Towards Effective Context for Meta-Reinforcement Learning: an Approach based on Contrastive Learning

Haotian Fu[1], **Hongyao Tang**[1], **Jianye Hao**[1,2*], **Chen Chen**[2], **Xidong Feng**[3], **Dong Li**[2], **Wulong Liu**[2]
[1]College of Intelligence and Computing, Tianjin University,
[2]Noah's Ark Lab, Huawei,
[3]Department of Automation, Tsinghua University,
{haotianfu, bluecontra}@tju.edu.cn,
{haojianye, chenchen9, lidong106, liuwulong}@huawei.com, fengxidongwh@gmail.com

Abstract

Context, the embedding of previous collected trajectories, is a powerful construct for Meta-Reinforcement Learning (Meta-RL) algorithms. By conditioning on an effective context, Meta-RL policies can easily generalize to new tasks within a few adaptation steps. We argue that improving the quality of context involves answering two questions: 1. How to train a compact and sufficient encoder that can embed the task-specific information contained in prior trajectories? 2. How to collect informative trajectories of which the corresponding context reflects the specification of tasks? To this end, we propose a novel Meta-RL framework called CCM (**C**ontrastive learning augmented **C**ontext-based **M**eta-RL). We first focus on the contrastive nature behind different tasks and leverage it to train a compact and sufficient context encoder. Further, we train a separate exploration policy and theoretically derive a new information-gain-based objective which aims to collect informative trajectories in a few steps. Empirically, we evaluate our approaches on common benchmarks as well as several complex sparse-reward environments. The experimental results show that CCM outperforms state-of-the-art algorithms by addressing previously mentioned problems respectively.

Introduction

Reinforcement Learning (RL) combined with deep neural networks has achieved impressive results on various complex tasks (Mnih et al. 2015; Lillicrap et al. 2016; Schulman et al. 2015). Conventional RL agents need large amount of environmental interactions to train a single policy for one task. However, in real-world problems many tasks share similar internal structures and we expect agents to adapt to such tasks quickly based on prior experiences. Meta-Reinforcement Learning (Meta-RL) proposes to address such problems by learning how to learn (Wang et al. 2016). Given a number of tasks with similar structures, Meta-RL methods aim to capture such common knowledge from previous experience on training tasks and adapt to a new task with only a small amount of interactions.

Based on this idea, many Meta-RL methods try to learn a general model initialization and update the parameters during adaptation (Finn, Abbeel, and Levine 2017; Rothfuss et al. 2019). Such methods require on-policy meta-training and are empirically proved to be sample inefficient. To alleviate this problem, a number of methods (Rakelly et al. 2019; Fakoor et al. 2020) are proposed to meta-learn a policy that is able to adapt with off-policy data by leveraging context information. Typically, an agent adapts to a new environment by inferring latent context from a small number of interactions with the environment. The latent context is expected to be able to capture the distribution of tasks and efficiently infer new tasks. Context-based Meta-RL methods then train a policy conditioned on the latent context to improve generalization.

As the key component of context-based Meta-RL, the quality of latent context can affect algorithms' performance significantly. However, current algorithms are sub-optimal in two aspects. Firstly, the training strategy for context encoder is flawed. A desirable context is expected to only extract task-specific information from trajectories and throw away other information. However, the latent context learned by existing methods (i.e. recovering value function (Rakelly et al. 2019) or dynamics prediction (Lee et al. 2020; Zhou, Pinto, and Gupta 2019)) are quite noisy as it may model irrelevant dependencies and ignore some task-specific information. Instead, we propose to directly analyze and discriminate the underlying structure behind different tasks' trajectories by leveraging contrastive learning. Secondly, prior context-based Meta-RL methods ignore the importance of collecting informative trajectories for generating distinctive context. If the exploration process does not collect transitions that are able to reflect the task's individual property and distinguish it from dissimilar tasks, the latent context would be ineffective. For instance, in many cases, tasks in one distribution only vary in the final goals, which means the transition dynamics remains the same in most places of the state space. Without a good exploration policy it is hard to obtain information that is able to distinguish tasks from each other, which leads to a bad context. A comparison of different learned latent embeddings are shown in Figure 6.

In this paper, we propose a novel off-policy Meta-RL algorithm CCM (**C**ontrastive learning augmented **C**ontext-based **M**eta-RL), aiming to improve the quality of context

*Corresponding author.
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

by tackling the two aforementioned problems. Our first contribution is an unsupervised training framework for context encoder by leveraging contrastive learning. The main insight is that by setting transitions from the same task as positive samples and the ones from different tasks as negative samples, contrastive learning is able to directly distinguish context in the original latent space without modeling irrelevant dependencies. The second contribution is an information-gain-based exploration strategy. With the purpose of collecting trajectories as informative as possible, we theoretically obtain a lower bound estimation of the exploration objective in contrastive learning framework. Then it is employed as an intrinsic reward, based on which a separate exploration agent is trained. The effectiveness of CCM is validated on a variety of continuous control tasks. The experimental results show that CCM outperforms state-of-the-art Meta-RL methods through generating high-quality latent context.

Preliminaries

Meta-Reinforcement Learning

In meta-reinforcement learning (Meta-RL) scenario, we assume a distribution of tasks $p(\mu)$. Each task $\mu \sim p(\mu)$ shares similar structures and corresponds to a different Markov Decision Process (MDP), $M_\mu = \{S, A, T_\mu, R_\mu\}$, with state space S, action space A, transition distribution T_μ, and reward function R_μ. We assume one or both of transition dynamics and reward function vary across tasks. Following prior problem settings in (Duan et al. 2016; Fakoor et al. 2020; Rakelly et al. 2019), we define a meta-test trial as N episodes in the same MDP, with an initial exploration of K episodes, followed by $N - K$ execution episodes leveraging the data collected in exploration phase. We further define a transition batch sampled from task μ as $b = \{\tau_i\}_{i=1}^k$. A trajectory consists of T transitions $\tau_{1:T}$ is a special case of transition batch when all transitions are consecutive. For context-based Meta-RL, the agent's policy typically depends on all prior transitions $\tau_{1:\Gamma} = \{(s_1, a_1, r_1, s_1') \cdots (s_\Gamma, a_\Gamma, r_\Gamma, s_\Gamma')\}$ collected by exploration policy π_{exp}. The agent firstly consumes the collected trajectories and outputs a latent context z through context encoder $q(z|\tau_{1:\Gamma})$, then executes policy π_{exe} conditioned on the current state and latent context z. The goal of the agent is therefore to maximize the expected return, $\mathbb{E}_{\mu\sim p(\mu)}\big[\mathbb{E}_{\tau_{1:\Gamma}\sim\pi_{exp}}\big[V_{\pi_{exe}}(q(z|\tau_{1:\Gamma}), \mu)\big]\big]$.

Contrastive Representation Learning

The key component of representation learning is the way to efficiently learn rich representations of given input data. Contrastive learning has recently been widely used to achieve such purpose. The core idea is to learn representation function that maps semantically similar data closer in the embedding space. Given a query q and keys $\{k_0, k_1, \cdots\}$, the goal of contrastive representation learning is to ensure q matches with positive key k_i more than any other keys in the data set. Empirically, positive keys and query are often obtained by taking two augmented versions of the same image, and negative keys are obtained from other images.

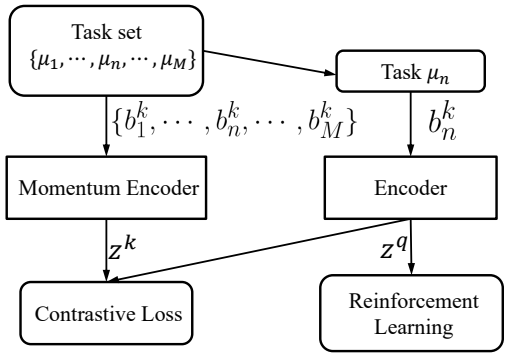

Figure 1: Contrastive Context Encoder

Previous work proposes InfoNCE loss(van den Oord, Li, and Vinyals 2018) to score positive keys $k_i \sim p(k|q)$ higher compared with a set of K distractors $k_j \sim p(k)$:

$$L_{NCE} = -\mathbb{E}\Big[\log \frac{\exp(f(q, k_i))}{\sum_{j=1}^K \exp(f(q, k_j))}\Big], \tag{1}$$

where function f calculates the similarity score between query and key data, and is usually modeled as bilinear products, i.e. $q^T W k$ (Hénaff et al. 2019). As proposed and proved in (van den Oord, Li, and Vinyals 2018), minimizing the InfoNCE loss is equivalent to maximizing a lower bound of the mutual information between q and k:

$$I(q; k) \geqslant \log(K) - L_{NCE} \tag{2}$$

The lower bound becomes tighter as K increases.

Algorithms

In this section, we describe our algorithm CCM by first introducing a novel context encoder training strategy, then we propose an information-gain-based exploration strategy to collect informative trajectories for effective task inference.

Contrastive Context Encoder

As the key component in context-based Meta-RL framework, how to train a powerful context encoder is non-trivial. One straightforward method is to train the encoder in an end-to-end fashion from RL loss (i.e. recover the state-action value function (Rakelly et al. 2019)). However, the update signals from value function is stochastic and weak that may not capture the similarity relations among tasks. Moreover, recovering value function is only able to implicitly capture high-level task-specific features and may ignore low-level detailed transition difference that contains relevant information as well. Another kind of methods resorts to dynamics prediction (Lee et al. 2020; Zhou, Pinto, and Gupta 2019). The main insight is to capture the task-specific features via distinguishing varying dynamics among tasks. However, entirely depending on low-level reconstructing states or actions is prone to over-fit on the commonly-shared dynamics transitions and model irrelevant dependencies which may hinder the learning process.

These two existing methods of training the context encoder are either not sufficient because of ignoring useful information or not compact because of modeling irrelevant dependencies. To this end, here we aim to train a compact and sufficient encoder through extracting mid-level features. We propose to directly extract the underlying task-specific information behind trajectories by performing contrastive learning on the nature distinctions of trajectories. Through explicitly comparing different tasks' trajectories as a whole instead of each transition within the trajectories, the encoder is prevented from modeling commonly-shared information while still be able to capture all the relevant task-specific structures by leveraging the contrastive nature behind different tasks.

Contrastive learning methods learn representations by pulling together semantically similar data points (positive data pairs) while pushing apart dissimilar ones (negative data pairs). We treat the trajectories sampled from same tasks as the positive data pairs and trajectories from different tasks as negative data pairs. Then the contrastive loss is minimized to gather the latent context of trajectories from same tasks closer in embedding space while pushing apart the context from other tasks.

Concretely, assuming a training task set containing M different tasks from task distribution $p(\mu)$. We first generate trajectories with current policy for each task and store them in replay buffer. At each training step, we first sample a task μ_n from the training task set, and then randomly sample two batches of transitions b_n^q, b_n^k from task μ_n independently. b_n^q serves as a query in contrastive learning framework while b_n^k is the corresponding positive key. We also randomly sample $M-1$ batches of transitions from the other tasks as negative keys. Note that previous work (e.g. CURL (Srinivas, Laskin, and Abbeel 2020)) relies on data augmentation on images like random crop to generate positive observations. In contrast, independent transitions sampled from the same task replay buffer naturally construct the positive samples.

As shown in Figure 1, after obtaining query and key data, we map them into latent context z^q and z^k respectively, on which we calculate similarity score and contrastive loss to train the encoder:

$$L_{contrastive} = -\mathbb{E}\Big[\log \frac{\exp(f(z^q, z_n^k))}{\sum_{j=1}^{M} \exp(f(z^q, z_j^k))}\Big] \tag{3}$$

Following the settings in (He et al. 2020; Srinivas, Laskin, and Abbeel 2020), we use momentum averaged version of the query encoder e^q for encoding the keys. The Reinforcement Learning part takes in the latent context as an additional observation, then executes policy and updates separately. Note that the Contrastive Context Encoder is a generic framework and can be integrated with any context-based Meta-RL algorithm.

Information-gain-based Exploration

Even with a compact and sufficient encoder, the context is still ineffective if the exploration process does not collect enough transitions that is able to reflect the new task's specific property and distinguish it from dissimilar tasks. Some previous approaches (Rakelly et al. 2019; Rothfuss et al. 2019) utilize Thompson-sampling (Thompson 1933) to explore, in which an agent needs to explore in the initial few episodes and execute optimally in the subsequent episode. However, these two processes is actually conducted by one single policy and is trained in an end-to-end fashion by maximizing expected return. This means the exploration policy is limited to the learned execution policy. When adapting to a new task, the agent tends to only explore experiences which are useful for solving previously trained tasks, making the adaptation process less effective.

Instead, we decouple the exploration and execution policy and define an exploration agent aiming to collect trajectories as informative as possible. We achieve this goal by encouraging the exploration policy to maximize the *information gain* from collecting new transition τ_i at time step i of task μ_n:

$$I(z|\tau_{1:i-1}; \tau_i) = H(z|\tau_{1:i-1}) - H(z|\tau_{1:i}), \tag{4}$$

where $\tau_{1:i-1}$ denotes the collected $i-1$ transitions before time step i. The above equation can also be interpreted as how much task belief z has changed given the newly collected transition. We further transform the equation as follows:

$$\begin{aligned} I(z|\tau_{1:i-1}; \tau_i) &= H(z|\tau_{1:i-1}) + H(z) - H(z) - H(z|\tau_{1:i}) \\ &= I(z; \tau_{1:i}) - I(z; \tau_{1:i-1}) \end{aligned} \tag{5}$$

Equation (5) implies that the information gain from collecting new transition τ_i can be written as the temporal difference of the mutual information between task belief and collected trajectories. This indicate that we expect the exploration agent to collect informative experience that form a solid hypothesis for task μ_n.

Although theoretically sound, the mutual information of latent context and collected trajectories is hard to directly estimate. We expect a tractable form of Equation (5) without losing information. To this end, we first theoretically define the sufficient encoder for context in Meta-RL framework. Given two batches of transitions b_1 and b_2 from one task, the encoders e_1 and e_2 extract representations $c_1 = e_1(b_1)$ and $c_2 = e_2(b_2)$, respectively.

Definition 4.1 (Context-Sufficient Encoder) The context encoder e of b_1 is sufficient if and only if $I(b_1, b_2) = I(e(b_1), b_2)$.

Intuitively, the mutual information remains the same as the encoder e does not change the amount of information contained in b_1, which means the encoder is context-sufficient.

Assuming the context encoder trained in previous section is context-sufficient, we can utilize this property to further transform the information gain objective in Equation (5) in a view at the latent context level. Given batches of transitions from the same task $b_{pos}, \tau_{1:i} \sim \mu_n$ and a context-sufficient encoder $e(\cdot)$, the latent context can be computed from the corresponding trajectories $c_{1:i} = e(\tau_{1:i})$, $c_{1:i-1} = e(\tau_{1:i-1})$. As for the latent context z, we can approximate it as the embedding of positive transition batch as well:

$z \approx e(b_{pos}) = c_{pos}$. Then (5) becomes:

$$\begin{aligned} I(z|\tau_{1:i-1};\tau_i) &= I(z;\tau_{1:i}) - I(z;\tau_{1:i-1}) \\ &= I[z;e(\tau_{1:i})] - I[z;e(\tau_{1:i-1})] \\ &= I(c_{pos};c_{1:i}) - I(c_{pos};c_{1:i-1}) \end{aligned} \tag{6}$$

Given this form of equation, we can further derive a tractable lower bound of $I(z|\tau_{1:i-1};\tau_i)$. The calculation process for the lower bound of Equation (6) can be decomposed into computing the lower bound of $I(c_{pos};c_{1:i})$ and the upper bound of $I(c_{pos};c_{1:i-1})$.

As mentioned in Preliminaries, a commonly used lower bound of $I(c_{pos},c_{1:i})$ can be written as:

$$I(c_{pos};c_{1:i}) \geqslant -L_{lower} + \log(W) \tag{7}$$

where,

$$L_{lower} = -\mathop{\mathbb{E}}_{C} \log\left[\frac{\exp(f(c_{1:i},c_{pos}))}{\sum_{j=1}^{M}\exp(f(c_{1:i},c_j))}\right] \tag{8}$$

W denotes the number of tasks. $C = C_{pos} \cup C_{neg}$, where C_{pos} contains latent context from the same tasks while C_{neg} contains latent context from different ones. As stated in previous section, we optimize the former term in (7) to train the context encoder and similarity function f.

We also need to find a tractable form for the upper bound of $I[c_{pos};c_{1:i-1}]$. Leveraging current contrastive learning framework, we make the following proposition:

Proposition 4.2

$$I(c_{pos};c_{1:i-1}) \leqslant -L_{upper} + \log(W) \tag{9}$$

where,

$$L_{upper} = -\mathop{\mathbb{E}}_{C_{pos}} log\left[\frac{\exp(f(c_{1:i-1},c_{pos}))}{\sum_{c_j\in C_{neg}}\exp(f(c_{1:i-1},c_j))}\right] \tag{10}$$

Proof. As defined in (van den Oord, Li, and Vinyals 2018), the optimal value for $exp(f(c_{1:i-1},c_{pos}))$ is given by $\frac{p(c_{pos}|c_{1:i-1})}{p(c_{pos})}$, inserting this back into Equation (10) results in:

$$\begin{aligned} L_{upper} &= -\mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{\frac{p(c_{pos}|c_{1:i-1})}{p(c_{pos})}}{\sum_{c_j\in C_{neg}}\frac{p(c_j|c_{1:i-1})}{p(c_j)}}\right] \\ &= \mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{p(c_{pos})}{p(c_{pos}|c_{1:i-1})}\sum_{c_j\in C_{neg}}\frac{p(c_j|c_{1:i-1})}{p(c_j)}\right] \\ &\approx \mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{p(c_{pos})}{p(c_{pos}|c_{1:i-1})}(W-1)\mathop{\mathbb{E}}_{c_{neg}}\frac{p(c_j|c_{1:i-1})}{p(c_j)}\right] \\ &= \mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{p(c_{pos})}{p(c_{pos}|c_{1:i-1})}(W-1)\right] \\ &\leqslant \mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{p(c_{pos})}{p(c_{pos}|c_{1:i-1})}W\right] \\ &= -I(c_{pos};c_{1:i-1}) + \log(W) \end{aligned} \tag{11}$$

Compared with InfoNCE loss, the only difference between these two bounds is whether the similarity relations $exp(f(c_{1:i};c_{pos}))$ between query and positive key is included in denominator. Then we can further derive the lower bound of Equation (6) as:

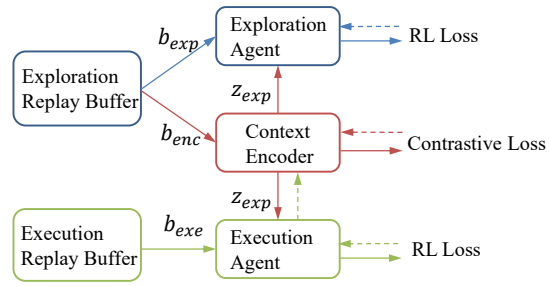

Figure 2: CCM training procedure. Dashed lines denote backward gradients.

Proposition 4.3

$$\begin{aligned} &I(z|\tau_{1:i-1};\tau_i) \geqslant L_{upper} - L_{lower} \\ &= \mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{\exp(f(c_{1:i},c_{pos}))}{\sum_{c_j\in C}\exp(f(c_{1:i},c_j))}\right] \\ &\quad - \mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{\exp(f(c_{1:i-1},c_{pos}))}{\sum_{c_j\in C_{neg}}\exp(f(c_{1:i-1},c_j))}\right] \end{aligned} \tag{12}$$

Proof. This can be derived by simply calculating the difference between Equation (7) and (9).

Thus, we obtain an estimate of the lower bound of mutual information $I(z|c_{1:i-1};c_i)$ and we further use it as an intrinsic reward r^{aux} for the independent exploration agent:

$$r^e = r^{env} + \alpha r^{aux}, where\ r^{aux} = L_{upper} - L_{lower} \tag{13}$$

, where r^{env} is the extrinsic reward from environment. The intrinsic reward can be interpreted as the difference between two contrastive loss. It measures how much the inference for current task has been improved after collecting new transition c_i. To make this objective more clear and comprehensible, we can transform the lower bound (12) as:

$$\begin{aligned} &L_{upper} - L_{lower} \\ &= \mathop{\mathbb{E}}_{C_{pos}}\left[f(c_{1:i},c_{pos}) - f(c_{1:i-1},c_{pos})\right] \\ &\quad - \mathop{\mathbb{E}}_{C_{pos}} \log\left[\frac{\sum_{c_j\in C}\exp(f(c_{1:i},c_j))}{\sum_{c_j\in C_{neg}}\exp(f(c_{1:i-1},c_j))}\right] \end{aligned} \tag{14}$$

Intuitively, the first term is an estimation of how much the similarity score for positive pairs (the correct task inference) has improved after collecting new transition. Maximizing this term means that we want the exploration policy to collect transitions which make it easy to distinguish between tasks and make correct and solid task hypothesis. The second term can be interpreted as a regularization term. Maximizing this term means that while the agent tries to visit places which may result in enhancement of positive score, we want to limit the policy not visiting places where the negative score (the similarity with other tasks in embedding space) enhances as well.

We summarize our training procedure in Figure 2. b_{exp}, b_{exe}, b_{enc} stand for the batch of transitions sampled for training the exploration agent, execution agent and context

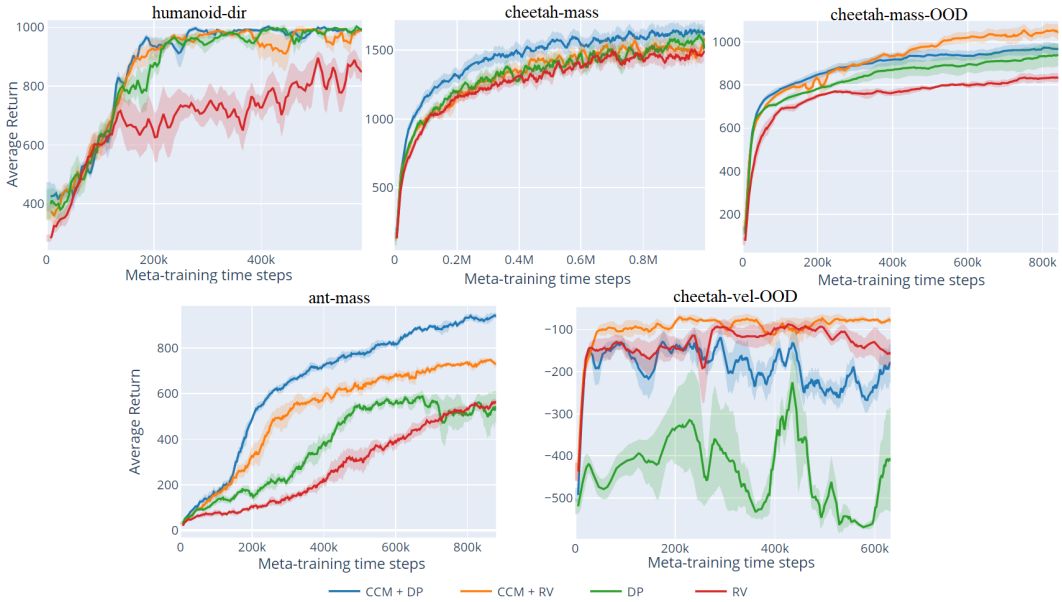

Figure 3: Comparison for different context encoder training strategies. Our methods CCM + DP and CCM + RV achieve consistently better performance compared to existing methods. The error bar shows 1 standard deviation.

encoder respectively. The exploration agent and execution agent interact with environment separately to get their own replay buffer. The execution replay buffer also contains transitions from exploration agent that do not add on the intrinsic reward term. At the beginning of each meta-training episode, the two agents sample transition data b_{exp} and b_{exe} to train their policy separately. We independently sample from exploration buffer to obtain transition batches for calculating latent context. Note that the reward terms in b_{enc} do not add on the intrinsic reward and only use the original environmental reward for computing latent context. In practice, we utilize Soft Actor-Critic (Haarnoja et al. 2018) for both exploration and execution agent. We first pretrain the context encoder with contrastive loss for a few episodes to avoid the intrinsic reward being too noisy at the beginning of training. During meta-testing, we first let the exploration agent collect trajectories for a few episodes then compute the latent context based on these experience. The execution agent then acts conditioned on the latent context. The pseudo-code for CCM during training and adaptation can be found in our appendix.

Experiments

In this section, we evaluate the performance of our proposed CCM to answer the following questions:

- Does CCM's contrastive context encoder improve the performance of state-of-the-art context-based Meta-RL methods?
- After combining with information-gain-based exploration policy, does CCM improve the overall adaptation performance compared with prior Meta-RL algorithms?
- How does regularization term (second term in (14)) influence the exploration policy?
- Is CCM able to extract effective and reasonable context information?

Comparison of Context Encoder Training Strategy

We first evaluate the performance of context-based Meta-RL methods after combining with contrastive context encoder on several continuous control tasks simulated via MuJoCo physics simulator (Todorov, Erez, and Tassa 2012), which are standard Meta-RL benchmarks used in prior works (Fakoor et al. 2020; Rakelly et al. 2019). We also evaluate its performance on out-of-distribution (OOD) tasks, which are more challenging cases for testing encoder's capacity of extracting useful information. We compare to the existing two kinds of training strategy for context encoder: 1) RV (Recovering Value-function) (Rakelly et al. 2019), in which the encoder is trained with gradients from recovering the state-action value function. 2) DP (Dynamics Prediction) (Lee et al. 2020; Zhou, Pinto, and Gupta 2019), in which the encoder is trained by performing forward or backward prediction. Here, we follow the settings in CaDM (Lee et al. 2020) which uses the combined prediction loss of forward prediction and backward prediction to update the context encoder[1]. Our method CCM+RV denotes the encoder receives gradients from both the contrastive loss and value function, while CCM+DP denotes the encoder receives gradients from both the contrastive loss and dynamics prediction loss. To maintain consistency, we does not consider extra exploration policy in this part of evaluation and all three methods use the same network structure (i.e. actor-critic, context encoder) and evaluation settings as state-of-the-art context-based Meta-RL method PEARL.

[1]For environments that have fixed dynamics, we further modify the model to predict the reward as well.

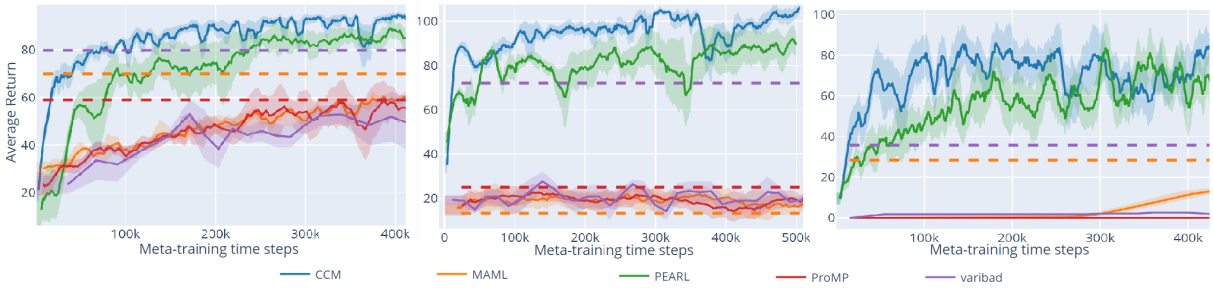

Figure 4: CCM's overall performance compared with state-of-the-art Meta-RL methods on complex sparse-reward environments. From left to right: *walker-sparse*, *cheetah-sparse*, *hard-point-robot*. Asymptotic performance is plotted in dash lines. The error bar shows 1 standard deviation.

The results are shown in Figure 3. Both CCM+DP and CCM+RV achieve comparably or better performance than existing algorithms, implying that the proposed contrastive context encoder can help to extract contextual information more effectively. For in-distribution tasks that vary in dynamics (i.e. ant-mass and cheetah-mass), CCM+DP obtains better returns and converges faster than CCM+RV. This is consistent to what is empirically found in CaDM (Lee et al. 2020): prediction models is more effective when the transition function changes across tasks. Moreover, recall that contrastive context encoder focus on the difference between tasks at trajectory level. Such property may compensate for DP's drawbacks which can embed commonly-shared dynamics knowledge that hinder the meta-learning process. We assume that this property leads to the better performance of CCM+DP on in-distribution tasks.

For OOD tasks, CCM+RV outperforms other methods including CCM+DP, implying that contrastive loss combined with loss from recovering value function obtains better generalization for different tasks. Recovering value function focuses on extracting high-level task-specific features. This property avoids the overfitting problem of dynamics prediction (limited to common dynamics in training tasks), of which is amplified in this case. However, solely using RV to train the encoder can perform poorly due to its noisy updating signal as well as the ignorance of detailed transition difference that contain task-specific information. Combining with CCM addresses such problems through the usage of low-variance InfoNCE loss as well as forcing the encoder explicitly find the trajectory-level information that varies tasks from each other.

Comparison of Overall Adaptation Performance on Complex Environments

We then consider both components of CCM algorithm and evaluate on several complex sparse-reward environments to compare the performance of CCM with prior Meta-RL algorithms and show whether the information-gain-based exploration strategy improve its adaptation performance. For cheetah-sparse and walker-sparse environments, agent receives a reward when it is close to the target velocity. In hard-point-robot, agent needs to reach two randomly selected goals one after another, and receives a reward when it's close enough to the goals. We compare CCM with state-of-the-art Meta-RL approach PEARL, and we further modify it with a contrastive context encoder for a fair comparison. We further consider MAML (Finn, Abbeel, and Levine 2017) and ProMP (Rothfuss et al. 2019), which adapt with on-policy methods. We also compare to varibad (Zintgraf et al. 2020), which can be viewed as a stable version of RL2 (Duan et al. 2016).

As shown in Figure 4, CCM achieves better performance than prior Meta-RL algorithms in both final returns and learning speed in such complex sparse-reward environments. Within relatively small number of environment interactions, on policy methods MAML, varibad and ProMP struggle to learn effective policies on these complex environment while off-policy context-based methods CCM and PEARL-CL generally achieves higher performance.

As the only difference between CCM and PEARL-CL here is whether to use the proposed extra exploration policy, we can conclude that CCM's information-gain-based exploration strategy enables fast adaptation by collecting more informative experience and further improving the quality of context when carrying out execution policies. Note that during training phases, trajectories collected by the exploration agent are used for updating the context encoder and the execution policy as well. As shown in experimental results, this may lead to faster training speed of context encoder and execution policy because of the more informative replay buffer.

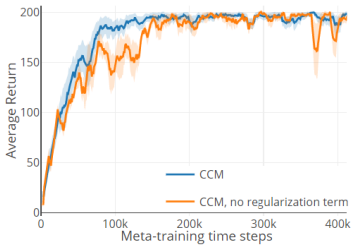

Figure 5: Influence of the regularization term

Influence of the Regularization Term

We perform ablation experiments in order to investigate the influence of the regularization term in the objective func-

tion which reflects the change in similarity score with all the tasks in embedding space (the second term in (14)). We compare against CCM using intrinsic reward without the regularization term on in-distribution version of hard-point-robot. The results are shown in Figure 5. The regularization term improves CCM's performance and we assume that without this term the exploration agent may try to collect transitions that the similarity score with all the tasks in embedding space increases as well, which will result in a noisy updating signal. Due to space limit, we put other ablation experiments in the appendix.

Visualization for Context

Finally, we visualize CCM's learned context during adaptation phases via t-SNE (Maaten and Hinton 2008) and compare with PEARL. We run the learned policies on ten randomly sampled test tasks multiple times to collect trajectories. Further, we encode the collected trajectories into latent context in embedding space with the learned context encoder and visualize via t-SNE in Figure 6. We find that the latent context generated by CCM from the same tasks is close together in embedding space while maintains clear boundaries between different tasks. In contrast, the latent context generated by PEARL shows globally messy clustering results with only a few reasonable patterns in local region. This indicates that CCM extracts high-quality (i.e. compact and sufficient) task-specific information from the environment compared with PEARL. As a result, the policy conditioned on the high-quality latent context is more likely to get a higher return on those meta-testing tasks, which is consistent to our prior empirical results.

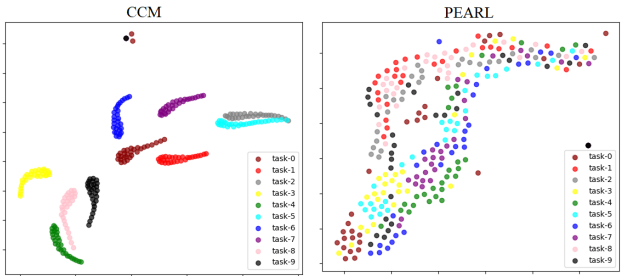

Figure 6: Visualization of context in embedding space. Different color represents context from different tasks.

Related Work

Contrastive Learning

Contrastive learning has recently achieved great success in learning representations for image or sequential data (van den Oord, Li, and Vinyals 2018; Hénaff et al. 2019; He et al. 2020; Chen et al. 2020). In RL, it has been used to extract reward signals in latent space (Sermanet et al. 2018; Dwibedi et al. 2018), or used as an auxiliary task to study representations for high-dimensional data (Srinivas, Laskin, and Abbeel 2020; Anand et al. 2019). Prior work (Srinivas, Laskin, and Abbeel 2020; Hénaff et al. 2019) has shown various methods of generating positive and negative pairs for image-based input data. The standard approach is to create multiple views of each datapoint like random crops and data augmentations (Wu et al. 2018; Chen et al. 2020; He et al. 2020). However, in this work we focus on low dimensional input data and leverage natural discrimination inside the trajectories of different tasks to generate positive and negative data. The selection of contrastive loss function is also various and the most competitive one is InfoNCE (van den Oord, Li, and Vinyals 2018). The motivation behind contrastive loss is the InfoMax principle (Linsker 1988), which can be interpreted as maximizing the mutual information between two views of data. The relationships between InfoNCE loss and mutual information is conprehensively explained in (Poole et al. 2019).

Meta-Reinforcement Learning

Meta-RL extends the framework of meta-learning (Thrun and Pratt 1998; Naik and Mammone 1992) to the reinforcement learning setting. We here consider off-policy context-based Meta-RL (Rakelly et al. 2019; Fu et al. 2019; Fakoor et al. 2020). Rakelly et al. (2019) propose PEARL that adapts to a new environment by inferring latent context variables from a small number of trajectories. Lee et al. (2020) learn a global model that generalizes across tasks by training a latent context to capture the local dynamics. In contrast, our approach directly focuses on the context itself, which motivates an algorithm to improve the quality of latent context. On the other hand, Some recent work like VariBAD (Zintgraf et al. 2020) focuses on-policy context-based Meta-RL methods.

Some prior research has proposed to explore to obtain the most informative trajectories in Meta-RL (Rakelly et al. 2019; Rothfuss et al. 2019; Gupta et al. 2018). The common idea behind these approaches is Thompson-sampling (Thompson 1933). However, this kind of posterior sampling is conditioned on learned execution policy which may lead to the exploration behavior only limited to the learned tasks. In contrast, we theoretically obtain a lower bound estimation of the exploration objective, based on which a separate exploration agent is trained.

Conclusion

In this paper, we propose that constructing a powerful context for Meta-RL involves two problems: 1) How to collect informative trajectories of which the corresponding context reflects the specification of tasks? 2) How to train a compact and sufficient encoder that can embed the task-specific information contained in prior trajectories? We then propose CCM which tackles them respectively. Firstly, CCM focuses on the underlying structure behind collected transitions and trains the encoder by leveraging contrastive learning. CCM further learns a separate exploration agent with an information-theoretical objective that aims to maximize the improvement of inference after collecting new transitions. The empirical results on several complex simulated control tasks show that CCM outperforms state-of-the-art Meta-RL methods by addressing the aforementioned problems.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (Grant Nos.: 61702362, U1836214, U1813204), Special Program of Artificial Intelligence and Special Program of Artificial Intelligence of Tianjin Municipal Science and Technology Commission (No.: 56917ZXRGGX00150), Tianjin Natural Science Fund (No.: 19JCYBJC16300), Research on Data Platform Technology Based on Automotive Electronic Identification System. We thank Chenjia Bai, Qianli Shen, and Weixun Wang for useful discussions and suggestions.

References

Anand, A.; Racah, E.; Ozair, S.; Bengio, Y.; Côté, M.; and Hjelm, R. D. 2019. Unsupervised State Representation Learning in Atari. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, 8766–8779.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *CoRR* abs/1606.01540.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *CoRR* abs/2002.05709.

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning. *CoRR* abs/1611.02779.

Dwibedi, D.; Tompson, J.; Lynch, C.; and Sermanet, P. 2018. Learning Actionable Representations from Visual Observations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018*, 1577–1584. IEEE.

Fakoor, R.; Chaudhari, P.; Soatto, S.; and Smola, A. J. 2020. Meta-Q-Learning. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, 1126–1135.

Fu, H.; Tang, H.; Hao, J.; Liu, W.; and Chen, C. 2019. MGHRL: Meta Goal-generation for Hierarchical Reinforcement Learning. *ArXiv* abs/1909.13607.

Gupta, A.; Mendonca, R.; Liu, Y.; Abbeel, P.; and Levine, S. 2018. Meta-Reinforcement Learning of Structured Exploration Strategies. In *NeurIPS*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 1856–1865.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 9726–9735. IEEE.

Hénaff, O. J.; Srinivas, A.; Fauw, J. D.; Razavi, A.; Doersch, C.; Eslami, S. M. A.; and van den Oord, A. 2019. Data-Efficient Image Recognition with Contrastive Predictive Coding. *CoRR* abs/1905.09272.

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

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016*.

Linsker, R. 1988. Self-Organization in a Perceptual Network. *Computer* 21(3): 105–117.

Liu, E. Z.; Raghunathan, A.; Liang, P.; and Finn, C. 2020. Explore then Execute: Adapting without Rewards via Factorized Meta-Reinforcement Learning. *CoRR* abs/2008.02790.

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

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540): 529–533. doi:10.1038/nature14236.

Naik, D. K.; and Mammone, R. 1992. Meta-neural networks that learn by learning. *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks* 1: 437–442 vol.1.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, 2778–2787. PMLR.

Poole, B.; Ozair, S.; van den Oord, A.; Alemi, A.; and Tucker, G. 2019. On Variational Bounds of Mutual Information. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, 5171–5180. PMLR.

Rakelly, K.; Zhou, A.; Finn, C.; Levine, S.; and Quillen, D. 2019. Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, 5331–5340.

Rothfuss, J.; Lee, D.; Clavera, I.; Asfour, T.; and Abbeel, P. 2019. ProMP: Proximal Meta-Policy Search. In *7th International Conference on Learning Representations, ICLR 2019*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M. I.; and Moritz, P. 2015. Trust Region Policy Optimization. In *Pro-*

ceedings of the 32nd International Conference on Machine Learning, ICML 2015, 1889–1897.

Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; and Levine, S. 2018. Time-Contrastive Networks: Self-Supervised Learning from Video. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018*, 1134–1141. IEEE.

Srinivas, A.; Laskin, M.; and Abbeel, P. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. *CoRR* abs/2004.04136.

Thompson, W. 1933. on the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25: 285–294.

Thrun, S.; and Pratt, L. Y. 1998. Learning to Learn. In *Springer US*.

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

van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748.

Wang, J. X.; Kurth-Nelson, Z.; Tirumala, D.; Soyer, H.; Leibo, J. Z.; Munos, R.; Blundell, C.; Kumaran, D.; and Botvinick, M. 2016. Learning to reinforcement learn. *CoRR* abs/1611.05763.

Wu, Z.; Xiong, Y.; Yu, S.; and Lin, D. 2018. Unsupervised Feature Learning via Non-parametric Instance Discrimination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 3733–3742.

Zhang, J.; Wang, J.; Hu, H.; Chen, Y.; Fan, C.; and Zhang, C. 2020. Learn to Effectively Explore in Context-Based Meta-RL. *CoRR* abs/2006.08170.

Zhou, W.; Pinto, L.; and Gupta, A. 2019. Environment Probing Interaction Policies. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Zintgraf, L. M.; Shiarlis, K.; Igl, M.; Schulze, S.; Gal, Y.; Hofmann, K.; and Whiteson, S. 2020. VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.