Continuous Adaptation via Meta-Learning in Nonstationary and Competitive Environments

Maruan Al-Shedivat [1] Trapit Bansal [2] Yuri Burda [4] Ilya Sutskever [4] Igor Mordatch [4] Pieter Abbeel [3,4]

Abstract

The ability to continuously learn and adapt from limited experience in nonstationary environments is an important milestone on the path towards general intelligence.

Approach:

Problem. We define a setup for continuous adaptation in a realistic few-shot regime.

Algorithm. A variant of gradient-based meta-learning. Training is done on pairs of temporally shifted tasks. The agent learns to anticipate and adapt to nonstationary transitions.

Evaluation. Use nonstationary locomotion and competitive multi-agent environments. Define iterated games to consistently evaluate adaptation in the multi-agent setting.

1. Motivation

- **Nonstationary worlds require fast continuous adaptation.** Multi-agent systems, any machine learning systems in the wild.
- **A step towards continual, never-ending learning [1].** A system that can keep learning and improving over a lifetime.

2. Background

Learning to learn for fast adaptation

Given a task description, a good adaptation rule must generate a model suitable for the task at hand:

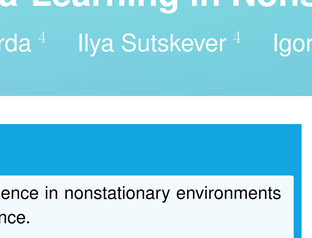
distribution over tasks/datasets

adaptation rule takes a task description and outputs a model

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{T\sim\mathcal{P}}\left[\mathcal{L}_T\left[g_\theta(T)\right]\right], \text{ where}$$

$$\mathcal{L}_T\left[g_\theta(T)\right] := \mathbb{E}_{z\sim\mathcal{D}_T}\left[L\left(f_\phi(z)\right)\right],\ \phi := g_\theta(T)$$

distribution over examples for task T



In few-shot classification, tasks are described by small labeled datasets.

Model-agnostic meta-learning (MAML)

- Adaptation via a gradient steps on a task-specific loss:

$$\phi_i = g_\theta(T_i) := \theta - \alpha\nabla_\theta\mathcal{L}_{T_i}(f_\theta)$$

- At meta-training, search for a good parameter initialization:

$$\min_\theta \sum_{T_i\sim\mathcal{P}} \mathcal{L}^{\text{tst}}_{T_i}\left(f_{\theta-\alpha\nabla_\theta\mathcal{L}^{\text{trn}}_{T_i}(f_\theta)}\right)$$

Intuition:

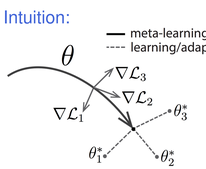
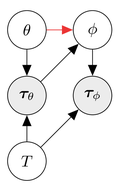


figure from [2]

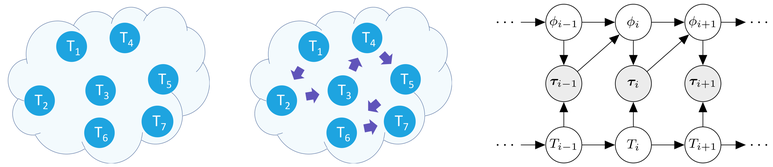
3. Adaptation as Inference



Meta-learning for RL. The data are trajectories: $\boldsymbol{\tau} := (x_1, a_1, r_1, \ldots, x_H, a_H, r_H)$.

- Treat policy parameters, tasks, and all trajectories as **random variables**.
- In this view, adaptation = inference and meta-learning = learning a prior.
- Brings in compositionality of probabilistic modeling:
  - Different priors and inference algorithms ⇒ new meta-learning methods (*cf.* [3]).
  - **Different dependencies between the variables ⇒ new adaptation methods.**

4. Meta-learning for Continuous Adaptation



Real tasks are rarely i.i.d. There are often relationships that we can to exploit. Assuming that the tasks change over time consistently, we can learn to anticipate the changes and adapt to the temporal shifts.

Meta-learn on **pairs of tasks** by solving:

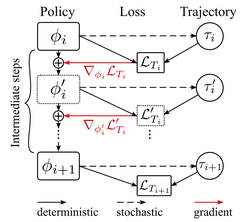
$$\min_\theta \mathbb{E}_{\mathcal{P}(T_0),\mathcal{P}(T_{i+1}|T_i)}\left[\sum_{i=1}^{L}\mathcal{L}_{T_i,T_{i+1}}(\theta)\right], \text{ where}$$

$$\mathcal{L}_{T_i,T_{i+1}}(\theta) := \mathbb{E}_{\boldsymbol{\tau}^{1:K}_{i,\theta}\sim P_{T_i}(\boldsymbol{\tau}|\theta)}\left[\mathbb{E}_{\boldsymbol{\tau}_{i+1,\phi}\sim P_{T_{i+1}}(\boldsymbol{\tau}|\phi)}\left[L_{T_{i+1}}(\boldsymbol{\tau}_{i+1,\phi}) \mid \boldsymbol{\tau}^{1:K}_{i,\theta}, \theta\right]\right]$$

The algorithm

Meta-learning at training time:

- Sample a batch of task pairs, $\{(T_i, T_{i+1})\}_{i=1}^n$.
- Rollout trajectories $\boldsymbol{\tau}^{1:K}_\theta$ for T_i (the first task in each pair) using π_θ.
- Compute $\phi\left(\boldsymbol{\tau}^{1:K}_\theta, \theta, \alpha\right)$ and rollout τ_ϕ for each T_{i+1} using π_ϕ.
- Update θ and α using the stochastic gradient of the meta-loss.



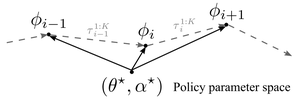
Unbiased estimator of the gradient of the meta-loss

$$\nabla_{\theta,\alpha}\mathcal{L}_{T_i,T_{i+1}}(\theta,\alpha) = \mathbb{E}_{\substack{\boldsymbol{\tau}^{1:K}_{i,\theta}\sim P_{T_i}(\boldsymbol{\tau}|\theta)\\ \boldsymbol{\tau}_{i+1,\phi}\sim P_{T_{i+1}}(\boldsymbol{\tau}|\phi)}}\left[L_{T_{i+1}}(\boldsymbol{\tau}_{i+1,\phi})\left[\nabla_{\theta,\alpha}\log\pi_\phi(\boldsymbol{\tau}_{i+1,\phi}) + \nabla_\theta\sum_{k=1}^{K}\log\pi_\theta(\boldsymbol{\tau}^k_{i,\theta})\right]\right]$$

N.B.: The highlighted term was missing in the original derivation of the policy gradients for MAML-RL, which made the gradient estimators biased [2]. A general solution for such issues is developed in [4].

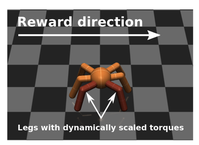
Adaptation at execution time:

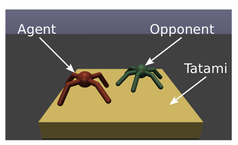
- Interact with the environment using π_ϕ. Store all trajectories and importance weights, π_θ/π_ϕ, in the experience buffer.
- Before each episode, compute ϕ using importance-corrected adaptation updates using trajectories from the buffer.



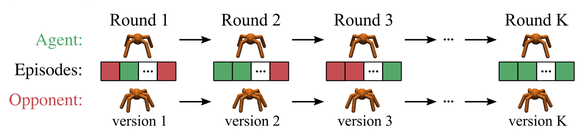
$$\phi_i := \theta - \alpha\frac{1}{K}\sum_{k=1}^{K}\left(\frac{\pi_\theta(\boldsymbol{\tau}^k)}{\pi_{\phi_{i-1}}(\boldsymbol{\tau}^k)}\right)\nabla_\theta L(\boldsymbol{\tau}^k), \quad \boldsymbol{\tau}^{1:K}\sim\text{ExperienceBuffer},$$

5. Environments & Setup





Iterated adaptation games



A multi-round game where an agent must adapt to opponents of increasing competence. The outcome of each round is either win, loss, or draw. Opponents are either pre-trained or also adapting.

6. Experiments

Nonstationary locomotion

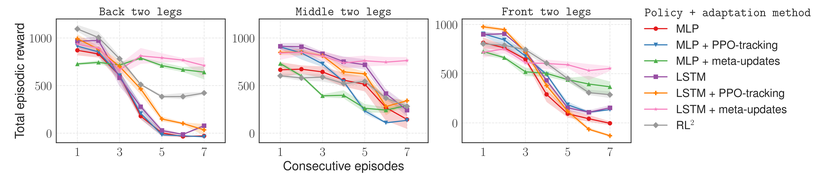


Figure 1: Episodic rewards for 7 consecutive episodes in 3 held out nonstationary locomotion environments.

Multi-agent competition

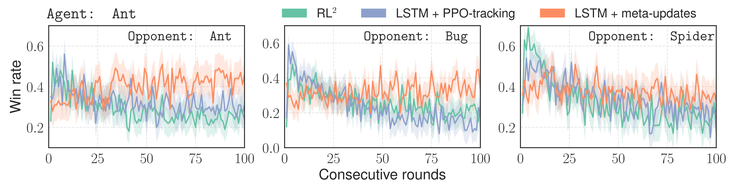


Figure 2: Win rates for different adaptation strategies in iterated games with opponents pretrained via self-play. Competence of the opponents was increasing from round to round based on the precomputed policies at different stages of self-play.

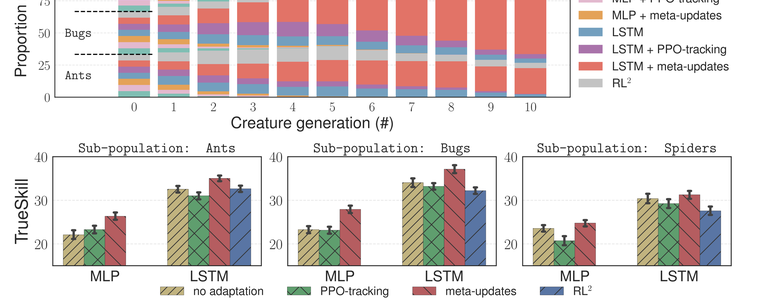


Figure 3: **Top:** Evolution of a population of 1050 agents. **Bottom:** TrueSkill of the top-performing agents in the population.

Discussion

Limitations:

- Gradient-based adaptation requires estimating second-order derivatives. This is computation- and sample-inefficient (needs large batches).
- Unlikely to work with sparse rewards.

Future work:

- Adaptation + model-based RL.
- Adaptation + curriculum learning/generation.
- Multi-step adaptation (i.e., planning with tasks); better use of historical information.

References

[1] Ring '94, '97, Mitchell et al. '15.

[2] Finn et al.: Model-agnostic meta-learning for fast adaptation of deep networks. ICML 2017.

[3] Grant et al.: Recasting Gradient-Based Meta-Learning as Hierarchical Bayes. ICLR 2018.

[4] Foerster et al.: DiCE: The Infinitely Differentiable Monte-Carlo Estimator. ICLR WS 2018.

Acknowledgements

Harri Edwards, Jakob Foerster, Aditya Grover, Aravind Rajeswaran, Vikash Kumar, Yuhuai Wu, Carlos Florensa, anonymous reviewers, and the OpenAI team.

Videos & highlights:

International Conference on Learning Representations, ICLR 2018, Vancouver, Canada

[1] Carnegie Mellon University, [2] UMass Amherst, [3] UC Berkeley, [4] OpenAI

Correspondence: alshedivat@cs.cmu.edu