Generalizable Task Representation Learning for Offline Meta-Reinforcement Learning with Data Limitations

Renzhe Zhou, Chen-Xiao Gao, Zongzhang Zhang*, Yang Yu

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
{zhourz, gaocx}@lamda.nju.edu.cn, {zzzhang, yuy}@nju.edu.cn

Abstract

Generalization and sample efficiency have been long-standing issues concerning reinforcement learning, and thus the field of Offline Meta-Reinforcement Learning (OMRL) has gained increasing attention due to its potential of solving a wide range of problems with static and limited offline data. Existing OMRL methods often assume sufficient training tasks and data coverage to apply contrastive learning to extract task representations. However, such assumptions are not applicable in several real-world applications and thus undermine the generalization ability of the representations. In this paper, we consider OMRL with two types of data limitations: limited training tasks and limited behavior diversity and propose a novel algorithm called GENTLE for learning generalizable task representations in the face of data limitations. GENTLE employs Task Auto-Encoder (TAE), which is an encoder-decoder architecture to extract the characteristics of the tasks. Unlike existing methods, TAE is optimized solely by reconstruction of the state transition and reward, which captures the generative structure of the task models and produces generalizable representations when training tasks are limited. To alleviate the effect of limited behavior diversity, we consistently construct pseudo-transitions to align the data distribution used to train TAE with the data distribution encountered during testing. Empirically, GENTLE significantly outperforms existing OMRL methods on both in-distribution tasks and out-of-distribution tasks across both the given-context protocol and the one-shot protocol.

1 Introduction

Despite the success of Reinforcement Learning (RL) in scenarios where online interaction is consistently available, RL is hampered from real-world applications such as healthcare and robotics controlling due to its sample complexity (Haarnoja et al. 2018; Wu and Zhang 2023) and inferior generalization ability (Kirk et al. 2023). The past decade witnessed tremendous effort from researchers to pave the path for RL toward real-world applications. For example, offline RL (Fujimoto, Meger, and Precup 2019; Kumar et al. 2020; Fujimoto and Gu 2021; Kostrikov, Nair, and Levine 2022; Ran et al. 2023; Gao et al. 2024), which optimizes the policies with a pre-collected and static dataset, provides a solution to relieving RL from costly online interactions, whereas meta-RL (Duan et al. 2016; Finn, Abbeel, and Levine 2017; Rakelly et al. 2019; Zintgraf et al. 2020; Fu et al. 2021; Xu et al. 2021; Luo et al. 2022), which involves training policies over a wide range of tasks, significantly enhances the generalization ability of the learned policies.

Offline Meta-Reinforcement Learning (OMRL) (Li et al. 2020; Li, Yang, and Luo 2021; Dorfman, Shenfeld, and Tamar 2021; Mitchell et al. 2021; Yuan and Lu 2022), as an intersection of offline RL and meta-RL, is promising to combine the good of both worlds. In OMRL, we are provided with datasets collected in various tasks which share some similarity in the underlying structures in dynamics or reward mechanisms, and aim to optimize the meta-policy. The meta-policy is later tested in tasks drawn from the same task distribution. Previous related methods (Li et al. 2020; Li, Yang, and Luo 2021; Yuan and Lu 2022) often interpret the OMRL challenge as task representation learning and meta-policy optimization. The former step aims to obtain indicative task representations from the dataset, while the latter optimizes a meta-policy on top of the learned representation. However, existing methods often assume a sufficient number of training tasks as well as sufficient diversity of behavior policy that collects the datasets, which is not realistic in real-world applications. We find that when the assumptions are not satisfied, the representations tend to overfit and fail to generalize on unseen testing tasks.

In light of this, we propose a new approach to **Gen**eralizable **T**ask representations **Le**arning (GENTLE) to enable effective task recognition in the face of limitations in training task quantity and behavior diversity. GENTLE follows the existing paradigm of OMRL and consists of two interleaving optimization stages: (1) task representation learning and (2) offline meta-policy optimization on top of the learned representations. For (1), we introduce a novel structure, Task Auto-Encoder (TAE) to extract representations from the context information. TAE is optimized to reconstruct the state transition and rewards on the probing data rather than contrastive loss, which models the generative structure of the environment and prevents the encoder from overfitting to miscellaneous features when the number of training tasks is limited. To alleviate TAE's training from overfitting to the behavior policy distribution, we augment the training data via policy, dynamics, and reward relabeling,

*Zongzhang Zhang is the corresponding author.

forcing TAE to learn to exploit the difference in dynamics and rewards rather than input data distributions. For (2), we adopt TD3+BC for its simplicity to optimize a meta-policy with task representations predicted by the TAE.

For evaluations, we compare GENTLE against other baseline algorithms in a set of continuous control tasks with two types of evaluation protocols: *given-context* protocol where the context is collected by an ad-hoc expert policy in the target environment, and *one-shot* protocol where the context is collected by the meta-policy. Experimental results demonstrate the superiority of GENTLE over the baseline methods, and the ablation study also discloses the necessity of each component of GENTLE.

2 Related Work

Offline Meta-Reinforcement Learning. Generalization is a known issue about RL agents (Kirk et al. 2023), and thus meta-RL is proposed to enhance the generalization ability of RL agents. Current meta-RL research can be categorized into two types: gradient-based approaches (Finn, Abbeel, and Levine 2017; Mitchell et al. 2021; Lin et al. 2022), which focus on fast adaptation to new tasks via few-shot gradient descent, and context-based approaches (Duan et al. 2016; Rakelly et al. 2019), which formalize the meta-RL tasks as contextual Markov Decision Processes (MDPs) and learn to encode task representations from histories. The combination of meta-RL and offline setting leads to Offline Meta-Reinforcement Learning (OMRL), a framework where only static task datasets are available to learn a meta-policy. Most of the previous OMRL methods follow the context-based approach (Li et al. 2020; Li, Yang, and Luo 2021; Dorfman, Shenfeld, and Tamar 2021; Pong et al. 2022; Yuan and Lu 2022). Overall, the workflow of these methods can break down into two procedures. The first is to learn a task representation encoder with the offline dataset and augment the states with the learned representations, while the second step is to optimize the meta-policy with offline RL algorithms. MACAW (Mitchell et al. 2021), on the other hand, follows the gradient-based approach and extends MAML (Finn, Abbeel, and Levine 2017) to the offline setting. Our method falls in the first category, with additional considerations for the data limitations. Similar to our motivations, MBML (Li et al. 2020), BOReL (Dorfman, Shenfeld, and Tamar 2021), and CORRO (Yuan and Lu 2022) also identify the impact of behavior policy on task identification, and alleviate this issue either by reward relabeling or generative relabeling on offline datasets. MIER (Mendonca et al. 2020) and SMAC (Pong et al. 2022), on the other hand, focus on the meta-testing stage and mitigate this issue by relabeling the context collected online.

Task Representation Learning. Successful meta-RL agent relies on the learned task representations to make adaptive decisions in different tasks. On how to encode the context and derive task representations, various methods differ in their learning objectives. Earlier methods, such as RL^2 (Duan et al. 2016) and PEARL (Rakelly et al. 2019), apply the same RL objective for the representation encoder. Specifically, RL^2 passes the gradient of the encoder through the representation and optimizes the encoder in an end-to-end fashion, while PEARL trains the encoder via critic loss combined with an additional information bottleneck term. Alternative approaches, like ESCP (Luo et al. 2022), employ the objective of maximizing relational matrix determinant of latent representations. In offline RL, a predominant approach to task representation learning is contrastive-style training (Li et al. 2020; Li, Yang, and Luo 2021; Yuan and Lu 2022). These methods take advantage of the static datasets, construct positive pairs and negative pairs via relabeling or generative augmenting, and afterward apply contrastive objectives to optimize the encoder. In this paper, we propose to extract representations via reconstruction, thus utilizing the generative structure of the underlying dynamics and facilitating the generalization of the representation. This idea has also been explored in past literature (Zintgraf et al. 2020; Yang et al. 2020; Dorfman, Shenfeld, and Tamar 2021; Cho, Jung, and Sung 2022; Ni et al. 2023). However, we apply this idea in the offline setting with off-policy data and characterize it theoretically.

3 Preliminaries

3.1 Problem Formulation

The RL problem can be formulated as a Markov Decision Process (MDP), which can be characterized by a tuple $M = (\mathcal{S}, \mathcal{A}, T, R, \mu_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $T(s'|s,a)$ is the transition function, $R(s,a)$ is the reward function, $\mu_0(s)$ is the initial state distribution, and $\gamma \in [0,1]$ is the discount factor. The policy $\pi(a|s)$ is a distribution over actions. The agent's goal is to find the optimal policy that maximizes the expected cumulative reward (a.k.a. return) $\max_\pi \eta(M,\pi) = \mathbb{E}_{\pi,M}[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)]$, where the expectation is taken over the trajectory distribution which is induced by π in M. The Q-function is defined as the expected return starting from state s, taking action a, and thereafter following policy π: $Q_\pi(s,a) = \mathbb{E}_{\pi,M}[\sum_{t'=t}^{\infty} \gamma^{t'-t} R(s_{t'},a_{t'})|s_t = s, a_t = a]$.

In Offline Meta-Reinforcement Learning (OMRL), we consider a set of tasks where each task is an MDP $M_i = (\mathcal{S}, \mathcal{A}, T_i, R_i, \mu_0, \gamma)$ and sampled from a task distribution $M_i \sim P(\mathcal{M})$. We assume the tasks only differ in transition functions and reward functions, and abbreviate them as $M = (T, R)$. We will use the term *model* to refer to M hereafter. During offline meta-training, we are given N training tasks $\{M_i\}_{i=1}^N$ sampled from $P(\mathcal{M})$ and the corresponding offline datasets $\{D_i\}_{i=1}^N$ generated by behavior policies. Using the fixed offline datasets, the algorithm needs to train a meta-policy π_{meta}. During meta-testing, given a testing task $M \sim P(\mathcal{M})$, the agent first needs to identify the environment with context information B^c before evaluation. Finally, the goal of meta-RL is to find the optimal meta-policy that maximizes the expected return over the task distribution:

$$\max_{\pi_{\text{meta}}} \eta(\pi_{\text{meta}}) = \mathbb{E}_{M\sim P(\mathcal{M})}[\eta(M, \pi_{\text{meta}})]. \tag{1}$$

3.2 TD3+BC

Offline RL trains a policy solely on a fixed dataset D. Due to the mismatch between the distributions of the dataset and the

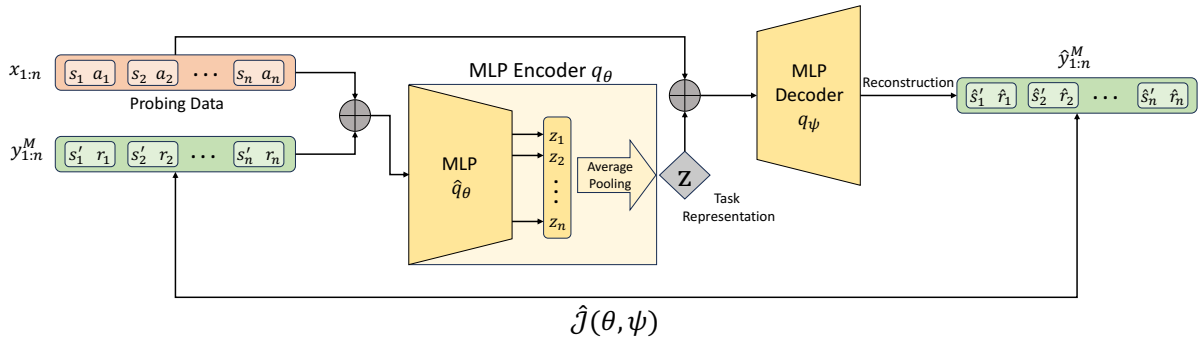

Figure 1: Illustration of TAE. The inputs of TAE are the concatenation of $x_{1:n}$ and $y^M_{1:n}$. The task representation z is obtained by average pooling over the n outputs of $\hat{q}_\theta$. And q_ψ seeks to reconstruct $y^M_{1:n}$ given the probing data $x_{1:n}$ and representation z.

current policy, the agent tends to falsely evaluate the value of out-of-distribution actions and thus misleads the policy optimization (Fujimoto, Meger, and Precup 2019; Kumar et al. 2019). To address the problem, TD3+BC (Fujimoto and Gu 2021) adds a regularization term to the objective in TD3 (Fujimoto, van Hoof, and Meger 2018) to constrain the policy around the dataset:

$$\max_\pi \mathbb{E}_{(s,a)\sim D}[\lambda Q(s, \pi(s)) - (\pi(s) - a)^2], \tag{2}$$

where D is the offline dataset and λ is the coefficient balancing between the TD3's objective and the regularization term. We choose TD3+BC as our backbone offline RL algorithm due to its simplicity and effectiveness. And for generality, we use the stochastic form of policy to represent the meta-policy as $\pi(\cdot|s, z)$ hereafter.

4 Method

In this section, we will elaborate on the core ingredients of our GENTLE method, designed to learn generalizable task representations from offline datasets.

4.1 Task Auto-Encoder

We begin with our TAE, which applies an encoder-decoder architecture to learn task representations. First, we define $x \in \mathcal{S} \times \mathcal{A}$ as the probing data and $\rho(x)$ as its distribution. We additionally assume that the distribution of probing data is task-agnostic:

Assumption 4.1. *(Independency of Probing Data) The distribution of probing data is independent of the model, i.e., $p(x) = p(x|M)$ and $p(M|x) = p(M)$.*

For each entry of the probing data $x = (s, a)$, we first sample a model $M = (T_M, R_M)$ from the task distribution, and continue to sample a state transition $s' \sim T_M(\cdot|s, a)$ and the reward $r = R_M(s, a)$ to assign the label of x as $y = (s', r)$. We use $x_{1:n}$ to represent n i.i.d. samples from $\rho(x)$ and $y_{1:n}$ to represent the corresponding labels of $x_{1:n}$. Throughout this paper, we make a further assumption that the model of the environment is deterministic, i.e.:

Assumption 4.2. *(Deterministic Model) For input $x = (s, a) \in \mathcal{S} \times \mathcal{A}$ and model $M = (T_M, R_M) \in \mathcal{M}$, $p(y|x, M) = \delta(y = y^M)$. Here $y^M = M(x)$ and $M(x) := (T_M(s, a), R_M(s, a))$.*

With all these prepared, we now describe TAE's architecture. In TAE, the encoder $q_\theta(z|x_{1:n}, y_{1:n})$ takes a batch of probing data $x_{1:n}$ and their labels $y_{1:n}$ as inputs, and outputs a distribution of the representation vector. The decoder $q_\psi(\hat{y}_k|x_k, z)$, on the other hand, takes the predicted representation and one of the probing data x_k as input, and finally outputs the distribution of the predicted label.

We train our TAE by maximizing the log-likelihood of the ground-truth label. Specifically, we maximize the following objective in terms of the parameters θ and ψ:

$$\mathcal{J}(\theta, \psi) = \mathbb{E}_{x_{1:n}, M, z}\left[\sum_{k=1}^{n} \log q_\psi(y^M_k|z, x_k)\right], \tag{3}$$

where $x_{1:n} \sim \rho(x), M \sim P(\mathcal{M}), z \sim q_\theta(z|x_{1:n}, y^M_{1:n})$. Intuitively, the encoder takes in the labeled data and predicts the task representation z, while the decoder seeks to reconstruct the ground-truth label via the maximum log-likelihood over the batch of probing data.

We now provide a theoretical characterization of the training process.

Theorem 4.1. *Let $x_{1:n}$ denote i.i.d. probing data sampled from distribution $\rho(x)$ and the probing data is labeled with sampled models from $P(\mathcal{M})$ to construct $y_{1:n}$. With Assumptions 4.1 and 4.2, optimizing $\mathcal{J}(\theta, \psi)$ in terms of the representation encoder q_θ and the decoder q_ψ corresponds to optimizing a lower bound of the mutual information between the task representation and the model $I(M; z)$:*

$$I(M; z) \geq I(M; y_{1:n}|x_{1:n}) + \mathcal{J}(\theta, \psi).$$

The proof is deferred to the appendix[1]. In Theorem 4.1, we show that we can lower-bound the original mutual information $I(M; z)$ via two terms. The first term, $I(M; y_{1:n}|x_{1:n})$, measures how discriminative the probing data $x_{1:n}$ is in terms of models, while the second term is precisely the training objective of TAE which is a reconstruction-oriented objective. Thus, optimizing Equation (3) corresponds to optimizing the lower bound of the

[1]https://www.lamda.nju.edu.cn/zhourz/AAAI24-supp.pdf

mutual information between the extracted representation z and the tasks, which justifies the effectiveness of our objective. It is noteworthy that optimizing mutual information is intractable in practice. A prior method, CORRO (Yuan and Lu 2022), derives a tractable lower bound via InfoNCE (van den Oord, Li, and Vinyals 2018), which employs a generative model to construct negative samples conditioned on the data within the task. Such a method focuses on all of the discriminative aspects of the input data, without consideration for the generative structure shared by different models. In contrast, our approach explicitly learns a decoder to account for this, thus enabling the encoder to closely approximate the intrinsic characteristics of the task.

4.2 Practical Implementation of TAE

Section 4.1 presents a principal framework of TAE, while in this section, we will elaborate on the practical implementation of TAE, as illustrated in Figure 1.

The encoder q_θ is implemented as a deterministic mapping from a batch of probing data to a representation vector $z \in \mathbb{R}^m$. This module consists of a feature transformation network $\hat{q}_\theta$ and an average-pooling layer. For each pair of the input probing data (x_k, y_k), $\hat{q}_\theta$ processes and projects the input into an intermediate embedding vector $z_k \in \mathbb{R}^m$, and the average-pooling layer aggregates the embeddings of each pair by taking the average element-wisely to obtain the final embedding $z \in \mathbb{R}^m$.

Considering that the models of the environment are deterministic, we also implement the decoder q_ψ as a deterministic mapping. However, when q_ψ is deterministic, the probability for the predicted label used in Equation (3) is undefined. To tractably estimate the log probabilities, we follow the common approach of using L2 distances as an approximation for the log probability (Chung et al. 2015; Babaeizadeh et al. 2018; Zhang et al. 2021). The original objective $\mathcal{J}(\theta, \psi)$ can thus be equivalently transformed into:

$$\hat{\mathcal{J}}(\theta, \psi) = \mathbb{E}_{x_{1:n}, M}\Big[\sum_{k=1}^{n}\Big(y_k^M - q_\psi(q_\theta(x_{1:n}, y_{1:n}), x_k)\Big)^2\Big], \tag{4}$$

where $x_{1:n} \sim \rho(x)$, $M \sim P(\mathcal{M})$. Finally, we use $\hat{\mathcal{J}}(\theta, \psi)$ as the training objective of the TAE.

4.3 Constructing the Probing Distribution

The training of TAE is also affected by the distribution of the probing data, whose distribution $\rho(x)$ is desired to satisfy certain requirements. Thus in this section, we investigate how to construct the probing distribution.

Generally, we expect $\rho(x)$ to satisfy the following properties:

1) Independent of models. This is required by Assumption 4.1, which states that the probing data should be sampled from an invariant distribution regardless of models M.

2) Consistent for training and evaluation. This property requires that $\rho(x)$ should resemble the distribution which the meta-policy may encounter during evaluation. We can perceive the probing distribution $\rho(x)$ as some *attention* over all possible aspects of the models, and the extracted representation z is the most discriminative over $\rho(x)$.

Given the above two desiderata, we propose to construct the training data of TAE by policy-relabeling, dynamics-relabeling, and reward-relabeling jointly. Suppose the offline datasets are represented by $\{D_i\}_{i=1}^N$, we first pretrain an estimated model $\widehat{M}_i$ for each task M_i via supervised learning. At the beginning of each iteration, for each task M_i, we randomly pick K_1 states from D_i and K_2 states from other datasets $\cup_j D_j \setminus D_i$ respectively. For each state $\hat{s}$, we first label its action by sampling $\hat{a}$ from the update-to-date meta-policy parameterized by ϕ: $\hat{a} \sim \pi_\phi(a|s, z_i)$. The state transition $\hat{s}'$ and the reward $\hat{r}$ are further predicted by the pre-trained model $\widehat{M}_i$. The constructed tuple $\langle \hat{s}, \hat{a}, \hat{s}', \hat{r} \rangle$ is thus an augmentation sample used to train TAE. The pseudo-code for the augmentation process is listed in Algorithm 1.

Algorithm 1: Data Augmentation via Relabeling

Input: Offline datasets $\{D_i\}_{i=1}^N$, pre-trained models $\{\widehat{M}_i\}_{i=1}^N$, meta-policy π_ϕ, task representations $\{z_i\}_{i=1}^N$

Initialize augmentation buffers $\{D_i^{\text{aug}}\}_{i=1}^N$ as $\emptyset$
for $i = 1, 2, \ldots, N$ **do**
  for $k = 1, 2, ..., K_1$ **do**
    Sample state s_k from D_i
    Sample $a_k \sim \pi_\phi(\cdot|s_k, z_i)$ and $(s'_k, r_k) = \widehat{M}_i(s_k, a_k)$
    $D_i^{\text{aug}} \leftarrow D_i^{\text{aug}} \cup \{(s_k, a_k, s'_k, r_k)\}$
  end for
  for $k = 1, 2, ..., K_2$ **do**
    Sample state s_k from $\{D_j\}_{j=1}^N \setminus D_i$
    Sample $a_k \sim \pi_\phi(\cdot|s_k, z_i)$ and $(s'_k, r_k) = \widehat{M}_i(s_k, a_k)$
    $D_i^{\text{aug}} \leftarrow D_i^{\text{aug}} \cup \{(s_k, a_k, s'_k, r_k)\}$
  end for
end for
return $\{D_i^{\text{aug}}\}_{i=1}^N$

By randomly sampling states across all of the datasets and relabeling the actions with the same meta-policy, we align the probing distribution $\rho(x)$ for each task so that the training procedure approximately satisfies property 1). Note that in the actual implementations, we are sampling from the ego dataset and other datasets with a ratio of $K_1 : K_2$. Theoretically, the ratio should be set to $1 : N-1$ precisely. However, in the practical implementation, we prefer a little biased ratio towards the ego dataset. This is because the estimated model $\widehat{M}_i$ may produce erroneous predictions on the states from other datasets, so we choose to strike a balance with such a biased ratio. For property 2), this is ensured by relabeling actions with the meta-policy. More details about the experiments can be found in the appendix.

4.4 Overall Framework of GENTLE

We summarize the overall meta-training framework of GENTLE in Algorithm 2. At the beginning of each itera-

Environment	Task Set	FOCAL	CORRO	BOReL	GENTLE (Ours)
Point-Robot	Train	-10.04 ± 3.68	$\mathbf{-5.76} \pm 1.02$	-15.38 ± 3.37	-6.46 ± 1.57
Ant-Dir		490.21 ± 73.80	-2.48 ± 14.30	70.32 ± 48.77	$\mathbf{570.20} \pm 60.81$
Cheetah-Vel		-221.97 ± 44.04	-384.03 ± 28.34	-257.82 ± 23.87	$\mathbf{-210.77} \pm 42.32$
Cheetah-Dir		1449.08 ± 182.48	1350.09 ± 124.88	770.86 ± 25.98	$\mathbf{1559.95} \pm 33.89$
Hopper-Params		343.82 ± 27.83	154.62 ± 23.86	185.33 ± 36.91	$\mathbf{354.58} \pm 35.77$
Walker-Params		578.42 ± 53.58	295.91 ± 34.11	213.21 ± 45.96	$\mathbf{627.77} \pm 41.10$
Point-Robot	Test	-13.94 ± 2.66	-11.38 ± 0.67	-16.33 ± 2.74	$\mathbf{-9.71} \pm 1.31$
Ant-Dir		451.63 ± 80.82	20.70 ± 17.28	87.57 ± 38.90	$\mathbf{501.67} \pm 98.49$
Cheetah-Vel		$\mathbf{-342.14} \pm 66.90$	-552.49 ± 34.44	-451.63 ± 29.05	-362.83 ± 46.08
Hopper-Params		224.56 ± 48.78	125.75 ± 39.81	149.74 ± 14.44	$\mathbf{251.60} \pm 14.81$
Walker-Params		277.71 ± 40.32	245.75 ± 19.05	175.24 ± 25.14	$\mathbf{335.59} \pm 51.55$

Environment	Task Set	FOCAL	CORRO	BOReL	GENTLE (Ours)
Point-Robot	Train	-17.44 ± 3.77	-15.12 ± 1.68	-21.64 ± 6.23	$\mathbf{-13.50} \pm 3.26$
Ant-Dir		188.25 ± 54.58	5.68 ± 30.59	96.71 ± 17.34	$\mathbf{596.06} \pm 78.20$
Cheetah-Vel		-301.06 ± 32.43	-450.85 ± 23.98	$\mathbf{-278.21} \pm 27.74$	-278.95 ± 71.80
Cheetah-Dir		75.28 ± 108.24	1099.45 ± 546.08	764.59 ± 12.80	$\mathbf{1525.66} \pm 70.05$
Hopper-Params		192.08 ± 45.43	129.44 ± 22.15	44.95 ± 16.24	$\mathbf{234.63} \pm 24.74$
Walker-Params		294.75 ± 34.19	279.76 ± 52.59	207.87 ± 47.08	$\mathbf{356.70} \pm 40.32$
Point-Robot	Test	-18.39 ± 3.47	-17.16 ± 1.56	-20.37 ± 2.05	$\mathbf{-17.02} \pm 2.60$
Ant-Dir		103.20 ± 56.39	16.44 ± 21.47	63.08 ± 33.67	$\mathbf{464.11} \pm 95.74$
Cheetah-Vel		-377.62 ± 86.74	-616.02 ± 31.93	-466.94 ± 39.26	$\mathbf{-369.93} \pm 64.77$
Hopper-Params		194.61 ± 65.53	135.77 ± 21.46	53.58 ± 21.00	$\mathbf{221.35} \pm 27.32$
Walker-Params		244.03 ± 31.35	217.82 ± 38.90	174.85 ± 20.93	$\mathbf{300.66} \pm 48.78$

Table 1: Performance on the benchmarks. Each number represents the return of the last checkpoint of the meta-policy, averaged over 8 random seeds, $\pm$ represents standard deviation. Top: given-context performance. Bottom: one-shot performance.

tion, we augment context data with the update-to-date meta-policy and the pre-trained models to construct the probing data, which is used to optimize the TAE as well as to compute the task representation. After detaching the gradient w.r.t. the encoder, the representation will be concatenated to raw observations for downstream offline policy optimization. GENTLE iterates between the construction of probing data and the optimization of meta policy until convergence.

At test time, we test GENTLE with both *given-context* protocol and *one-shot* protocol. The former assumes that the meta-policy is given access to a dataset collected in testing task to serve as context B^c. In the latter, we first collect a trajectory as context B^c with z_{prior} sampled from a prior distribution, and calculate task representation $z = q_\theta(B^c)$. Then the meta-policy is evaluated by conditioning on the calculated representation. In the practical implementation, we scale the range of z to $(-1, 1)$ and set z_{prior} to all zeros.

5 Experiments

In this section, we carry out extensive experiments to evaluate GENTLE. We release our code at https://github.com/LAMDA-RL/GENTLE.

5.1 Baselines and Benchmarks

Following the experimental setup in prior studies (Li, Yang, and Luo 2021; Yuan and Lu 2022), we construct a 2D navigation environment and several multi-task MuJoCo (Todorov, Erez, and Tassa 2012) environments to evaluate our algorithm. To comply with the data limitation on the number of training tasks, we sample 10 training tasks and 10 testing tasks for each environment (except for Cheetah-Dir which only has two tasks). For offline dataset generation, we train a SAC (Haarnoja et al. 2018) agent to expert level on each task and then collect trajectories as the offline datasets to simulate the data limitation on behavior diversity.

To evaluate the performance of GENTLE, We compare it with the following OMRL methods: **FOCAL** (Li, Yang, and Luo 2021), **CORRO** (Yuan and Lu 2022), and **BOReL** (Dorfman, Shenfeld, and Tamar 2021). Note that in the original implementation, FOCAL uses BRAC (Wu, Tucker, and Nachum 2019), CORRO and BOReL use SAC as their backbone offline RL algorithms, while we use TD3+BC as the backbone algorithm. To provide a fair comparison, we also implement them with TD3+BC, and conduct the experiments with both the original baselines and the re-implemented baselines. We place the results of the original baselines in the appendix. Finally, we use a variant of BOReL without oracle reward relabeling in the experiments for a fair comparison.

5.2 Main Results

We evaluate GENTLE alongside other baselines across the given-context protocol and the one-shot protocol. As shown in Table 1, it is evident that GENTLE significantly outperforms other baselines in almost all scenarios. When the context is given, all considered methods exhibit reasonable performance. However, it is noteworthy that GENTLE slightly

Environment	Task Set	GENTLE Contrastive	GENTLE w/o Relabel	GENTLE w/o PolicyRelabel	GENTLE
Point-Robot	Train	-18.50 ± 3.20	$\mathbf{-12.80} \pm 0.98$	-16.91 ± 3.59	-13.50 ± 3.26
Ant-Dir		509.20 ± 106.19	206.56 ± 78.00	437.68 ± 91.65	$\mathbf{596.06} \pm 78.20$
Cheetah-Vel		$\mathbf{-263.57} \pm 39.01$	-367.84 ± 51.50	-304.29 ± 60.17	-278.95 ± 71.80
Cheetah-Dir		1465.21 ± 141.88	91.33 ± 726.57	1512.32 ± 89.38	$\mathbf{1525.66} \pm 70.05$
Hopper-Params		199.94 ± 34.45	164.95 ± 51.91	200.13 ± 35.19	$\mathbf{234.63} \pm 24.74$
Walker-Params		284.66 ± 53.11	329.40 ± 46.80	336.03 ± 50.99	$\mathbf{356.70} \pm 40.32$
Point-Robot	Test	-19.21 ± 1.47	$\mathbf{-15.51} \pm 1.36$	-17.71 ± 3.32	-17.02 ± 2.60
Ant-Dir		376.37 ± 110.18	125.99 ± 61.42	356.20 ± 75.99	$\mathbf{464.11} \pm 95.74$
Cheetah-Vel		-415.44 ± 45.16	-439.77 ± 63.57	-448.27 ± 76.54	$\mathbf{-369.93} \pm 64.77$
Hopper-Params		199.78 ± 31.75	144.22 ± 56.86	188.71 ± 29.90	$\mathbf{221.35} \pm 27.32$
Walker-Params		232.79 ± 36.21	279.40 ± 66.80	292.05 ± 61.89	$\mathbf{300.66} \pm 48.78$

Table 2: Performance of GENTLE variants on one-shot protocol. Each number represents the return of the last checkpoint of the meta-policy, averaged over 8 random seeds, $\pm$ represents the standard deviation.

Algorithm 2: Meta Training of GENTLE

Input: Offline datasets $\{D_i\}_{i=1}^N$, models $\{\widehat{M}_i\}_{i=1}^N$, encoder-decoder q_θ, q_ψ, meta-policy π_ϕ, Q-function Q_ω

Pre-train models $\{\widehat{M}_i\}_{i=1}^N$ on $\{D_i\}_{i=1}^N$ independently by supervised learning
for epoch $= 1, 2, \ldots, E$ **do**
  Augment context data via relabeling: $\{D_i^{\text{aug}}\}_{i=1}^N =$ Algorithm 1
  for gradient step $= 1, 2, \cdots, S$ **do**
    // TAE training
    Sample context batches $\{B_i^c\}_{i=1}^N$ from $\{D_i^{\text{aug}} \cup D_i\}_{i=1}^N$
    Update θ, ψ by maximizing Eq. 4 on $\{B_i^c\}_{i=1}^N$
    // Policy optimization
    Compute representation for each task $\{z_i = q_\theta(B_i^c)\}_{i=1}^N$
    Sample RL batches $\{B_i\}_{i=1}^N$ from offline datasets and concatenate the states in $\{B_i\}_{i=1}^N$ with $\{z_i\}_{i=1}^N$
    Update ϕ, ω on $\{B_i\}_{i=1}^N$ by TD3+BC
  end for
end for
return π_ϕ

surpasses the performance of the baseline methods, which indicates the efficacy of the representations extracted by GENTLE. We witness a sharp drop in the baseline methods such as FOCAL when switching to one-shot protocol, which is primarily attributed to the context distribution shift between training and online adaptation. In contrast, GENTLE remarkably sustains a high-performing policy under the one-shot protocol, thereby showcasing the remarkable generalization capability of GENTLE's representation encoder during online adaptation.

5.3 Illustration of the Representations

We dive deeper to examine the learned representations of each algorithm. To illustrate the quality of the learned representations, we use the meta-policy to collect context data in the testing tasks and employ the learned encoder to predict the task representations. For each task, we obtain a total number of 400 representation vectors and employ t-SNE (Van der Maaten and Hinton 2008) to project them onto a two-dimensional plane. The results, as depicted in Figure 2, reveal that the representations predicted by GENTLE naturally form distinct clusters based on their task IDs, signifying the proficiency of GENTLE's encoder in deriving effective task representation even for the testing tasks. On the contrary, FOCAL, CORRO, and BOReL fail to distinguish the representations with online context. The predicted task representations are intertwined and lack clear distinctions in the projected space.

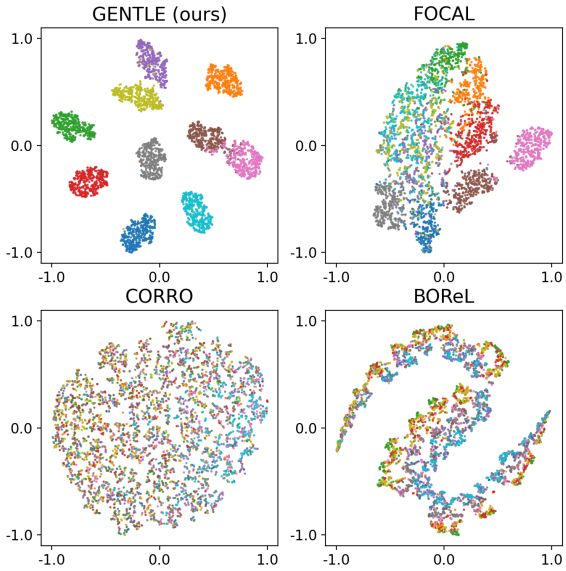

Figure 2: Visualization of the learned representations drawn from 10 testing tasks in Ant-Dir. Each point represents an embedding vector extracted from online context, which is color-coded according to task identity.

5.4 Ablation Study

Ablation on algorithm components. The core ingredients of GENTLE are the TAE structure and the construction of probing data. We investigate the necessity of these components. We introduce several variants of GENTLE. Specifically, we replace TAE's objective with the contrastive-style objective used in FOCAL, and term it as GENTLE-Contrastive. We create another variant, GENTLE without Relabel, by skipping the relabeling process and training TAE directly with the offline datasets. The last variant, GENTLE without PolicyRelabel, skips policy-relabeling and uses the dataset action for relabeling. The results under one-shot protocol are listed in Table 2. By comparing GENTLE and GENTLE-Contrastive, we find that the reconstruction objective does offer benefits over the contrastive-style objective. The variant without relabeling shows severely degenerated performance in certain tasks, particularly in Ant-Dir and Cheetah-Dir. Without the process of relabeling, the context encoder tends to overfit to training data distribution. Finally, although the variant without policy-relabeling shows favorable results on all tasks, its performance still lags behind GENTLE. This exemplifies the importance of the consistency property for the probing data distribution.

Ablation on sampling ratio. To construct the probing data, we sample them from the ego dataset and other datasets with a ratio of $K_1 : K_2$. We investigate the influence of this ratio. We conduct experiments on Ant-Dir, with a series of sampling ratios: 1:0, 1:1, 1:3, 1:6, 1:9, 1:12, 1:15. Specifically, a ratio of 1:0 signifies exclusive sampling from the ego task dataset, while a ratio of 1:9 implies comprehensive sampling from all other task datasets. And for ratios 1:12 and 1:15, we downsample the ego task dataset. The results are depicted in Figure 3. The ratio of 1:0 exhibits the poorest performance, attributed to its reliance solely on the ego dataset which fails to ensure property 1) in Section 4.3. Increasing the sampled number of other task datasets leads to improved performance. Notably, larger ratios yield similar or slightly worse performance, and ratios 1:12 and 1:15 result in instability and performance drop, which can be attributed to the estimation error of the pre-trained dynamics models. Besides, a larger ratio also requires more computation. Based on the above considerations, we opt for a balanced ratio of 1:3 in all of the experiments.

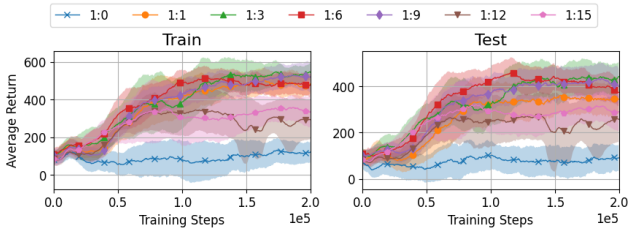

Figure 3: Return curves of GENTLE on training and testing tasks in Ant-Dir across 7 sampling ratios.

Ablation on training tasks and behavior diversity. GENTLE is proposed to tackle data limitations on training tasks and behavior diversity. To inspect how GENTLE adapts to the former, we vary the number of training tasks between 4-10 while leaving testing tasks unchanged. As shown in Figure 4, a small number of training tasks significantly diminishes GENTLE's generalization over testing tasks. With an increase in the number of training tasks, GENTLE exhibits enhanced generalization performance.

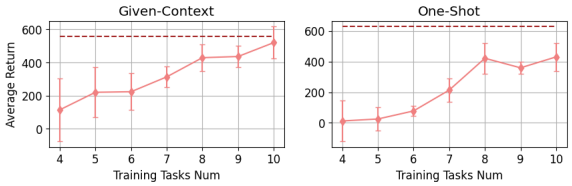

Figure 4: Performance of GENTLE on Ant-Dir testing tasks under different numbers of training tasks across 8 seeds. Dotted line represents the performance on training tasks.

Lastly, we illustrate the capability of handling limited behavior diversity by inspecting how algorithms adapt to improved diversity. We use a medium-level policy to collect *medium* context data, and use 5 logged checkpoints of policy to collect *mixed* context data. The *medium* and *mixed* data are only used to train the context encoder, while the policy is still optimized with expert datasets. As shown in Table 3, FOCAL's performance witnessed significant improvement as the behavior diversity improves (from *expert* to *mixed*), while GENTLE remains approximately the same since the data-relabeling process already enriches the behavior diversity and aligns the distribution even with the *expert* context.

Task Set	Dataset	FOCAL	GENTLE
Train	Expert	188.25 ± 54.58	$\mathbf{596.06} \pm 78.20$
	Medium	235.69 ± 97.81	507.32 ± 50.18
	Mixed	$\mathbf{353.43} \pm 68.28$	571.41 ± 97.11
Test	Expert	103.20 ± 56.39	464.11 ± 95.74
	Medium	124.58 ± 58.95	475.11 ± 85.40
	Mixed	$\mathbf{213.48} \pm 81.40$	$\mathbf{481.10} \pm 80.07$

Table 3: Performance on one-shot protocol in Ant-Dir across 8 seeds under different types of context training datasets.

6 Conclusion and Future Work

In this paper, we propose an innovative OMRL algorithm called GENTLE. We adopt a novel structure of Task Auto-Encoder (TAE), which incorporates an encoder-decoder framework trained by reconstruction of rewards and transitions. We also employ relabeling to construct pseudo-transitions, which aligns the TAE's training data distribution with the testing data distribution during meta-adaptation. Our experimental results show GENTLE's superior performance in diverse environments and tasks.

Notwithstanding the achievements, our work leaves some aspects unaddressed. We lack provisions for sparse reward settings, and do not tackle the development of exploration policy during meta-testing. We leave these for future work.

Acknowledgements

This work is supported by the National Key R&D Program of China (2022ZD0114804), the National Science Foundation of China (62276126, 62250069), and the Fundamental Research Funds for the Central Universities (14380010).

References

Babaeizadeh, M.; Finn, C.; Erhan, D.; Campbell, R. H.; and Levine, S. 2018. Stochastic Variational Video Prediction. In *International Conference on Learning Representations (ICLR)*.

Cho, M.; Jung, W.; and Sung, Y. 2022. Multi-Task Reinforcement Learning with Task Representation Method. In *ICLR Workshop on Generalizable Policy Learning in Physical World*.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A. C.; and Bengio, Y. 2015. A Recurrent Latent Variable Model for Sequential Data. In *Advances in Neural Information Processing Systems (NIPS)*, 2980–2988.

Dorfman, R.; Shenfeld, I.; and Tamar, A. 2021. Offline Meta Reinforcement Learning - Identifiability Challenges and Effective Data Collection Strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 4607–4618.

Duan, Y.; Schulman, J.; Chen, X.; Bartlett, P. L.; Sutskever, I.; and Abbeel, P. 2016. RLˆ2: Fast Reinforcement Learning via Slow Reinforcement Learning. *arXiv preprint arXiv:1611.02779*.

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

Fu, H.; Tang, H.; Hao, J.; Chen, C.; Feng, X.; Li, D.; and Liu, W. 2021. Towards Effective Context for Meta-Reinforcement Learning: An Approach Based on Contrastive Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 7457–7465.

Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 20132–20145.

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In *International Conference on Machine Learning (ICML)*, 2052–2062.

Fujimoto, S.; van Hoof, H.; and Meger, D. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *International Conference on Machine Learning (ICML)*, 1582–1591.

Gao, C.; Wu, C.; Cao, M.; Kong, R.; Zhang, Z.; and Yu, Y. 2024. ACT: Empowering Decision Transformer with Dynamic Programming via Advantage Conditioning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

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 *International Conference on Machine Learning (ICML)*, 1856–1865.

Kirk, R.; Zhang, A.; Grefenstette, E.; and Rocktäschel, T. 2023. A Survey of Zero-Shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76: 201–264.

Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations (ICLR)*.

Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 11761–11771.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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 *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Lin, S.; Wan, J.; Xu, T.; Liang, Y.; and Zhang, J. 2022. Model-Based Offline Meta-Reinforcement Learning with Regularization. In *International Conference on Learning Representations (ICLR)*.

Luo, F.; Jiang, S.; Yu, Y.; Zhang, Z.; and Zhang, Y. 2022. Adapt to Environment Sudden Changes by Learning a Context Sensitive Policy. In *AAAI Conference on Artificial Intelligence (AAAI)*, 7637–7646.

Mendonca, R.; Geng, X.; Finn, C.; and Levine, S. 2020. Meta-Reinforcement Learning Robust to Distributional Shift via Model Identification and Experience Relabeling. *arXiv preprint arXiv:2006.07178*.

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

Ni, F.; Hao, J.; Mu, Y.; Yuan, Y.; Zheng, Y.; Wang, B.; and Liang, Z. 2023. MetaDiffuser: Diffusion Model as Conditional Planner for Offline Meta-RL. In *International Conference on Machine Learning (ICML)*, 26087–26105.

Pong, V. H.; Nair, A. V.; Smith, L. M.; Huang, C.; and Levine, S. 2022. Offline Meta-Reinforcement Learning with Online Self-Supervision. In *International Conference on Machine Learning (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 *International Conference on Machine Learning (ICML)*, 5331–5340.

Ran, Y.; Li, Y.; Zhang, F.; Zhang, Z.; and Yu, Y. 2023. Policy Regularization with Dataset Constraint for Offline Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, 28701–28717.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A Physics Engine for Model-Based Control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5026–5033.

van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Wu, C.; and Zhang, Z. 2023. Surfing Information: The Challenge of Intelligent Decision-Making. *Intelligent Computing*, 2: Article 0041.

Wu, Y.; Tucker, G.; and Nachum, O. 2019. Behavior Regularized Offline Reinforcement Learning. *arXiv preprint arXiv:1911.11361*.

Xu, F.; Jiang, S.; Yin, H.; Zhang, Z.; Yu, Y.; Li, M.; Li, D.; and Liu, W. 2021. Enhancing Context-Based Meta-Reinforcement Learning Algorithms via An Efficient Task Encoder. In *AAAI Conference on Artificial Intelligence (AAAI)*, 15937–15938.

Yang, J.; Petersen, B. K.; Zha, H.; and Faissol, D. M. 2020. Single Episode Policy Transfer in Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

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

Zhang, J.; Wang, J.; Hu, H.; Chen, T.; Chen, Y.; Fan, C.; and Zhang, C. 2021. MetaCURE: Meta Reinforcement Learning with Empowerment-Driven Exploration. In *International Conference on Machine Learning (ICML)*, 12600–12610.

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 *International Conference on Learning Representations (ICLR)*.