Meta-Normalizing Flow for Data-Limited Offline Meta-Reinforcement Learning (Student Abstract)

Lianghui Liu, Zongzhang Zhang*

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
School of Artificial Intelligence, Nanjing University, Nanjing 210023, China
liulh@lamda.nju.edu.cn, zzzhang@nju.edu.cn

Abstract

Offline Meta-Reinforcement Learning (OMRL) leverages pre-collected data to adapt to new tasks. Context-based methods learn task representations from contexts. However, the context is influenced by both the task and the behavior policy. The mismatch between the behavior policy and the testing policy causes a context distribution shift problem, which results in poor task representations and degraded performance. This problem is exacerbated in settings with data limitations. To address this, we propose a novel approach called Meta-Normalizing Flow (Meta-NF). First, it employs a highly expressive and sample-efficient normalizing flow policy. Second, it incorporates a metric for testing-time task representation selection to effectively mitigate the context shift problem. Empirical results demonstrate that Meta-NF outperforms existing OMRL methods, with both components contributing to its strong performance.

Introduction

Offline Meta-Reinforcement Learning (OMRL) learns to generalize to new tasks from pre-collected data. Most OMRL methods (Li, Yang, and Luo 2021; Zhou et al. 2024) employ a context-based framework, learning compact task representations from contexts, which are then used to condition the policy. These representations are typically learned using contrastive or reconstruction methods. In this work, we consider two types of data limitations: limited training tasks and limited behavior diversity. For evaluation, we adopt the zero-shot protocol where the context is current collected data by the testing policy. The context shift problem caused by the mismatch between the behavior policy and the testing policy poses a severe challenge.

To solve this, we propose a novel algorithm called Meta-Normalizing Flow (Meta-NF). Since normalizing flow (Rezende and Mohamed 2015) is capable of modeling complex data distributions, Meta-NF employs a normalizing flow policy to mitigate the extrapolation error inherent in offline reinforcement learning. Furthermore, Meta-NF incorporates a selection mechanism that filters out inaccurate task representations generated from out-of-distribution contexts during meta-testing.

*Corresponding author: Zongzhang Zhang.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Our Method

This section provides a detailed description of our method.

Task Representation Learning

We adopt the task representation learning approach from UNICORN (Li et al. 2024). Accordingly, we provide only a brief review of the method and omit the exact loss function. A transition $\tau_t = (s_t, a_t, r_t, s'_t)$ is comprised of the state, action, reward, and next state. A context encoder maps each τ_t to a latent task representation z_t. A context decoder reconstructs r_t and s'_t from s_t, a_t, and z_t. We sample a context of m transitions from the replay buffer $\mathcal{D}_i$ of task i and compute the mean representation, denoted as z^i. This mean representation z^i is used to reconstruct the batch of transitions, forming the reconstruction loss $\mathcal{L}_{\text{RECON}}$. Another loss $\mathcal{L}_{\text{FOCAL}}$ encourages clustering by pushing the representations z^i from the same task closer together while pushing apart representations from different tasks. Finally, the overall loss combines these two objectives:

$$\mathcal{L} = \mathcal{L}_{\text{RECON}} + \mathcal{L}_{\text{FOCAL}}. \tag{1}$$

Policy Learning

A normalizing flow $\boldsymbol{f}$ maps a d-dimensional random variable $\boldsymbol{x} \in \mathbb{R}^d$ to $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^d$. Because its Jacobian $\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ is invertible, the density $p(\boldsymbol{y})$ can be exactly computed using the change of variables formula: $p(\boldsymbol{y}) = p(\boldsymbol{f}^{-1}(\boldsymbol{y}))/|\det(\boldsymbol{J})|$. The $\det$ means determinant.

Papers such as (Dinh, Krueger, and Bengio 2015; Dinh, Sohl-Dickstein, and Bengio 2017; Kingma and Dhariwal 2018) often construct flows whose Jacobian determinants can be computed efficiently. By composing multiple such flows, we obtain a powerful flow: $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}_h \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x})$. This flow $\boldsymbol{f}$ can transform a simple prior density into a complex target density. We use $\boldsymbol{f}$ as our policy and adopt a Gaussian prior. As a policy, it samples an action $\boldsymbol{a}$ by first drawing from Gaussian distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and then computing $\boldsymbol{a} = \boldsymbol{f}(\boldsymbol{x})$. The log probability for any action $\boldsymbol{a}$ is $\log p(\boldsymbol{a}) = \log p_0(\boldsymbol{f}^{-1}(\boldsymbol{a})) - \sum_{i=1}^{h} \log|\det(\boldsymbol{J}_i)|$, where $\boldsymbol{J}_i$ is the Jacobian of $\boldsymbol{f}_i$. For simplicity, conditioning variables are omitted in the notation.

We can employ the normalizing flow policy similarly to a Gaussian policy, yet with significantly greater expressive

Environment	Task Set	FOCAL	CSRO	GENTLE	UNICORN	**Meta-NF** (Ours)
Point-Robot	Train	-12.11 ± 0.88	-11.93 ± 0.59	-11.55 ± 1.01	-12.18 ± 1.35	$\mathbf{-10.11} \pm 0.76$
Ant-Dir		203.34 ± 36.54	253.44 ± 113.60	105.08 ± 90.99	231.81 ± 50.20	$\mathbf{574.63} \pm 57.27$
Cheetah-Vel		-238.32 ± 37.28	-202.56 ± 38.98	-286.74 ± 29.63	-213.45 ± 37.17	$\mathbf{-84.50} \pm 3.77$
Cheetah-Dir		-165.93 ± 774.17	-159.06 ± 288.47	-463.55 ± 349.32	-89.72 ± 682.70	$\mathbf{1083.13} \pm 432.55$
Hopper-Params		175.01 ± 77.42	143.05 ± 62.02	263.16 ± 63.22	177.33 ± 35.92	$\mathbf{365.34} \pm 26.81$
Walker-Params		372.79 ± 92.46	231.13 ± 122.50	340.04 ± 55.43	375.52 ± 96.00	$\mathbf{448.46} \pm 26.53$
Point-Robot	Test	-17.16 ± 0.70	-16.51 ± 1.02	-16.67 ± 0.73	-17.28 ± 1.59	$\mathbf{-14.53} \pm 0.81$
Ant-Dir		11.10 ± 52.62	124.09 ± 62.42	113.43 ± 39.81	31.72 ± 88.57	$\mathbf{433.45} \pm 67.79$
Cheetah-Vel		-307.21 ± 29.83	-299.40 ± 92.25	-385.48 ± 49.95	-310.48 ± 65.35	$\mathbf{-148.89} \pm 10.61$
Hopper-Params		137.53 ± 54.30	112.95 ± 42.37	204.00 ± 38.98	135.88 ± 35.90	$\mathbf{233.21} \pm 22.39$
Walker-Params		330.48 ± 89.03	229.68 ± 128.88	318.72 ± 54.98	288.37 ± 115.46	$\mathbf{346.96} \pm 45.95$

Environment	Task Set	Meta-NF w/o Both	Meta-NF w/o Selection	Meta-NF w/o NF	**Meta-NF** (Ours)
Point-Robot	Train	-12.18 ± 1.35	-12.40 ± 0.59	-10.22 ± 0.59	$\mathbf{-10.11} \pm 0.76$
Ant-Dir		231.81 ± 50.20	348.91 ± 54.56	329.69 ± 80.81	$\mathbf{574.63} \pm 57.27$
Cheetah-Vel		-213.45 ± 37.17	-85.30 ± 14.64	-207.11 ± 38.29	$\mathbf{-84.50} \pm 3.77$
Cheetah-Dir		-89.72 ± 682.70	127.52 ± 219.01	$\mathbf{1286.77} \pm 244.96$	1083.13 ± 432.55
Hopper-Params		177.33 ± 35.92	300.53 ± 25.12	219.91 ± 72.66	$\mathbf{365.34} \pm 26.81$
Walker-Params		375.52 ± 96.00	$\mathbf{500.46} \pm 70.91$	346.66 ± 87.79	448.46 ± 26.53
Point-Robot	Test	-17.28 ± 1.59	-17.38 ± 1.21	-15.17 ± 0.89	$\mathbf{-14.53} \pm 0.81$
Ant-Dir		31.72 ± 88.57	115.55 ± 83.32	221.55 ± 59.38	$\mathbf{433.45} \pm 67.79$
Cheetah-Vel		-310.48 ± 65.35	-162.31 ± 20.27	-347.47 ± 57.86	$\mathbf{-148.89} \pm 10.61$
Hopper-Params		135.88 ± 35.90	$\mathbf{255.77} \pm 30.51$	137.03 ± 27.65	233.21 ± 22.39
Walker-Params		288.37 ± 115.46	$\mathbf{383.58} \pm 63.19$	268.97 ± 31.83	346.96 ± 45.95

Table 1: Experiment results. Each number represents the return of the last checkpoint of the meta-policy, averaged over 5 random seeds, $\pm$ represents standard deviation. Top: main results under the zero-shot protocol. Bottom: ablation results.

power. It more effectively constrains the policy's deviation from the behavior policy and also ensures that the context remains closer to the training distribution during meta-testing.

We train the policy similarly to TD3+BC (Fujimoto and Gu 2021). The Q-function is optimized via Bellman updates, while the policy loss is a weighted combination of a behavior cloning term with the negative expected Q-value.

Task Representation Selection in Testing Time

We consider the zero-shot protocol. Assume the agent has collected a partial trajectory $\tau_{1:t} = (\tau_1, \cdots, \tau_t)$ as context. We encode each τ_i into z_i. For each candidate representation z_i, we use it to reconstruct the entire observed context $\tau_{1:t}$. We then select the representation z_j that yields the lowest reconstruction loss and use it to condition the policy for sampling the next action.

Experiment

Following the settings of previous studies, we evaluate algorithms on a 2D navigation environment and several multi-task MuJoCo environments. For the majority of tasks, we sample 10 training tasks and train an expert policy to collect 50 trajectories per task, simulating data-limited conditions. Detailed experimental settings and additional results are provided in the appendix[1].

We compare Meta-NF with the following OMRL methods: FOCAL (Li, Yang, and Luo 2021), CSRO (Gao et al. 2023), GENTLE (Zhou et al. 2024), and UNICORN. To provide a fair comparison, we remove the data augmentation in GENTLE, reimplement them with TD3+BC, and use mean task representation during meta-testing.

The results are summarized in Table 1. Meta-NF consistently outperforms all baseline methods across all tasks by a substantial margin. We evaluate the following ablations: T1 (Meta-NF w/o Both), T2 (Meta-NF w/o Selection), T3 (Meta-NF w/o NF), and T4 (full Meta-NF). Comparisons between T1 and T2 or between T3 and T4 demonstrate that the normalizing flow component significantly improves performance across nearly all tasks. Similarly, comparisons between T1 and T3 or between T2 and T4 indicate that task representation selection also contributes to some tasks, such as Cheetah-Dir and Ant-Dir tasks.

Conclusion

This paper tackles zero-shot adaptation in OMRL under data-limited settings. It leverages a powerful normalizing flow policy alongside a task representation selection mechanism to address the context shift problem. Experimental results demonstrate the effectiveness of this approach. For future work, developing more robust task representation selection mechanisms would be a promising direction.

Acknowledgments

This work is supported by the National Science Foundation of China (No. 62276126).

References

Dinh, L.; Krueger, D.; and Bengio, Y. 2015. NICE: Nonlinear Independent Components Estimation. In *ICLR*.

[1]https://www.lamda.nju.edu.cn/liulh/files/Meta-NF.pdf

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using Real NVP. In *ICLR*.

Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. In *NeurIPS*, 20132–20145.

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*, 80024–80043.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In *NeurIPS*, 10236–10245.

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.; Yu, Y.; Zhao, J.; and Heng, P. 2024. Towards an Information Theoretic Framework of Context-Based Offline Meta-Reinforcement Learning. In *NeurIPS*, 75642–75667.

Rezende, D. J.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. In *ICML*, 1530–1538.

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.