenging Meta-World and conventional MuJoCo benchmarks with changing dynamics or/and reward functions. The results demonstrate the superior adaptability of our model.

Related Work

Meta-RL aims to learn a policy capable of adapting efficiently to any new task from a given task distribution with similar structures, thereby minimizing the requirement for extensive data. Gradient-based Meta-RL methods (Finn, Abbeel, and Levine 2017; Liu et al. 2022; Tang 2022) aim to optimize initial parameters with policy gradient algorithms, offering a favorable starting point for new tasks. While these methods generate an improved policy that is applicable across task distributions, their policy gradient updates tend to be sample inefficient.

Context-based Meta-RL methods (Rakelly et al. 2019) provide some relief to this inefficiency. They infer the latent context of the current task from freshly acquired trajectories and train a latent context-conditioned policy to achieve rapid adaptation. To improve sample efficiency, PEARL (Rakelly et al. 2019) combines online inference of latent context with off-policy RL algorithms. To enable the context encoder to recognize unpredictable and swiftly changing environments, (Luo et al. 2022) proposes to minimize the variance and maximize the relational matrix determinant of the latent context embeddings. (Lee et al. 2020; Seo et al. 2020; Mu et al. 2022) concentrate on meta-learning dynamics and the separation of context encoding from the inference of the next state, ensuring flexible adaptation to changes in dynamics. However, the latent context derived from the above methods remains entangled, lacking the ability to differentiate between changes in dynamics and reward functions.

The integration of Successor Features (SFs) (Barreto et al. 2017) with the generalized policy improvement (GPI) method has demonstrated remarkable transfer capabilities across various RL tasks (Barreto et al. 2020; Alegre, Bazzan, and Da Silva 2022). SFs decouple the dynamics and the reward functions of an environment, thereby enabling flexible transfer learning across tasks with different reward structures but unchanged dynamics. Moreover, USFs (Ma et al. 2020) explores a specific scenario where tasks share the same dynamics while having different goals for goal-conditioned RL. (Mozifian et al. 2022) further extends USFs to multi-task settings with continuous goals.

In our work, we leverage decomposition forms similar to SFs not only in the representation space (Han and Wu 2022) but also extend them to the policy space, ensuring more accurate task inference and efficient policy improvement. We draw inspiration from (Zhang, Satija, and Pineau 2018) to predict actions and next-state embeddings to assist context-aware features in learning changed dynamics, making the process more flexible when integrated with Meta-RL.

Problem Statement

Meta-RL is defined on a distribution of tasks $p(\mathcal{T})$, where each task is a Markov Decision Process (MDP) (Sutton and Barto 1998) represented by a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma, \rho_0)$, in which $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $p(s'|s,a)$ the transition dynamics, $r(s,a,s')$ the reward function, ρ_0 the initial state distribution, and $\gamma \in [0,1)$ the discount factor. In Meta-RL, a task $\mathcal{T}$ is defined as $\{p(s_0), p(s'|s,a), r(s,a,s')\}$, where different tasks can be obtained by altering dynamics or reward functions. During the meta-learning process, a policy $\pi(a|s,\mathbf{z})$ is conditioned on both the state s and the latent context embedding $\mathbf{z}$. And $\mathbf{z}$ is learned based on a set of training tasks sampled from $p(\mathcal{T})$. This policy adapts to the present task by considering the past transition history, which is referred to as context c. We define $c_k^{\mathcal{T}}$ as (s_k, a_k, s'_k, r_k), representing one transition in task $\mathcal{T}$, and $c_{1:K}^{\mathcal{T}}$ comprises the experience collected so far. At test time, the policy should adapt to a new task sam-

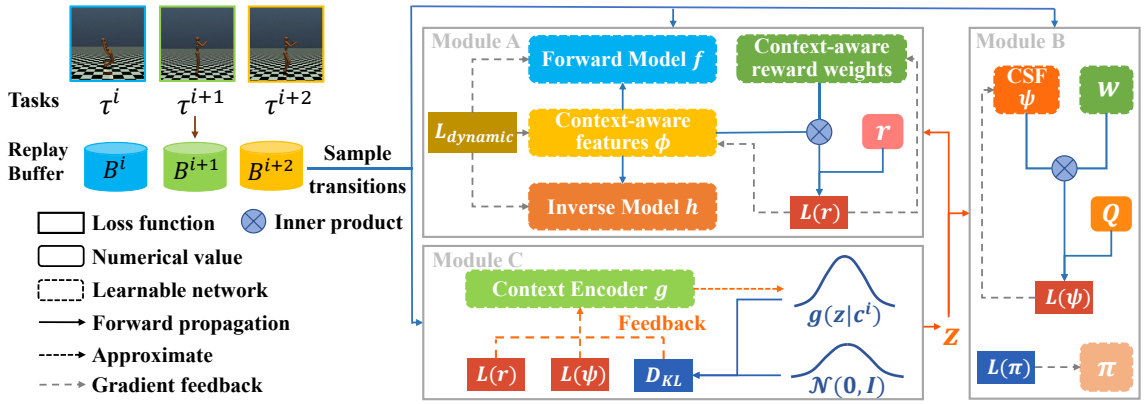


Figure 1: The architecture of MetaCARD. The MetaCARD framework consists of three main modules: (1) Module A: Context-aware feature acquisition, which learns context-aware reward weights w and features ϕ; (2) Module B: Policy learning, which the agent learns/refines its policy; and (3) Module C: Task representation learning, which derives task representations with uncertainty feedback.

pled from $p(\mathcal{T})$. The overall objective is formulated as:

$$max_\pi \left\{ \mathcal{R}_\star = \mathbb{E}_{\mathcal{T}_\star \sim p(\mathcal{T})} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \right\}, \quad (1)$$
$$a_t \sim \pi(s_t, \mathbf{z}), \quad \star = \{\text{“}train\text{”}, \text{“}test\text{”}\}$$

Methodology

As illustrated in Fig.1, MetaCARD starts by decomposing the reward function into two components: context-aware features and context-aware reward weights. Such decomposition is then applied in both Q-value estimation and policy learning. Moreover, updates to these two components also facilitate the context encoder to extract richer task-pertinent information. This allows the model to individually predict the changes in reward functions and dynamics, resulting in enhanced adaptability when dealing with more challenging Meta-RL tasks.

Decoupled Context-Aware Reward and Dynamics Components

Let us first introduce a context encoder g parameterized by η to infer task-relevant information. This encoder generates a latent context embedding $\mathbf{z}$ by processing a recent history of transitions sampled from task $\mathcal{T}$. Specifically, we define the latent context embedding as:

$$\mathbf{z} = g_\eta(c^{\mathcal{T}}_{1:K}), \quad (2)$$

where $c^{\mathcal{T}}_k = (s_k, a_k, s'_k, r_k)$ represents the k-th transition in the history, consisting of the current state s_k, action a_k, next state s'_k, and reward r_k. The history involves K transitions.

We propose that the reward function can be linearly decomposed as follows:

$$r(s, a, s', \mathbf{z}) = \phi(s, a, s', \mathbf{z})^\mathsf{T} w(\mathbf{z}), \quad (3)$$

where $\phi(s, a, s', \mathbf{z}) \in \mathbb{R}^d$ denotes the context-aware features of (s, a, s') and represents the dynamics of the environment, w is a fully-connected network and its output $w(\mathbf{z}) \in \mathbb{R}^d$ represents the context-aware reward weights. By concatenating the latent context embedding $\mathbf{z}$ and (s, a, s') as inputs to the ϕ network, the context-aware features are enriched by incorporating task-relevant information, better reflecting the inherent variations within the tasks. Based on Eq.(3), the Q-value can be rewritten as

$$\begin{aligned} Q_\pi(s, a, \mathbf{z}) &= \mathbb{E}_\pi \left[r_{t+1} + \gamma r_{t+2} + \ldots | s_t = s, a_t = a \right] \\ &= \mathbb{E}_\pi [\phi^\mathsf{T}_{t+1} w(\mathbf{z}) + \gamma \phi^\mathsf{T}_{t+2} w(\mathbf{z}) + \\ &\ldots | s_t = s, a_t = a] \\ &= \mathbb{E}_\pi \left[\sum_{m=t}^{\infty} \gamma^{m-t} \phi_{m+1} \Big| s, a \right]^\mathsf{T} w(\mathbf{z}) \\ &= \psi^\pi(s, a, \mathbf{z})^\mathsf{T} w(\mathbf{z}). \end{aligned} \quad (4)$$

The context-aware successor feature $\psi^\pi(s, a, \mathbf{z})$ (CSF) under policy π equates to the expected discounted sum of ϕ. As shown in Eq.(3) and (4), the dynamics are disentangled from the associated reward in the feature space, and both become correlated with the inferred task information. Consequently, the agent is able to perceive these two distinct types of environmental changes. For the simplicity of the notation, we will use $\phi(s', \mathbf{z})$ hereafter, which is technically a simplified version of $\phi(s, a, s', \mathbf{z})$.

The w network is trained with off-policy data sampled from the replay buffer $\mathcal{B}$ to minimize the prediction error of rewards:

$$\mathcal{L}_r = \mathbb{E}_{\substack{(s,a,s',r)\sim\mathcal{B} \\ \mathbf{z}\sim g_\eta}} \left[\frac{1}{2} (r - \phi_\mu(s', \mathbf{z})^\mathsf{T} w_\varphi(\mathbf{z}))^2 \right]. \quad (5)$$

Besides, we utilize a forward model f_ξ and an inverse model h_ρ to assist in learning $\phi_\mu(s', \mathbf{z})$. The forward model takes $\phi_\mu(s, \mathbf{z})$ of the current state and the action a as inputs, and predicts the context-aware features of the next state:

$$\hat{\phi}(s', \mathbf{z}) = f_\xi(\phi_\mu(s, \mathbf{z}), a). \quad (6)$$

The inverse model takes $\phi_\mu(s, \mathbf{z})$ and $\phi_\mu(s', \mathbf{z})$ of two consecutive states as inputs, and predicts the action $\hat{a}$, which is

Algorithm 1: MetaCARD algorithm

 # Meta-training Process

Input: training tasks $\mathcal{T}_i^{train}$ from $p(\mathcal{T})$, replay buffers $\mathcal{B}_i$, context encoder $g_\eta(\mathbf{z}|c^i)$, contextual policy $\pi_\theta(a|s,\mathbf{z})$, context-aware features $\phi_\mu(s,a,s',\mathbf{z})$, CSF $\psi_{\zeta_1}(s,a,\mathbf{z})$ and $\psi_{\zeta_2}(s,a,\mathbf{z})$, context-aware reward weights $w_\varphi(\mathbf{z})$, forward model f_ξ, and inverse model h_ρ

while not done **do**
  for each training task $\mathcal{T}_i^{train}$ **do**
    Sample latent context $\mathbf{z} \sim g_\eta(\mathbf{z}|c^i)$
    Roll-out policy $\pi_\theta(a|s,\mathbf{z})$ and add transitions $(s_k,a_k,s'_k,r_k)_{k:1\cdots K}$ to $\mathcal{B}_i$
  end for
  for each training step **do**
    for each training task $\mathcal{T}_i^{train}$ **do**
      Sample $c^i = (s_i,a_i,s'_i,r_i)_{i:1\cdots I} \sim \mathcal{B}_i$ for context encoder and RL batch $b_i \sim \mathcal{B}_i$
      Sample latent context $\mathbf{z} \sim g_\eta(\mathbf{z}|c^i)$
      Train w: $\mathbb{E}_{b_i^w,\mathbf{z}}[\mathcal{L}_r]$, $b_i^w \sim \mathcal{B}_i$
      Train ϕ, f, and h: $\mathbb{E}_{b_i^\phi,\mathbf{z}}[\mathcal{L}_{\text{dynamics}}]$, $b_i^\phi \sim \mathcal{B}_i$
      Train CSF ψ_{ζ_1} and ψ_{ζ_2}: $\mathbb{E}_{b_i,\mathbf{z}}[\mathcal{L}_{\text{critic}}]$
      Train contextual policy π: $\mathbb{E}_{b_i,\mathbf{z}}[\mathcal{L}_{\text{actor}}]$
      Train context encoder g: $\mathbb{E}_{b_i,\mathbf{z}}[\mathcal{J}_{\text{encoder}}]$
    end for
  end for
end while

 # Meta-testing Process

Input: test tasks $\mathcal{T}^{test}$ from $p(\mathcal{T})$

Initialize transitions $c^{\mathcal{T}} = \{\}$
for $m = 1,\ldots,M$ **do**
  Sample latent context $\mathbf{z} \sim g_\eta(\mathbf{z}|c^m)$
  Roll out policy $\pi_\theta(a|s,\mathbf{z})$ to generate transitions $D_m^{\mathcal{T}} = \{(s_n,a_n,s'_n,r)\}$
  Store the transitions: $c^{\mathcal{T}} = c^{\mathcal{T}} \cup D_m^{\mathcal{T}}$
end for

executed by the agent to transit from s to s':

$$\hat{a} = h_\rho(\phi_\mu(s,\mathbf{z}),\phi_\mu(s',\mathbf{z})). \tag{7}$$

Then the auxiliary loss is defined as follows:

$$\mathcal{L}_{\text{auxiliary}} = \mathbb{E}_{\substack{(s,a,s')\sim\mathcal{B}\\ \mathbf{z}\sim g_\eta}}\big[|\hat{\phi}(s',\mathbf{z}) - \phi_\mu(s',\mathbf{z})| + |\hat{a}-a|\big]. \tag{8}$$

We sample transitions from the replay buffer $\mathcal{B}$ and jointly update the parameters μ,ξ, and ρ as follows:

$$\mathcal{L}_{\text{dynamics}} = \mathcal{L}_r + \lambda\mathcal{L}_{\text{auxiliary}}, \tag{9}$$

where λ is a hyperparameter ranging from 0 to 1.

Specifically, in environments where dynamics remain constant but rewards changes, we can substitute $\phi(s',\mathbf{z})$ with $\phi(s,a,s')$ and only train the ϕ and w networks using Eq.(5). During meta-training, we can promptly evaluate policy π on different tasks by approximating context-aware reward weights $w(\mathbf{z})$. This approach not only simplifies the training of the Q-value but also facilitates the learning of shared dynamics structures, enabling rapid adaptation.

Policy Learning

To establish more intimate relationships between the trajectories generated by the policy and the changes in the environment, we further incorporate context-aware reward weights and successor features into the meta-training of the policy. Our algorithm is built on the Soft Actor-Critic (SAC) algorithm (Haarnoja et al. 2018), an off-policy actor-critic method that utilizes the maximum entropy RL objective.

Considering $\phi(s',\mathbf{z})$ as the role of immediate reward, we can estimate the CSF[1] as:

$$\hat{\psi}^\pi(s,a,\mathbf{z}) = \mathbb{E}_\pi\left[\phi_\mu(s',\mathbf{z}) + \gamma\psi^\pi_{\zeta_j}(s',\pi(s'),\mathbf{z})\right], j = 1,2. \tag{10}$$

During the training of the ψ network, the latent context embedding $\mathbf{z}$ serves as a special state input, thereby facilitating the acquisition of a more generalized representation of the dynamics. Consequently, the objective of training the ψ network is to minimize the squared residual error:

$$\begin{aligned}\mathcal{L}_{\text{critic}} = \mathbb{E}_{(s,a)\sim\mathcal{B},\mathbf{z}\sim g_\eta}\Big[\frac{1}{2}(\psi_{\zeta_j}(s,a,\mathbf{z})^\mathsf{T} w_\varphi(\mathbf{z}) - \\ \hat{\psi}(s,a,\overline{\mathbf{z}})^\mathsf{T} w_\varphi(\overline{\mathbf{z}}))^2\Big],\end{aligned} \tag{11}$$

where $\overline{\mathbf{z}}$ denotes stopping the back propagation of gradients.

To mitigate the issue of Q-value overestimation and stabilize the training process, double ψ networks are leveraged for the estimation of CSF. The Q-value is then estimated by computing the minimum product of ψ and w:

$$Q(s,a,\overline{\mathbf{z}}) = \min_{j=1,2}\psi_{\zeta_j}(s,a,\overline{\mathbf{z}})^\mathsf{T} w_\varphi(\overline{\mathbf{z}}). \tag{12}$$

The policy network can be updated by minimizing the expected KL-divergence:

$$\mathcal{L}_{\text{actor}} = \mathbb{E}_{\substack{s\sim\mathcal{B}\\ a\sim\pi_\theta\\ \mathbf{z}\sim g_\eta}}\left[D_{\text{KL}}\left(\pi_\theta(a \mid s,\overline{\mathbf{z}})\middle\| \frac{\exp\left(Q(s,a,\overline{\mathbf{z}})\right)}{\mathcal{Z}_v(s)}\right)\right], \tag{13}$$

where the policy π_θ is also conditioned on the latent context embedding $\overline{\mathbf{z}}$ and $\mathcal{Z}_v(s)$ is a partition function used to normalize the distribution. The policy evaluation is achieved through the context-aware dynamics and reward components. Thus, the reward and dynamics structure across different tasks can be separately learned during policy training.

Context Encoder Optimization with Uncertainty Feedback

It is crucial for the latent context to adeptly reflect changes in the reward function and dynamics of the environment. Therefore, the context encoder is designed to precisely capture alterations in both reward functions and dynamics from past transitions. In addition to leveraging the KL-divergence, as inspired by (Rakelly et al. 2019), to enrich the latent context embedding with more task-specific information, we also utilize the training losses of reward and CSF, denoted

[1] In our experiments, the update procedure includes target CSF networks, whose parameters are updated with an exponential moving average derived from the weights of the ψ network.

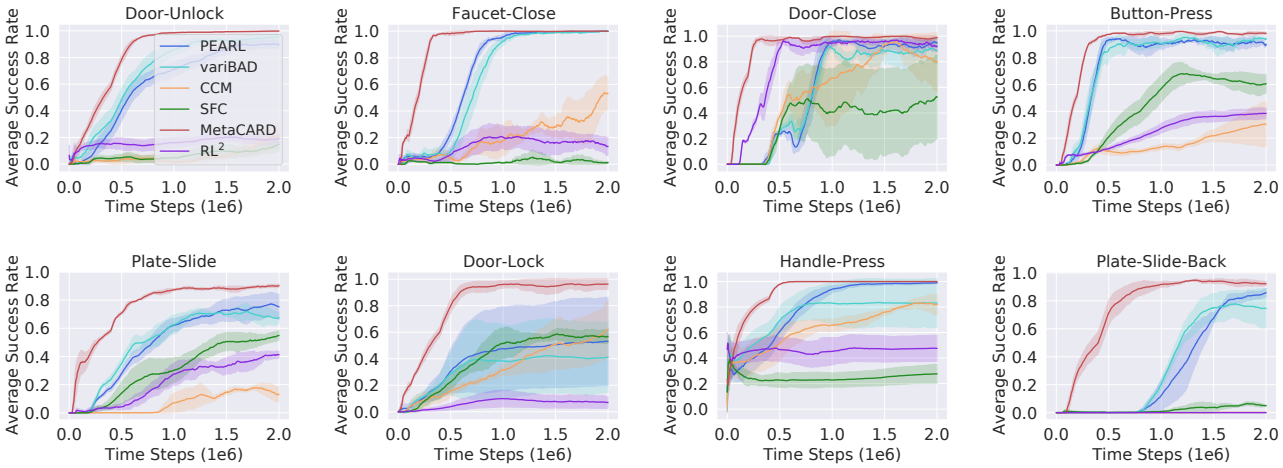


Figure 2: Meta-testing average performance on 8 ML1 environments over 3 random seeds. For each seed, the average success rate is computed every 600 testing steps, averaging over 10 episodes.

as $\mathcal{L}_{\text{CARD}}$, regarded as uncertainty feedback to jointly constrain the context encoder, thus extracting more compact and accurate task representations.

$$\mathcal{L}_{\text{CARD}} = \mathcal{L}_r + \mathbb{E}_{\substack{(s,a)\sim\mathcal{B} \\ \mathbf{z}\sim g_\eta}}\Big[\Big|\psi_{\zeta_j}(s,a,\mathbf{z}) - \hat{\psi}(s,a,\bar{\mathbf{z}})\Big|\Big]. \tag{14}$$

Then, the optimization objective of the context encoder $\mathcal{J}_{\text{encoder}}$ is formulated as:

$$\mathbb{E}_{\substack{c\sim\mathcal{B} \\ \mathbf{z}\sim g_\eta}}[\log p(Q) - \alpha D_{\text{KL}}(g_\eta(\mathbf{z}|c^{\mathcal{T}}_{1:K}))\|p(\mathbf{z})] - \beta\mathcal{L}_{\text{CARD}}, \tag{15}$$

where α and β are hyperparameters, and β is set to 0.01 in all experiments. As $\mathcal{L}_{\text{CARD}}$ enables the context encoder to promptly perceive and adapt to task variations, it fosters the training of more robust policies capable of effectively accommodating diverse tasks. The efficacy of $\mathcal{L}_{\text{CARD}}$ is further confirmed through the ablation study.

Meta-Testing

During the Meta-testing phase, we randomly sample various testing tasks from the task distribution $p(\mathcal{T})$. Task-specific transitions c^m are generated by performing a random policy. Subsequently, the context encoder utilizes these transitions for task inference, resulting in the derivation of latent context embeddings $\mathbf{z}$. Conditioned on $\mathbf{z}$, the well-trained policy $\pi_\theta(a|s,\mathbf{z})$ is evaluated on test tasks, thereby generating a sequence of new transitions $D^{\mathcal{T}}_m$. Through such iterations, we test the performance and adaptation ability of the policy. Alg. 1 presents the detailed procedures of MetaCARD.

Experiments

We experimented on a wide range of tasks to evaluate the adaptability of MetaCARD, with a focus on the following questions: 1) How does MetaCARD perform against the latest baselines when dealing with more challenging tasks? (Table 1 and Fig.2) 2) How proficiently can MetaCARD adapt to tasks with simultaneous changes in dynamics and rewards? (Fig.5) 3) Is MetaCARD capable of learning a compact and distinct task representation? (Fig.6 and Fig.7)

Experimental Setups

Task settings Our experiments include two benchmarks: 1) Meta-World[2] (Yu et al. 2019) consisting of 50 different robotic manipulation tasks; 2) Classical MuJoCo (Todorov, Erez, and Tassa 2012) with only parametric diversity, a benchmark widely adopted in Meta-RL. The experiments are mainly in three distinct scenarios: a) Tasks with sole changes in reward functions (e.g., walking direction for Half-Cheetah-Dir and goal location for Ant-Goal); b) Tasks with sole changes in dynamics (e.g., body mass for Half-Cheetah-Mass and random parameters for Walker-Rand-Params); c) Tasks with changes in both reward functions and dynamics (e.g., walking direction and body mass for Half-Cheetah-Dir-Mass, and walking direction and damping for Half-Cheetah-Dir-Damp).

Baselines We compared our method with five prevailing and competitive baselines: (1) **PEARL** (Rakelly et al. 2019) integrates task inference with off-policy RL algorithms; (2) **SFC** (Han and Wu 2022) utilizes SFs to train the context encoder; (3) **RL^2** (Duan et al. 2016) employs a recurrent neural network (RNN) to encode states; (4) **VariBAD** (Zintgraf et al. 2019) employs a decoder to train the posterior over the latent context; (5) **CCM** (Fu et al. 2020) leverages contrastive learning to learn the latent context.

Evaluation metrics We adhered to the Meta-Learning 1 (ML1) evaluation protocol in the Meta-World benchmark, which uses few-shot adaptation to goal variations within a single task. All models were trained with equal time steps,

[2]Generally, Meta-World is considered to be a more challenging benchmark due to its broader task distribution. (Beck et al. 2023)

	Door-Unlock		Faucet-Close		Door-Close		Button-Press	
	Train	Test	Train	Test	Train	Test	Train	Test
RL^2	19.80±5.41	20.60±5.61	18.00±14.34	13.27±10.24	90.27±7.33	91.80±6.35	39.93±4.72	39.27±5.74
variBAD	94.10±2.53	95.30±2.78	99.47±0.46	99.80±**0.00**	93.73±7.37	88.73±15.95	95.00±1.74	94.13±2.02
CCM	24.67±18.20	21.73±15.02	52.80±17.03	53.20±17.17	80.47±33.83	79.53±35.45	41.07±32.13	35.27±32.86
PEARL	90.20±6.39	90.40±4.21	99.67±0.42	99.87±0.23	92.70±5.13	94.80±4.85	89.00±1.06	89.93±1.62
SFC	16.93±7.17	17.00±11.64	1.40±1.51	1.27±1.30	51.13±50.04	52.67±50.21	61.27±9.64	61.27±9.41
MetaCARD	**99.73±0.31**	**99.93±0.12**	**100.00±0.00**	**99.93**±0.12	**98.20±3.12**	**98.87±1.96**	**98.53±0.99**	**98.07±1.55**
	Plate-Slide		Door-Lock		Handle-Press		Plate-Slide-Back	
	Train	Test	Train	Test	Train	Test	Train	Test
RL^2	37.27±4.00	44.60±**2.82**	12.33±14.70	6.33±6.30	46.80±1.64	48.00±15.87	0.00±0.00	0.00±0.00
variBAD	72.07±3.70	69.13±7.70	47.13±38.12	42.53±38.69	79.87±34.87	83.07±29.33	73.13±16.07	73.40±22.19
CCM	9.53±8.73	10.27±9.70	68.93±23.55	66.60±30.38	79.13±13.12	79.53±18.13	0.00±0.00	0.00±0.00
PEARL	76.40±10.53	74.30±13.83	55.95±41.87	53.10±48.56	99.07±0.64	99.33±0.31	88.87±3.21	88.20±7.99
SFC	64.13±7.60	55.73±5.26	54.13±4.73	55.33±9.26	37.87±20.61	34.27±20.34	2.27±**2.20**	3.27±1.70
MetaCARD	**89.80±2.82**	**89.33**±3.63	**98.93±1.68**	**97.67±3.70**	**99.87±0.23**	**99.93±0.12**	**92.00**±2.43	**92.33±0.61**

Table 1: Average success rate ± standard error (%) on Meta-World.

and subsequent policy evaluations were performed with the same number of time steps. Following (Zhang et al. 2021), the performance of the policy learned with the ML1 protocol was evaluated by computing the average success rate across three random seeds. In the classical MuJoCo benchmark, we computed the average return across three random seeds.

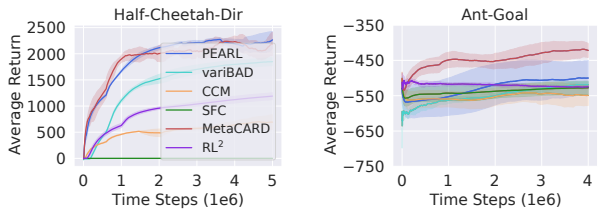


Figure 3: Meta-testing average return on Half-Cheetah-Dir and Ant-Goal with changed reward functions.

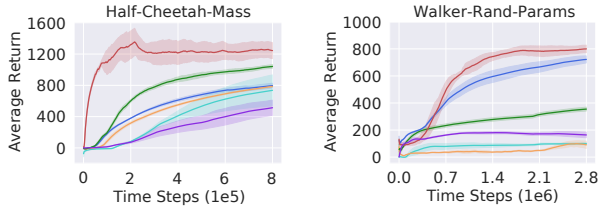


Figure 4: Meta-testing average return on Half-Cheetah-Mass and Walker-Rand-Params with changed dynamics.

Adaptation Efficiency and Performance

Performance on tasks with sole changes in reward functions or dynamics We experimented with 8 ML1 environments and 2 MuJoCo environments, all of which varied in reward functions, to evaluate the performance of all models. The testing curves and converged performances are depicted in Fig.2 and Fig.3, respectively. As illustrated in Fig.2, MetaCARD consistently achieves state-of-the-art (SOTA) adaptation efficiency across all 8 ML1 environments.

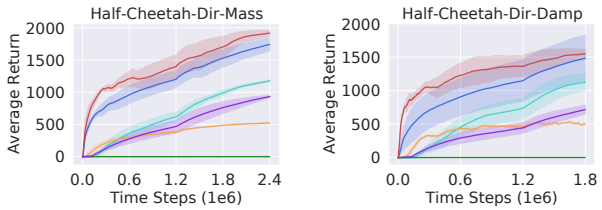
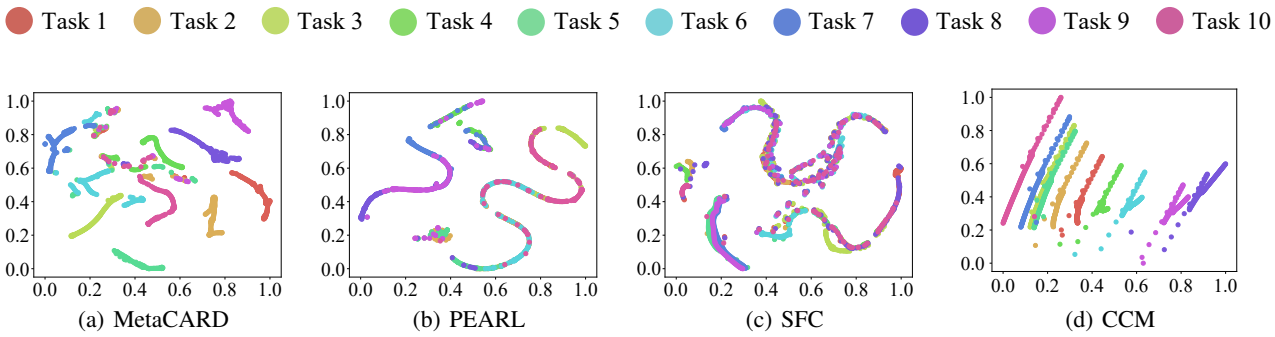


Figure 5: Meta-testing average return on Half-Cheetah-Dir-Mass and Half-Cheetah-Dir-Damp with both changed reward functions and dynamics.

As presented in Table 1, MetaCARD also achieves the highest average success rate with lowest variance in most ML1 environments. Although SFC outperforms PEARL in Door-Lock, it fails in the majority of the other environments. In comparison with MetaCARD, PEARL exhibits impressive asymptotic performance in Faucet-Close and Handle-Press. However, PEARL adapts more slowly to test tasks in other environments, which may be attributed to its coupled task representation that makes discerning shifts in reward functions across different tasks challenging. On the contrary, variBAD outperforms PEARL in both Door-Unlock and Button-Press by leveraging a disentangled latent context. Although CCM has learned excellent task representations (Fig.6), it performs significantly worse. Thus, a contrastive-only context cannot perceive the relations between tasks adequately. In contrast, MetaCARD's explicit consideration of dynamic and reward components enables it to learn more meaningful representations, which is critical for better policy learning in such complex task distribution.

In MuJoCo environments, MetaCARD shows comparable performance in Half-Cheetah-Dir and significantly outperforms other baselines in Ant-Goal, as illustrated in Fig.3.



(a) MetaCARD (b) PEARL (c) SFC (d) CCM

Figure 6: t-SNE visualization of the latent context extracted from trajectories collected in 10 diverse tasks.

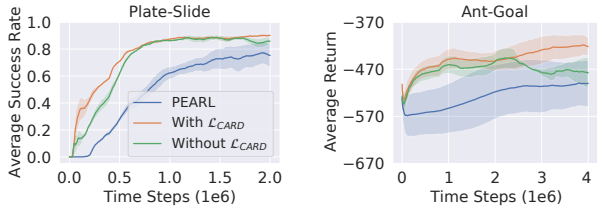


Figure 7: Ablation of $\mathcal{L}_{\text{CARD}}$ for context encoder training.

Notably, RL^2 outperforms CCM and SFC possibly due to PPO optimization. Furthermore, MetaCARD achieves performance that is superior to the baselines in Half-Cheetah-Mass and Walker-Rand-Params, as presented in Fig.4. These results confirm that MetaCARD effectively captures task-specific information and fits better with the task distribution. Although variBAD and SFC learn task representations that decouple dynamics and reward functions, their policies cannot form a connection between the intrinsic attributes of these two types of changes and the true task distribution.

Performance on tasks with changes in both dynamics and reward functions This part examines the models' adaptability to more complex Meta-RL settings with both dynamics and reward functions of tasks modified. In Fig.5, MetaCARD demonstrates a notably more efficient adaptation ability than other methods, indicating that the context-aware reward and dynamics components can effectively capture the causality of dynamics and reward functions. Conversely, variBAD and CCM appear to struggle with adapting to new tasks, possibly because their policies cannot accurately detect changes between tasks, despite having learned distinct or otherwise excellent task representations.

Ablation Study

We conducted an ablation experiment to assess the effectiveness of $\mathcal{L}_{\text{CARD}}$. As illustrated in Fig.7, the incorporation of the $\mathcal{L}_{\text{CARD}}$ (Eq.(14)) into the update of the context encoder yields substantial improvements in both adaptation efficiency and final performance. The possible reason may stem from the fact that $\mathcal{L}_{\text{CARD}}$ enables the context encoder to accurately mirror changes in both reward functions and dynamics, which, in turn, facilitates the rapid adaptation of the agent to novel tasks. Interestingly, across most environments, even in the absence of $\mathcal{L}_{\text{CARD}}$, MetaCARD continues to outperform PEARL by efficiently adapting to new tasks and achieving superior asymptotic performance. This proves that the policy learned via context-aware successor features and context-aware reward weights can discern the difference between reward functions and dynamics across tasks, setting it apart from the Actor-Critic structure utilized in PEARL.

Visualization for Context

To provide a visual insight into the learned latent context of MetaCARD during the meta-testing phase, We employed t-SNE (van der Maaten and Hinton 2008) for comparison with PEARL, SFC and CCM. We initiated the learned policies on ten randomly sampled test tasks derived from Faucet-Close in the Meta-World benchmark to generate trajectory data. These trajectories were encoded by the context encoder to output latent context embeddings, which were then visualized with t-SNE as depicted in Fig.6.

The results demonstrate that the context encoder trained by the MetaCARD is highly effective in distinguishing different trajectories from distinct tasks, indicating the extraction of compact and disentangled task representations. In contrast, the latent context embeddings for different tasks trained by PEARL and SFC tend to overlap, while embeddings from the same tasks cluster together. Consequently, in more complex environments, PEARL and SFC struggle to accurately discern the variations in task structure as reflected in trajectory changes during task inference.

Conclusion

In this paper, we propose MetaCARD to effectively resolve the MDP ambiguity problem by: 1) inferring tasks and learning the policy through context-aware reward and dynamics components to accurately capture the unique attributes of decoupled dynamics and rewards, thereby promoting a clearer understanding of task specifics, 2) imparting task uncertainty feedback from policy evaluation to context encoder to avoid overfitting the task distribution. The experiment results demonstrate that our model outperforms baseline methods by a large margin in accurately identifying tasks and effectively adapting to complex environments.

Acknowledgments

This work was partially supported by the NSFC under Grants No. 92270125 and No. 62276024, as well as the National Key R&D Program of China under Grant No. 2022YFC3302101.

References

Alegre, L. N.; Bazzan, A.; and Da Silva, B. C. 2022. Optimistic linear support and successor features as a basis for optimal policy transfer. In *International Conference on Machine Learning*, 394–413. PMLR.

Barreto, A.; Dabney, W.; Munos, R.; Hunt, J. J.; Schaul, T.; van Hasselt, H. P.; and Silver, D. 2017. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30.

Barreto, A.; Hou, S.; Borsa, D.; Silver, D.; and Precup, D. 2020. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, 117(48): 30079–30087.

Beck, J.; Vuorio, R.; Liu, E. Z.; Xiong, Z.; Zintgraf, L.; Finn, C.; and Whiteson, S. 2023. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*.

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

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Fu, H.; Tang, H.; Hao, J.; Chen, C.; Feng, X.; Li, D.; and Liu, W. 2020. Towards Effective Context for Meta-Reinforcement Learning: an Approach based on Contrastive Learning. In *AAAI Conference on Artificial Intelligence*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *ArXiv*, abs/1801.01290.

Han, X.; and Wu, F. 2022. Meta Reinforcement Learning with Successor Feature Based Context. *arXiv preprint arXiv:2207.14723*.

Lee, K.; Seo, Y.; Lee, S.; Lee, H.; and Shin, J. 2020. Context-aware Dynamics Model for Generalization in Model-Based Reinforcement Learning. *ArXiv*, abs/2005.06800.

Li, J.; Vuong, Q. H.; Liu, S.; Liu, M.; Ciosek, K.; Christensen, H. I.; and Su, H. 2019. Multi-task Batch Reinforcement Learning with Metric Learning. *arXiv: Learning*.

Liu, B.; Feng, X.; Ren, J.; Mai, L.; Zhu, R.; Zhang, H.; Wang, J.; and Yang, Y. 2022. A theoretical understanding of gradient bias in meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 31059–31072.

Luo, F.; Jiang, S.; Yu, Y.; Zhang, Z.; and Zhang, Y.-F. 2022. Adapt to Environment Sudden Changes by Learning a Context Sensitive Policy. In *AAAI Conference on Artificial Intelligence*.

Ma, C.; Ashley, D. R.; Wen, J.; and Bengio, Y. 2020. Universal successor features for transfer reinforcement learning. *arXiv preprint arXiv:2001.04025*.

Mozifian, M.; Fox, D.; Meger, D.; Ramos, F.; and Garg, A. 2022. Learning Successor Feature Representations to Train Robust Policies for Multi-task Learning. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

Mu, Y.; Zhuang, Y.; Ni, F.; Wang, B.; Chen, J.; Hao, J.; and Luo, P. 2022. DOMINO: Decomposed Mutual Information Optimization for Generalized Context in Meta-Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35: 27563–27575.

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

Seo, Y.; Lee, K.; Clavera, I.; Kurutach, T.; Shin, J.; and Abbeel, P. 2020. Trajectory-wise Multiple Choice Learning for Dynamics Generalization in Reinforcement Learning. *ArXiv*, abs/2010.13303.

Sutton, R. S.; and Barto, A. G. 1998. Reinforcement Learning: An Introduction. *IEEE Transactions on Neural Networks*, 9(5): 1054.

Tang, Y. 2022. Biased Gradient Estimate with Drastic Variance Reduction for Meta Reinforcement Learning. In *International Conference on Machine Learning*, 21050–21075. PMLR.

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

van der Maaten, L.; and Hinton, G. E. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605.

Wang, B.; Xu, S.; Keutzer, K.; Gao, Y.; and Wu, B. 2021. Improving Context-Based Meta-Reinforcement Learning with Self-Supervised Trajectory Contrastive Learning. *CoRR*, abs/2103.06386.

Yu, T.; Quillen, D.; He, Z.; Julian, R. C.; Hausman, K.; Finn, C.; and Levine, S. 2019. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. *ArXiv*, abs/1910.10897.

Yuan, H.; and Lu, Z. 2022. Robust task representations for offline meta-reinforcement learning via contrastive learning. In *International Conference on Machine Learning*, 25747–25759. PMLR.

Zhang, A.; Satija, H.; and Pineau, J. 2018. Decoupling Dynamics and Reward for Transfer Learning. *ArXiv*, abs/1804.10689.

Zhang, J.; Wang, J.; Hu, H.; Chen, T.; Chen, Y.; Fan, C.; and Zhang, C. 2021. Metacure: Meta reinforcement learning with empowerment-driven exploration. In *International Conference on Machine Learning*, 12600–12610. PMLR.

Zintgraf, L.; Shiarlis, K.; Igl, M.; Schulze, S.; Gal, Y.; Hofmann, K.; and Whiteson, S. 2019. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*.