# Decoupling Meta-Reinforcement Learning with Gaussian Task Contexts and Skills

**Hongcai He[1], Anjie Zhu[1], Shuang Liang[1], Feiyu Chen[1,2], Jie Shao[1,2*]**

[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Sichuan Artificial Intelligence Research Institute, Yibin, China
{hehongcai,anjiezhu}@std.uestc.edu.cn, {shuangliang,chenfeiyu,shaojie}@uestc.edu.cn

## Abstract

Offline meta-reinforcement learning (meta-RL) methods, which adapt to unseen target tasks with prior experience, are essential in robot control tasks. Current methods typically utilize task contexts and skills as prior experience, where task contexts are related to the information within each task and skills represent a set of temporally extended actions for solving subtasks. However, these methods still suffer from limited performance when adapting to unseen target tasks, mainly because the learned prior experience lacks generalization, i.e., they are unable to extract effective prior experience from meta-training tasks by exploration and learning of continuous latent spaces. We propose a framework called decoupled meta-reinforcement learning (DCMRL), which (1) contrastively restricts the learning of task contexts through pulling in similar task contexts within the same task and pushing away different task contexts of different tasks, and (2) utilizes a Gaussian quantization variational autoencoder (GQ-VAE) for clustering the Gaussian distributions of the task contexts and skills respectively, and decoupling the exploration and learning processes of their spaces. These cluster centers which serve as representative and discrete distributions of task context and skill are stored in task context codebook and skill codebook, respectively. DCMRL can acquire generalizable prior experience and achieve effective adaptation to unseen target tasks during the meta-testing phase. Experiments in the navigation and robot manipulation continuous control tasks show that DCMRL is more effective than previous meta-RL methods with more generalizable prior experience.

## Introduction

Current offline meta-reinforcement learning (meta-RL) methods have been widely adopted across various domains and produced notable results, particularly regarding the robot control task (Nam et al. 2022; Rakelly et al. 2019). These meta-RL methods acquire prior experience from a series of tasks during the meta-training phase and then employ the prior experience to the unseen target tasks which have implicit relationships with the training tasks during the meta-testing phase. There are two frequently utilized forms of prior experience, task contexts and skills (Nam et al. 2022; Rakelly et al. 2019; Pertsch, Lee, and Lim 2020). Task contexts are related to the vital statistical information of tasks, which are obtained from past trajectories generated by agents. Additionally, when meeting an unseen target task, task contexts that are extracted from its trajectories will enable agents to acquire its information and achieve adaptation to the unseen target task (Nam et al. 2022; Rakelly et al. 2019). On the other hand, skills represent the means of useful behaviors that can solve subtasks. As temporal behaviors, skills can be learned from various forms of data and can be transferred to new tasks and even new environment configurations. Moreover, a series of skills for solving different subtasks can be combined to achieve solutions to complex tasks (Nam et al. 2022; Pertsch, Lee, and Lim 2020).

However, most offline meta-RL methods suffer from poor generalization issues, hindering them to achieve robust adaptation to unseen target tasks. This is due to the limited prior experience, which is caused by the coupled exploration and learning processes of continuous latent space. More specifically, exploration and learning processes are interconnected for extracting prior experience from continuous latent space. Insufficient exploration in the initial stage often leads to limited learning, which in turn results in inadequate exploration in subsequent stages. This ultimately results in both the exploration and learning processes being limited to a small portion of the entire continuous latent space of prior experience, leading to sub-optimal decisions (Campos et al. 2020).

Moreover, the existing methods (Chebotar et al. 2021; Lynch et al. 2019; Pertsch, Lee, and Lim 2020; Pong et al. 2022; Nam et al. 2022) suffer from another limitation as they just model task contexts and/or skills as continuous latent spaces without considering their inherent characteristics. A task corresponds to a series of similar task contexts, since agents usually generate diverse trajectories resulting from variations in their execution processes and levels of success. For example, in the maze navigation tasks, agents can start from a fixed point and take different paths to the specified endpoint. However, failing to consider the relationships between task contexts of the same and different tasks will result in unclear and ambiguous learning. In addition, there are a series of similar skills in the continuous latent space due to the similarity between subtasks. For example, in the kitchen manipulation tasks, opening the door of the microwave and

*Corresponding author: Jie Shao.

opening the hinge cabinet are similar. Nevertheless, without consideration for relationships between skills, it will be hard to select the most accurate skill.

To this end, we propose a framework called decoupled meta-reinforcement learning (DCMRL) using both task contexts and skills as prior experience so as to acquire task contexts and skills with generalization, additionally achieving effective adaptation to unseen target tasks. Specifically, we model the distributions of task context and skill as Gaussian distributions, instead of just representing task contexts and skills as simple vectors, since Gaussian distributions can capture the uncertainty in their respective spaces and provide more robust representations. Firstly, DCMRL utilizes our proposed Gaussian quantization variational autoencoder (GQ-VAE) to perform online clustering on the Gaussian distributions of task contexts and skills in their respective continuous latent spaces, generating a set of discrete cluster centers. These cluster centers are stored in the form of learnable codebooks, as the representative distributions of task contexts and skills with generalization. Additionally, exploration of the continuous latent spaces and learning of discrete cluster centers inside the codebooks achieve decoupling of exploration and learning processes, therefore we apply this decoupling operation to both task contexts and skills via GQ-VAEs. Moreover, task contexts are vulnerable to the distribution mismatch of meta-training tasks and unseen target tasks during meta-testing. To solve this issue, DCMRL contrastively restricts task contexts through the dissimilarity of task contexts for different tasks and the similarity of different task contexts for the same task, leading to task contexts with generalization. In essence, DCMRL enhances the generalizability of the task contexts and skills acquired as prior experience during the meta-training phase, thereby enabling more effective adaptation to unseen target tasks during the meta-testing phase.

The main contributions of our method are threefold:

- We propose DCMRL, a novel framework that enhances the generalizability of task contexts and skills by contrastively restricting task contexts and decoupling the exploration and learning of their respective spaces, leading to more effective adaptation to unseen target tasks.
- We propose a novel GQ-VAE that clusters on Gaussian distributions of task context and skill distributions in their corresponding continuous latent spaces and decouples the exploration and learning of their respective spaces, enhancing their generalizability.
- We evaluate DCMRL in two challenging continuous robot control environments, i.e., maze navigation and kitchen manipulation, which are long-horizon and sparse-reward. The results show that DCMRL outperforms previous meta-RL methods, achieving more effective adaptation to unseen target tasks.

## Related Work

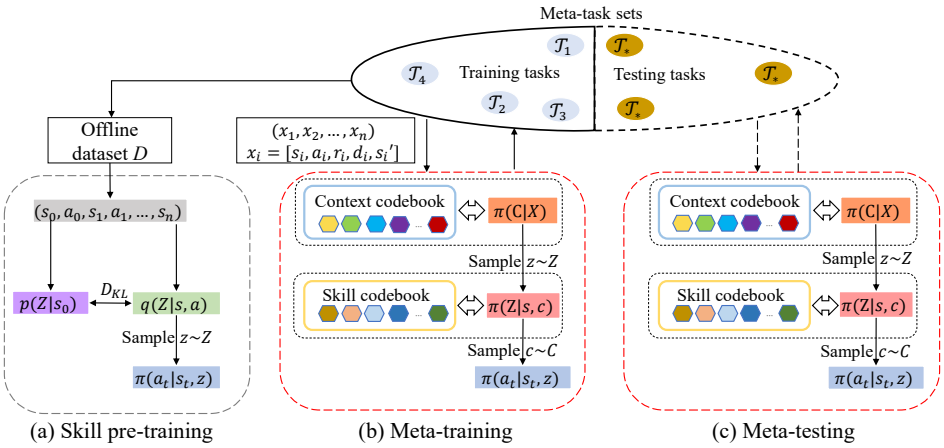
**Offline Meta-reinforcement Learning.** The primary goal of offline meta-reinforcement learning (meta-RL) is the acquisition of learning strategies from offline datasets, allowing more effective learning in new tasks through appropriate prior experience. Due to a distributional shift between offline and online data during testing, it is critical to obtain robust task representations that generalize well while learning. Most previous methods (Dorfman, Shenfeld, and Tamar 2021; Mitchell et al. 2021; Pong et al. 2022; Siegel et al. 2020) meta-learn from offline datasets, including reward and task annotations, and adapt to a new task with only a small amount of new data. However, some meta-training tasks are hard to annotate due to the lack of corresponding prior task experience. Moreover, Pong et al. (2022) utilize semi-supervised learning with both offline and online data for distributional shift, but heavily relying on annotation functions from offline data. In contrast, our proposed DCMRL leverages a large offline dataset across many tasks without rewards or task annotations for extracting skills.

**Context-based Meta-RL.** Context-based methods train a module to take prior experience as a form of task-specific context. Some methods (Duan et al. 2016; Finn, Abbeel, and Levine 2017; Humplik et al. 2019; Liu et al. 2021; Rothfuss et al. 2019; Wang et al. 2017; Yu et al. 2018; Yang et al. 2019; Zintgraf et al. 2019) have been proposed for meta-learning dynamic models and policies that can quickly adapt to unseen target tasks. In contrast, other recursive methods (Fakoor et al. 2020; Lee et al. 2020; Mishra et al. 2018; Rakelly et al. 2019; Seo et al. 2020) make fast adaptation by aggregating experience into a latent representation on which the policy is conditioned. Additionally, some methods train recurrent Q-function with off-policy Q-learning approaches which are often used on simple tasks (Heess et al. 2015) or in discrete environments (Hausknecht and Stone 2015). As a context-based method, DCMRL decouples the exploration and learning processes of task contexts during meta-training for improving the generalization of task contexts, and achieves effective adaptation to unseen target tasks during meta-testing.

**Skill-based Meta-RL.** Another method for exploiting offline data without requiring reward or task annotations is extracting skills as the identification of reusable, short-horizon behaviors. Skill-based learning methods learn unseen long-horizon target tasks by transferring these learned skills and converge significantly faster than learning from scratch (Hausman et al. 2018; Lee et al. 2019). Previous works (Ajay et al. 2021; Chebotar et al. 2021; Lynch et al. 2019; Merel et al. 2020; Pertsch, Lee, and Lim 2020; Pertsch et al. 2021) have shown that skill-based learning methods can learn a broad range of skills with diverse datasets and accomplish long-horizon tasks. However, these methods still require a substantial number of interactions with environment to learn enough skills or new skills. SiMPL (Nam et al. 2022) learns skills by combining meta-learning process and offline dataset but still suffers from limited generalization. As a skill-based method, DCMRL further applies the decoupling operation to the exploration and learning processes of skills, generating more generalized and representative skills.

## Method

Decoupled meta-reinforcement learning (DCMRL) consists of three phases: skill pre-training, meta-training and meta-testing. We mainly focus on the meta-training phase and




(a) Skill pre-training (b) Meta-training (c) Meta-testing

Figure 1: Method overview. DCMRL as a hierarchical framework, has three phases. (a) Skill pre-training learns a prior skill $p(Z|s)$ (purple) as a constraint and a low-level skill-based policy $\pi(a_t|s_t, z)$ (blue) trained with $q(Z|s, a)$ (green) through the offline dataset $D$. (b) Meta-training meta-trains a high-level skill policy $\pi(Z|s, c)$ (red) for skill distribution $Z$ and a task context policy $\pi(C|X)$ (orange) for task context distributions $C$ through meta-training tasks $T = \{\mathcal{T}_1, ..., \mathcal{T}_{n_{task}}\}$, while the pre-trained low-level skill-based policy $\pi(a_t|s_t, z)$ remains fixed. (c) Meta-testing utilizes the meta-trained modules for effective adaptation to an unseen target task $\mathcal{T}^* \in T^*$ with task contexts generated from a few transitions of it. Additionally, $\pi(C|X)$ and $\pi(a_t|s_t, z)$ are fixed and $\pi(Z|s, c)$ remains under fine-tuning.

aim to acquire more generalizable task contexts and skills through: (1) contrastively restricting task contexts by the relationships between tasks for enhancing their generalization and (2) applying our proposed GQ-VAEs to cluster the distributions of task contexts and skills respectively, and decouple the exploration and learning processes of their respective spaces. An illustration of DCMRL is given in Figure 1.

## Problem Formulation

An offline meta-RL task is typically formalized as a fully observable Markov decision process (MDP), defined as a tuple $< \mathcal{S}, \mathcal{A}, p, r, \gamma, \rho_0 >$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p(s'|s, a)$ is the transition dynamics, $r(s, a)$ is the reward function, $\rho_0$ is the initial state distribution, and $\gamma \in [0, 1)$ is the factor discounting the future reward. The policy is a distribution $\pi(a|s)$ over actions. In a complete MDP, the agent is initialized in a given state and selects an action at each time step by sampling from a fixed policy $\pi$. Meanwhile, the environment responds by updating the state using transition probabilities $p$ and providing a reward $r$ and a boolean flag of done $d$. Additionally, the marginal state distribution at time step $t$ is defined as $\mu_\pi^t(s)$ and the objective of the agent is to maximize the expected cumulative rewards $max_\pi \mathcal{J}_\mathcal{M}(\pi) = \mathbb{E}_{s_t \sim \mu_\pi^t, a_t \sim \pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. More details of problem formulation and preliminaries can be found in Appendix of our full version (He et al. 2023).

As an offline meta-RL method, DCMRL assumes access to a *task-agnostic* dataset of state-action trajectories $D = (s_0, a_0, \ldots, s_t, a_t)$, collected from various tasks or unlabeled data. With a wide variety of behaviors, $D$ is able to accelerate learning of different tasks (Nam et al. 2022). We also assume access to a set of $n_{task}$ meta-training tasks $T = \{\mathcal{T}_1, \ldots, \mathcal{T}_{n_{task}}\}$, simultaneously with a set of unseen target tasks $T^*$, and represent each task as an MDP respectively. Notably, we do not assume that there are direct relationships between either $T^*$ and $D$ or $T^*$ and $T$. Specifically, the offline dataset $D$ does not contain any demonstrations for solving tasks in $T^*$, and there are no intersections between $T^*$ and $T$.

DCMRL aims to extract skills from the offline dataset $D$, perform the meta-training phase on the set of meta-training tasks $T$ and handle the set of unseen target tasks $T^*$ in the meta-testing phase. Moreover, we represent the distributions of task context and skill as Gaussian distributions, denoted by $C$ and $Z$, while $c$ and $z$ are the task context embeddings and skill embeddings sampled from them.

## Skill Pre-training

During the skill pre-training phase, DCMRL comprises a skill prior $p(Z|s_0)$, a skill encoder $q(Z|s, a)$ and a skill-based policy $\pi(a_t|s_t, z)$. Specifically, our primary focus lies on the skill prior $p(Z|s_0)$ and skill-based policy $\pi(a_t|s_t, z)$. Both the skill prior $p(Z|s_0)$ and skill encoder $q(Z|s, a)$ are implemented as deep neural networks that output skill distributions $Z$ in the form of Gaussian distributions. For a K-step trajectory randomly drawn from the sequences in offline dataset $D$, the skill prior $p(Z|s_0)$ predicts a skill distribution $Z$ based on the initial state $s_0$ of the trajectory, while the skill encoder $q(Z|s, a)$ aligns the sequence of full state-action pairs to a skill distribution $Z$. Moreover, the skill prior $p(Z|s_0)$ is trained by matching to the skill distribution encoded by the skill encoder $q(Z|s, a)$ as follows:

$$\min_p \mathcal{D}_{KL}(sg[q(Z|s, a)], p(Z|s_0)), \tag{1}$$

where $sg[\cdot]$ denotes the stop-gradient operation and $\mathcal{D}_{KL}$ denotes the Kullback-Leibler divergence. Furthermore, skill-based policy $\pi(a_t|s_t, z)$ is fine-tuned through behavioral

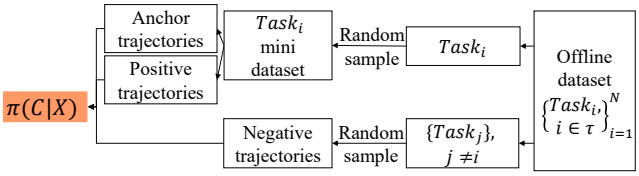



Figure 2: Contrastive task context learning. DCMRL samples a long trajectory data as the mini dataset and generates anchor trajectories and positive trajectories from it, while negative trajectories are from other tasks.

cloning to replicate the action sequence $a_{0:K-1}$ corresponding to the given skill embedding $z$ which is sample from the skill distribution $Z$ output by skill encoder $q(Z|s,a)$. Moreover, we leverage a unit Gaussian prior distribution of skill inspired by Higgins et al. (2017) for regularization:

$$\max_{q,\pi} \mathbb{E}_{z\sim q}[\prod_{t=0}^{K-1} \log \pi(a_t|s_t,z) - \alpha \mathcal{D}_{KL}(q(Z|s,a),\mathcal{N}(0,\mathcal{I}))], \tag{2}$$

where $\alpha$ is a weight coefficient of the constraint. In summary, we make a skill pre-training phase for skill prior $p(Z|s_0)$, which is utilized for imposing constraints on high-level skill policy $\pi(Z|s,c)$, and skill-based policy $\pi(a|s,z)$, which serves as the low-level policy and remains fixed during meta-training and meta-testing.

## Meta-training

### Contrastive Task Context Learning

Task context as a kind of prior experience, aims to specify which task from the distribution the agent should focus on, and provides information that helps the agent adapt to its strategy when a new task is encountered. We contrastively enhance task context representations by distinguishing unique task features, thereby promoting effective and human-like adaptability in diverse tasks.

We employ a specific sampling strategy on trajectories from meta-training tasks. Traditional methods such as Yuan and Lu (2022) sample positive samples from the same task, but this can result in chaos and significant disparity from the anchor trajectory due to variations in tuples and their orders. Hence, DCMRL samples anchor and positive samples in two stages: first, a longer trajectory is sampled as a mini dataset from current task; then, two different trajectories of the same and fixed length are sampled from this long trajectory, maintaining tuples' relative orders, to serve as the anchor and positive samples respectively. Negative samples are randomly sourced from other tasks, as traditional methods (see Figure 2).

We utilize the classic contrastive learning loss function, triplet loss (Schroff, Kalenichenko, and Philbin 2015), for anchor, positive, and negative samples. These trajectories are processed with a high-level task context policy $\pi(C|X)$ to generate distributions of task contexts, $\tilde{C}$, $\tilde{C}^+$, and $\tilde{C}^-$. The triplet loss aims to minimize the similarity between $\tilde{C}$ and $\tilde{C}^-$ and maximize the similarity between $\tilde{C}$ and $\tilde{C}^+$ as follows:

$$\mathcal{L}_{triplet}(\tilde{C},\tilde{C}^+,\tilde{C}^-) = \sum_{\tilde{C}\in\mathcal{C}} max(0, sim(\tilde{C},\tilde{C}^-) - sim(\tilde{C},\tilde{C}^+) + \epsilon), \tag{3}$$

where $sim(\cdot)$ is the similarity function, and we utilize the cosine similarity here. $\mathcal{C}$ is the continuous latent space of task contexts and the margin parameter $\epsilon$ is configured as the minimum offset between distances of similar and dissimilar pairs.

### GQ-VAE

We propose Gaussian quantization variational autoencoder (GQ-VAE), which consists of three main parts: an encoder, a learnable codebook and a decoder, where the encoder and decoder are deep neural networks. The codebook $\mathcal{CB} = \{O^1,\ldots,O^K\}$ contains $K$ cluster centers as discrete codes, where $K$ is a hyperparameter.

Specifically, the encoder maps the input trajectory $X$ to a Gaussian distribution $\tilde{O}$. Codes within $\mathcal{CB}$ are modeled as Gaussian distributions, as the targets for clustering originate from the Gaussian distributions encoded by the encoder. Once a Gaussian distribution $\tilde{O}$ is outputted by the encoder, it will be matched to its closest code $O^k$ within $\mathcal{CB}$ through a match operation $\mathbf{m}(\cdot)$. Moreover, Euclidean distance is utilized to measure the distance between current $\tilde{O}$ and each $O^i$, where $1 \leq i \leq K$ in the codebook $\mathcal{CB}$. The complete process of $\mathbf{m}(\cdot)$ is as follows:

$$O = \mathbf{m}(\tilde{O}) := \underset{O^k\in\mathcal{CB}, 1\leq k\leq K}{argmin} \|\tilde{O} - O^k\|_2. \tag{4}$$

Finally, the decoder will take the code $O^k$ that matches $\tilde{O}$ as input to output a reconstructed trajectory $\tilde{X}$.

In essence, exploration of the latent space and learning of codebook $\mathcal{CB}$ in DCMRL interacts to yield complementary and reinforcing effects. Directly, exploring the continuous latent space forms its composition in Gaussian distributions, while learning in codebook $\mathcal{CB}$ achieves discretization of the continuous latent space by deriving $K$ codes as cluster centers. Moreover, the match operation $\mathbf{m}(\cdot)$ integrates the two stages by matching the initialized codes in the codebook $\mathcal{CB}$ with different $\tilde{O}$. The codes learn from different $\tilde{O}$ they match and optimize their positions in the continuous latent space. Overall, this online clustering procedure resembles a classical $K$-means algorithm. The loss function used to update GQ-VAE is as follows:

$$\mathcal{L}_{GQ} = \|sg[\mathbf{m}(\tilde{O})] - \tilde{O}\|_2 + \mu\|\mathbf{m}(\tilde{O}) - sg[\tilde{O}]\|_2 + \|\tilde{X} - X\|_2, \tag{5}$$

where $\mu$ is a weight coefficient.

The loss function comprises three terms. The first two terms differ solely in the objects of the $sg[\cdot]$ operation, updating the encoder and matched code in $\mathcal{CB}$, respectively, and $\mu$ balances them. The third term is derived by inputting the matched code after the $sg[\cdot]$ operation to the decoder, which optimizes the decoder through quantifying the disparity between reconstructed and input trajectories.

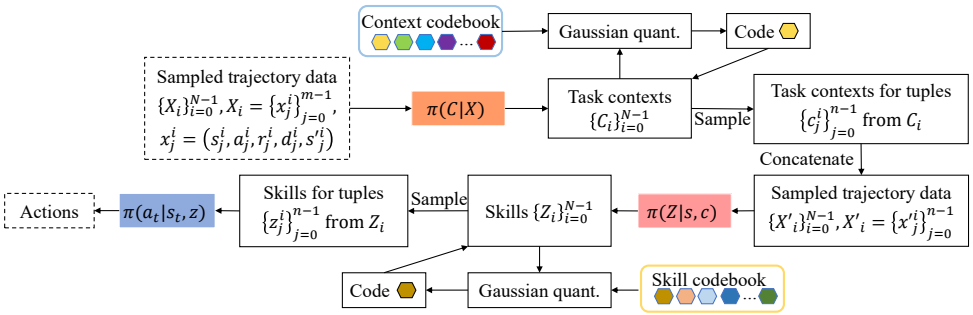



Figure 3: Task context and skill learning architecture. The task context distribution $C$ and skill distribution $Z$ generated by task policy $\pi(C|X)$ and skill policy $\pi(Z|s,c)$ respectively, are both implemented in the form of Gaussian distributions. The task context embeddings $c$ and skill embeddings $z$ are sampled from these distributions. In addition, the learning processes of $\pi(C|X)$ and $\pi(Z|s,c)$ are accompanied by corresponding learnable codebooks that correspond to their respective spaces.

In summary, GQ-VAE decouples the exploration over the continuous latent space and the learning of the codebook $\mathcal{CB}$. Its bidirectional updating mechanism achieves better learning of each code within $\mathcal{CB}$ through exploration, facilitating discretization. Meanwhile, exploration guided by learned codes attains deeper characterization. DCMRL simultaneously utilizes task contexts and skills as prior experience, and applies GQ-VAEs on both task contexts and skills as shown in Figure 3. Next, we introduce the exploration of the two continuous latent spaces of task contexts and skills, as well as the learning of corresponding task context codebook and skill codebook.

**Task Context Learning.** GQ-VAE for the task context learning stage includes three modules: a high-level task context policy $\pi(C|X)$ as task context encoder, a learnable codebook of task context with $K_\mathcal{C}$ quantized task context codes $\mathcal{CB}_\mathcal{C} = \{C^1, ..., C^{K_\mathcal{C}}\}$ and a corresponding decoder.

Given a batch of sampled trajectory data $\mathcal{X} = \{X_1, \ldots, X_N\}$, where $N$ is the task batch size that means the sampled number of tasks and $X_i = \{(s_j^i, a_j^i, r_j^i, d_j^i, s_j'^i)\}_{j=0}^{n_c}$ is input task trajectory data whose length is $n_c$, the task context encoder $\pi(C|X)$ outputs a task context distribution $\tilde{C}$. The complete process of matching is as follows:

$$C = \mathbf{m}(\tilde{C}) := \underset{C^k \in \mathcal{CB}_\mathcal{C}, 1 \le k \le K_\mathcal{C}}{argmin} \|\tilde{C} - C^k\|_2. \tag{6}$$

We utilize an objective $\mathcal{L}_{Context} = \mathcal{L}_{BC} + \lambda \mathcal{L}_{GQ_{Context}}$ for updating, where $\mathcal{L}_{BC}$ is the behavior-cloning loss generated by the skill-based policy $\pi(a_t|s_t, z)$, $\lambda$ is a weight coefficient of loss and the formulation of $\mathcal{L}_{GQ_{Context}}$ is as follows:

$$\mathcal{L}_{GQ_{Context}} = \|sg[\mathbf{m}(\tilde{C})] - \tilde{C}\|_2 + \eta \|\mathbf{m}(\tilde{C}) - sg[\tilde{C}]\|_2 + \|\tilde{X} - X\|_2, \tag{7}$$

where $\eta$ is a weight coefficient.

**Skill Learning.** GQ-VAE for the skill context learning stage includes a high-level skill policy $\pi(Z|s,c)$ as skill encoder, a learnable codebook of skills with $K_\mathcal{Z}$ quantized skill codes $\mathcal{CB}_\mathcal{Z} = \{Z^1, \ldots, Z^{K_\mathcal{Z}}\}$ and a decoder.

Given an additional batch of sampled trajectories $\mathcal{X}' = \{X'_1, \ldots, X'_N\}$ from the same tasks, it is different from that used in the task context stage and the length of sampled trajectories $n_z$ is often different from $n_c$. The skill policy $\pi(Z|s,c)$ inputs a sampled trajectory $X'_i$ and task context $c$ sampled from $C$ to output a skill distribution $\tilde{Z}$. Furthermore, the complete process of matching is as follows:

$$Z = \mathbf{m}(\tilde{Z}) := \underset{Z^k \in \mathcal{CB}_\mathcal{Z}, 1 \le k \le K_\mathcal{Z}}{argmin} \|\tilde{Z} - Z^k\|_2. \tag{8}$$

The objective is $\mathcal{L}_{Skill} = \mathcal{L}_{BC} + \gamma \mathcal{L}_{GQ_{Skill}}$, where $\gamma$ is a weight coefficient and $\mathcal{L}_{GQ_{Skill}}$ is the skill quantization loss:

$$\mathcal{L}_{GQ_{Skill}} = \|sg[\mathbf{m}(\tilde{Z})] - \tilde{Z}\|_2 + \iota \|\mathbf{m}(\tilde{Z}) - sg[\tilde{Z}]\|_2 + \|\tilde{X}' - X'\|_2, \tag{9}$$

where $\iota$ is also a weight coefficient.

Besides applying GQ-VAE, the skill policy updating also leverages the skill prior $p(Z|s_0)$ from the skill pre-training phase, as Nam et al. (2022), with the objective as follows:

$$\max_\pi \mathbb{E}_{c \sim \pi(\cdot|X)} [\sum_t \mathbb{E}_{(s_t, z) \sim \rho_{\pi|c}} [r_\mathcal{T}(s_t, z) - \beta \mathcal{D}_{KL}(\pi(Z|s,c), p(Z|s_0))]], \tag{10}$$

where $\beta$ is a weight coefficient of the constraint.

In the subsequent process, skill-based policy $\pi(a_t|s_t, z)$ uses specific skill $z$, which is sampled from $Z$ and current state, to general corresponding action.

## Meta-testing

DCMRL has trained the high-level task context policy $\pi(C|X)$ and skill policy $\pi(Z|s,c)$ on the set of meta-training tasks during meta-training phase. When facing unseen target tasks in the meta-testing phase, we first collect a few trajectories $X^*$ and extract the task context distribution $C^*$ from them through the task context policy $\pi(C|X)$. Then, we utilize the skill policy $\pi(Z|s,c)$ under $c^*$ sampled from $C^*$ to generate $Z^*$ and sample $z^*$ from it. In order to refine DCMRL on the unseen target tasks, we continue to optimize the skill policy through Eq. (9) and Eq. (10).

# Experiments

Our experiments are mainly based on long-horizon and sparse-reward tasks and evaluate DCMRL on two key issues: (1) whether better prior experience can be learned and

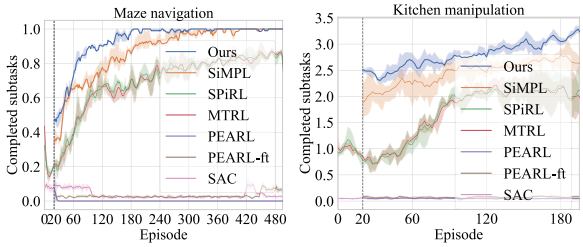



Figure 4: Comparisons of sample efficiency. We evaluate DCMRL, SiMPL, SPiRL and MTRL in maze navigation and kitchen manipulation. In both environments, we individually train each model for every target task using five distinct random seeds. Prior to fine-tuning on the target tasks, our method, SiMPL, PEARL and PEARL-ft employ 20 episodes of environment interactions for conditioning the meta-trained policies.

(2) whether the effective adaptation of unseen target tasks can be achieved. Our code is available at https://github.com/hehongc/DCMRL/.

## Experimental Setup

We compare DCMRL with SiMPL (Nam et al. 2022), SAC (Haarnoja et al. 2018), SPiRL (Pertsch, Lee, and Lim 2020), PEARL (Rakelly et al. 2019), PEARL-ft (Nam et al. 2022) and MTRL (Teh et al. 2017). We evaluate in two complex continuous control environments: maze navigation and kitchen manipulation. Maze navigation is a 2D environment, in which the agent typically requires hundreds of time steps to complete a task, and only sparse rewards are provided upon success. Kitchen manipulation involves a 7-DoF robotic arm for executing a task consisting of four subtasks, in which the agent generally takes 300-500 time steps to complete a task, and only sparse rewards are provided after complete subtasks in order. More details about the experimental environments and baselines are in He et al. (2023).

## Comparison with State-of-the-art Methods

We report both quantitative performance and qualitative adaptation experimental results, presented in Figures 4 and 5 respectively. Specifically, Figure 4 shows key insights on adaptation to unseen target tasks and is used to evaluate the performance of DCMRL in a quantitative manner. Meanwhile, Figure 5 offers additional inspection and verification of DCMRL for an intricate maze navigation domain in terms of qualitative analysis. DCMRL exhibits superior performance and sample efficiency compared with all baselines in Figure 4 for adapting to unseen target tasks. Additionally, ablation experiments can be found in He et al. (2023).

We delve into the impact of leveraging prior experience in reinforcement learning methods to adapt to unseen target tasks. Without leveraging prior experience, SAC exhibits constrained adaptation. While PEARL and PEARL-ft learn task contexts as prior experience from meta-training tasks, they struggle to effectively adapt to unseen target tasks even with fine-tuning. In addition, SPiRL leverages a series of

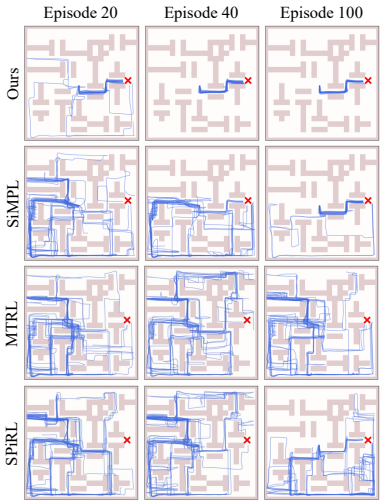



Figure 5: Visualization of the adaptation process. DCMRL, SiMPL, SPiRL and MTRL can make effective adaptation of the maze navigation environment with prior experience. We present their adaptation situations in episodes 20, 40 and 100.

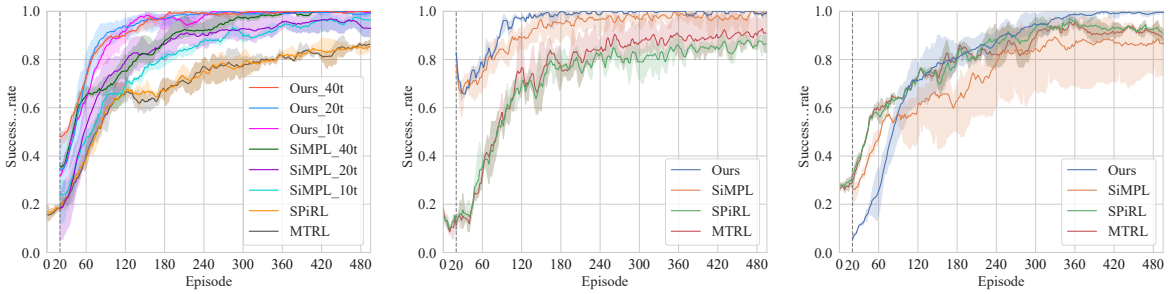
continuous skills extracted from offline datasets as prior experience, which provide limited assistance for adaptation. Moreover, MTRL trains a multi-task agent from meta-training tasks, exhibiting adaptation performance comparable to SPiRL. Furthermore, SiMPL utilizes both skills and task contexts, which are extracted respectively from offline datasets and meta-training tasks, as prior experience. However, its adaptation remains limited due to the constrained generalizability of prior experience.

Generally, DCMRL exhibits significantly quicker and better unseen target task adaptation than other methods. In just a few episodes, it achieves policy convergence to solve nearly 90% of the unseen target tasks in the maze environment and nearly three out of four subtasks in the kitchen manipulation environment. Subsequently, the further learning of skills for adaptation is able to achieve better performance. Finally, DCMRL can solve nearly 100% of the unseen target tasks in the maze environment and over three out of four subtasks in the kitchen manipulation environment.

The visual representations depicted in Figure 5 demonstrate that the application of offline datasets by DCMRL, SiMPL, SPiRL, and MTRL lead to effective adaptation of the maze environment in the early episodes. DCMRL outperforms SiMPL, SPiRL and MTRL in terms of convergence speed, achieving higher sample efficiency.

## Meta-training & Target Task Distribution Analysis

In this section, we explore how the meta-training task distribution impacts the adaptation of unseen target tasks. Our evaluation focuses on two specific factors: (1) the quantity of tasks in the meta-training task distribution, and (2) the alignment of the meta-training task distribution with the target task distribution. Our experiments are conducted within the context of the maze navigation task.




(a) Sparser task distribution (b) $\mathcal{T}_{Train-top} \rightarrow \mathcal{T}_{Target-top}$ (c) $\mathcal{T}_{Train-top} \rightarrow \mathcal{T}_{Target-bottom}$

Figure 6: Meta-training and target task distribution analysis. (a) DCMRL and SiMPL are trained on both the original meta-training task number setup (40) and lower setup (i.e., 10, 20), denoted as 10t, 20t and 40t, while SPiRL and MTRL are only trained on the original setup. All models are evaluated with the same set of unseen target tasks. (b) DCMRL, SiMPL, SPiRL and MTRL are trained on a meta-training task distribution ($\mathcal{T}_{Train-top}$) that exhibits higher alignment with the target task distribution ($\mathcal{T}_{Target-top}$). (c) DCMRL, SiMPL, SPiRL and MTRL are trained on a meta-training task distribution ($\mathcal{T}_{Train-top}$) which is misaligned with the target task distribution ($\mathcal{T}_{Target-bottom}$). To comprehensively evaluate the efficacy of our approach, we train each model on each target task using five distinct random seeds.

**The Quantity of Meta-training Tasks.** We aim to gauge the extent to which varying the number of meta-training tasks influences adaptation. Following Nam et al. (2022), we train DCMRL with a lower quantity of meta-training tasks (i.e., 10 and 20) in addition to the original number setup (40), and evaluate these models with the same set of unseen target tasks. The quantitative results presented in Figure 6(a) indicate that even with fewer numbers of meta-training tasks, DCMRL exhibits similar performance and exceeds the performance of best baseline in all settings (i.e., SiMPL).

**Alignment of Meta-training and Target Tasks.** The focus of this investigation is to determine the extent to which a model's performance would improve or deteriorate when trained on a meta-training task distribution that aligns differently with the target tasks. To achieve this objective, we implement task distributions that possess varied degrees of bias towards either meta-training or target tasks. Specifically, the meta-training set is generated by exclusively drawing goal locations from the top 25% of the maze ($\mathcal{T}_{Train-top}$), which means that there are 10 meta-training tasks (i.e., $40 \times 25\%$). Subsequently, to obtain the relevant results, we formulate two target task distributions, one characterized by excellent alignment and the other by weak alignment with the meta-training distribution, since they are sampled respectively from the top 25% portion of the maze ($\mathcal{T}_{Target-top}$) and the bottom 25% portion of the maze ($\mathcal{T}_{Target-bottom}$). Moreover, to alleviate spurious biases resulting from uneven density in the task distribution, we employ density-balanced sampling tactics throughout the experimental procedure.

Figure 6(b) and Figure 6(c) respectively depict the target task adaptation process with models trained under good task alignment conditions (meta-train on $\mathcal{T}_{Train-top}$ and meta-test on $\mathcal{T}_{Target-top}$) and bad task alignment conditions (meta-train on $\mathcal{T}_{Train-top}$ and meta-test on $\mathcal{T}_{Target-bottom}$). The results reveal that DCMRL indeed can achieve superior performance with good task alignment conditions (see Figure 6(b)). In addition, unlike SiMPL our model trained under a misaligned meta-training task distribution, though exhibiting initially lower performance, eventually achieves similarly superior performance (see Figure 6(c)). To summarize, DCMRL exhibits strong generalization, achieving high performance with minimal meta-training tasks and demonstrating robustness to variations in the quality of task alignment.

## Conclusion

We propose DCMRL, an offline meta-RL framework, which can acquire more generalizable prior experience to achieve effective adaptation to unseen target tasks. Specially, we utilize both task contexts and skills as prior experience and use Gaussian distributions for their representations. We extract the skills from offline datasets, and perform exploration and learning of the continuous latent spaces of task contexts and skills with meta-training tasks. In addition, we propose GQ-VAE, which clusters the Gaussian distributions of task contexts and skills in their respective continuous latent spaces and decouples the exploration and learning processes of task contexts and skills, enhancing their generalization. These cluster centers which serve as representative and discrete distributions of task context and skill are respectively stored in task context codebook and skill codebook. Moreover, we sample positive samples, negative samples and anchor samples through a specific sampling strategy, and contrastively restrict the task contexts, leading to more appropriate representations of task contexts. Experiments on challenging continuous control navigation and manipulation tasks that are long-horizon and sparse-reward demonstrate that DCMRL outperforms the prior methods in meta-RL.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (grants No. 62276047 and No. 62302080).

## References


Ajay, A.; Kumar, A.; Agrawal, P.; Levine, S.; and Nachum, O. 2021. OPAL: Offline Primitive Discovery for Accelerating Offline Reinforcement Learning. In *ICLR*.

Campos, V.; Trott, A.; Xiong, C.; Socher, R.; Giró-i-Nieto, X.; and Torres, J. 2020. Explore, Discover and Learn: Unsupervised Discovery of State-Covering Skills. In *ICML*, 1317–1327.

Chebotar, Y.; Hausman, K.; Lu, Y.; Xiao, T.; Kalashnikov, D.; Varley, J.; Irpan, A.; Eysenbach, B.; Julian, R.; Finn, C.; and Levine, S. 2021. Actionable Models: Unsupervised Offline Reinforcement Learning of Robotic Skills. In *ICML*, 1518–1528.

Dorfman, R.; Shenfeld, I.; and Tamar, A. 2021. Offline Meta Reinforcement Learning - Identifiability Challenges and Effective Data Collection Strategies. In *NeurIPS*, 4607–4618.

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. RL$ˆ2$: Fast Reinforcement Learning via Slow Reinforcement Learning. *CoRR*, abs/1611.02779.

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.

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 *ICML*, 1856–1865.

Hausknecht, M. J.; and Stone, P. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. In *AAAI*, 29–37.

Hausman, K.; Springenberg, J. T.; Wang, Z.; Heess, N.; and Riedmiller, M. A. 2018. Learning an Embedding Space for Transferable Robot Skills. In *ICLR*.

He, H.; Zhu, A.; Liang, S.; Chen, F.; and Shao, J. 2023. Decoupling Meta-Reinforcement Learning with Gaussian Task Contexts and Skills. *CoRR*, abs/2312.06518.

Heess, N.; Hunt, J. J.; Lillicrap, T. P.; and Silver, D. 2015. Memory-based control with recurrent neural networks. *CoRR*, abs/1512.04455.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C. P.; Glorot, X.; Botvinick, M. M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.

Humplik, J.; Galashov, A.; Hasenclever, L.; Ortega, P. A.; Teh, Y. W.; and Heess, N. 2019. Meta reinforcement learning as task inference. *CoRR*, abs/1905.06424.

Lee, K.; Seo, Y.; Lee, S.; Lee, H.; and Shin, J. 2020. Context-aware Dynamics Model for Generalization in Model-Based Reinforcement Learning. In *ICML*, 5757–5766.

Lee, Y.; Sun, S.; Somasundaram, S.; Hu, E. S.; and Lim, J. J. 2019. Composing Complex Skills by Learning Transition Policies. In *ICLR*.

Liu, E. Z.; Raghunathan, A.; Liang, P.; and Finn, C. 2021. Decoupling Exploration and Exploitation for Meta-Reinforcement Learning without Sacrifices. In *ICML*, 6925–6935.

Lynch, C.; Khansari, M.; Xiao, T.; Kumar, V.; Tompson, J.; Levine, S.; and Sermanet, P. 2019. Learning Latent Plans from Play. In *CoRL*, 1113–1132.

Merel, J.; Tunyasuvunakool, S.; Ahuja, A.; Tassa, Y.; Hasenclever, L.; Pham, V.; Erez, T.; Wayne, G.; and Heess, N. 2020. Catch & Carry: reusable neural controllers for vision-guided whole-body tasks. 39.

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

Mitchell, E.; Rafailov, R.; Peng, X. B.; Levine, S.; and Finn, C. 2021. Offline Meta-Reinforcement Learning with Advantage Weighting. In *ICML*, 7780–7791.

Nam, T.; Sun, S.; Pertsch, K.; Hwang, S. J.; and Lim, J. J. 2022. Skill-based Meta-Reinforcement Learning. In *ICLR*.

Pertsch, K.; Lee, Y.; and Lim, J. J. 2020. Accelerating Reinforcement Learning with Learned Skill Priors. In *CoRL*, 188–204.

Pertsch, K.; Lee, Y.; Wu, Y.; and Lim, J. J. 2021. Demonstration-Guided Reinforcement Learning with Learned Skills. In *CoRL*, 729–739.

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.

Rothfuss, J.; Lee, D.; Clavera, I.; Asfour, T.; and Abbeel, P. 2019. ProMP: Proximal Meta-Policy Search. In *ICLR*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 815–823.

Seo, Y.; Lee, K.; Gilaberte, I. C.; Kurutach, T.; Shin, J.; and Abbeel, P. 2020. Trajectory-wise Multiple Choice Learning for Dynamics Generalization in Reinforcement Learning. In *NeurIPS*.

Siegel, N. Y.; Springenberg, J. T.; Berkenkamp, F.; Abdolmaleki, A.; Neunert, M.; Lampe, T.; Hafner, R.; Heess, N.; and Riedmiller, M. A. 2020. Keep Doing What Worked: Behavior Modelling Priors for Offline Reinforcement Learning. In *ICLR*.

Teh, Y. W.; Bapst, V.; Czarnecki, W. M.; Quan, J.; Kirkpatrick, J.; Hadsell, R.; Heess, N.; and Pascanu, R. 2017. Distral: Robust multitask reinforcement learning. In *NeurIPS*, 4496–4506.

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*.

Yang, Y.; Caluwaerts, K.; Iscen, A.; Tan, J.; and Finn, C. 2019. NoRML: No-Reward Meta Learning. In *AAMAS*, 323–331.

Yu, T.; Finn, C.; Xie, A.; Dasari, S.; Zhang, T.; Abbeel, P.; and Levine, S. 2018. One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning. In *ICLR (Workshop)*.

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

Zintgraf, L. M.; Shiarlis, K.; Kurin, V.; Hofmann, K.; and Whiteson, S. 2019. Fast Context Adaptation via Meta-Learning. In *ICML*, 7693–7702.