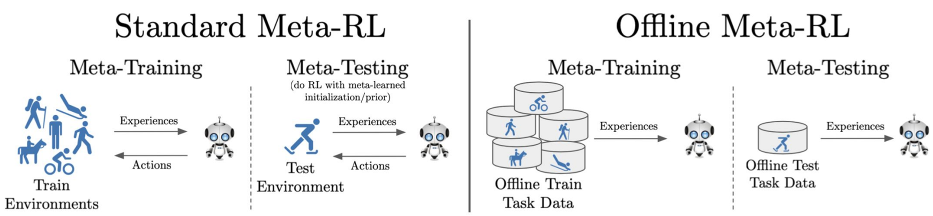
Offline Meta-Reinforcement Learning with Advantage Weighting

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, Chelsea Finn

Stanford University · Berkeley University of California



Overview

We introduce the **offline meta-reinforcement learning** (offline meta-RL) problem setting and **propose an algorithm** that performs well in this setting. Offline meta-RL is analogous to the widely successful supervised **pre-train + fine-tune** transfer learning strategy, in which a model is pre-trained on a large batch of fixed, pre-collected data (possibly from various tasks) and fine-tuned to a new task using relatively little data. That is, in offline meta-RL, we **meta-train on fixed, pre-collected data** from several tasks and **adapt to a new task with a very small amount** of data from the new task. By nature of being offline, algorithms for offline meta-RL can utilize the **largest possible pool of training data** available and eliminate potentially unsafe or costly data collection during meta-training.

This setting inherits the challenges of offline RL, but it **differs significantly** because offline RL does not generally consider a) **transfer to new tasks** or b) **limited data from the test task**, both of which we face in offline meta-RL. Targeting the offline meta-RL setting, we propose an algorithm, **Meta-Actor Critic with Advantage Weighting (MACAW)**. MACAW is an optimization-based meta-learning algorithm that uses simple, supervised regression objectives for both the inner and outer loop of meta-training. Our experiments show that MACAW enables fully offline meta-RL and demonstrates superior performance in a variety of settings including **offline variants of standard meta-RL benchmarks, adapting from sub-optimal data, and learning from a limited number of training tasks.**

Key Contributions

1. The **fully offline meta-reinforcement learning** problem setting

2. An algorithm for offline meta-reinforcement learning, called **Meta-Actor Critic with Advantage Weighting** or **MACAW**

Offline Meta-RL Problem Setting

We consider the **offline meta-reinforcement learning setting.** An offline meta-RL problem consists of:

- Training tasks $\mathcal{T}_i$ sampled from a distribution $p(\mathcal{T})$

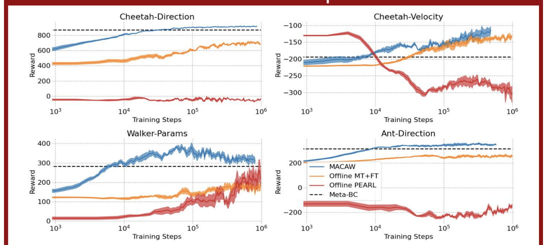
- Fixed buffers of offline data $D_i = \{s_{i,j}, a_{i,j}, s'_{i,j}, r_{i,j}\}$ for each training task

Each D_i is populated with trajectories sampled from a corresponding behavior policy μ_i

During offline **meta-training**, an agent trains on the data in the fixed training buffers **without interacting with the environment.**

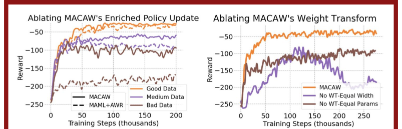
During **fully offline meta-testing**, the agent is given a small amount of interaction data from a test task sampled from $p(\mathcal{T})$ which is used for adaptation **without interacting with the test environment**. The agent is then evaluated by its **average on-policy return** on the test task.

Benchmark Comparison



We evaluate MACAW on offline variants of **standard meta-RL problems** [1], [2]. We compare with an offline variant of **PEARL** [3], an offline **multi-task + fine-tuning** baseline based on AWR [4], and a **meta-imitation** baseline.

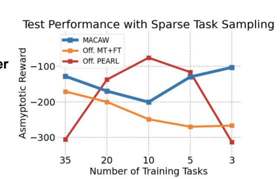
Ablation Studies



The left setting compares MACAW's performance with and without the **enriched policy update** when adaptation data **quality** is varied. When adaptation data is **sub-optimal**, MACAW's enriched policy update enables a **significant improvement** in performance. The right setting compares MACAW's performance with and without **weight transform layers.** The weight transform significantly improves both **speed of learning** as well as **asymptotic performance** of the policy. Both settings utilize the **Cheetah-Vel** problem.

Training Task Sparsity Experiment

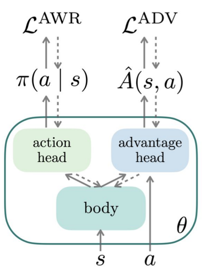
Generally, we prefer an offline meta-RL algorithm that can generalize to new tasks when presented with only a **small number of meta-training tasks**. This experiment evaluates the extent to which various algorithms rely on **dense sampling of the space of tasks** during training in order to generalize well.



Surprisingly, Offline PEARL **completely fails** to learn both when training tasks are plentiful and when they are scarce, but learns **relatively effectively** in the middle regime (5-20 tasks). In contrast, MACAW finds a solution of reasonable quality for **any sampling of the task space**, even for very dense or very sparse samplings of the training tasks. In practice, this property is desirable, because it allows the same algorithm to **scale to very large offline datasets while still producing useful adaptation behaviors for small datasets**. Ultimately, MACAW effectively exploits the available data when meta-training tasks are plentiful and shows by far the greatest robustness when tasks are scarce.

MACAW Policy Architecture

MACAW's policy uses a **multi-headed** architecture in order to accommodate an **auxiliary policy loss** used to enrich the standard AWR [4] policy loss gradient. See **Theorems 1&2 in the paper.** During adaptation in both meta-training and meta-testing, the policy outputs actions and **action advantage estimates**; this eliminates the **policy gradient ambiguity problem** present when adapting using only the standard AWR policy loss.



Method

MACAW meta-learns **initializations** for a **value function** and **policy** during **meta-training** (see Algorithm 1 below). During **meta-testing**, MACAW takes a small number of gradient steps on the inner loop **value function** and **policy** objectives using a small batch of offline data (see Algorithm 2 below).

The **value function** objective is defined as (as in [4]):

$$\mathcal{L}_V(\phi, D) \triangleq \mathbb{E}_{\mathbf{s},\mathbf{a}\sim D}\left[(V_\phi(\mathbf{s}) - \mathcal{R}_D(\mathbf{s},\mathbf{a}))^2\right]$$

The **inner loop** policy objective is defined as:

$$\mathcal{L}_\pi \triangleq \mathcal{L}^{\text{AWR}} + \lambda \mathcal{L}^{\text{ADV}}$$

Where $\mathcal{L}^{\text{ADV}}$ is an auxiliary **advantage regression** loss designed to **increase inner loop policy update expressiveness**, defined as

$$\mathcal{L}^{\text{ADV}}(\theta, \phi'_i, D) \triangleq \mathop{\mathbb{E}}_{\mathbf{s},\mathbf{a}\sim D}\left[(\hat{A}(\mathbf{s},\mathbf{a}) - (\mathcal{R}_D(\mathbf{s},\mathbf{a}) - V_{\phi'_i}(\mathbf{s})))^2\right]$$

and $\mathcal{L}^{\text{AWR}}$ is the Advantage-Weighted Regression [4] policy loss:

$$\mathcal{L}^{\text{AWR}}(\vartheta, \varphi, D) = \mathbb{E}_{\mathbf{s},\mathbf{a}\sim D}\left[-\log \pi_\vartheta(\mathbf{a}|\mathbf{s}) \exp\left(\frac{1}{T}(\mathcal{R}_D(\mathbf{s},\mathbf{a}) - V_\varphi(\mathbf{s}))\right)\right]$$

The complete MACAW algorithm for both meta-training and meta-testing is described below.

Algorithm 1 MACAW Meta-Training

```
Input: Tasks {T_i}, offline buffers {D_i}
Hyperparameters: learning rates α1, α2, η1, η2, training iterations n, temperature T
Randomly initialize meta-parameters θ, φ
for n steps do
    for task T_i ∈ {T_i} do
        Sample disjoint batches D_i^tr, D_i^ts ~ D_i
        φ'_i ← φ − η1 ∇_φ L_V(φ, D_i^tr)
        θ'_i ← θ − α1 ∇_θ L_π(θ, φ'_i, D_i^tr)
    φ ← φ − η2 Σ_i [∇_φ L_V(φ'_i, D_i^ts)]
    θ ← θ − α2 Σ_i [∇_θ L^AWR(θ'_i, φ'_i, D_i^ts)]
```

Algorithm 2 MACAW Meta-Testing

```
Input: Test task T_j, offline experience D, meta-policy π_θ, meta-value function V_φ
Hyperparameters: learning rates α1, η, adaptation iterations n, temperature T
Initialize θ_0 ← θ, φ_0 ← φ.
for n steps do
    φ_{t+1} ← φ_t − η1 ∇_{φ_t} L_V(φ_t, D)
    θ_{t+1} ← θ_t − α1 ∇_{θ_t} L_π(θ_t, φ_{t+1}, D)
```

References

[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning, 2017a.

[2] Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. arXiv preprint arXiv:1810.06784, 2018.

[3] Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In International Conference on Machine Learning, 2019.

[4] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

Correspondence to **eric.mitchell@cs.stanford.edu**

Paper link: