Improving Generalization in Offline Meta-Reinforcement Learning via Cross-task Contexts

Hongcai He[1], **Zetao Zheng**[1,2], **Anjie Zhu**[1], **Deqiang Ouyang**[3,*] **Jie Shao**[1,2]
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Sichuan Artificial Intelligence Research Institute, Yibin, China
[3]Chongqing University, Chongqing, China
{hehongcai,anjiezhu}@std.uestc.edu.cn, {ztzheng,shaojie}@uestc.edu.cn, deqiangouyang@cqu.edu.cn

Abstract

Context-based offline meta-reinforcement learning (meta-RL) is a paradigm that integrates meta-learning with offline reinforcement learning. It learns a strategy to extract task-specific contexts from trajectories of meta-training tasks and leverages this strategy for adapting to unseen target tasks. However, existing methods struggle to generate generalizable contexts for adaptations due to context shift, which arises from the context-based policy overfitting to offline data. We argue that leveraging the internal relationships among tasks, rather than treating each task in isolation, is crucial for mitigating the impact of context shift. Hence, we propose a framework called cross-task contexts for improving generalization in meta-RL (CTMRL). Specifically, we design a context quantization variational auto-encoder (CQ-VAE), which clusters task-specific contexts of meta-training tasks into discrete codes based on the internal relationships among tasks. Cross-task contexts are constructed with these codes, reflecting shared information across similar tasks. These cross-task contexts not only serve as high-level structures to capture similarity across tasks but also provide a foundation for hard contrastive learning that enhances the distinguishability of similar yet distinct tasks, thereby improving the generalization of contexts and facilitating adaptation to unseen target tasks. The evaluation in meta-environments confirms the performance advantage of CTMRL over existing methods.

Code — https://github.com/hehongc/CTMRL
Extended version — https://github.com/hehongc/CTMRL

Introduction

Context-based offline meta-reinforcement learning (meta-RL) is a paradigm that integrates meta-learning and offline reinforcement learning, aiming to apply the generalization capability of meta-learning to reinforcement learning tasks. Methods under this paradigm focus on learning a strategy to extract contexts from historical trajectories and leverage contexts to address tasks. These contexts capture information about task characteristics and decision-making patterns. Then, this strategy is used to leverage the learned patterns and adapt to new tasks. Existing methods (Li, Yang, and Luo 2021; Li et al. 2021; Yuan and Lu 2022; Gao et al. 2023; Zhou et al. 2024) learn the strategy from trajectories sampled from offline datasets of meta-training tasks during the meta-training phase to avoid expensive online interactions with real or simulated environments. Subsequently, they collect a small number of online trajectories from unseen target tasks (meta-testing tasks) and leverage the extracted contexts to make adaptations during the meta-testing phase. Moreover, these methods have achieved notable progress in the domain of continuous control.

However, most context-based offline meta-RL methods (Zhou et al. 2024; Gao et al. 2023; Yuan and Lu 2022) treat individual tasks in isolation and rely solely on the superficial differences indicated by task labels, without considering deeper relationships among tasks. These methods only try to generate task-specific contexts for task solutions, overlooking the *inherent similarities and differences among tasks*. This operation makes them particularly vulnerable to *context shift* (Pong et al. 2022; Wang et al. 2023; Gao et al. 2023). This challenge limits the generalization of the generated contexts and ultimately degrades adaptation performance on unseen target tasks. More specifically, since meta-RL shares connections with both meta-learning and reinforcement learning, this challenge is closely related to the classical memorization challenge in meta-learning (Yin et al. 2020) and the ambiguity in Markov decision processes (MDPs) (Li et al. 2020, 2021). It arises when the context-based policy overfits to the offline datasets of meta-training tasks during the meta-training phase, making it difficult to adapt to the online trajectories collected from unseen target tasks during the meta-testing phase.

We argue that fully leveraging the internal relationships among tasks offers an effective perspective for mitigating the impact of context shift. Such relationships are essential for the generalization of contexts and the effective adaptation to unseen target tasks. Some tasks share similar structures with minimal differences, while others differ significantly. By exploiting these similarities, we construct a high-level structure that captures a range of relatively similar tasks, rather than being limited to each task individually. Meanwhile, distinguishability among these similar tasks is important to avoid potential confusion. Crucially, these relationships should be explicitly encoded into the learned contexts, as contexts inherently carry task information. In this work,

*Corresponding author: Deqiang Ouyang.

we aim to leverage the aforementioned structure, which captures finer-grained similarities and differences across tasks, to enhance the generalization of contexts and improve the adaptation to unseen target tasks. Specifically, we construct cross-task contexts by clustering task-specific contexts of meta-training tasks, assigning all task-specific contexts of each task to a single cluster. Training with cross-task contexts enables the optimization of task-specific contexts for a range of similar tasks, instead of treating each task as entirely separate and distinct. Meanwhile, these cross-task contexts also provide a foundation for measuring similarities across tasks, enabling us to further enhance the distinguishability among the task-specific contexts of tasks that are similar yet distinct.

To this end, we propose a framework called cross-task contexts for improving generalization in meta-RL (CTMRL). Specifically, inspired by the vector quantization variational auto-encoder (VQ-VAE) (van den Oord, Vinyals, and Kavukcuoglu 2017), which maps a continuous latent space of vectors into discrete codes, we design a context quantization variational auto-encoder (CQ-VAE), capturing the shared structure across similar tasks. CQ-VAE consists of a context encoder, a codebook with learnable and discrete codes, and context-based reward and state predictors. The context encoder is used to generate task-specific contexts. Then, the codebook stores codes generated by clustering the task-specific contexts of meta-training tasks, capturing the shared structure across similar tasks. These codes are used to construct cross-task contexts, which in turn help establish comprehensive relationships among task-specific contexts. Next, the context-based reward and state predictors as the decoder structure, optimize CQ-VAE in a supervised manner by accurately predicting rewards and next states based on task-specific and cross-task contexts. Furthermore, we design a contrastive loss that leverages task distances measured by CQ-VAE to construct hard negative samples, enhancing the distinguishability of task-specific contexts for similar yet distinct tasks. Overall, by modeling the varying similarity and distinctness degrees among tasks, CTMRL builds comprehensive relationships among contexts and effectively captures both intra-task and inter-task structures, generating generalizable task-specific contexts and adapting effectively to unseen target tasks. The main contributions of CTMRL are fourfold:

- We propose CTMRL, which constructs comprehensive relationships among contexts by considering varying degrees of similarity and distinctness among tasks, aiming to mitigate the impact of context shift from a novel perspective.
- We introduce CQ-VAE to capture the shared structure across similar tasks from the task-specific contexts, rather than just treating each task as separate and distinct.
- We design a novel contrastive loss that constructs hard negative samples based on task distances measured by CQ-VAE to enhance the distinguishability of task-specific contexts.
- We validate the effectiveness of CTMRL in the MuJoCo environments, and results show substantial performance improvements over existing meta-RL methods.

Related Work

Meta-reinforcement learning. Meta-RL focuses on learning a strategy from a series of meta-training tasks and applying it to effectively adapt to unseen target tasks (meta-testing tasks). Existing meta-RL methods can be broadly divided into two main categories: optimization-based and context-based methods. Specifically, optimization-based methods (Finn, Abbeel, and Levine 2017; Foerster et al. 2018; Houthooft et al. 2018) learn an optimal policy initialization for meta-training tasks by executing policy gradients on limited samples and leveraging the internal relationships among tasks. Meanwhile, context-based methods extract contexts with task information from historical trajectories through recurrent (Fakoor et al. 2020; Wang et al. 2017), recursive (Mishra et al. 2018), or probabilistic (Rakelly et al. 2019; Zintgraf et al. 2020) structures. These contexts are utilized to guide the meta-policy, which is designed to handle all tasks, in completing these tasks. CTMRL belongs to the category of context-based methods.

Context-based offline meta-RL. Context-based offline meta-RL extends context-based meta-RL by incorporating an offline setting during the meta-training phase. Specifically, it extracts contexts from trajectories sampled from offline datasets, rather than relying on online interactions with environments. FOCAL (Li, Yang, and Luo 2021) employs behavior regularization to constrain task inference. CORRO (Yuan and Lu 2022) introduces contrastive learning to generate robust task representations. IDAQ (Wang et al. 2023) constructs a return-based uncertainty quantification to generate in-distribution contexts for tasks. CSRO (Gao et al. 2023) tries to mitigate the effects of context shift through a proposed max-min mutual information representation learning mechanism. GENTLE (Zhou et al. 2024) further captures task information by reconstructing state transitions and rewards. UNICORN (Li et al. 2024a) aims to improve contexts by reconstructing task-specific models. Given the contexts, ER-TRL (Nakhaeinezhadfard, Scannell, and Pajarinen 2025) seeks to reduce the mutual information between tasks and behaviors by promoting higher conditional entropy of the behavior policy. However, these methods are limited in leveraging the internal relationships among tasks to construct comprehensive relationships among contexts. They suffer significantly from context shift as they rely solely on the explicit differences indicated by task labels and treat each task in isolation. To address this issue, we cluster the task-specific contexts of meta-training tasks to capture the shared structure across similar tasks and construct cross-task contexts accordingly. These cross-task contexts serve as a foundation for modeling the degrees of similarity and distinctness among tasks. Overall, CTMRL generates generalizable task-specific contexts and achieves effective adaptation to unseen target tasks.

Preliminaries

Reinforcement learning. A reinforcement learning (RL) task is typically modeled as a fully observable Markov decision process (MDP), represented as a tuple $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma, \rho_0 \rangle$. Specifically, $\mathcal{S}$ is the state space, $\mathcal{A}$ is

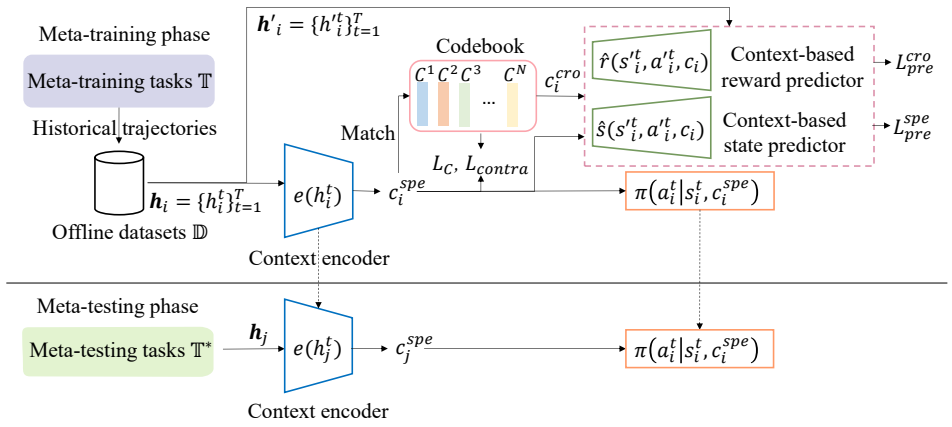

Figure 1: **Framework overview. (a) Meta-training** extracts task-specific contexts c_i^{spe} from offline trajectories $\boldsymbol{h}_i$ and uses them to guide decision-making. CQ-VAE, which consists of the context encoder $e(h_i^t)$, codebook $CB = \{C^1, \ldots, C^N\}$, and context-based reward and state predictors, $\hat{r}(s'^t_i, a'^t_i, c_i)$ and $\hat{s}(s'^t_i, a'^t_i, c_i)$, is used to generate cross-task contexts c_i^{cro} and improve the generalization of task-specific contexts. **(b) Meta-testing** extracts task-specific contexts c_j^{spe} from online trajectories $\boldsymbol{h}_j$ and leverages them to adapt to unseen target tasks $\mathcal{T}_j$.

the action space, and $s \in \mathcal{S}$ and $a \in \mathcal{A}$ represent the state and action at time-step t, respectively. $p(s^{t+1}|s^t, a^t)$ is the transition function, $r(s^t, a^t)$ is the reward function, ρ_0 is the initial distribution of states, and $\gamma \in [0, 1)$ is the discount factor for future rewards. A stochastic policy $\pi(a^t|s^t)$ is modeled as the distribution of actions. The marginal distribution of states at time-step t is defined as $\mu_\pi^t(s^t)$. In MDPs, the goal of agents is to maximize the expected cumulative rewards over time as: $max_\pi \mathcal{J}_\mathcal{M}(\pi) = \mathbb{E}_{s^t \sim \mu_\pi^t, a^t \sim \pi}[\sum_{t=0}^{\infty} \gamma^t r(s^t, a^t)]$.

Context-based offline meta-reinforcement learning. Context-based offline meta-RL is generally formalized as a partially observable Markov decision process (POMDP) (Kaelbling, Littman, and Cassandra 1998), where states in the environment are only partially observable. In contrast to s_i, the context, as the unobservable component of the complete state, corresponds to the task information. Agents need to extract the context from offline data and utilize it to make decisions: $a_i^t \sim \pi(a_i^t|s_i^t, c_i)$, where c_i represents the context associated with task $\mathcal{T}_i$. The objective of agents in POMDPs is similar to that in MDPs. Moreover, context-based offline meta-RL methods assume access to two sets of tasks, which differ in their transition or reward functions: a set of n_{task} meta-training tasks $\mathbb{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{n_{task}}\}$ and a set of unseen target tasks $\mathbb{T}^*$. Each task is modeled as a POMDP. $\mathbb{T}$ corresponds to a set of offline datasets $\mathbb{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_{n_{task}}\}$ for the meta-training phase.

Method

As illustrated in Figure 1, CTMRL operates in two main phases: meta-training and meta-testing. In the meta-training phase, CTMRL learns to extract task-specific contexts c_i^{spe} from historical trajectories $\boldsymbol{h}_i$ sampled from the offline dataset $\mathcal{D}_i$ associated with meta-training task $\mathcal{T}_i$. In the meta-testing phase, for an unseen target task $\mathcal{T}_j$, CTMRL extracts task-specific contexts c_j^{spe} from a few online trajectories $\boldsymbol{h}_j$ and uses c_j^{spe} as one of the decision conditions to guide effective adaptation to $\mathcal{T}_j$.

Meta-training

In the meta-training phase, CTMRL aims to capture the shared structure across similar tasks by clustering task-specific contexts from meta-training tasks and constructing cross-task contexts, which serve as the foundation for modeling varying degrees of task similarity and distinctness.

CQ-VAE Existing methods (Li, Yang, and Luo 2021; Li et al. 2021; Yuan and Lu 2022; Gao et al. 2023; Zhou et al. 2024) typically use a context encoder $e(h_i^t)$ to encode each transition h_i^t, where $t \in [1, T]$, within the trajectory $\boldsymbol{h}_i$ that consists of T time-steps and corresponds to the meta-training task $\mathcal{T}_i$, into a representation c_i^t. Then, the sequence of c_i^t is aggregated into the task-specific context c_i^{spe} of $\mathcal{T}_i$ by taking the mean: $c_i^{spe} = mean(\{c_i^t\}_{t=1}^T)$. Since c_i^{spe} represents the task information of $\mathcal{T}_i$, it is used to guide the context-based policy $\pi(a_i^t|s_i^t, c_i^{spe})$ in completing that task. However, we argue that existing methods overlook the internal relationships among tasks and fail to account for varying degrees of task similarity and distinctness. This limitation hinders the learning strategy from constructing comprehensive relationships among task-specific contexts, thereby impairing their generalization. Vector quantization-variational auto-encoder (VQ-VAE) is a module that discretizes the continuous representation space, outputting discrete codes instead of continuous vectors. Building on the concept of discretization, DCMRL (He et al. 2024) and Choreographer (Mazzaglia et al. 2023) decouple the exploration and learning of the continuous latent space, while LISA (Garg et al. 2022) enhances the interpretability of the learned skills. Inspired by these methods, we cluster the task-specific contexts of meta-training tasks into discrete centers to capture the shared structure across similar tasks. These centers serve as codes for constructing cross-task contexts, which

form the foundation for modeling varying degrees of similarity and distinctness among tasks. Therefore, we propose a context quantization variational auto-encoder (CQ-VAE), which employs $e(h_i^t)$ as the encoder structure to extract task-specific contexts. A codebook CB within learnable and discrete codes is introduced to represent shared structures and construct cross-task contexts c_i^{cro}. The decoder structure consists of context-based reward and state predictors $\hat{s}(s'^t_i, a'^t_i, c_i)$ and $\hat{r}(s'^t_i, a'^t_i, c_i)$, which are trained to reconstruct reward and next states based on c_i^{spe} or c_i^{cro}.

Codebook. Codebook $CB = \{C^1, \ldots, C^N\}$ comprises N learnable discrete codes formed as vectors, where N is a fixed hyperparameter. To ensure each code captures the shared structure across similar tasks, we employ a matching process similar to online clustering of c_i^{spe}. Specifically, each c_i^{spe} generated by $e(h_i^t)$ is matched with its closest code C^n within CB, where $1 \leq n \leq N$, based on cosine similarity. When a match occurs, the matched C^n is updated toward c_i^{spe} at a learning rate, allowing C^n to gradually evolve into a cluster center that captures shared information across similar tasks. The complete matching process is as follows:

$$c_i^{cro} = C^i = \underset{C^n \in CB, 1 \leq n \leq N}{\operatorname{argmax}} cos(c_i^{spe}, C^n), \tag{1}$$

$$C^i_{new} = (1 - \eta) \cdot C^i_{old} + \eta \cdot c_i^{spe}, \tag{2}$$

where $cos(\cdot, \cdot)$ denotes the cosine similarity between two vectors, C^i_{old} is the code matched to c_i^{spe} before the update, C^i_{new} is the updated version of C^i_{old} after matching, and η is the learning rate. Additionally, based on this hard assignment matching strategy, we construct a contrastive loss to model the connection between task-specific contexts and codes. Following the widely adopted InfoNCE loss formulation (van den Oord, Li, and Vinyals 2018), it is defined as:

$$L_C = -\sum_{i=1}^{B} log \frac{exp(\frac{cos(c_i^{spe}, C_i)}{\tau})}{\sum_{j=1}^{N} exp(\frac{cos(c_i^{spe}, C_j)}{\tau})}, \tag{3}$$

where B is the batch size of meta-training tasks. This loss further facilitates the optimization of C^i in CB, enabling more effective construction of the shared structure across similar tasks and reducing ambiguity in the matching process.

In contrast to existing methods (He et al. 2024; Mazzaglia et al. 2023; Garg et al. 2022) that randomly initialize codes within CB, we introduce a warm-up process to initialize CB with the K-means algorithm and task-specific contexts of meta-training tasks. In each warm-up episode, we sample trajectories from all offline datasets and generate corresponding task-specific contexts using $e(h_i^t)$. These contexts are then clustered into N centers with the K-means algorithm, which are used to initialize CB. Beyond initialization, we continue to refine the codes in CB by applying K-means during training. We set an interval of episodes and collect c_i^{spe} of all meta-training tasks after each episode within this interval. Each c_i^{spe} is then clustered into one of N centers with the K-means algorithm, and the same matching process (Eq. (1) and Eq. (2)) is applied for the learning of codes.

Context-based predictors. Inspired by GENTLE (Zhou et al. 2024) and UNICORN (Li et al. 2024a) that capture the generative structure of the task models through reconstruction, we propose a context-based reward predictor $\hat{r}(s'^t_i, a'^t_i, c_i)$ and a context-based state predictor $\hat{s}(s'^t_i, a'^t_i, c_i)$ as the decoder structure of CQ-VAE to achieve similar optimization. Specifically, $\hat{r}(s'^t_i, a'^t_i, c_i^{spe})$ and $\hat{s}(s'^t_i, a'^t_i, c_i^{spe})$ are used to predict r'^t_i and s'^{t+1}_i of h'^t_i within $\boldsymbol{h'_i}$, optimizing c_i^{spe} based on the prediction accuracy. These processes are as follows:

$$L_r^{spe} = \sum_{t=1}^{T} \|\hat{r}(s'^t_i, a'^t_i, c_i^{spe}) - r'^t_i\|_2, \tag{4}$$

$$L_s^{spe} = \sum_{t=1}^{T} \|\hat{s}(s'^t_i, a'^t_i, c_i^{spe}) - s'^{t+1}_i\|_2. \tag{5}$$

Considering the similar effects of these two losses and for simplicity, we sum them with equal weight to form the prediction objective as follows:

$$L_{pre}^{spe} = L_r^{spe} + L_s^{spe}. \tag{6}$$

Building on this, we use these two modules to refine the cross-task context c_i^{cro}, further optimizing the corresponding C^i in CB. We input c_i^{cro} into these two context-based predictors and aim for them to accurately predict rewards r'^t_i and next states s'^{t+1}_i given s'^t_i, a'^t_i, and the shared c_i^{cro} across a range of similar tasks. Therefore, this operation encourages c_i^{cro} to capture the shared structure across similar tasks and promotes the learning of their task-specific contexts. These processes are as follows:

$$L_r^{cro} = \sum_{t=1}^{T} \|\hat{r}(s'^t_i, a'^t_i, c_i^{cro}) - r'^t_i\|_2, \tag{7}$$

$$L_s^{cro} = \sum_{t=1}^{T} \|\hat{s}(s'^t_i, a'^t_i, c_i^{cro}) - s'^{t+1}_i\|_2, \tag{8}$$

$$L_{pre}^{cro} = L_r^{cro} + L_s^{cro}. \tag{9}$$

Notably, $e(h_i^t)$, $\hat{r}(s'^t_i, a'^t_i, c_i)$ and $\hat{s}(s'^t_i, a'^t_i, c_i)$ are implemented as learnable neural networks. Unlike methods such as CORRO and GENTLE, CTMRL is trained in an end-to-end manner without requiring any pretraining of individual modules. Our experiments confirm that this design has no impact on performance. In summary, CTMRL utilizes CQ-VAE to construct c_i^{cro}, which connects a range of similar tasks measured by C^i in CB, thereby capturing the shared structure rather than treating each task in isolation. By exploiting this shared structure, CTMRL facilitates the optimization process and improves the generalization of task-specific contexts.

Contrastive Learning based on CQ-VAE Since contexts encode task-specific information of different tasks, it is crucial to extend inter-task relationships to task-specific contexts. Since CTMRL captures the shared structure across similar tasks through cross-task contexts, it is critical to establish the distinguishability among the task-specific contexts of these similar yet distinct tasks for constructing comprehensive relationships among contexts. As an approach

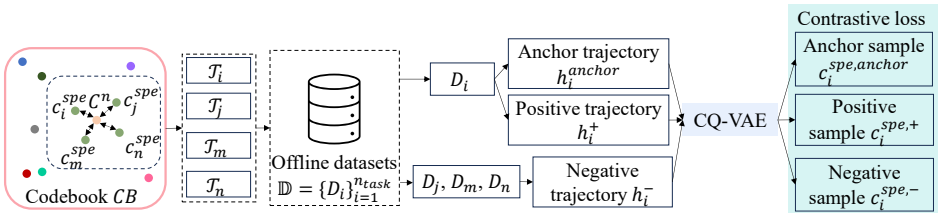

Figure 2: **Illustration of contrastive learning based on CB.** For a code $C^n \in CB$ and corresponding matching task-specific contexts c_i^{spe}, c_j^{spe}, c_m^{spe} and c_n^{spe} of $\mathcal{T}_i$, $\mathcal{T}_j$, $\mathcal{T}_m$ and $\mathcal{T}_n$, we construct contrastive loss to optimize CQ-VAE. Taking $\mathcal{T}_i$ as an example, we sample two different trajectories h_i^{anchor} and h_i^+ from D_i, and respectively sample negative trajectories h_i^- from D_j, D_m and D_n. These trajectories are used to generate $c_i^{spe,anchor}$, $c_i^{spe,+}$ and $c_i^{spe,-}$, and treated as components to construct contrastive loss. Moreover, this process will also be performed on $\mathcal{T}_j$, $\mathcal{T}_m$ and $\mathcal{T}_n$.

that pulls the positive sample close to the anchor sample and pushes the negative sample far away, contrastive learning can be used to enhance the generalization of task-specific contexts (Yuan and Lu 2022; Li et al. 2021; He et al. 2024; Li et al. 2024b). A common approach to constructing the anchor samples $\boldsymbol{c}_i^{spe,anchor}$ and positive samples $c_i^{spe,+}$ is to sample two different trajectories, $\boldsymbol{h}_i^{anchor}$ and $\boldsymbol{h}_i^+$ from D_i, and extract these samples from them. Meanwhile, $c_j^{spe,+}$ of $\mathcal{T}_j$ is treated as the negative sample of $c_i^{spe,anchor}$. R2PGO (Li et al. 2024b) constructs hard negative samples based on task distances measured by task parameters, which are often unavailable in real or simulated environments. To address this, we propose a novel approach to construct hard negative samples based on task distances measured by CB of CQ-VAE, thereby enhancing the distinguishability among the task-specific contexts of similar yet distinct tasks.

Codes within CB represent the shared structure across similar tasks, as task-specific contexts of similar tasks match the same code. We iterate through each code in CB, and treat the tasks whose task-specific contexts are matched to the same code as similar. Based on these groupings, we construct the contrastive loss. As illustrated in Figure 2, for a particular task $\mathcal{T}_i$, we extract $c_i^{spe,anchor}$ and $c_i^{spe,+}$ from $\boldsymbol{h}_i^{anchor}$ and $\boldsymbol{h}_i^+$. Then, except for $c_i^{spe,+}$, each $c_j^{spe,+}$, where $j \neq i$, is treated as the negative sample $c_i^{spe,-}$ for $c_i^{spe,anchor}$. If a particular code is matched to only one task $\mathcal{T}_i$, we randomly sample M additional tasks to construct the contrastive loss, where M is a fixed hyperparameter. In rare cases, some codes may not be matched to any task-specific contexts, typically due to an excessively large codebook size N. To avoid this, we set an appropriate value of N to ensure effective code utilization. We construct the contrastive loss as follows:

$$L_{contra} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\log\frac{\exp(\frac{\cos(c_k^{spe,anchor},c_k^{spe,+})}{\tau})}{\sum_{l=1}^{K}\exp(\frac{\cos(c_k^{spe,anchor},c_l^{spe,+})}{\tau})}, \tag{10}$$

where τ is a temperature parameter. K is the number of tasks whose task-specific contexts are matched to the same code, potentially varying across different codes. Overall, by further enhancing the distinguishability of task-specific contexts belonging to different yet similar tasks within the shared structure established by CQ-VAE, CTMRL constructs comprehensive relationships among contexts. These relationships mitigate task confusion and further improve the generalization of the task-specific contexts.

Meta-testing

CTMRL aims to achieve effective adaptation to unseen target tasks within $\mathbb{T}^*$ with our trained context encoder $e(h_j^t)$ from CQ-VAE and context-based policy $\pi(a_j^t|s_j^t, c_j^{spe})$ during the meta-testing phase. For an unseen target task $\mathcal{T}_j$, CTMRL first collects a limited number of online trajectories $\boldsymbol{h}_j = \{h_j^t\}_{t=1}^T$, and then encodes each h_j^t into c_j^t through $e(h_j^t)$. Next, $\{c_j^t\}_{t=1}^T$ can be aggregated into c_j^{spe} through $mean(\cdot)$ operation. Subsequently, CTMRL utilizes c_j^{spe} to guide $\pi(a_j^t|s_j^t, c_j^{spe})$ in generating actions. Notably, the data used for context generation is collected in two distinct phases: (1) In the first phase, the agent takes random actions to collect initial trajectories; (2) In the second phase, actions are selected using the $\pi(a_j^t|s_j^t, c_j^{spe})$ conditioned on c_j^{spe}. Pseudo-codes of meta-training and meta-testing phases, the theoretical analysis and more implementation details of CTMRL are in our online appendix.

Experiments

Experimental setup. We conduct a series of experiments in multiple meta-environments to evaluate CTMRL for generating generalizable task-specific contexts and achieving effective adaptation to unseen target tasks. We compare CTMRL with ER-TRL (Nakhaeinezhadfard, Scannell, and Pajarinen 2025), UNICORN (Li et al. 2024a), GENTLE (Zhou et al. 2024), IDAQ (Wang et al. 2023), CSRO (Gao et al. 2023), ANOLE (Ren et al. 2022) and CORRO (Yuan and Lu 2022) across seven benchmark environments: Sparse-Point-Robot, Point-Robot-Wind, Point-Robot, Half-Cheetah-Vel, Hopper-Rand-Params, Walker-Rand-Params and Ant-Goal. All methods are evaluated using the same offline datasets to ensure a fair comparison, although this may lead to deviations from the originally reported performance of some baselines. All experimental results are averaged across six random seeds and their variances are measured with a 95% bootstrap confidence interval. More details of environments and baselines are in our online appendix.

Environment	CTMRL (ours)	ER-TRL	UNICORN	GENTLE	IDAQ	CSRO	ANOLE	CORRO
	Out-of-distribution							
Half-Cheetanh-Vel	**-108.30±13.54**	-125.93±7.48	-124.15±8.47	-131.01±33.94	-127.00±21.03	-126.65±9.13	-121.77±17.55	-124.93±24.00
Point-Robot	-4.81±0.12	-4.81±0.14	-4.78±0.09	-7.31±1.22	**-4.76±0.07**	-4.78±0.14	-5.05±0.05	-5.82±0.50
Point-Robot-Wind	-5.69±0.43	-6.80±1.68	-15.61±1.15	-5.98±0.27	**-5.56±0.28**	-15.95±3.36	-5.81±0.39	-12.24±4.98
Sparse-Point-Robot	**12.45±0.33**	12.31±0.55	12.38±1.11	5.27±1.16	**12.45±0.22**	11.06±1.36	11.99±0.93	7.22±2.75
Hopper-Rand-Params	**349.30±36.51**	345.12±25.95	313.87±22.75	238.09±21.94	314.00±18.59	348.78±30.38	310.92±49.51	256.79±6.48
Walker-Rand-Params	**332.74±32.53**	317.30±9.32	319.35±14.79	320.04±7.35	303.43±31.98	317.39±19.52	315.00±19.64	319.01±14.79
Ant-Goal	**-378.81±4.43**	-616.43±8.17	-500.71±4.48	-500.84±2.79	-417.42±2.20	-524.41±92.41	-691.59±4.47	-614.59±4.10
	In-distribution							
Half-Cheetanh-Vel	**-111.68±15.89**	-125.55±6.06	-124.55±6.76	-132.26±24.00	-121.29±15.38	-120.18±14.75	-120.95±11.24	-119.26±12.50
Point-Robot	-4.84±0.10	-4.81±0.14	**-4.73±0.02**	-7.63±2.29	-4.76±0.04	-4.77±0.05	-5.11±0.02	-5.86±0.71
Point-Robot-Wind	-5.69±0.13	-6.63±1.75	-15.55±1.27	-5.80±0.16	-5.69±0.06	-15.82±3.55	**-5.67±0.18**	-12.13±4.85
Sparse-Point-Robot	12.52±0.21	12.55±0.19	12.29±1.34	6.14±0.85	**12.61±0.09**	11.18±1.31	12.12±0.84	5.89±2.07
Hopper-Rand-Params	**351.25±22.55**	306.10±24.36	286.83±4.49	244.51±14.66	284.99±6.93	286.69±1.01	276.64±16.32	263.66±12.94
Walker-Rand-Params	**352.24±28.44**	342.80±12.50	316.30±20.03	327.89±17.62	306.42±29.85	327.68±10.99	326.82±36.33	338.42±9.22
Ant-Goal	**-319.76±3.26**	-619.23±8.24	-517.66±4.13	-507.70±5.81	-403.12±6.55	-490.31±132.81	-697.59±4.69	-625.48±8.53

Table 1: Comparison results.

Comparison with baselines. We evaluate CTMRL for both meta-testing tasks (out-of-distribution) and meta-training tasks (in-distribution). As shown in Table 1, CTMRL consistently achieves strong adaptation performance across most environments, outperforming all baselines in both settings. In the Point-Robot and Point-Robot-Wind environments, CTMRL performs slightly worse than the best-performing baseline but remains highly competitive. These results indicate that our CQ-VAE can effectively capture the shared structure across similar tasks and apply it to construct comprehensive relationships among contexts. Therefore, CTMRL enhances the generalization of task-specific contexts and improves the adaptation to unseen target tasks. Additionally, we present the adaptation processes in our online appendix.

Ablation study. To validate the effectiveness of each component in CTMRL, we design four variants: CTMRL without L_C, CTMRL without L_{pre}^{cro}, CTMRL without K-means and CTMRL without L_{contra}. Each variant removes a single component from the original CTMRL. We evaluate these variants to analyze the individual impact of each component on overall performance. As shown in Table 2, all components are essential for maintaining strong performance on both meta-training and meta-testing tasks. The removal of any single component results in a degradation of the adaptation performance, highlighting its significance. These findings demonstrate that the full integration of all components is critical for effectively capturing the shared structure across tasks, enhancing the generalization of task-specific contexts, and improving the adaptation to unseen target tasks. However, for meta-training tasks, the removal of modules that promote generalization may result in performance gains, which are likely attributed to overfitting.

Effect of code number. The number of codes in the codebook CB is determined by a crucial and fixed hyperparameter N. To analyze the impact of different values of N, we conduct a series of experiments across multiple environments. Considering that each code in the codebook is designed to capture the shared structure across a range of similar tasks, we set the value of N to be smaller than the total number of meta-training tasks in the environments. Specifically, we evaluate $N \in \{2, 4, 8, 16, 32, 64\}$. Due to the limited number of meta-training tasks in certain environments, the settings $N = 32$ and $N = 64$ are inapplicable to the Hopper-Rand-Params, Walker-Rand-Params and Ant-Goal environments, while $N = 64$ is also inapplicable to the Point-Robot-Wind environment. As shown in Table 3, choosing an appropriate value of N is crucial for the performance of CTMRL. The optimal value of N varies across environments and may differ between meta-training and meta-testing tasks within the same environment. This suggests that the number of codes plays a key role in balancing context generalization and task-specific adaptation.

In adapting to unseen target tasks, CTMRL exhibits a consistent trend for the number of codes N. When N is too small, the captured shared structure becomes overly coarse, potentially introducing confusion among similar tasks and leading to suboptimal performance. As N increases, the performance improves and reaches its peak at an appropriate value, where the balance between generalization and task-specific representation is best maintained. However, further increasing N leads to overfitting to individual tasks, reducing context generalization and resulting in performance degradation. In the Hopper-Rand-Params environment, performance continues to improve within the tested range of N values, without showing a decline. When maintaining performance in meta-training tasks, the effect of the N value becomes more complex. In the Half-Cheetah-Vel and Point-Robot-Wind environments, it exhibits a trend similar to that observed in meta-testing tasks. Nevertheless, in other environments, performance typically peaks at a smaller N, followed by a decline and sometimes a recovery at larger values. We attribute this to the fact that, with sufficient offline data, a small N may already suffice to capture the necessary structure for meta-training tasks. In contrast, larger values

Environment	CTMRL (ours)	CTMRL w/o L_C	CTMRL w/o L_{pre}^{cro}	CTMRL w/o K-means	CTMRL w/o L_{contra}
	Out-of-distribution				
Half-Cheetanh-Vel	**-108.30±13.54**	-112.41±15.79	-135.42±20.72	-115.38±8.80	-132.16±19.71
Point-Robot	**-4.81±0.12**	-4.83±0.12	-4.93±0.15	-4.86±0.21	-4.82±0.11
Point-Robot-Wind	**-5.69±0.43**	-5.81±0.54	-5.96±0.55	-5.72±0.36	-5.75±0.28
Sparse-Point-Robot	**12.45±0.33**	12.40±0.27	12.34±0.38	11.83±0.72	12.09±0.51
Hopper-Rand-Params	**349.30±36.51**	342.32±37.48	343.60±20.19	298.31±19.81	336.76±49.86
Walker-Rand-Params	**332.74±32.53**	331.69±33.62	327.57±6.67	325.91±5.61	324.99±43.06
Ant-Goal	**-378.81±4.43**	-381.80±8.30	-382.38±8.14	-385.98±2.76	-414.74±3.67
	In-distribution				
Half-Cheetanh-Vel	**-111.68±15.89**	-114.59±15.24	-123.95±12.44	-114.21±8.10	-132.03±18.93
Point-Robot	**-4.84±0.10**	-4.86±0.05	-4.86±0.08	-4.85±0.09	-4.85±0.12
Point-Robot-Wind	-5.69±0.13	-5.71±0.16	-5.77±0.14	-5.85±0.14	**-5.64±0.10**
Sparse-Point-Robot	**12.52±0.21**	12.44±0.20	12.48±0.24	12.35±0.15	12.48±0.09
Hopper-Rand-Params	351.25±22.55	350.46±20.58	348.24±7.98	312.42±28.34	**374.69±26.57**
Walker-Rand-Params	352.24±28.44	349.93±34.56	**383.59±14.98**	361.24±12.04	331.47±33.55
Ant-Goal	-319.76±3.26	-320.97±4.08	-320.55±2.17	**-311.40±3.85**	-334.62±3.45

Table 2: Ablation results.

Environment	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$
	Out-of-distribution					
Half-Cheetanh-Vel	-151.46±5.84	-119.53±5.16	-113.96±15.29	-108.30±13.54	-127.97±23.73	-131.19±16.07
Point-Robot	-4.95±0.05	-4.91±0.17	-4.81±0.12	-4.81±0.08	-4.86±0.10	-4.86±0.12
Point-Robot-Wind	-5.86±0.35	-5.73±0.67	-5.69±0.43	-5.84±0.50	-6.00±0.55	–
Sparse-Point-Robot	12.26±0.44	12.45±0.33	12.38±0.57	12.36±0.37	12.34±0.35	12.19±0.43
Hopper-Rand-Params	331.68±22.90	338.21±10.12	341.50±21.70	349.30±36.51	–	–
Walker-Rand-Params	331.01±14.51	332.74±32.53	325.78±12.93	324.94±10.40	–	–
Ant-Goal	-378.81±4.43	-377.85±7.88	-383.70±5.69	-384.39±6.45	–	–
	In-distribution					
Half-Cheetanh-Vel	-142.55±12.58	-121.24±11.48	-115.07±6.75	-111.68±15.89	-127.21±22.32	-129.80±15.83
Point-Robot	-4.75±0.02	-4.81±0.04	-4.84±0.10	-4.74±0.06	-4.75±0.02	-4.75±0.06
Point-Robot-Wind	-5.59±0.16	-5.73±0.17	-5.69±0.13	-5.68±0.16	-5.65±0.14	–
Sparse-Point-Robot	12.58±0.09	12.52±0.21	12.56±0.15	12.41±0.31	12.65±0.07	12.45±0.17
Hopper-Rand-Params	377.18±11.04	366.07±19.75	358.04±15.66	351.25±22.55	–	–
Walker-Rand-Params	390.81±27.30	352.24±28.44	340.91±20.85	360.30±19.63	–	–
Ant-Goal	-319.76±3.26	-319.42±8.99	-320.16±5.16	-319.81±8.26	–	–

Table 3: Effect of the value of N.

of N may introduce confusion and lead to a decline in task-specific performance, as the increased number of clusters may cause the shared structure to exert excessive influence. Further increases in N may again isolate task-specific patterns, which can occasionally lead to partial performance recovery. Overall, our primary focus is on improving the adaptation of CTMRL to unseen target tasks. Furthermore, as shown in Table 1, CTMRL also achieves acceptable performance in meta-training tasks compared with the baselines. Analysis of distance between codes can be found in our online appendix.

Conclusion

We propose a context-based offline meta-RL framework called CTMRL, which primarily centers around our designed context quantization variational auto-encoder (CQ-VAE). CQ-VAE adopts the traditional context encoder as its encoder structure and additionally incorporates a codebook with learnable and discrete codes, and context-based reward and state predictors as its decoder structure. CQ-VAE captures the shared structure across similar tasks by clustering task-specific contexts of meta-training tasks into discrete codes, leveraging the internal relationships among tasks. This shared structure enables CTMRL to learn how to generate task-specific contexts with generalization, not for individual tasks, but for a range of similar tasks. Moreover, such a structure serves as a foundation for enhancing the distinguishability of task-specific contexts corresponding to similar yet distinct tasks through contrastive learning. Experiments across multiple meta-environments demonstrate that CTMRL outperforms existing context-based offline meta-RL baselines in terms of adaptation to unseen target tasks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62276047), Natural Science Foundation of Chongqing (No. CSTB2023NSCQ-MSX1020), China Postdoctoral Science Foundation (No. 2025M771590 and No. GZC20251104), and Yibin Science and Technology Program (No. 2024JC007).

References

Fakoor, R.; Chaudhari, P.; Soatto, S.; and Smola, A. J. 2020. Meta-Q-Learning. In *ICLR*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 1126–1135.

Foerster, J. N.; Farquhar, G.; Al-Shedivat, M.; Rocktäschel, T.; Xing, E. P.; and Whiteson, S. 2018. DiCE: The Infinitely Differentiable Monte Carlo Estimator. In *ICML*, 1524–1533.

Gao, Y.; Zhang, R.; Guo, J.; Wu, F.; Yi, Q.; Peng, S.; Lan, S.; Chen, R.; Du, Z.; Hu, X.; Guo, Q.; Li, L.; and Chen, Y. 2023. Context Shift Reduction for Offline Meta-Reinforcement Learning. In *NeurIPS*.

Garg, D.; Vaidyanath, S.; Kim, K.; Song, J.; and Ermon, S. 2022. LISA: Learning Interpretable Skill Abstractions from Language. In *NeurIPS*.

He, H.; Zhu, A.; Liang, S.; Chen, F.; and Shao, J. 2024. Decoupling Meta-Reinforcement Learning with Gaussian Task Contexts and Skills. In *AAAI*, 12358–12366.

Houthooft, R.; Chen, Y.; Isola, P.; Stadie, B. C.; Wolski, F.; Ho, J.; and Abbeel, P. 2018. Evolved Policy Gradients. In *NeurIPS*, 5405–5414.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artif. Intell.*, 101(1-2): 99–134.

Li, J.; Vuong, Q.; Liu, S.; Liu, M.; Ciosek, K.; Christensen, H. I.; and Su, H. 2020. Multi-task Batch Reinforcement Learning with Metric Learning. In *NeurIPS*.

Li, L.; Huang, Y.; Chen, M.; Luo, S.; Luo, D.; and Huang, J. 2021. Provably Improved Context-Based Offline Meta-RL with Attention and Contrastive Learning. *CoRR*, abs/2102.10774.

Li, L.; Yang, R.; and Luo, D. 2021. FOCAL: Efficient Fully-Offline Meta-Reinforcement Learning via Distance Metric Learning and Behavior Regularization. In *ICLR*.

Li, L.; Zhang, H.; Zhang, X.; Zhu, S.; Zhao, J.; and Heng, P. 2024a. Towards an Information Theoretic Framework of Context-Based Offline Meta-Reinforcement Learning. In *NeurIPS*.

Li, Z.; Lin, Z.; Chen, Y.; and Liu, Z. 2024b. Efficient Offline Meta-Reinforcement Learning via Robust Task Representations and Adaptive Policy Generation. In *IJCAI*, 4524–4532.

Mazzaglia, P.; Verbelen, T.; Dhoedt, B.; Lacoste, A.; and Rajeswar, S. 2023. Choreographer: Learning and Adapting Skills in Imagination. In *ICLR*.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A Simple Neural Attentive Meta-Learner. In *ICLR*.

Nakhaeinezhadfard, M.; Scannell, A.; and Pajarinen, J. 2025. Entropy Regularized Task Representation Learning for Offline Meta-Reinforcement Learning. In *AAAI*, 19616–19623.

Pong, V. H.; Nair, A. V.; Smith, L. M.; Huang, C.; and Levine, S. 2022. Offline Meta-Reinforcement Learning with Online Self-Supervision. In *ICML*, 17811–17829.

Rakelly, K.; Zhou, A.; Finn, C.; Levine, S.; and Quillen, D. 2019. Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables. In *ICML*, 5331–5340.

Ren, Z.; Liu, A.; Liang, Y.; Peng, J.; and Ma, J. 2022. Efficient Meta Reinforcement Learning for Preference-based Fast Adaptation. In *NeurIPS*.

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

van den Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural Discrete Representation Learning. In *NIPS*, 6306–6315.

Wang, J.; Kurth-Nelson, Z.; Soyer, H.; Leibo, J. Z.; Tirumala, D.; Munos, R.; Blundell, C.; Kumaran, D.; and Botvinick, M. M. 2017. Learning to reinforcement learn. In *CogSci*.

Wang, J.; Zhang, J.; Jiang, H.; Zhang, J.; Wang, L.; and Zhang, C. 2023. Offline Meta Reinforcement Learning with In-Distribution Online Adaptation. In *ICML*, 36626–36669.

Yin, M.; Tucker, G.; Zhou, M.; Levine, S.; and Finn, C. 2020. Meta-Learning without Memorization. In *ICLR*.

Yuan, H.; and Lu, Z. 2022. Robust Task Representations for Offline Meta-Reinforcement Learning via Contrastive Learning. In *ICML*, 25747–25759.

Zhou, R.; Gao, C.; Zhang, Z.; and Yu, Y. 2024. Generalizable Task Representation Learning for Offline Meta-Reinforcement Learning with Data Limitations. In *AAAI*, 17132–17140.

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 *ICLR*.