RL 논문 리뷰 스터디 5기

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

Chelsea Finn, Pieter Abbeel, Sergey Levine

SooHan Kang

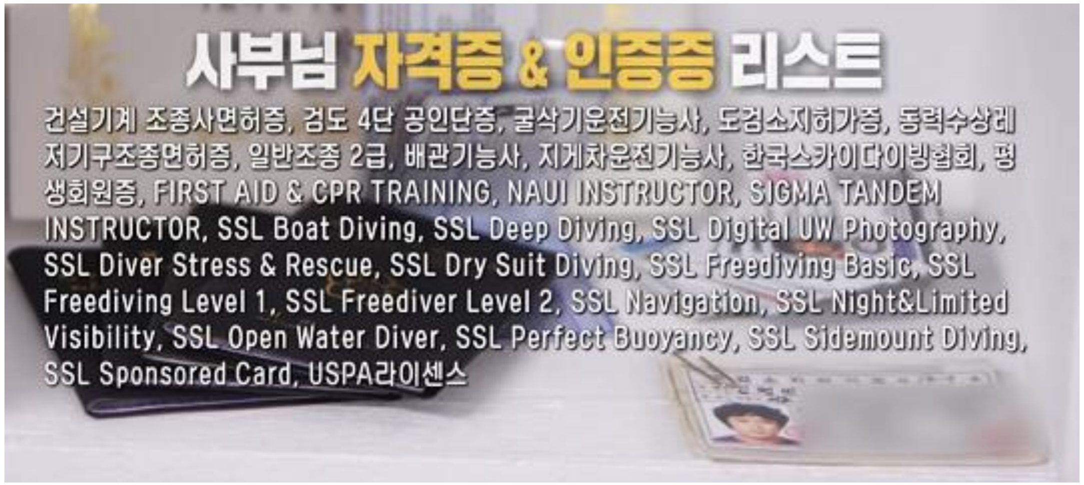
2021-07-05

Abstract

Abstract

- We propose an algorithm for meta-learning that is model-agnostic, in the sense that it is compatible with any model trained with gradient descent and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning.

- META – LEARNING?
  - Goal : Train a model on a variety of learning tasks, Solve new learning tasks using only a small number of training samples.



사부님 자격증 & 인증증 리스트

건설기계 조종사면허증, 검도 4단 공인단증, 굴삭기운전기능사, 도검소지허가증, 동력수상레저기구조종면허증, 일반조종 2급, 배관기능사, 지게차운전기능사, 한국스카이다이빙협회, 평생회원증, FIRST AID & CPR TRAINING, NAUI INSTRUCTOR, SIGMA TANDEM INSTRUCTOR, SSL Boat Diving, SSL Deep Diving, SSL Digital UW Photography, SSL Diver Stress & Rescue, SSL Dry Suit Diving, SSL Freediving Basic, SSL Freediving Level 1, SSL Freediver Level 2, SSL Navigation, SSL Night&Limited Visibility, SSL Open Water Diver, SSL Perfect Buoyancy, SSL Sidemount Diving, SSL Sponsored Card, USPA라이센스

Abstract

- In our approach, the parameters of the model are explicitly trained such that a small number of gradient steps with a small amount of training data from a new task will produce good generalization performance on that task.

- In effect, our method trains the model to be easy to fine-tune.

- We demonstrate that this approach leads to state-of-the-art performance on two few-shot image classification benchmarks, produces good results on few-shot regression, and accelerates fine-tuning for policy gradient reinforcement learning with neural network policies

MAML

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

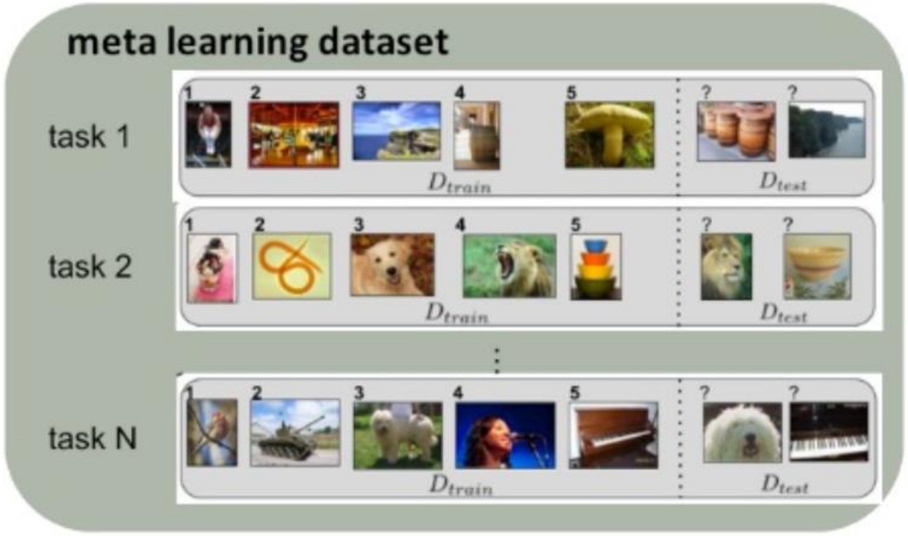
- META LEARNING
  - Goal : Train a model on a variety of learning tasks, Solve new learning tasks using only a small number of training samples.

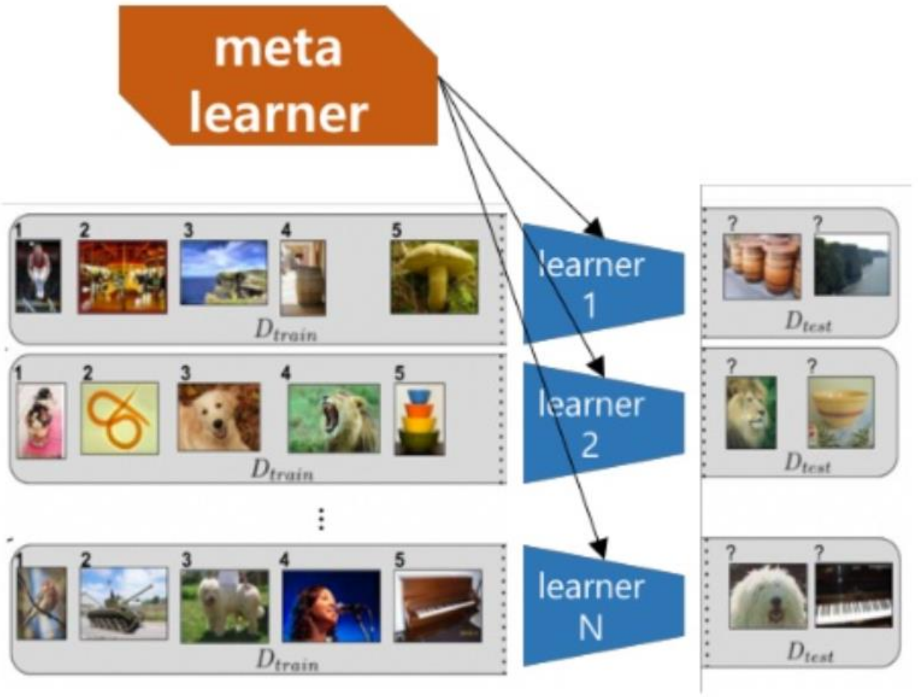
  - Algorithm : Meta-Learner, Good weight initialization(MAML)

  - Compatible with any model trained with gradient descent and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning.(Model-Agnostic!)

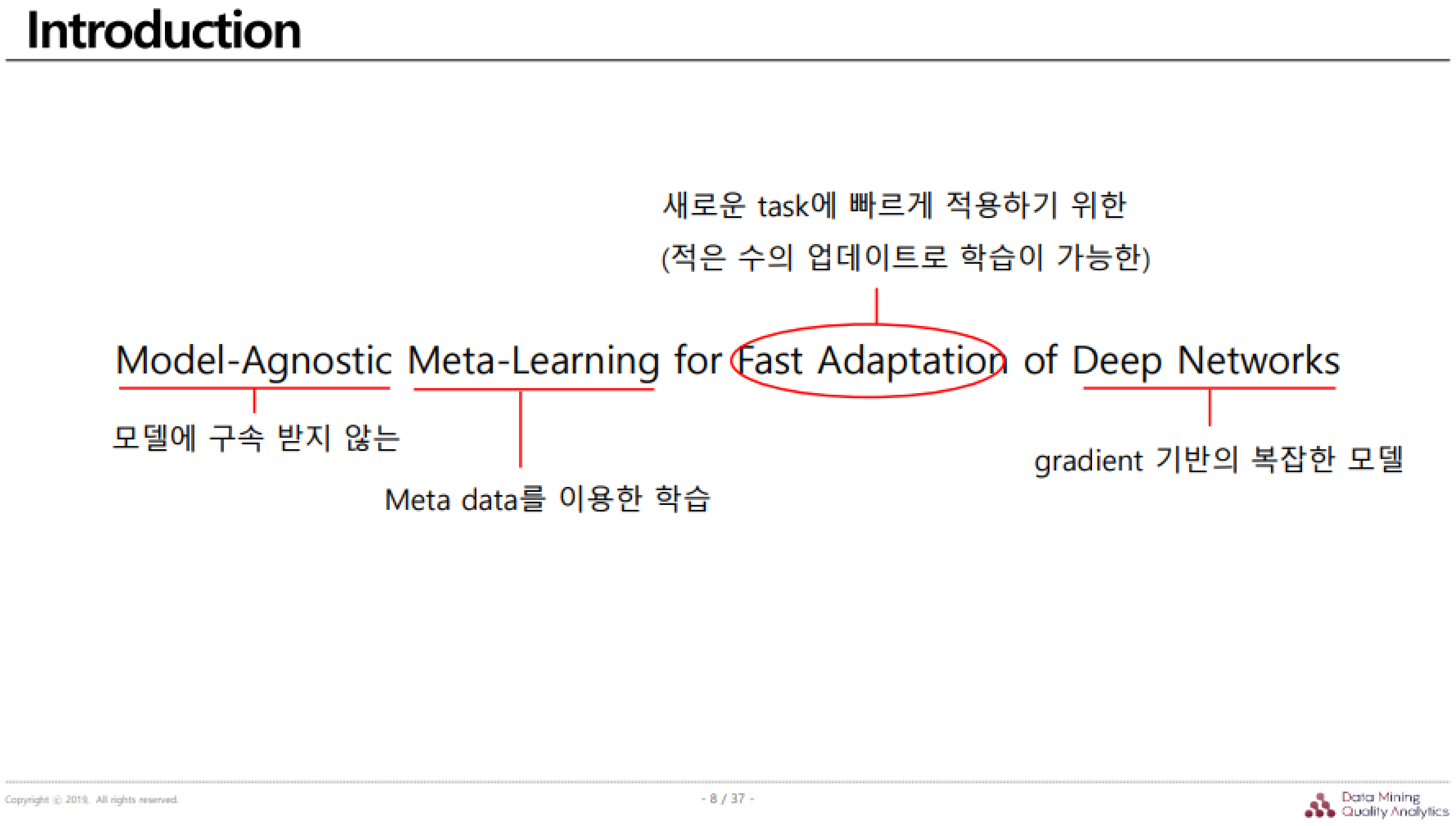
Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

Meta Supervised Learning





Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks



Introduction

새로운 task에 빠르게 적용하기 위한
(적은 수의 업데이트로 학습이 가능한)

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

모델에 구속 받지 않는

Meta data를 이용한 학습

gradient 기반의 복잡한 모델

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

- Model f, Observation x, Outputs a, Loss Function L, Distribution over initial observation $q(x_1)$, Transition distribution $q(x_{t+1}|x_t, a_t)$, Episode length H.

- Task $T = \{L(x_1, a_1, \dots, x_H, a_H), q(x_1), q(x_{t+1}|x_t, a_t), H\}$

- In i.i.d. supervised learning problem, the length H=1.

- The loss $L(x_1, a_1, \dots, x_H, a_H) \rightarrow R$, provides task-specific feedback, which might be in the form of a misclassification loss or a cost function in a Markov decision process.

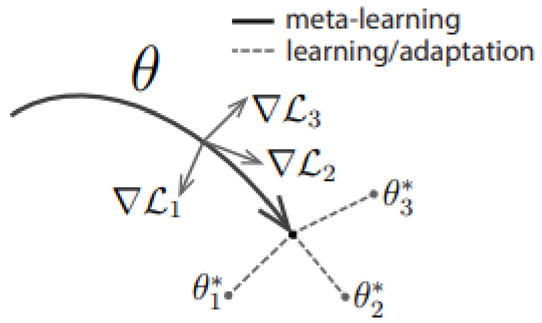


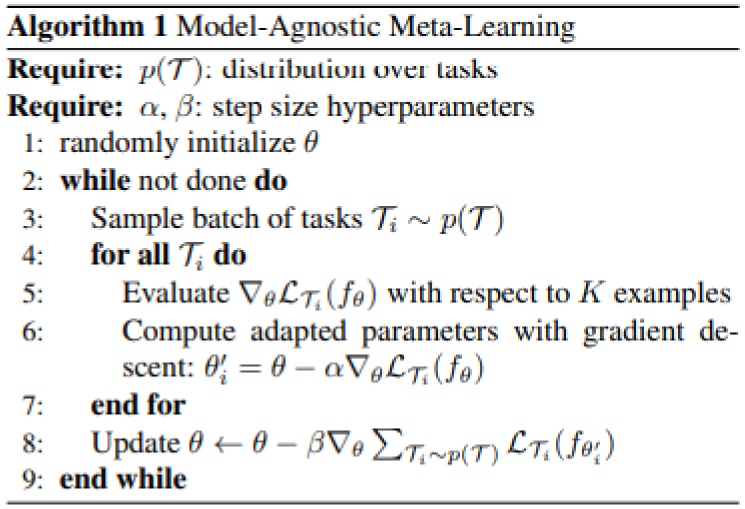
Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

- In our meta-learning scenario, we consider a distribution over tasks $p(T)$ that we want our model to be able to adapt to.

- In the K-shot learning setting, the model is trained to learn a new task T_i drawn from $p(T)$ from only K samples drawn from q_i and feedback L_{T_i} generated by T_i.

- In effect, the test error on sampled tasks T_i serves as the training error of the meta-learning process.



```
Algorithm 1 Model-Agnostic Meta-Learning
Require: $p(\mathcal{T})$: distribution over tasks
Require: $\alpha$, $\beta$: step size hyperparameters
1: randomly initialize $\theta$
2: while not done do
3:     Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4:     for all $\mathcal{T}_i$ do
5:         Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to $K$ examples
6:         Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7:     end for
8:     Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: end while
```

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

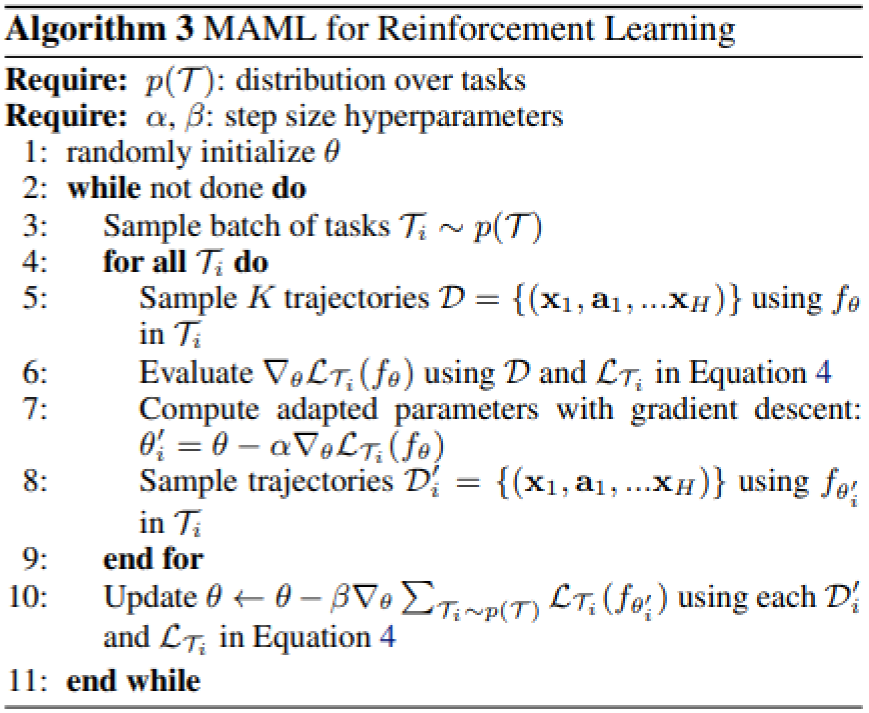
- In reinforcement learning (RL), the goal of few-shot meta-learning is to enable an agent to quickly acquire a policy for a new test task using only a small amount of experience in the test setting.

- A new task might involve achieving a new goal or succeeding on a previously trained goal in a new environment. For example, an agent might learn to quickly figure out how to navigate mazes so that, when faced with a new maze, it can determine how to reliably reach the exit with only a few samples. In this section, we will discuss how MAML can be applied to meta-learning for RL.

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

- Each RL task T_i contains an initial state distribution $q_i(x_1)$ and a transition distribution $q_i(x_{t+1}|x_t, a_t)$, and the loss L_{T_i} corresponds to the (negative) reward function R.

- The model being learned, f_θ, is a policy that maps from states x_t to a distribution over actions at a_t each timestep t ∈ {1, ..., H}.



Algorithm 3 MAML for Reinforcement Learning

```
Require: $p(\mathcal{T})$: distribution over tasks
Require: $\alpha$, $\beta$: step size hyperparameters
 1: randomly initialize $\theta$
 2: while not done do
 3:     Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
 4:     for all $\mathcal{T}_i$ do
 5:         Sample $K$ trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_\theta$ in $\mathcal{T}_i$
 6:         Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
 7:         Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
 8:         Sample trajectories $\mathcal{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta_i'}$ in $\mathcal{T}_i$
 9:     end for
10:     Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
11: end while
```

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = -\mathbb{E}_{\mathbf{x}_t, \mathbf{a}_t \sim f_\phi, q_{\mathcal{T}_i}} \left[\sum_{t=1}^{H} R_i(\mathbf{x}_t, \mathbf{a}_t) \right].$$

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

- The gradient updates are computed using vanilla policy gradient (REINFORCE) (Williams, 1992), and we use trust-region policy optimization (TRPO) as the meta-optimizer (Schulman et al., 2015).

- We compare to three baseline models: (a) pretraining one policy on all of the tasks and then fine-tuning, (b) training a policy from randomly initialized weights, and (c) an oracle policy which receives the parameters of the task as input, which for the tasks below corresponds to a goal position, goal direction, or goal velocity for the agent.

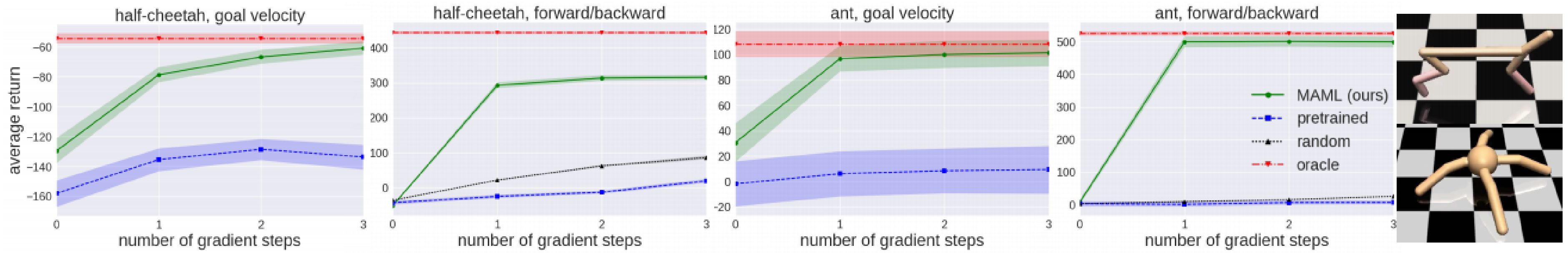


Figure 5. Reinforcement learning results for the half-cheetah and ant locomotion tasks, with the tasks shown on the far right. Each gradient step requires additional samples from the environment, unlike the supervised learning tasks. The results show that MAML can adapt to new goal velocities and directions substantially faster than conventional pretraining or random initialization, achieving good performs in just two or three gradient steps. We exclude the goal velocity, random baseline curves, since the returns are much worse (< -200 for cheetah and < -25 for ant).

Discussion and Future Work

- ❑ We introduced a meta-learning method based on learning easily adaptable model parameters through gradient descent.

- ❑ Our approach has a number of benefits.
  - It is simple and does not introduce any learned parameters for metalearning
  - It can be combined with any model representation that is amenable to gradient-based training, and any differentiable objective, including classification, regression, and reinforcement learning.
  - Lastly, since our method merely produces a weight initialization, adaptation can be performed with any amount of data and any number of gradient steps.

Thank you