https://arxiv.org/pdf/1703.03400.pdf
https://cs330.stanford.edu/

Paper review
Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks (ICML 2017)

Presentation: **Jeiyoon Park**
6th Generation, TAVE

8/31/2021

TAVE Research

Outline

Outline

1. Contribution
2. Method
3. Experiments
4. Conclusion

Contribution

1. Model-agnostic meta-learning

1) We humans are "generalists"

,which means we learn and adapt quickly from only a few examples

2) This kind of fast and flexible learning is challenging

since the agent must integrate its prior experience with a small amount of new information, while avoiding overfitting to the new data

3) The mechanism should be general to the task without constraints on the architecture

Contribution

2. So why this paper?

1) No expanding the # of learned parameters and constraints on the model architecture

2) Training model’s initial parameters such that the model has maximal performance on a new task after the parameters have been updated through one or more gradient steps

3) A small # of gradient steps with a small amount of training data from a new task will produce great generalization performance

4) The method is applicable to a variety of different learning (e.g. classification, regression, and reinforcement learning)

Outline

Detour: Meta-learning

1. What is a task?

For now: dataset $\mathcal{D}$ / loss function $\mathcal{L}$ $\longrightarrow$ model f_θ

Different tasks can vary based on:

- different objects
- different people
- different objectives
- different lighting conditions
- different words
- different languages
- …

Not *just* different "tasks"

Detour: Meta-learning

2. There are many tasks with shared structure!

Even if the tasks are seemingly unrelated:

- The **laws of physics** underly real data.
- People are **all organisms with intentions**.
- The **rules of English** underly English language data.
- Languages all develop for **similar purposes**.

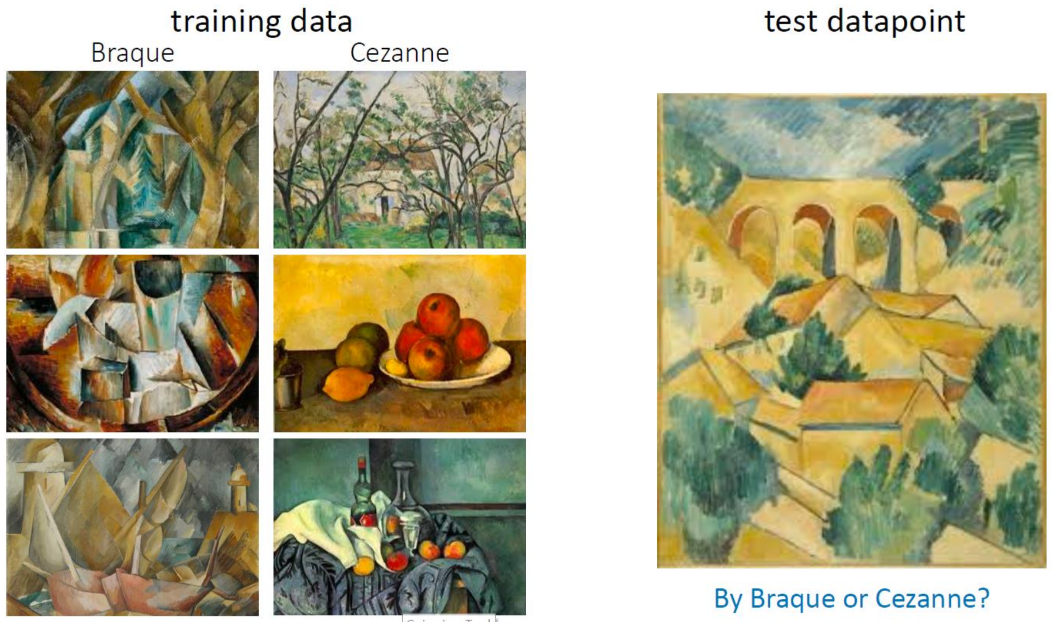
This leads to far greater structure than random tasks.

Detour: Meta-learning

3. Informal definitions

1) The multi-task learning problem: Learn all of the tasks more quickly or more proficiently than learning them independently

2) The meta-learning problem: Given data/experience on previous tasks, learn a new task more quickly and/or more proficiently



By Braque or Cezanne?

Method

1. Problem set-up

1) f: a model, X: observations, a: output

2) Each task: $\mathcal{T}$

$$\mathcal{T} = \{\mathcal{L}(X_1, a_1, \ldots, X_H, a_H), q(X_1), q(X_{t+1}|X_t, a_t), H\}$$

where, $\mathcal{T}$ consists of loss function $\mathcal{L}$, initial observation $q(X_1)$, transition distribution $q(X_{t+1}|X_t, a_t)$, and episode length H (e.g. H = 1 in classification setting)

3) The loss $\mathcal{L}$ provides task-specific feedback, which might be in the form of a misclassification loss

Method

2. Meta-learning scenario

1) We consider a distribution over task $p(\mathcal{T})$

2) In K-shot learning setting, the model is trained to learn a new task $\mathcal{T}_i$ drawn from $p(\mathcal{T})$ from only K samples drawn from q_i and feedback $\mathcal{L}_{\mathcal{T}_i}$ generated by $\mathcal{T}_i$

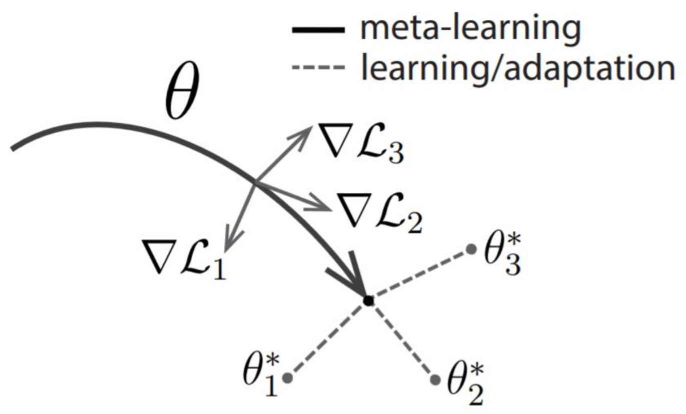


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

$$\mathcal{T} = \{\mathcal{L}(X_1, a_1, \dots, X_H, a_H), q(X_1), q(X_{t+1}|X_t, a_t), H\}$$

3) During meta-training, with sampled $\mathcal{T}_i$, the model is trained with K samples and feedback from the corresponding loss $\mathcal{L}_{\mathcal{T}_i}$, and then tested on new samples from $\mathcal{T}_i$

4) The test error serves as the training error of the meta-learning process

Method

3. Model-agnostic meta-learning

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
```

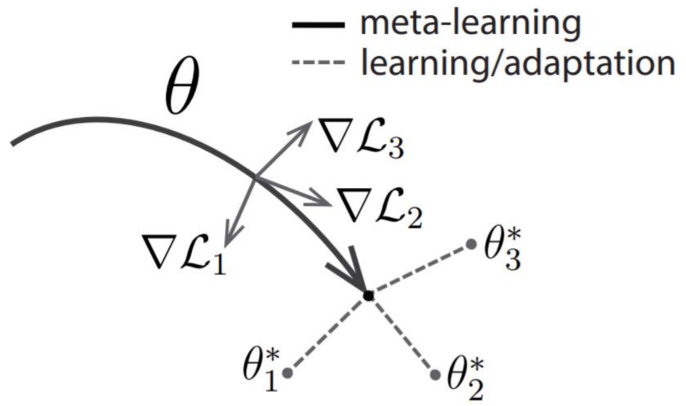
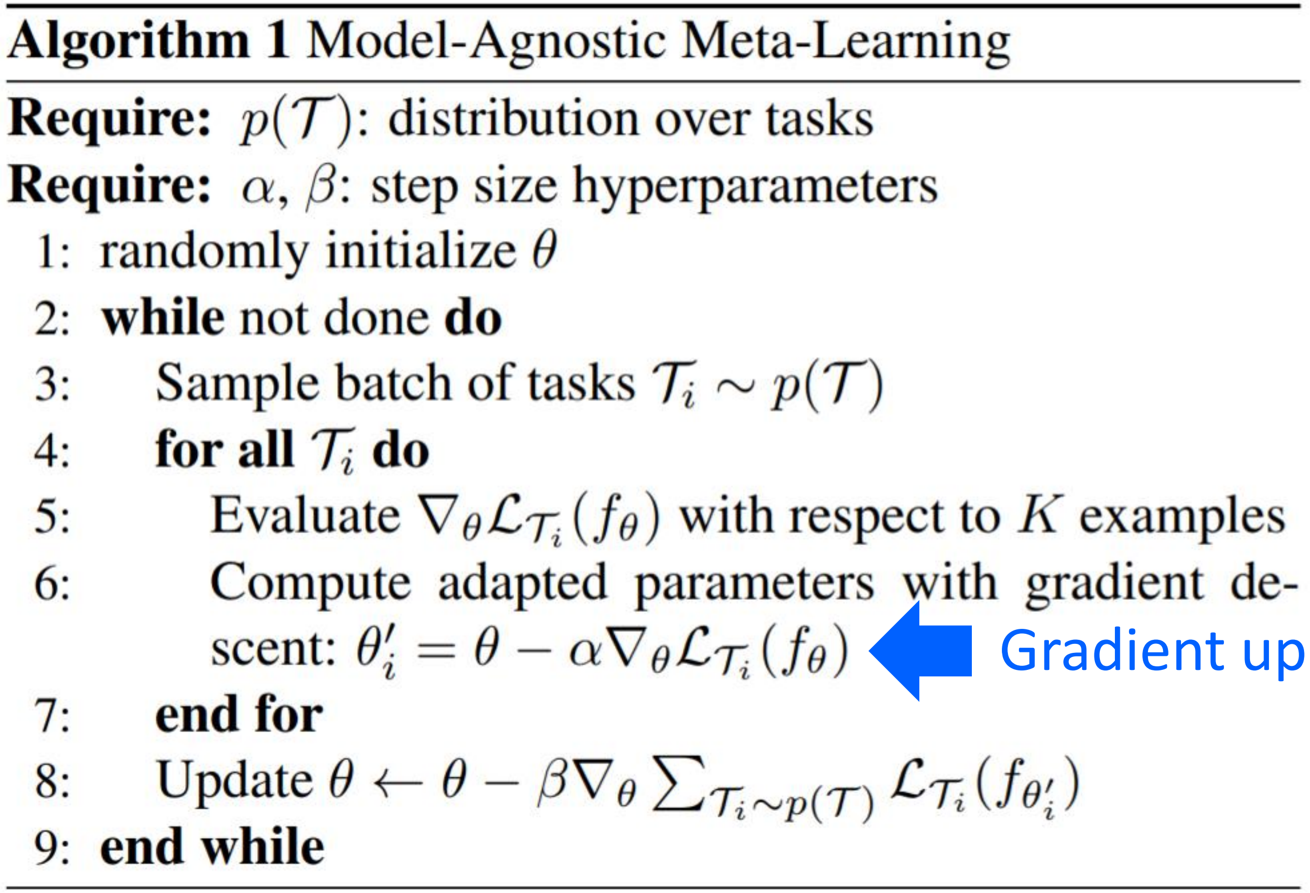


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Method

3. Model-agnostic meta-learning



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
```

Gradient update

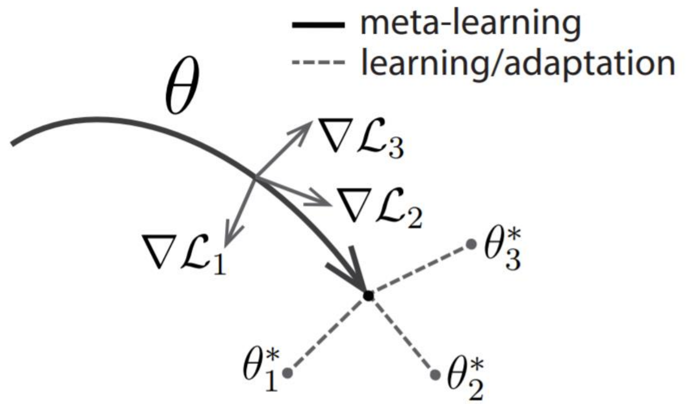
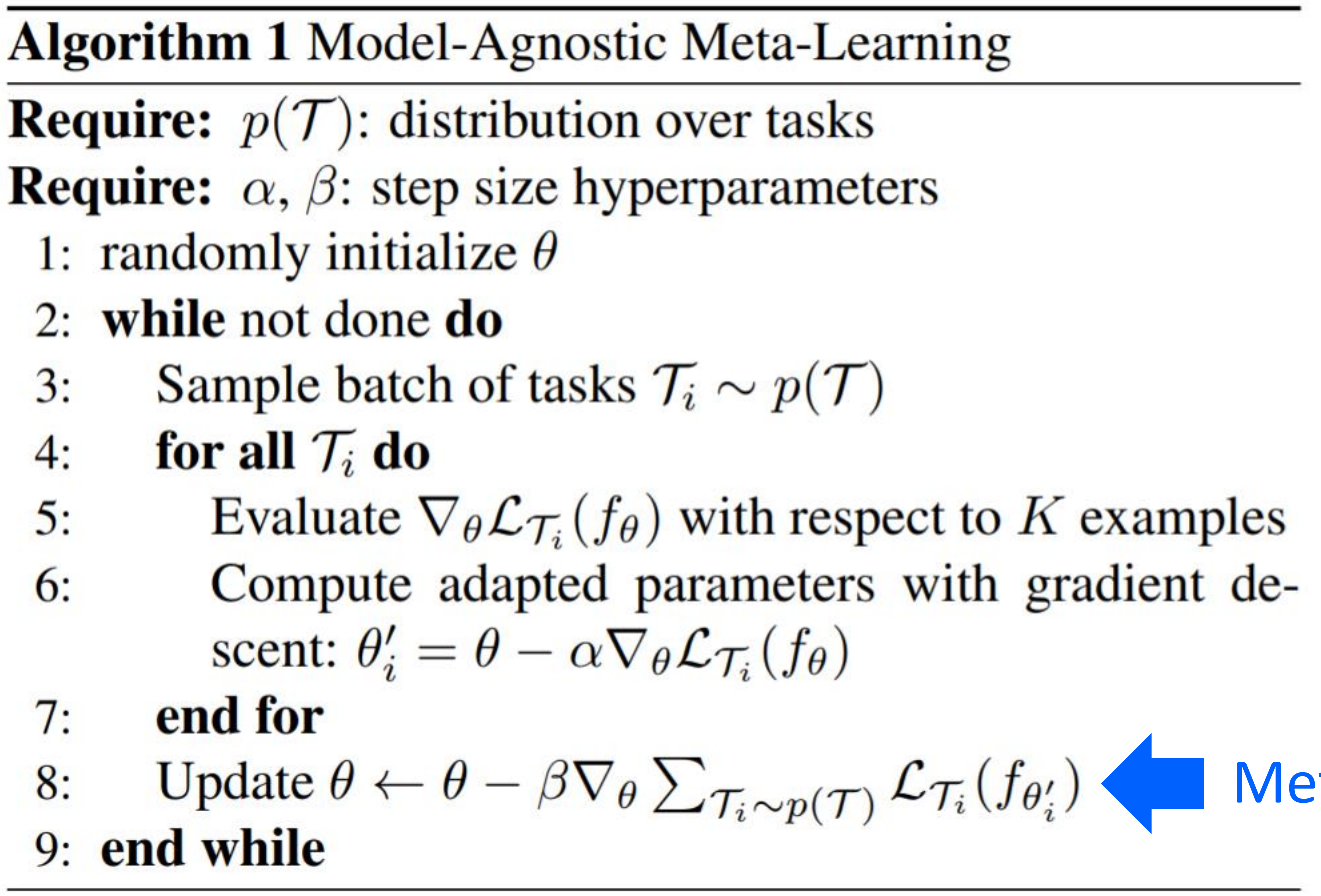


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Method

3. Model-agnostic meta-learning



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
```

Meta-optimization

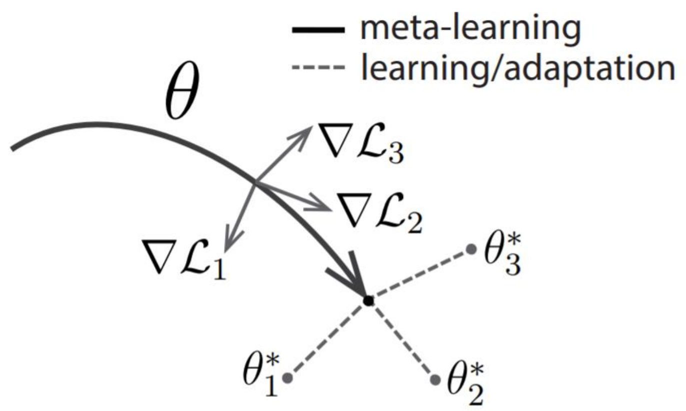
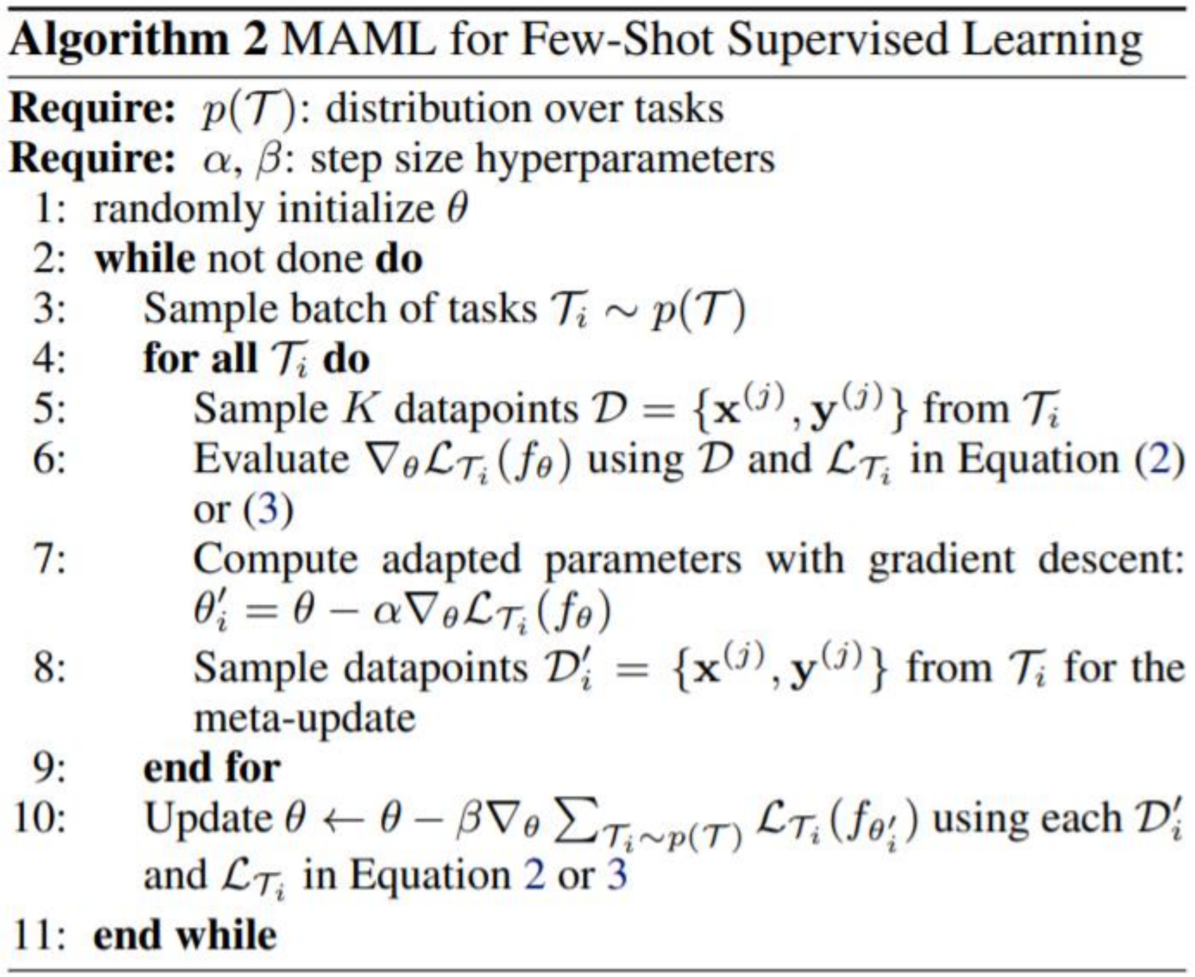


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Species of MAML

1. Supervised "Regression" and "Classification"



Algorithm 2 MAML for Few-Shot Supervised 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 $K$ datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
 6:         Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2)
            or (3)
 7:         Compute adapted parameters with gradient descent:
            $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
 8:         Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the
            meta-update
 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 2 or 3
11: end while
```

Note that

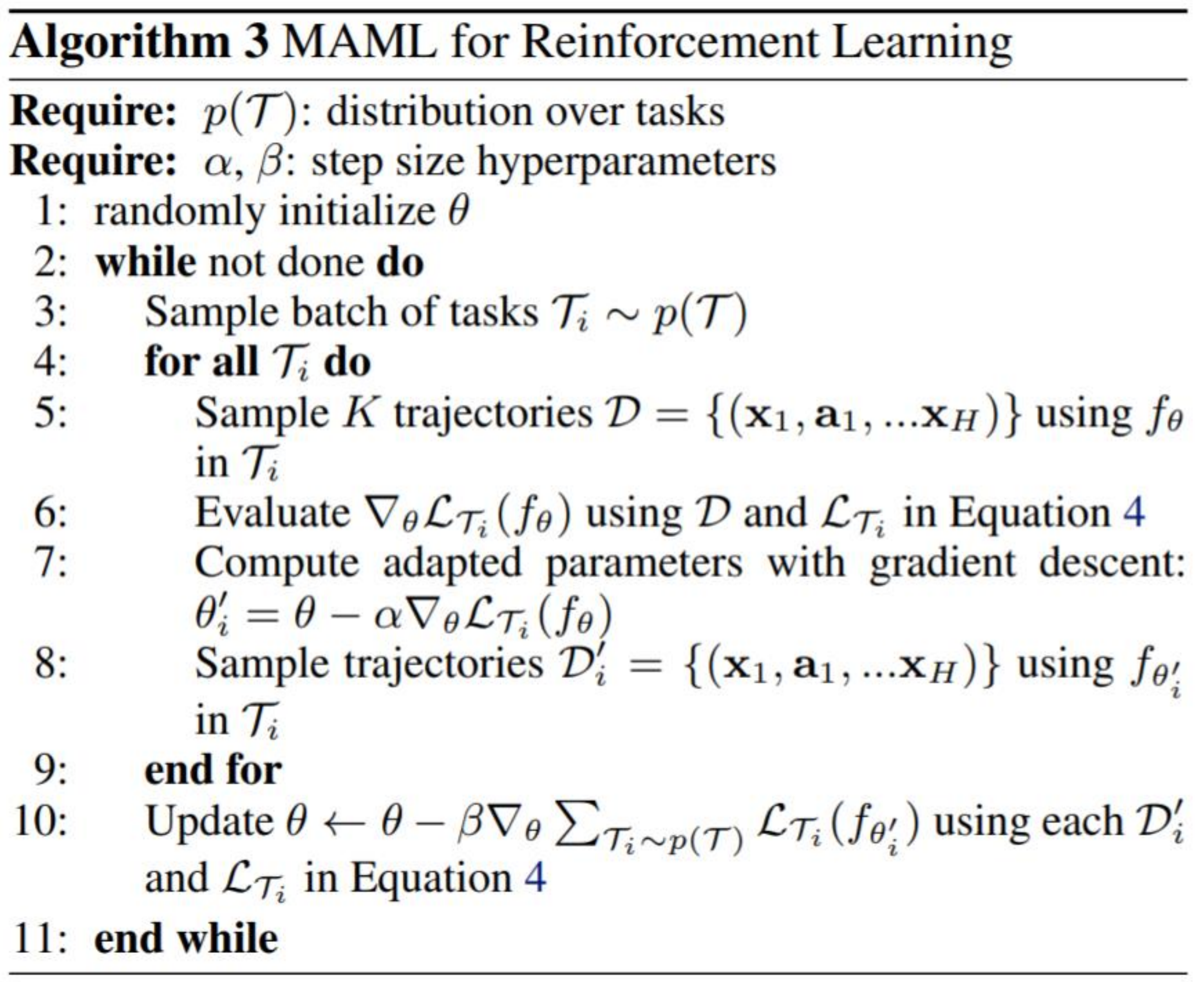
1) $H = 1$

2) Classification → Cross-entropy
   Regression → MSE

3) NK datapoints are needed for K-shot classification and N-way classification

Species of MAML

2. “Reinforcement Learning”



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
```

Note that

1) $H \neq 1$

2) Policy gradient method for not differentiable reward function R

Outline

1. Contribution
2. Method
3. Experiments
4. Conclusion

Experiments

1. To answer the following questions:

(1) Can MAML enable fast learning of new tasks?

(2) Can MAML be used for meta-learning in multiple different domains?

(3) Can a model learned with MAML continue to improve with additional gradient updates and/or examples?

Experiments: Regression

1. Task description

- $p(\mathcal{T})$ is continuous, where the amplitude varies within $[0.1, 0.5]$, phase varies within $[0, \pi]$, and datapoints X are sampled uniformly from $[-5.0, 5.0]$
- Loss function is MSE, regressor is NNs with 2 hidden layers
- Baselines: (a) pretraining on all of the tasks, which entails training a network to regress to random sinusoid functions and then, at test-time, fine-tuning with gradient descent on the K provided points, using an automatically tuned step size, and (b) an oracle

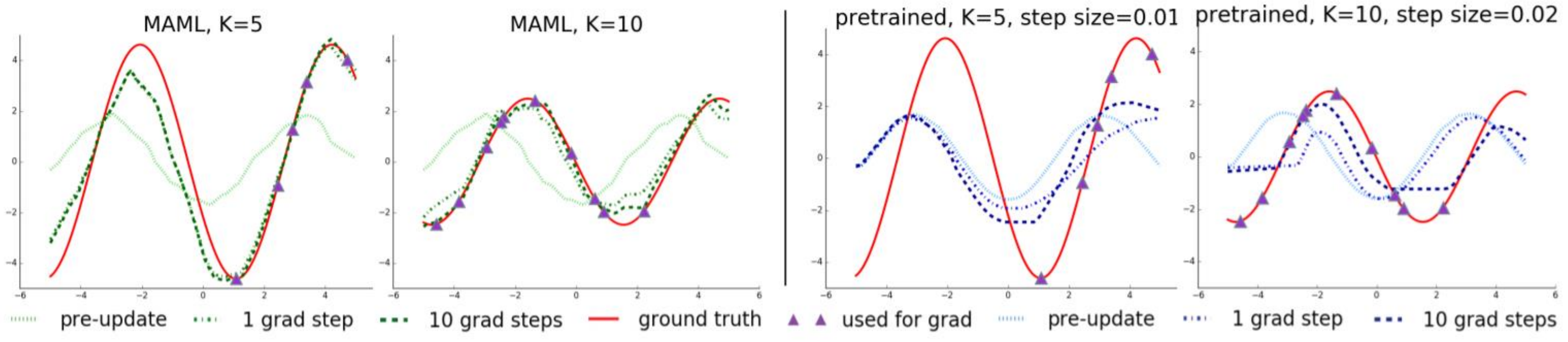


Figure 2. Few-shot adaptation for the simple regression task. Left: Note that MAML is able to estimate parts of the curve where there are no datapoints, indicating that the model has learned about the periodic structure of sine waves. Right: Fine-tuning of a model pretrained on the same distribution of tasks without MAML, with a tuned step size. Due to the often contradictory outputs on the pre-training tasks, this model is unable to recover a suitable representation and fails to extrapolate from the small number of test-time samples.

Experiments: Regression

1. To answer the following questions:

(1) Can MAML enable fast learning of new tasks?

(2) Can MAML be used for meta-learning in multiple different domains?

(3) Can a model learned with MAML continue to improve with additional gradient updates and/or examples?

2. Additional gradient steps

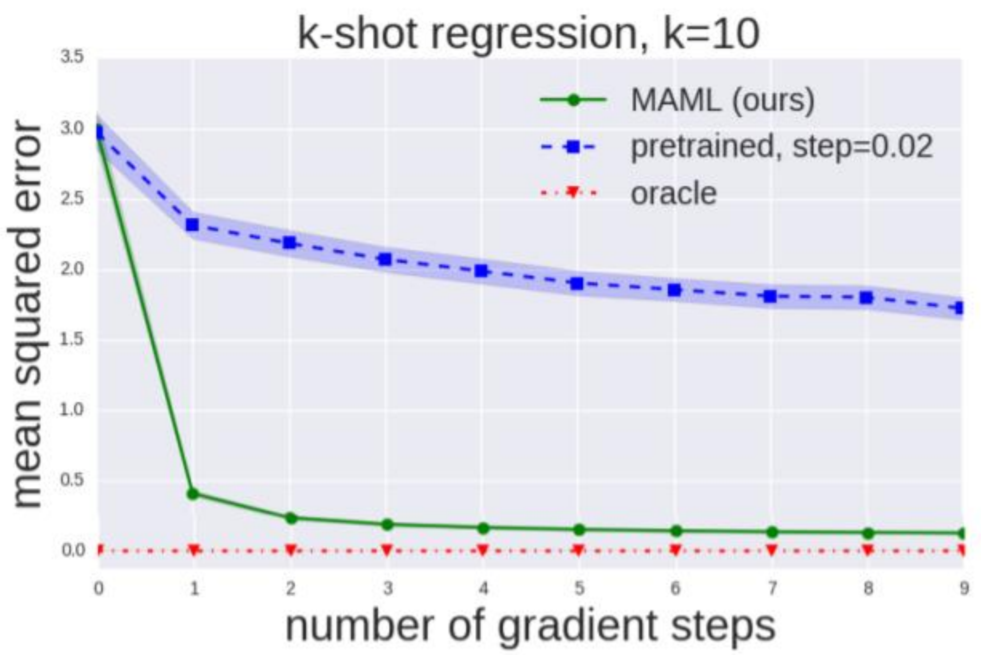
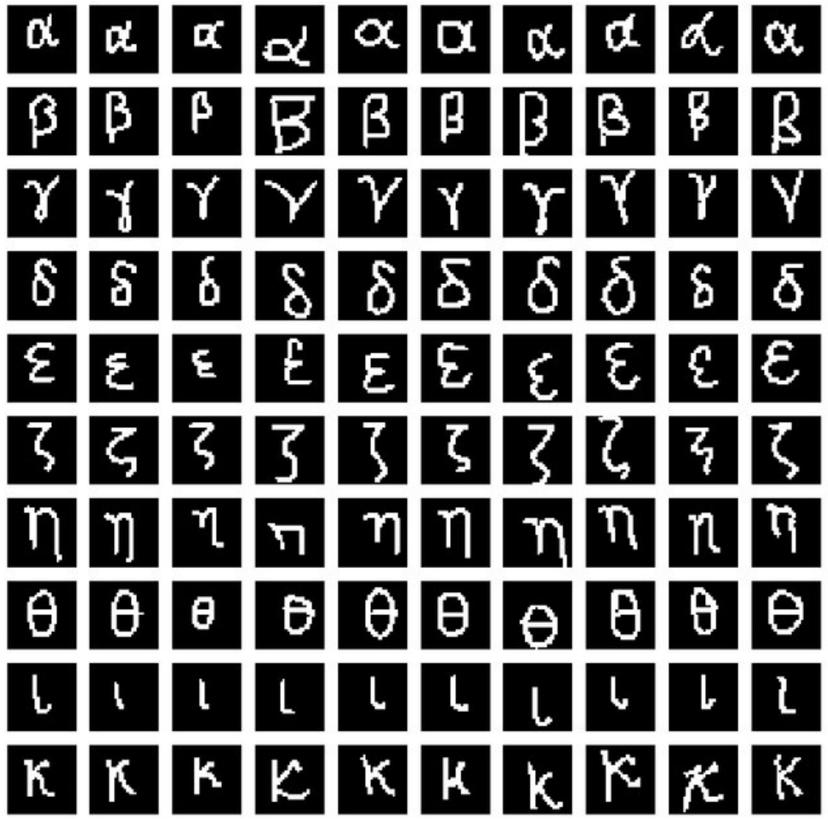


Figure 3. Quantitative sinusoid regression results showing the learning curve at meta test-time. Note that MAML continues to improve with additional gradient steps without overfitting to the extremely small dataset during meta-testing, achieving a loss that is substantially lower than the baseline fine-tuning approach.

Experiments: Classification

1. Task description

- Omniglot: 20 instances of 1623 characters from 50 different alphabets
- MiniImagenet: 64 training classes, 12 validation classes, and 24 test classes



Experiments: Classification

2. Result

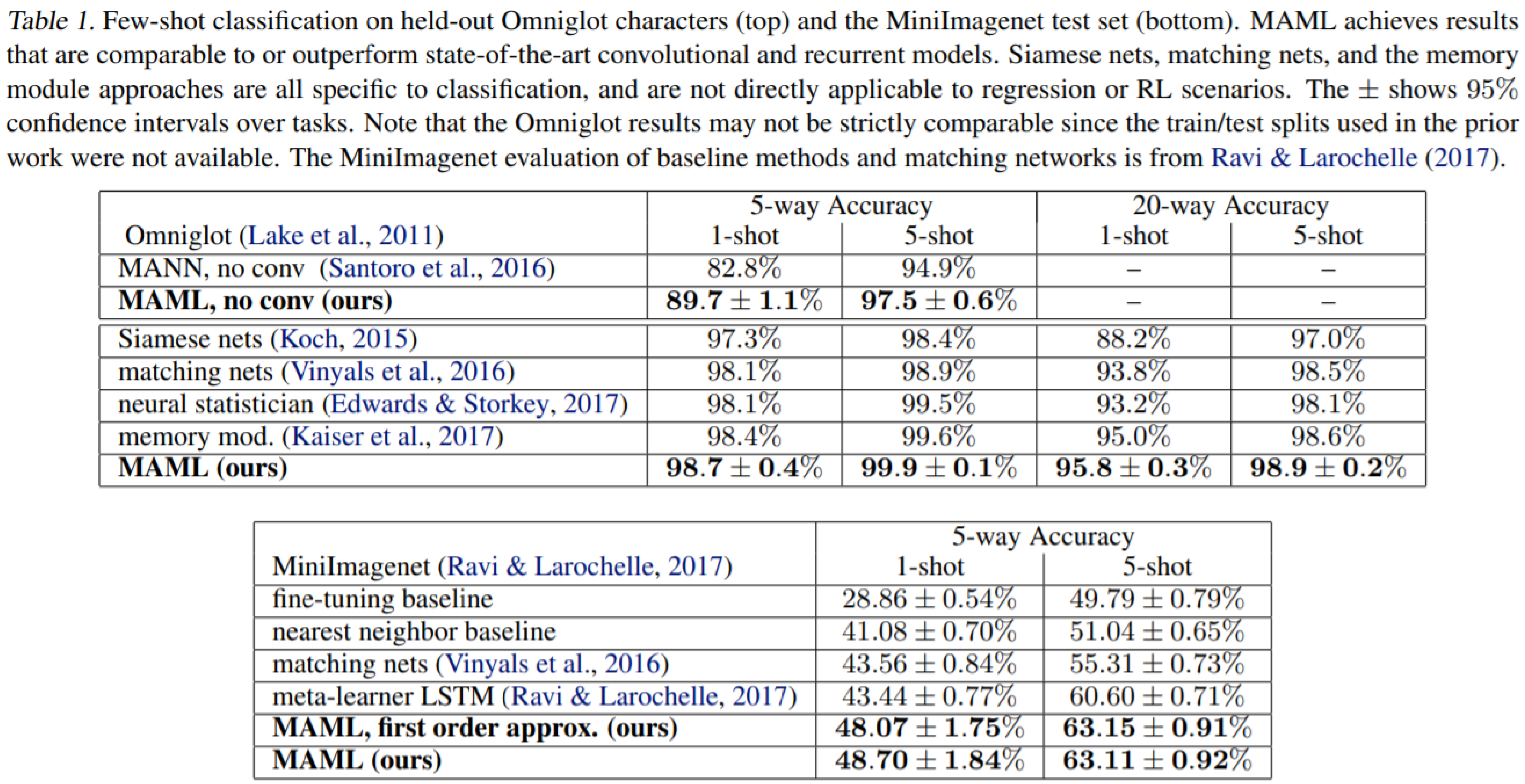


Table 1. Few-shot classification on held-out Omniglot characters (top) and the MiniImagenet test set (bottom). MAML achieves results that are comparable to or outperform state-of-the-art convolutional and recurrent models. Siamese nets, matching nets, and the memory module approaches are all specific to classification, and are not directly applicable to regression or RL scenarios. The $\pm$ shows 95% confidence intervals over tasks. Note that the Omniglot results may not be strictly comparable since the train/test splits used in the prior work were not available. The MiniImagenet evaluation of baseline methods and matching networks is from Ravi & Larochelle (2017).

	5-way Accuracy		20-way Accuracy	
Omniglot (Lake et al., 2011)	1-shot	5-shot	1-shot	5-shot
MANN, no conv (Santoro et al., 2016)	82.8%	94.9%	–	–
MAML, no conv (ours)	**89.7 $\pm$ 1.1%**	**97.5 $\pm$ 0.6%**	–	–
Siamese nets (Koch, 2015)	97.3%	98.4%	88.2%	97.0%
matching nets (Vinyals et al., 2016)	98.1%	98.9%	93.8%	98.5%
neural statistician (Edwards & Storkey, 2017)	98.1%	99.5%	93.2%	98.1%
memory mod. (Kaiser et al., 2017)	98.4%	99.6%	95.0%	98.6%
MAML (ours)	**98.7 $\pm$ 0.4%**	**99.9 $\pm$ 0.1%**	**95.8 $\pm$ 0.3%**	**98.9 $\pm$ 0.2%**

	5-way Accuracy	
MiniImagenet (Ravi & Larochelle, 2017)	1-shot	5-shot
fine-tuning baseline	28.86 $\pm$ 0.54%	49.79 $\pm$ 0.79%
nearest neighbor baseline	41.08 $\pm$ 0.70%	51.04 $\pm$ 0.65%
matching nets (Vinyals et al., 2016)	43.56 $\pm$ 0.84%	55.31 $\pm$ 0.73%
meta-learner LSTM (Ravi & Larochelle, 2017)	43.44 $\pm$ 0.77%	60.60 $\pm$ 0.71%
MAML, first order approx. (ours)	**48.07 $\pm$ 1.75%**	**63.15 $\pm$ 0.91%**
MAML (ours)	**48.70 $\pm$ 1.84%**	**63.11 $\pm$ 0.92%**

Experiments: Reinforcement Learning

1. Task Description: rllab benchmark suite

(1) 2D Navigation

(2) Locomotion (MuJoCo Simulator)

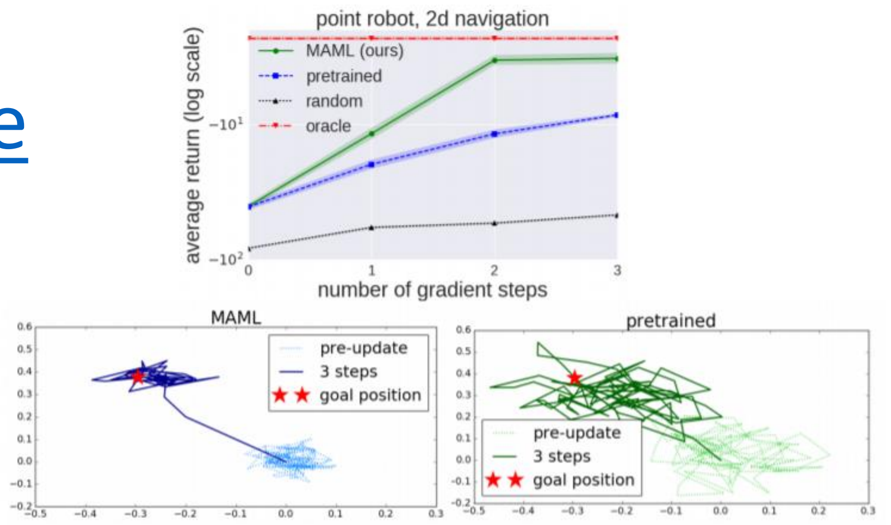


Figure 4. Top: quantitative results from 2D navigation task, Bottom: qualitative comparison between model learned with MAML and with fine-tuning from a pretrained network.

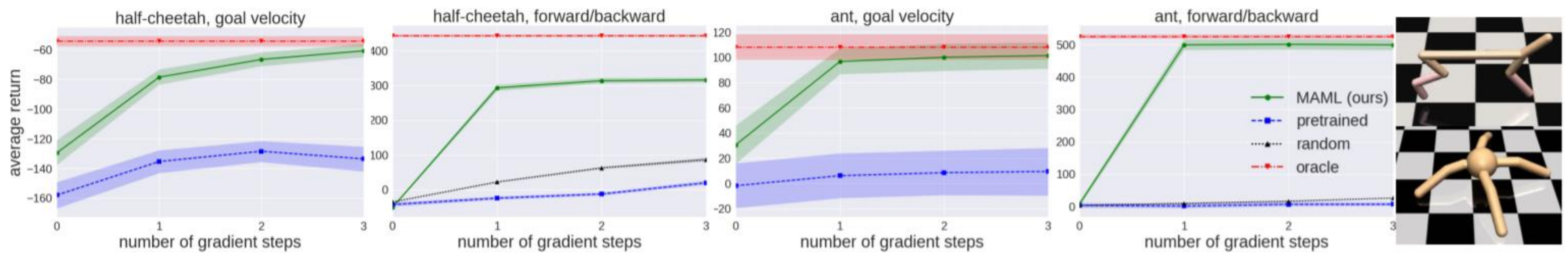
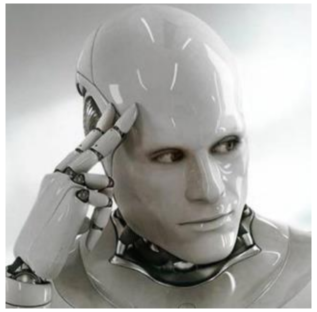


Figure 5. Reinforcement learning results for the half-cheetah and ant locomotion tasks, with the tasks shown on the far right. Each gradient step requires additional samples from the environment, unlike the supervised learning tasks. The results show that MAML can adapt to new goal velocities and directions 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).

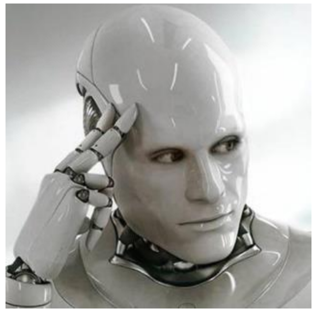
Outline

Conclusion

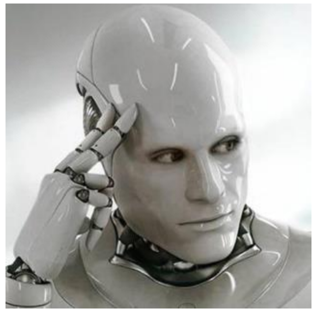
- Application: Evaluation metric for open domain dialog



Dataset A



Dataset B



Dataset C

Thank you

https://jeiyoon.github.io/

TAVE Research