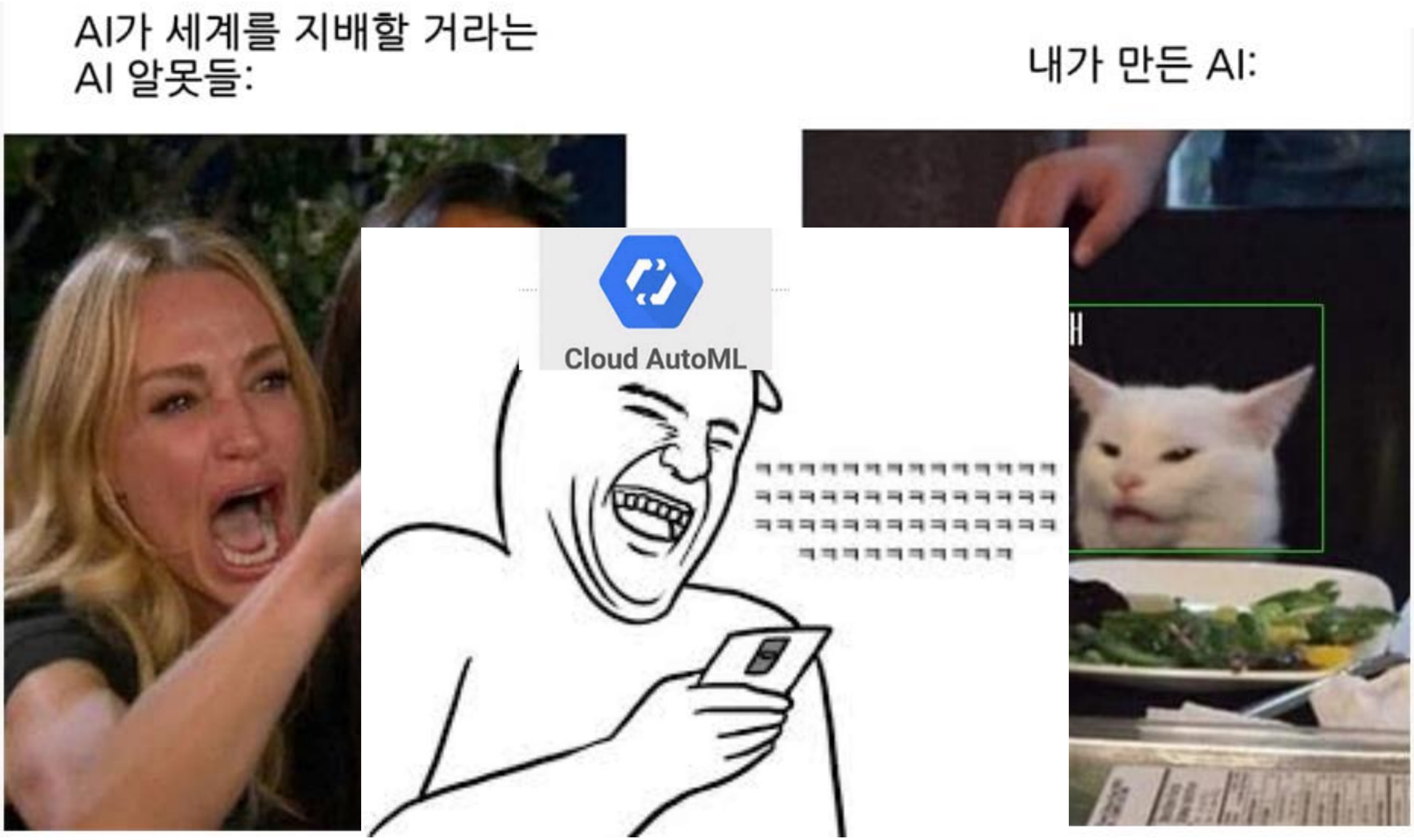
Recent Trends of Meta Learning

Kevin Jeong
DGU AI lab.
chjeong@dongguk.edu

AI가 세계를 지배할 거라는
AI 알못들:

AI가 세계를 지배할 거라는
AI 알못들:

내가 만든 AI:


AI가 세계를 지배할 거라는
AI 알못들:
내가 만든 AI:
Cloud AutoML
ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ
ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ
ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ
ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ

What is the Meta Learning?

Why is it fancy?

Research Category

- Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning

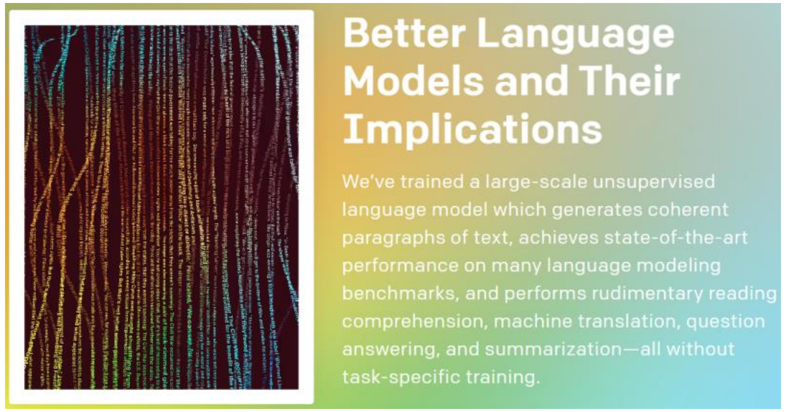
- Meta Learning
  - Meta Supervised Learning
  - Meta Reinforcement Learning

  - What about Unsupervised Meta Learning?
    - Not sure, Seems that not much research yet
    - But it is very important!

What is Meta Learning?

Large, diverse data → Broad Generalization

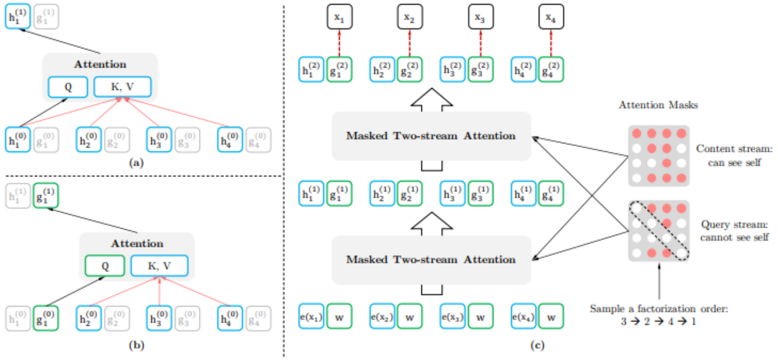
Russakovsky et al. '14



GPT-2

Radford et al. '19

XLNet

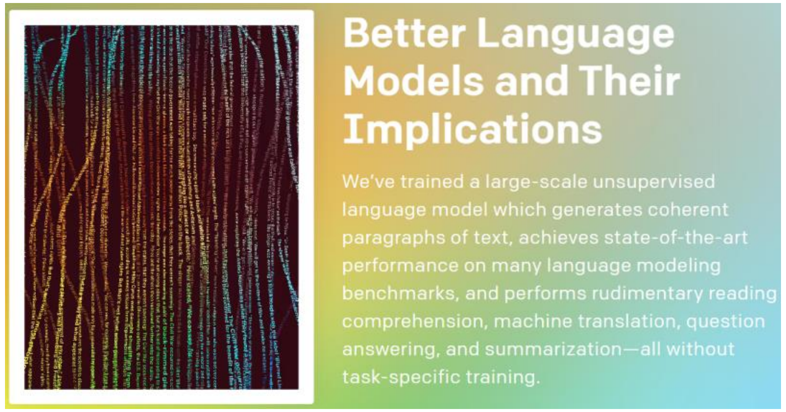


Yang et al. '19

What is Meta Learning?

Large, diverse data → Broad Generalization

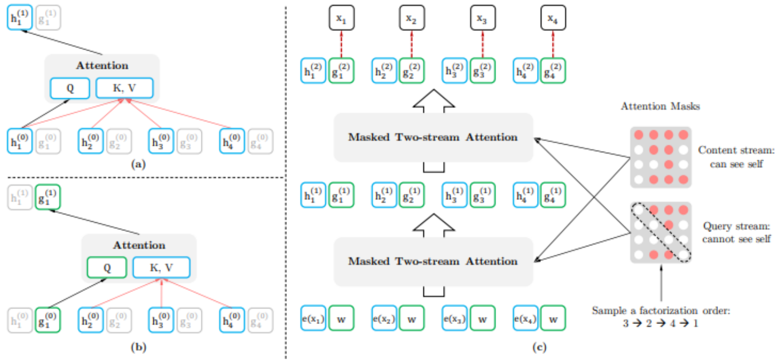
Russakovsky et al. '14



GPT-2

Radford et al. '19

XLNet



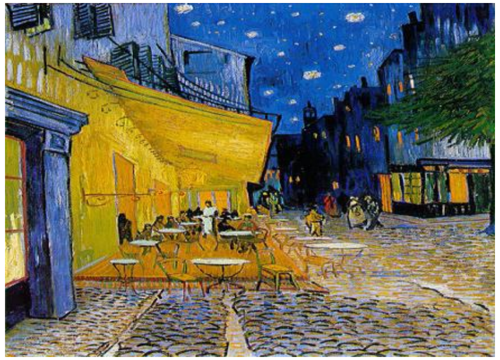
Yang et al. '19

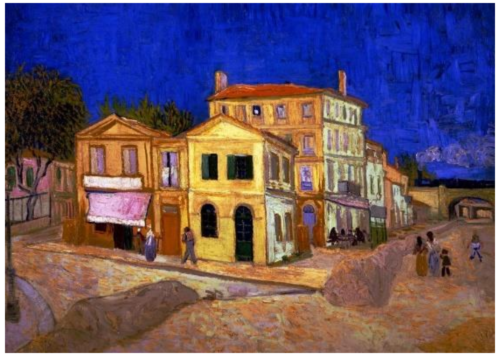
What if you don't have a large dataset?

What if you want a general-purpose AI system in the real world?

Training data

Van Gogh





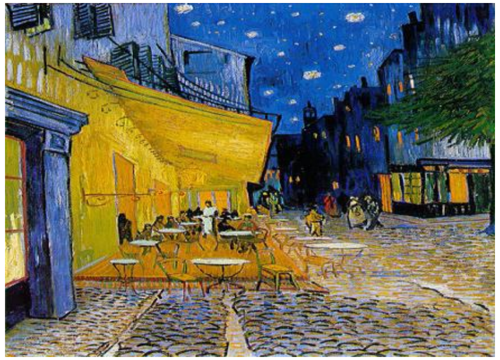
Paul Cezanne

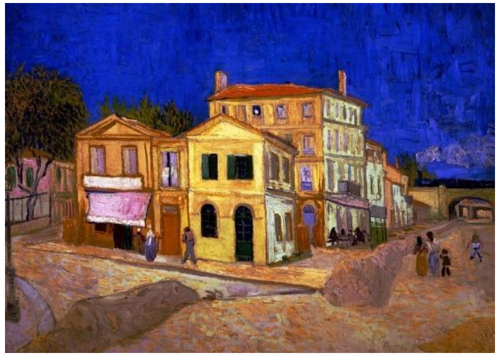
Training data

Test datapoint

Van Gogh

Paul Cezanne





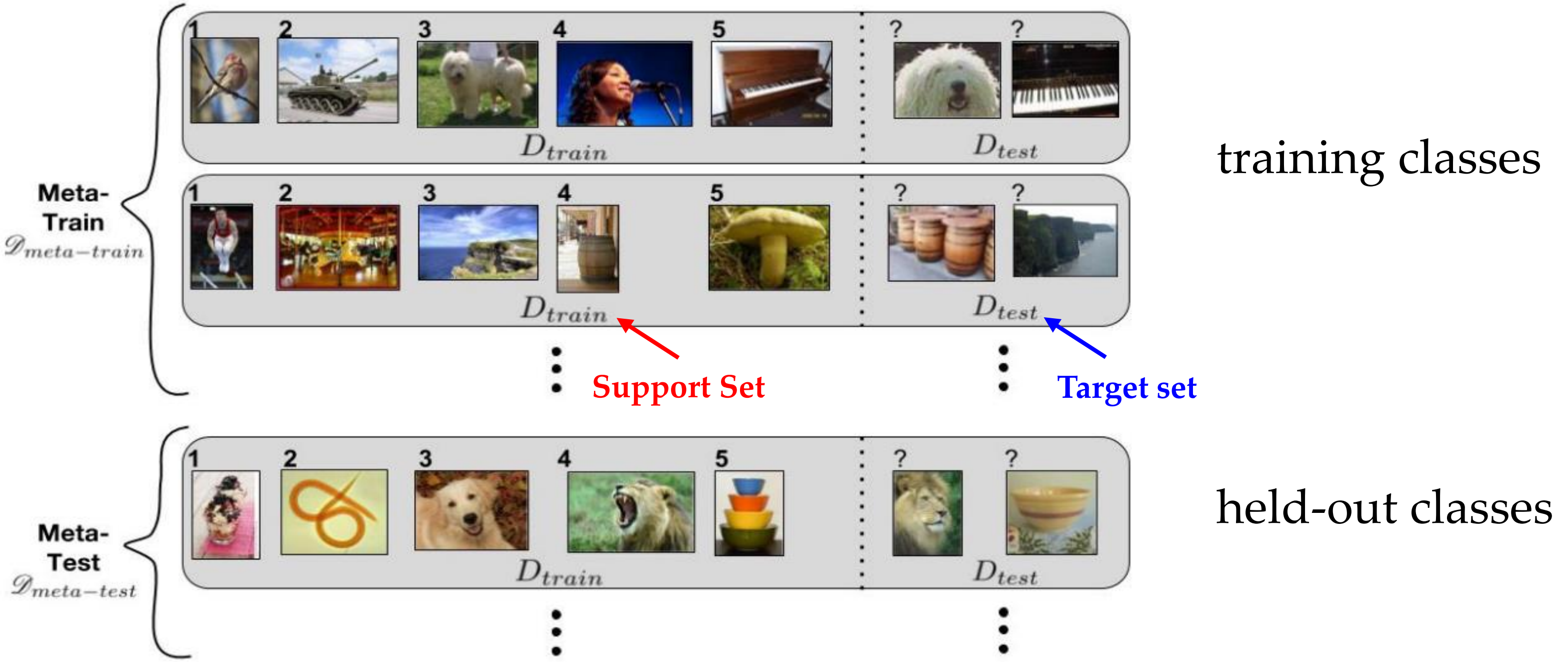
By Gogh or Cezanne?

→ Few-Shot Learning

Chelsea Finn, Sergey Levine. Meta Learning Tutorials, ICML 2019

Training Setup

- Meta Learning : Maximum Likelihood Estimation

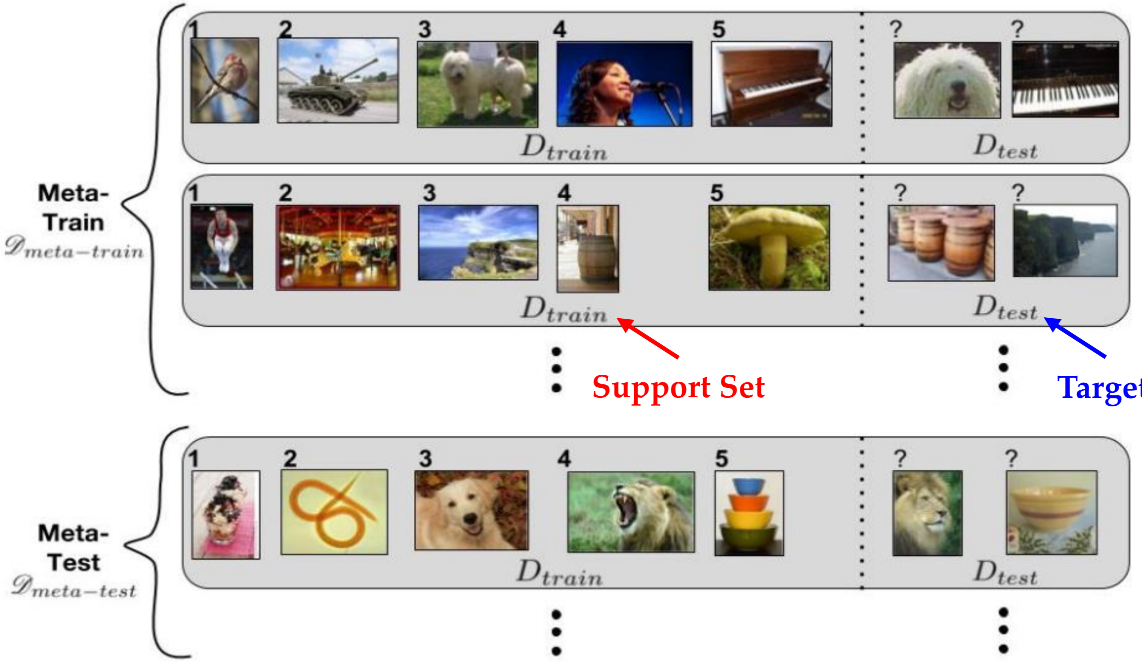


$$\theta = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{L\sim T}\left[\mathbb{E}_{S\sim L,\, B\sim ,L}\left[\sum_{(x,y)\in B} \log P_\theta(y|x,S)\right]\right]$$

Sachin Ravi and Hugo Larochelle, Optimization as a model for few-shot learning, ICLR 2017

Training Setup

- Meta Learning : Maximum Likelihood Estimation



training classes

held-out classes

$$\theta = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{L\sim T}\left[\mathbb{E}_{S\sim L,\, B\sim L}\left[\sum_{(x,y)\in B} \log P_\theta(y|x,S)\right]\right]$$

How to Train?
→ Algorithms

- Black-box adaptation
- Non-parametric methods
- Optimization-based Inference
- Bayesian methods
- …

Sachin Ravi and Hugo Larochelle, Optimization as a model for few-shot learning, ICLR 2017

Meta Learning

- Few-Shot Learning Perspective
  - Black-Box Adaptation(Model-based Approach)
    → Using Internal Memory like RNN(or External Memory) : SL(O), RL(O)
  - Non-Parametric Method(Metric-based Approach)
    → Learn the metric manifold space(kernel function) : SL(O), RL(X)
  - Optimization-based Inference
    → Optimization of the model parameters : SL(O), RL(O)

Black-Box Adaptation(Model-based Approach)

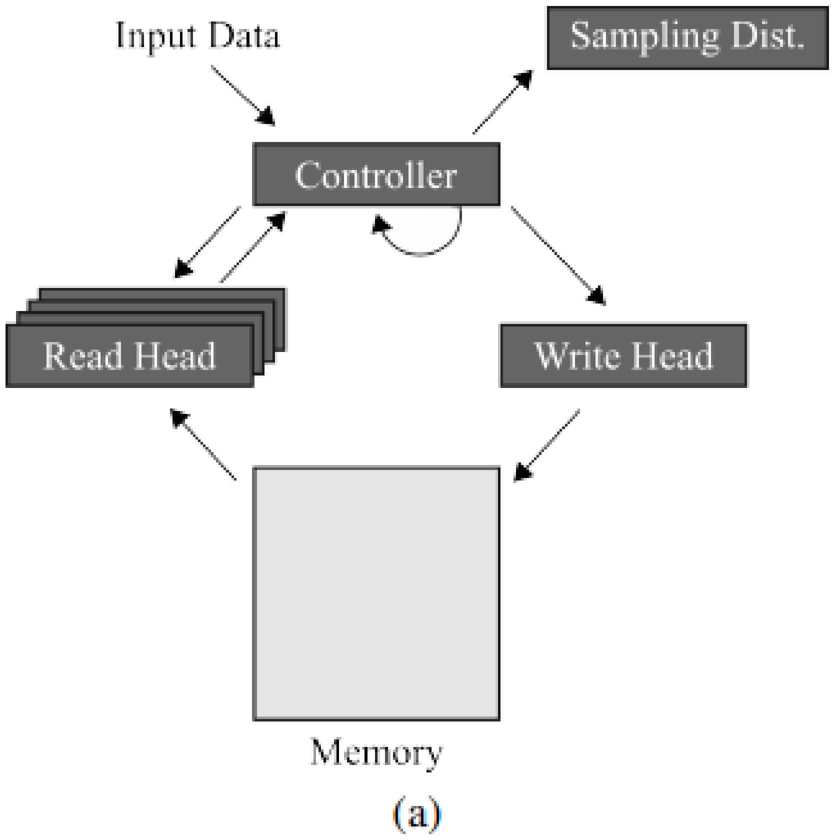
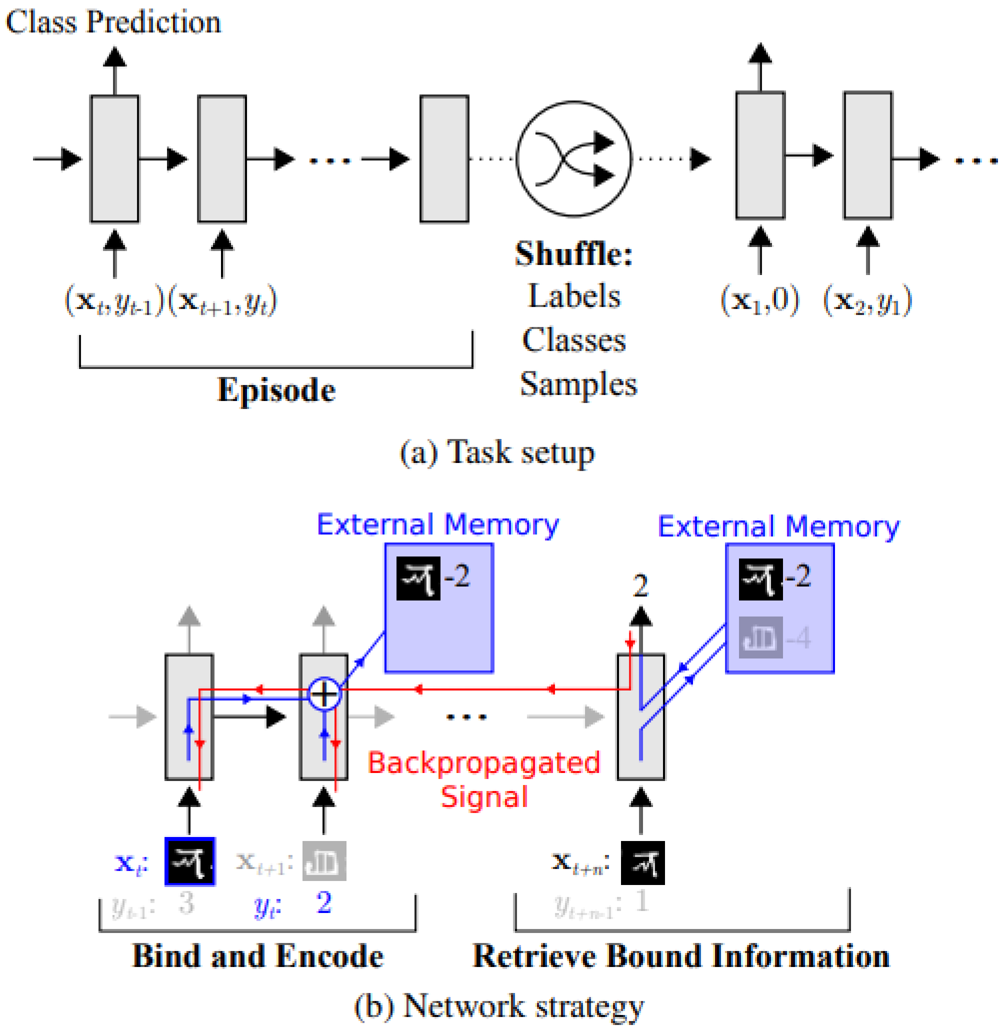


Figure 7. MANN Architecture.



Santoro, Adam, et al. "Meta-learning with memory-augmented neural networks." ICML 2016.

Non-Parametric Method(Metric-based Approach)

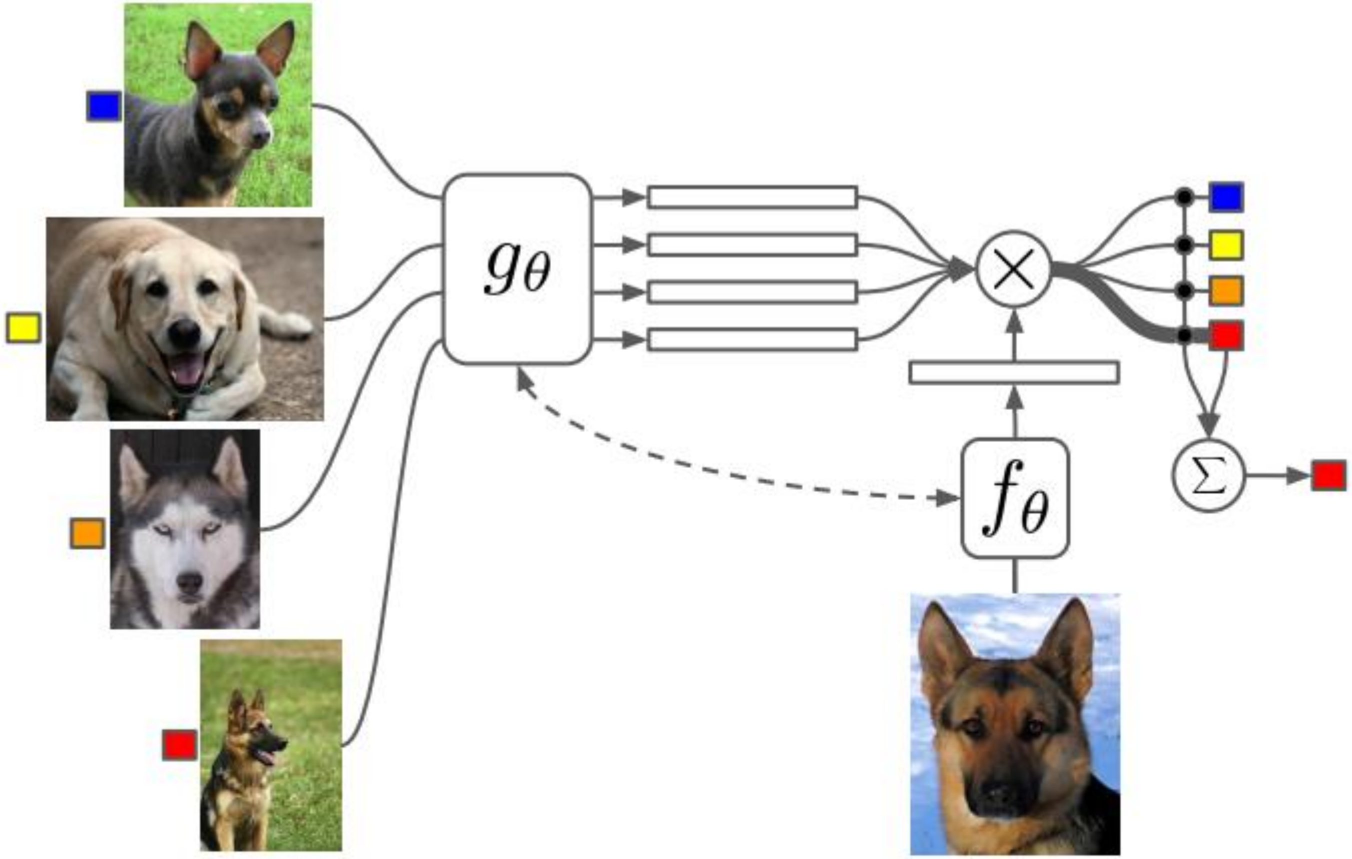
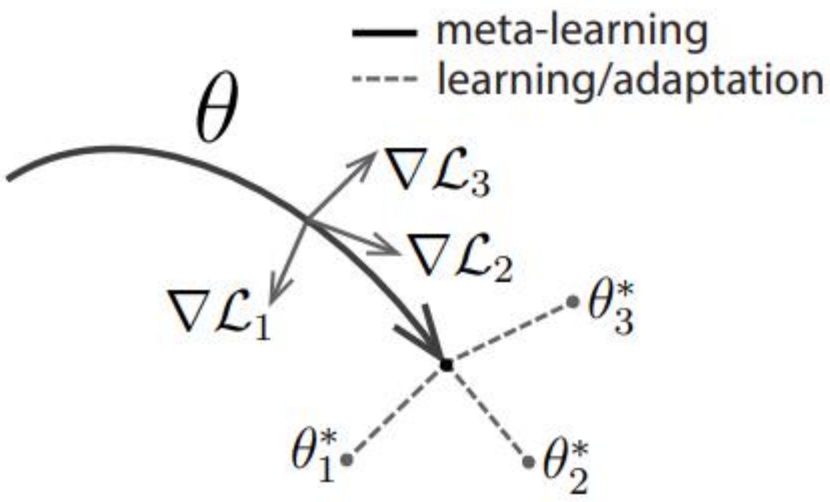


Figure 1: Matching Networks architecture

Vinyals, Oriol, et al. "Matching networks for one shot learning." NIPS 2016.

Optimization-based Inference



Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

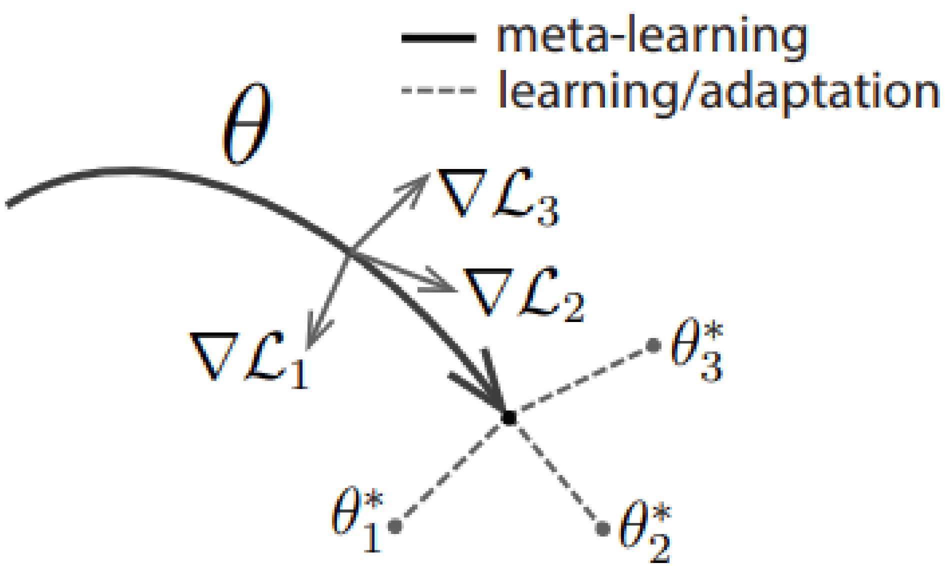
randomly initialize θ
while not done **do**
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ **do**
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
        Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
end while

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." ICML 2017

Model-Agnostic Meta Learning

Optimization-based Inference

Meta Supervised Learning



$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\mathbf{x}^{(j)}, \mathbf{y}^{(j)} \sim \mathcal{T}_i} \mathbf{y}^{(j)} \log f_\phi(\mathbf{x}^{(j)}) + (1 - \mathbf{y}^{(j)}) \log(1 - f_\phi(\mathbf{x}^{(j)}))$$

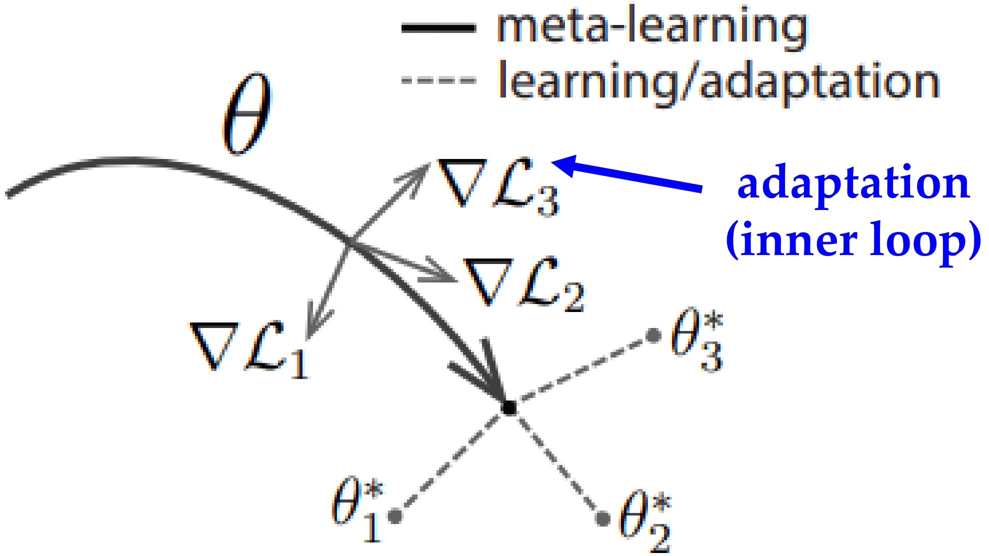
Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

```
randomly initialize θ
while not done do
    Sample batch of tasks T_i ∼ p(T)
    for all T_i do
        Sample K datapoints D = {x^(j), y^(j)} from T_i
        Evaluate ∇_θ L_{T_i}(f_θ) using D and L_{T_i} in Equation (2) or (3)
        Compute adapted parameters with gradient descent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
        Sample datapoints D'_i = {x^(j), y^(j)} from T_i for the meta-update
    end for
    Update θ ← θ − β∇_θ Σ_{T_i∼p(T)} L_{T_i}(f_{θ'_i}) using each D'_i and L_{T_i} in Equation 2 or 3
end while
```

Model-Agnostic Meta Learning

- Optimization-based Inference
  - Meta Supervised Learning



$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\mathbf{x}^{(j)},\mathbf{y}^{(j)}\sim\mathcal{T}_i} \mathbf{y}^{(j)} \log f_\phi(\mathbf{x}^{(j)}) + (1-\mathbf{y}^{(j)}) \log(1 - f_\phi(\mathbf{x}^{(j)}))$$

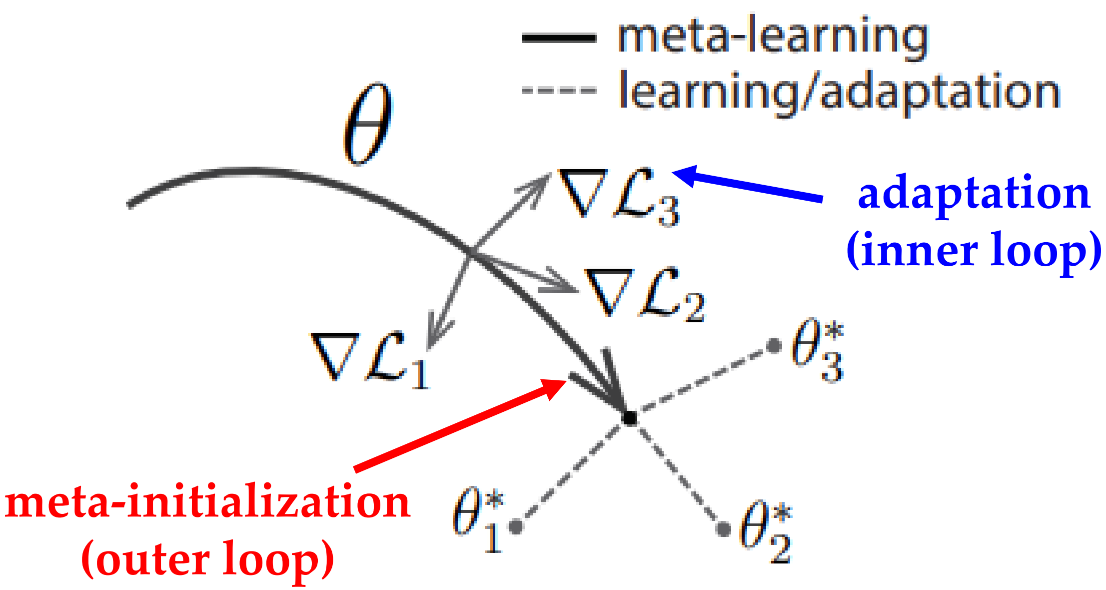
Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha\nabla_\theta\mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
  end for
  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
end while

Model-Agnostic Meta Learning

- Optimization-based Inference
  - Meta Supervised Learning



$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{\mathbf{x}^{(j)},\mathbf{y}^{(j)}\sim\mathcal{T}_i} \mathbf{y}^{(j)} \log f_\phi(\mathbf{x}^{(j)}) + (1-\mathbf{y}^{(j)}) \log(1 - f_\phi(\mathbf{x}^{(j)}))$$

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
  end for
  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
end while

Chelsea Finn, Pieter Abbeel, Sergey Levine, Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, ICML 2017

Meta Reinforcement Learning

- Meta Learning
  - Generic Learning

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta, \mathcal{D}^{train})$$
$$= f_{learn}(\mathcal{D}^{train})$$

  - Generic Meta Learning

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \mathcal{L}(\phi_i, \mathcal{D}_i^{test})$$

$$\text{where } \phi_i = f_\theta(\mathcal{D}_i^{test})$$

Meta Reinforcement Learning

- Meta Learning
  - Generic Learning

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \mathcal{L}(\theta, \mathcal{D}^{train})$$
$$= f_{learn}(\mathcal{D}^{train})$$

  - Generic Meta Learning

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \mathcal{L}(\phi_i, \mathcal{D}_i^{test})$$

$$\text{where } \phi_i = f_\theta(\mathcal{D}_i^{test})$$

- Meta Reinforcement Learning
  - Reinforcement Learning

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{\pi_\theta(\tau)}[R(\tau)]$$
$$= f_{RL}(\mathcal{M})$$
$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, r\}$$

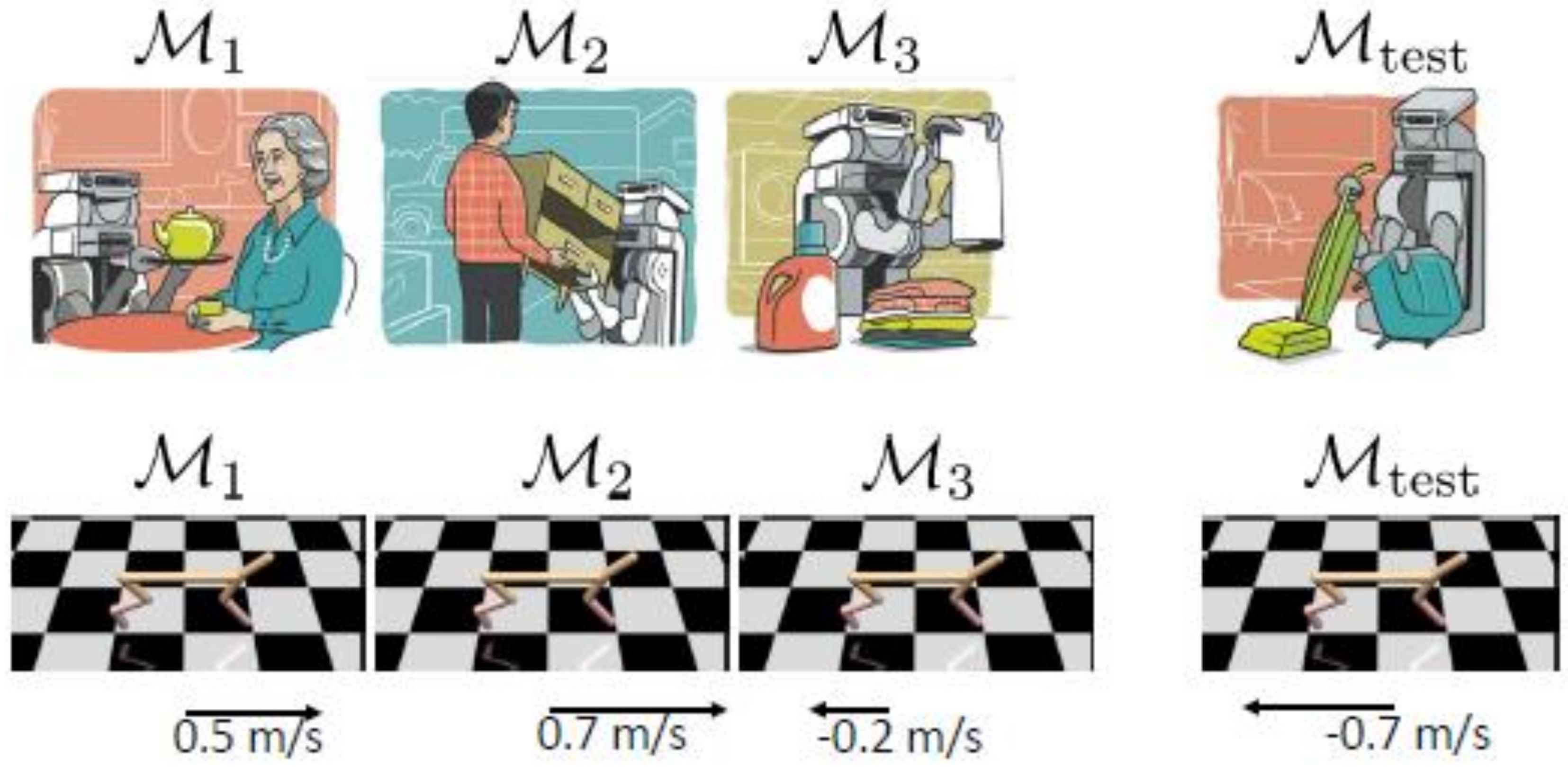
  - Meta Reinforcement Learning

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{n} \mathbb{E}_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

$$\text{where } \phi_i = f_\theta(\mathcal{M}_i)$$

Meta Reinforcement Learning

- What's the mathematics formulations meaning?



Meta Reinforcement Learning

- Meta-RL with recurrent policies(Black-box adaptation)

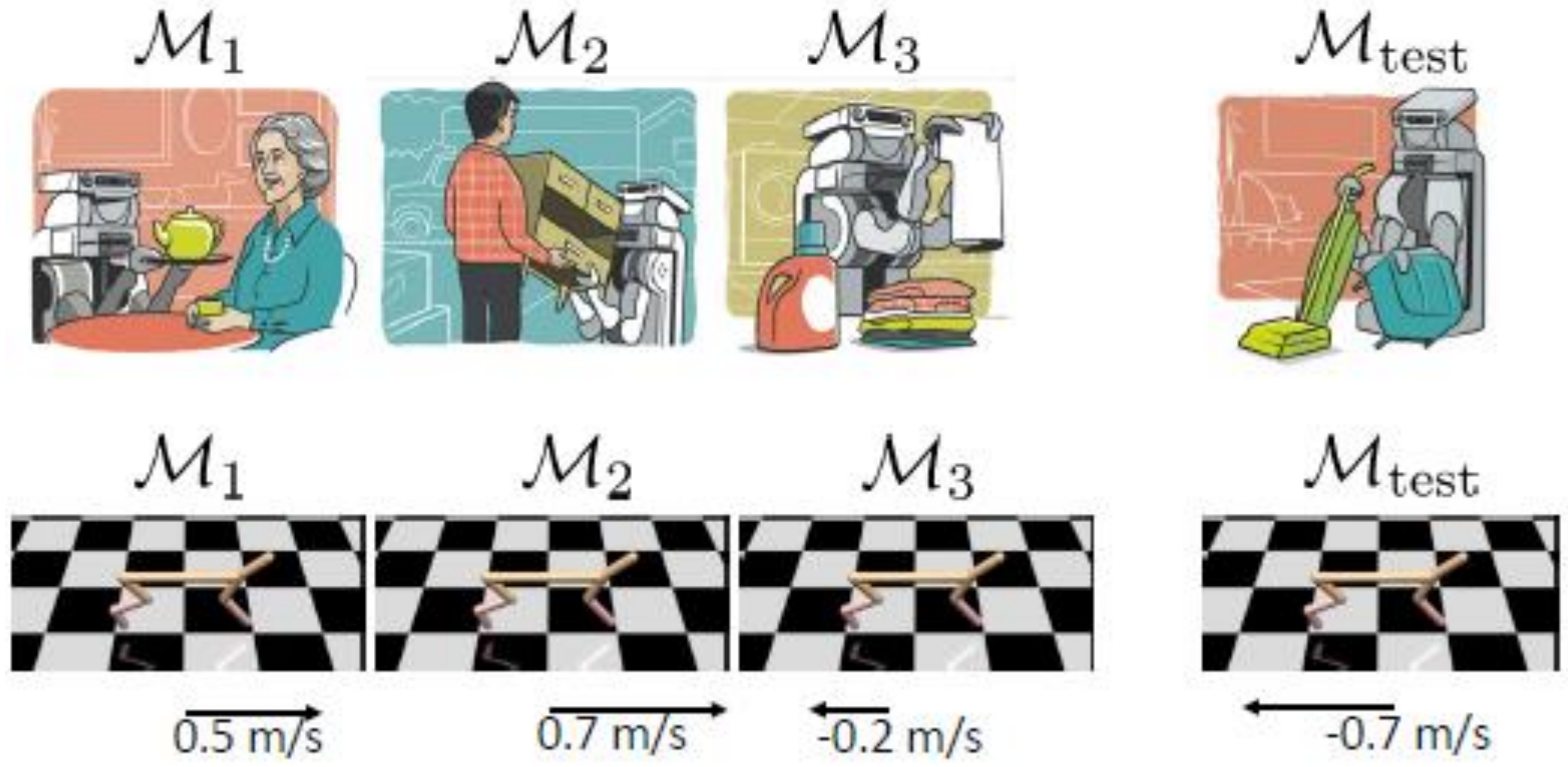
- Meta-RL as an optimization problem → MAML

- Meta-RL as partially observed RL(POMDP)
  - Variational inference for meta-RL
  - Specific instantiation : PEARL

- Challenge of meta learning(★★★★★)
  - Meta-Overfitting → One of strategies : Unsupervised meta learning
  - Memorization → One of strategies : Online meta learning

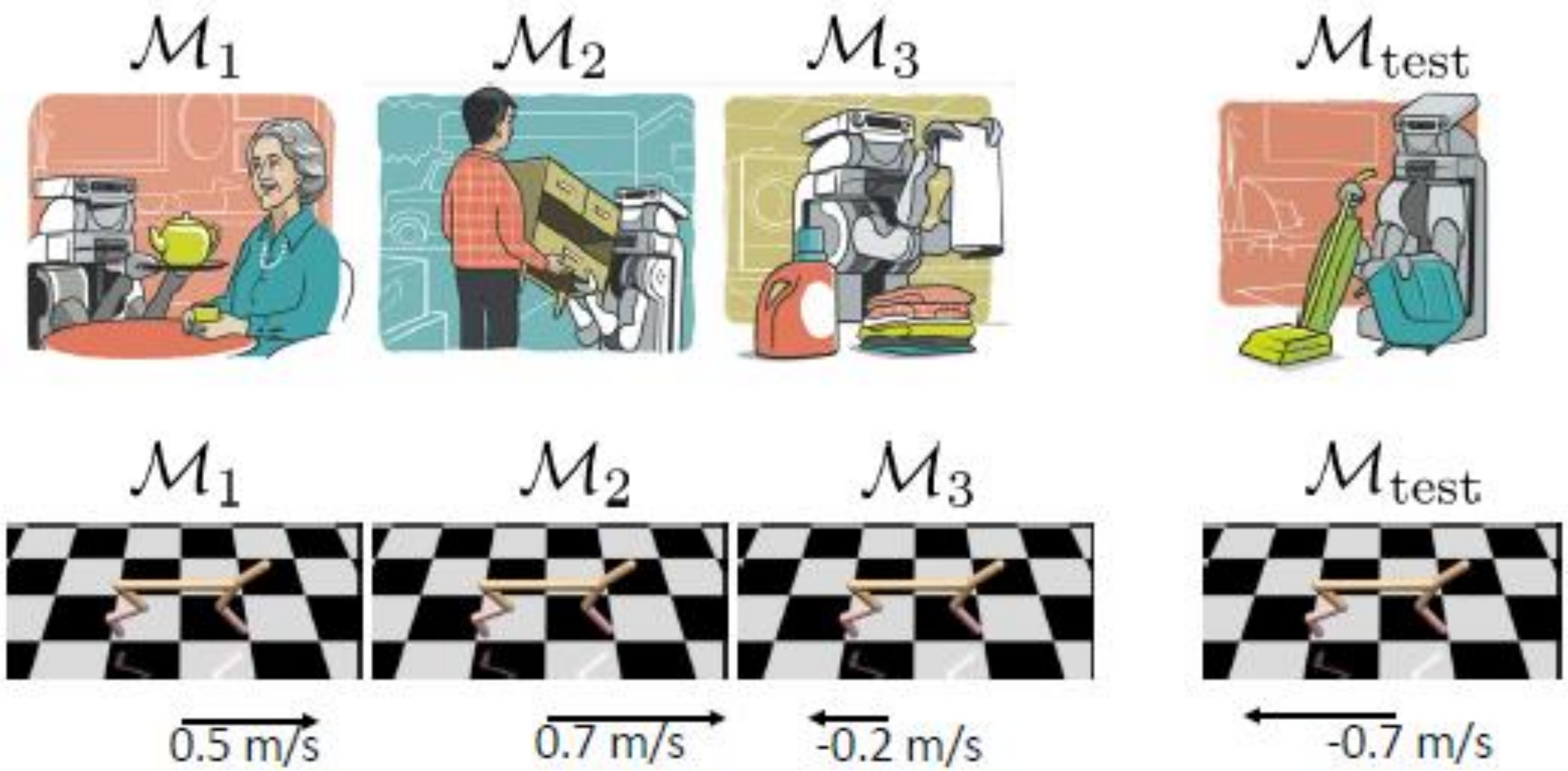
Challenge 1 : Meta Overfitting

- Are these really various tasks?



Challenge 1 : Meta Overfitting

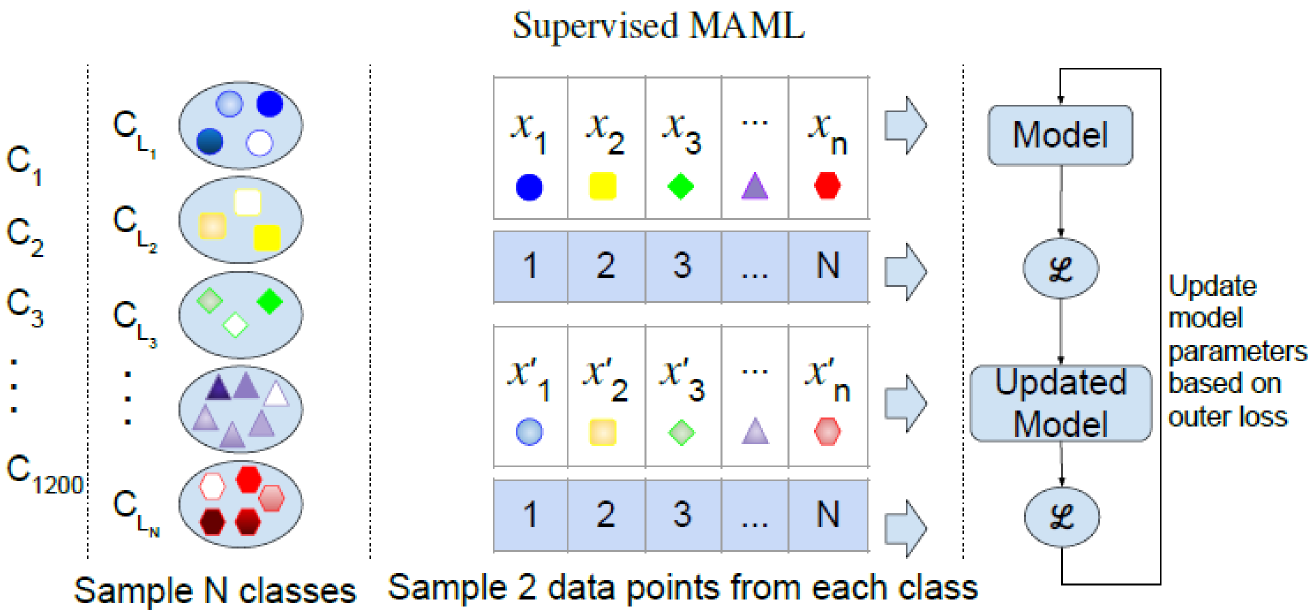
- Are these really various tasks?



We need Unsupervised Meta Learning!
→ Using Self-supervision, Active Learning etc.

Challenge 1 : Meta Overfitting

- Are these really various tasks?



Khodadadeh, Siavash, Ladislau Bölöni, and Mubarak Shah. "Unsupervised Meta-Learning For Few-Shot Image and Video Classification." arXiv preprint arXiv:1811.11819 (2018).

Challenge 1 : Meta Overfitting

- Are these really various tasks?

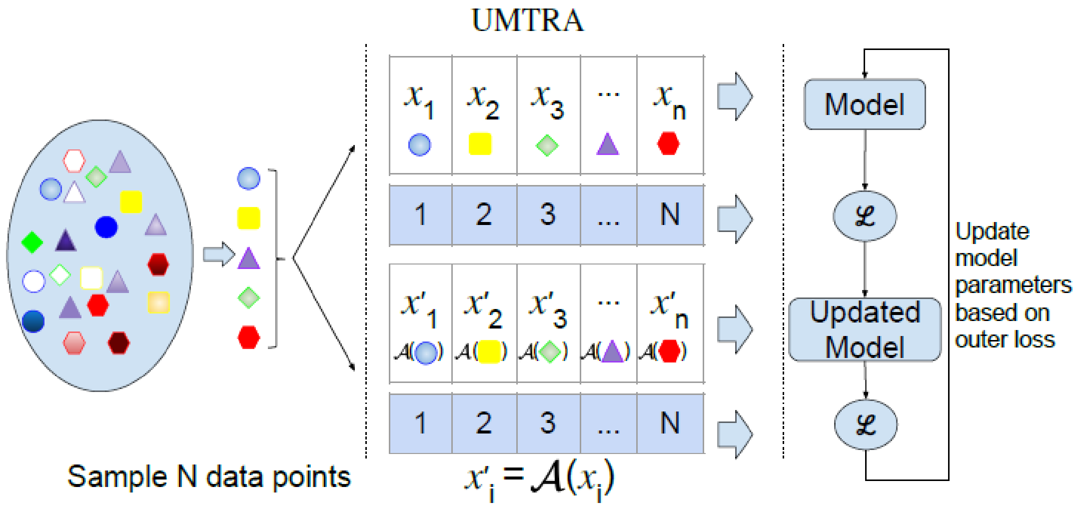
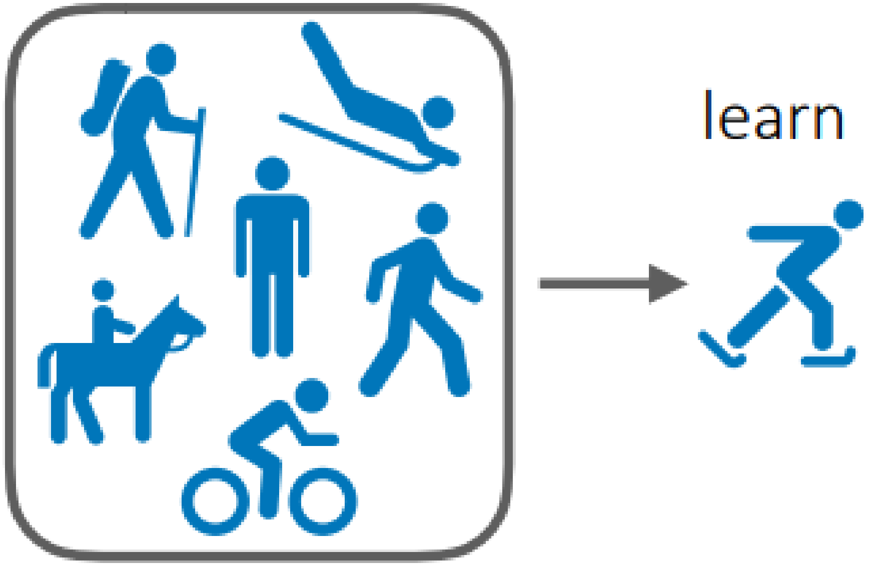


Figure 1: The process of creation of the training and validation data of the meta-training task $\mathcal{T}$. (top) Supervised MAML: We start from a dataset where the samples are labeled with their class. The training data is created by sampling N distinct classes C_{L_i}, and choosing a random sample x_i from each. The validation data is created by choosing a different sample x'_i from the same class. (bottom) UMTRA: We start from a dataset of unlabeled data. The training data is created by randomly choosing N samples x_i from the dataset. The validation data is created by applying the augmentation function $\mathcal{A}$ to each sample from the training data. For both MAML and UMTRA, artificial temporary labels $1, 2 \ldots N$ are used.

Khodadadeh, Siavash, Ladislau Bölöni, and Mubarak Shah. "Unsupervised Meta-Learning For Few-Shot Image and Video Classification." arXiv preprint arXiv:1811.11819 (2018).

Challenge 2 : Memorization

- Do persons or animals really learn that way?

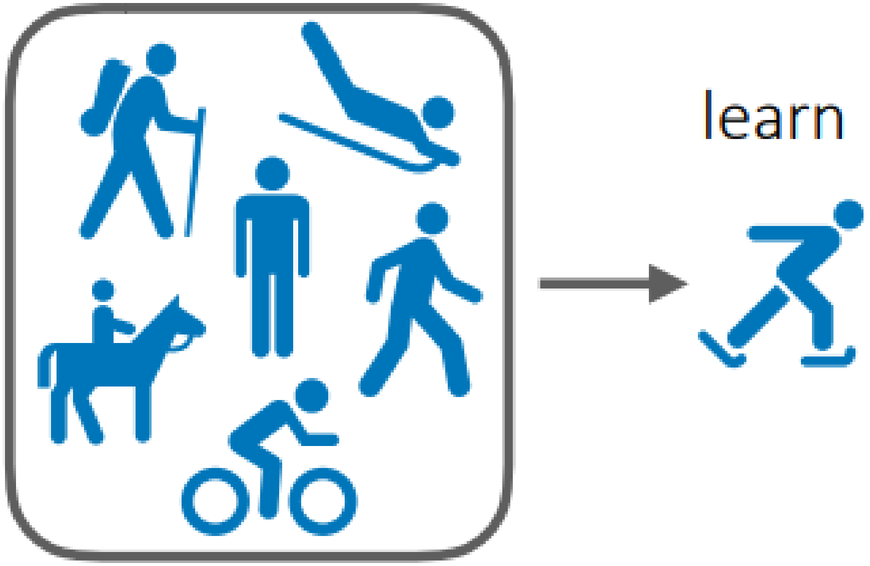


Meta Learning

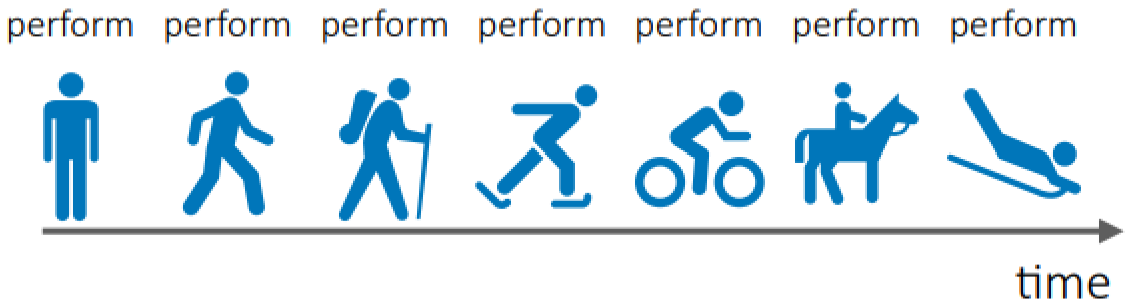
Finn, Chelsea, et al. "Online meta-learning." *arXiv preprint arXiv:1902.08438* (2019).

Challenge 2 : Memorization

- Do persons or animals really learn that way?



Meta Learning



Online Meta Learning

Finn, Chelsea, et al. "Online meta-learning." *arXiv preprint arXiv:1902.08438* (2019).

More Advanced Research Studies for Strong AI?

To be continue…!


Thank you!
Any Question?