
MAML is a Noisy Contrastive Learner

NewInML Workshop@NeuroIPS'21

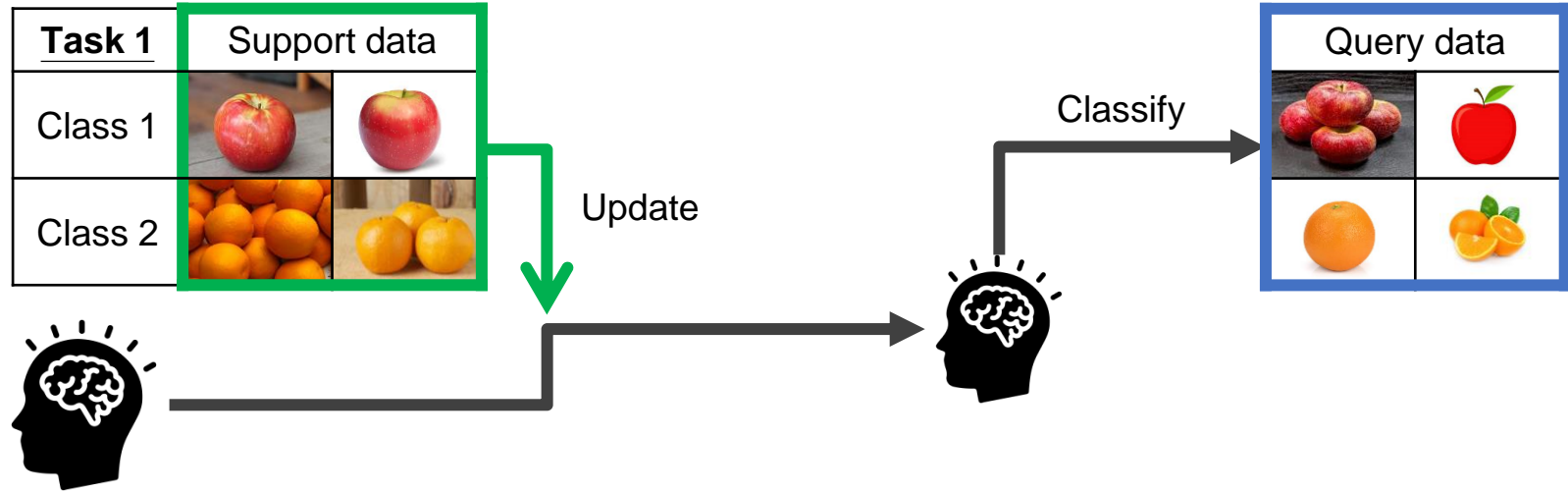
Chia-Hsiang Kao¹, Wei-Chen Chiu¹, Pin-Yu Chen²

¹National Yang Ming Chiao Tung University

²MIT-IBM Watson AI Lab

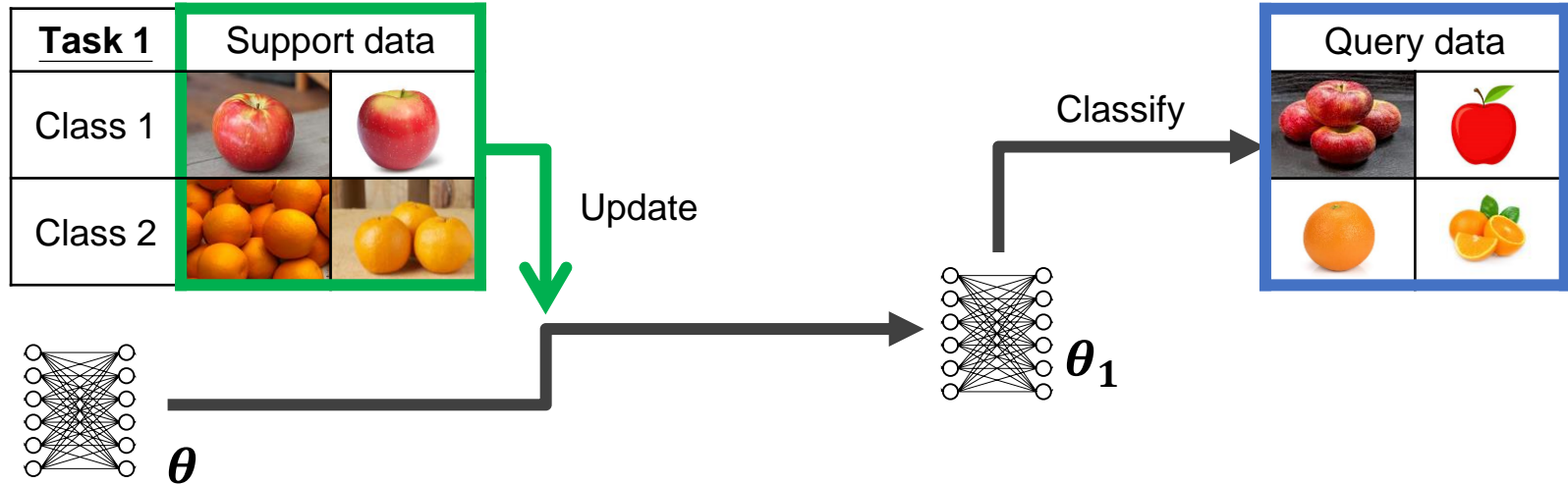
Introduction

Humans learn to classify even with limited experience.



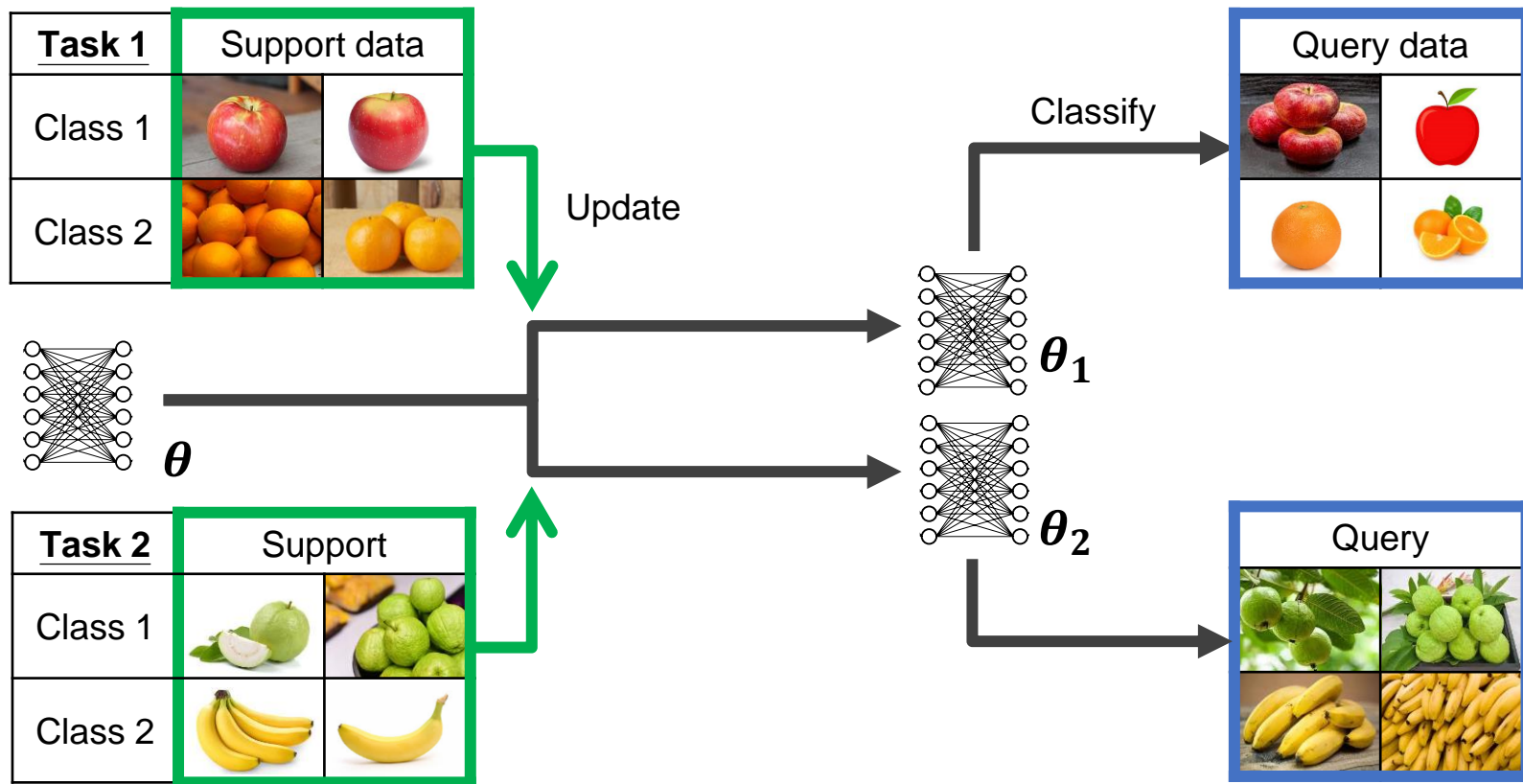
Introduction

Humans learn to classify even with limited experience.



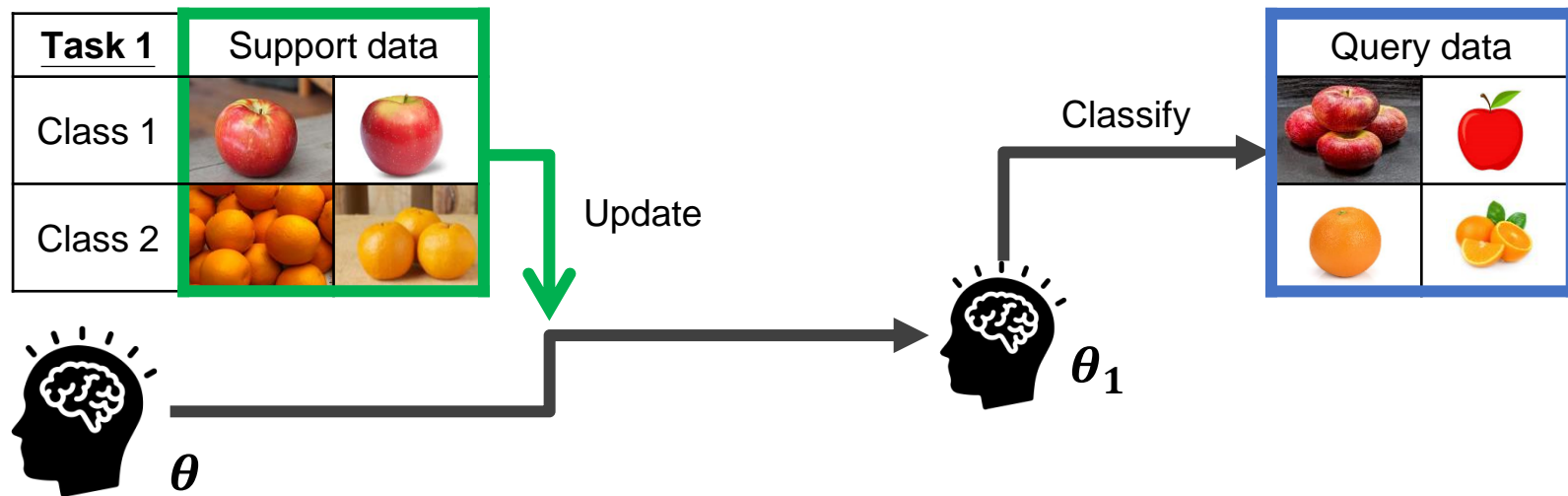
Introduction

Humans learn to classify even with limited experience.



Introduction

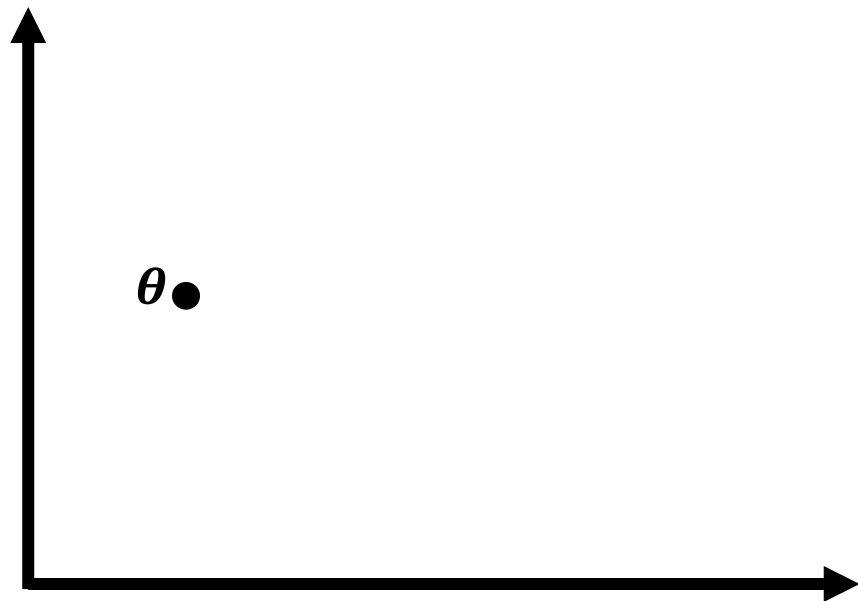
Humans learn to classify even with limited experience.



















- MAML is a gradient-based meta-learning algorithm that finds a good θ .

Introduction

MAML makes model learn to classify after seeing little data.

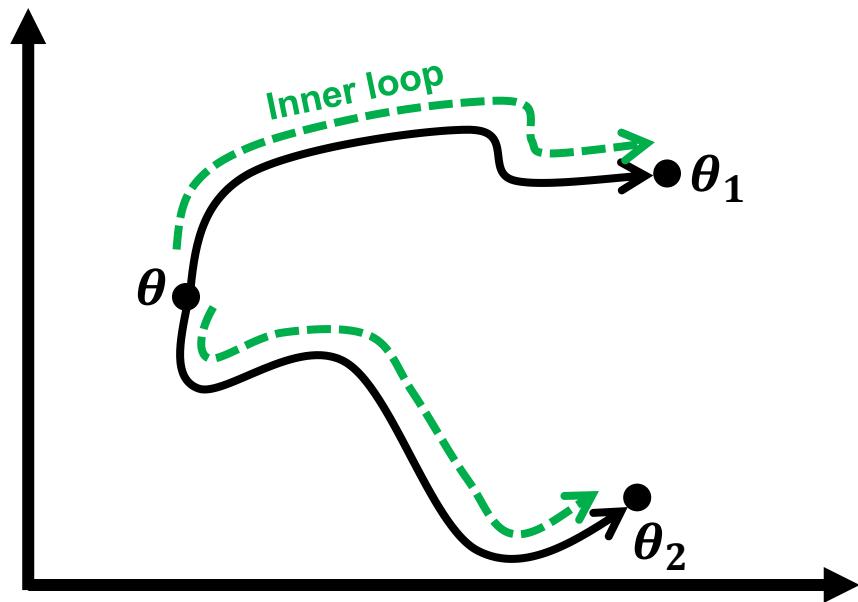





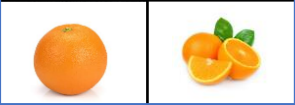




<u>Task 1</u>	Support S_1		Query Q_1	
Class 1				
Class 2				

<u>Task 2</u>	Support S_2		Query Q_2	
Class 1				
Class 2				

Introduction

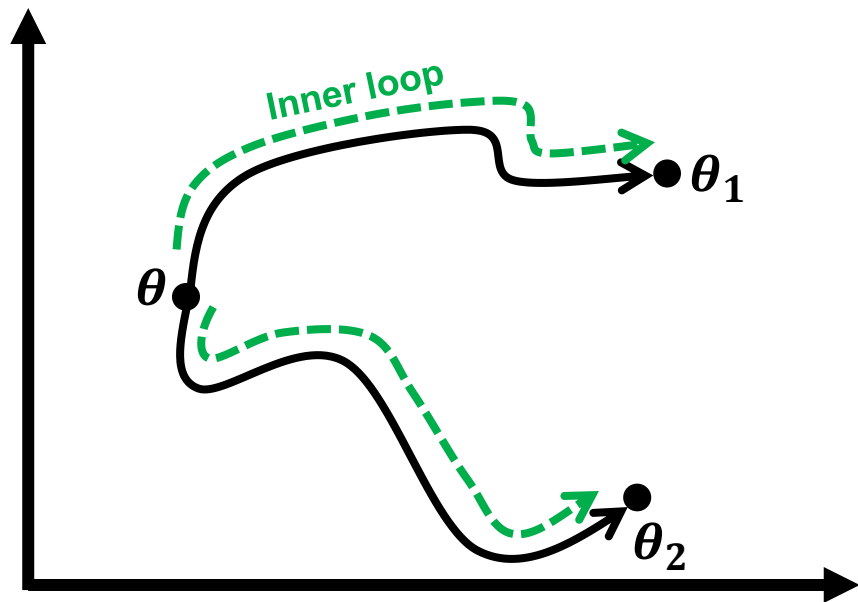
MAML makes model learn to classify after seeing little data.




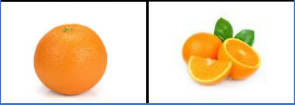





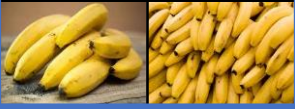
<u>Task 1</u>	Support S_1	Query Q_1
Class 1		
Class 2		
<u>Task 2</u>	Support S_2	Query Q_2
Class 1		
Class 2		

Introduction

MAML makes model learn to classify after seeing little data.



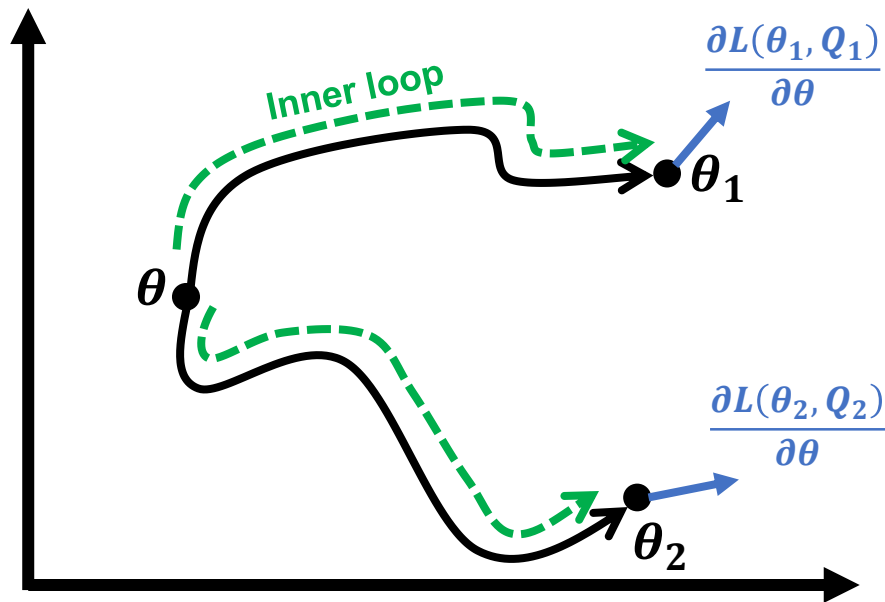
<u>Task 1</u>	Support S_1	Query Q_1
Class 1		
Class 2		









<u>Task 2</u>	Support S_2	Query Q_2
Class 1		
Class 2		









- Goal: Minimize $L(\theta_1, Q_1)$ and $L(\theta_2, Q_2)$ by finding best θ .

Introduction

MAML makes model learn to classify after seeing little data.



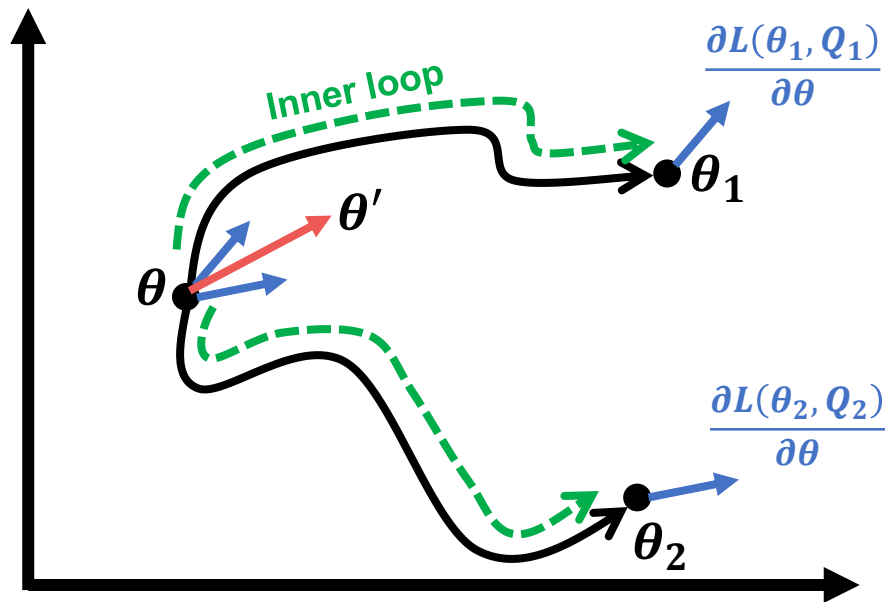
Task 1	Support S_1		Query Q_1	
Class 1				
Class 2				









Task 2	Support S_2		Query Q_2	
Class 1				
Class 2				









- Goal: Minimize $L(\theta_1, Q_1)$ and $L(\theta_2, Q_2)$ by finding best θ .

Introduction

MAML makes model learn to classify after seeing little data.



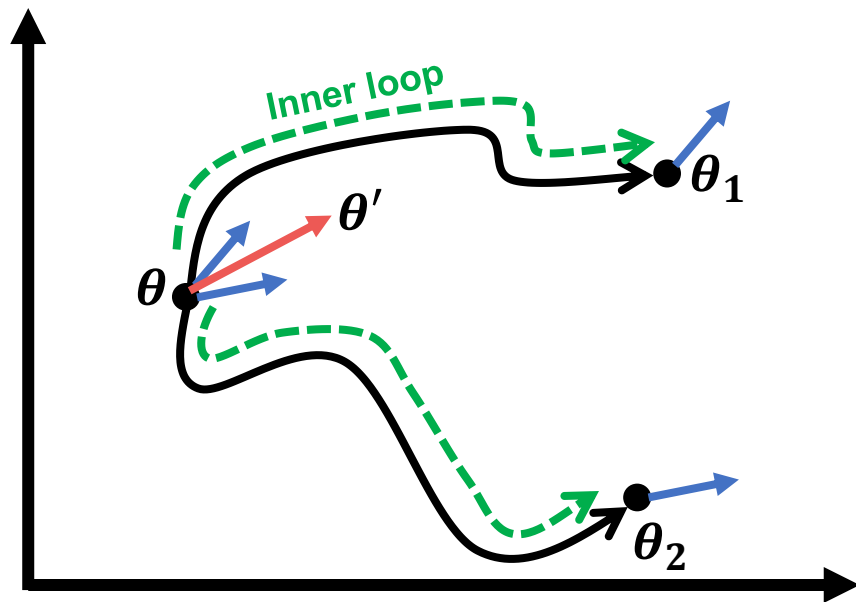
Task 1	Support S_1		Query Q_1	
Class 1				
Class 2				

Task 2	Support S_2		Query Q_2	
Class 1				
Class 2				

- Goal: Minimize $L(\theta_1, Q_1)$ and $L(\theta_2, Q_2)$ by finding best θ .

Introduction

MAML makes model learn to classify after seeing little data.



Algorithm 1 Second-order MAML

```
1: while not done do
2:   Sample tasks  $\{T_1, T_2\}$ 
3:   for  $n = 1, 2$  do
4:      $\{S_n, Q_n\} \leftarrow$  sample from  $T_n$ 
5:      $\theta_n = \theta$ 
6:     for  $i = 1, 2, \dots, N_{step}$  do
7:        $\theta_n \leftarrow \theta_n - \eta \nabla_{\theta_n} L(\theta_n, S_n)$ 
8:     end for
9:   end for
10:  Update  $\theta \leftarrow \theta - \rho \sum_{n=1}^{N_{batch}} \nabla_{\theta} L(\theta_n, Q_n)$ 
11: end while
```

- Goal: Minimize $L(\theta_1, Q_1)$ and $L(\theta_2, Q_2)$ by finding best θ .
- Method: update θ by $\theta' = \theta - \sum \frac{\partial L(\theta_n, Q_n)}{\partial \theta}$

Introduction

MAML makes model learn to classify after seeing little data.

Why is MAML successful?

- It is widely believed that MAML encourages models to learn a general-purpose representations which are applicable to novel tasks.

Introduction

MAML makes model learn to classify after seeing little data.

Why is MAML successful?

- It is widely believed that MAML encourages models to learn a general-purpose representations which are applicable to novel tasks.

In this paper, we step further and ask:

- How does MAML encourage any model to learn general-purpose representations?
- What is the role of the support and query data and how do they interact with each other?

Introduction

MAML makes model learn to classify after seeing little data.

Why is MAML successful?

- It is widely believed that MAML encourages models to learn a general-purpose representations which are applicable to novel tasks.

In this paper, we step further and ask:

- How does MAML encourage any model to learn general-purpose representations?
- What is the role of the support and query data and how do they interact with each other?

Our contribution:

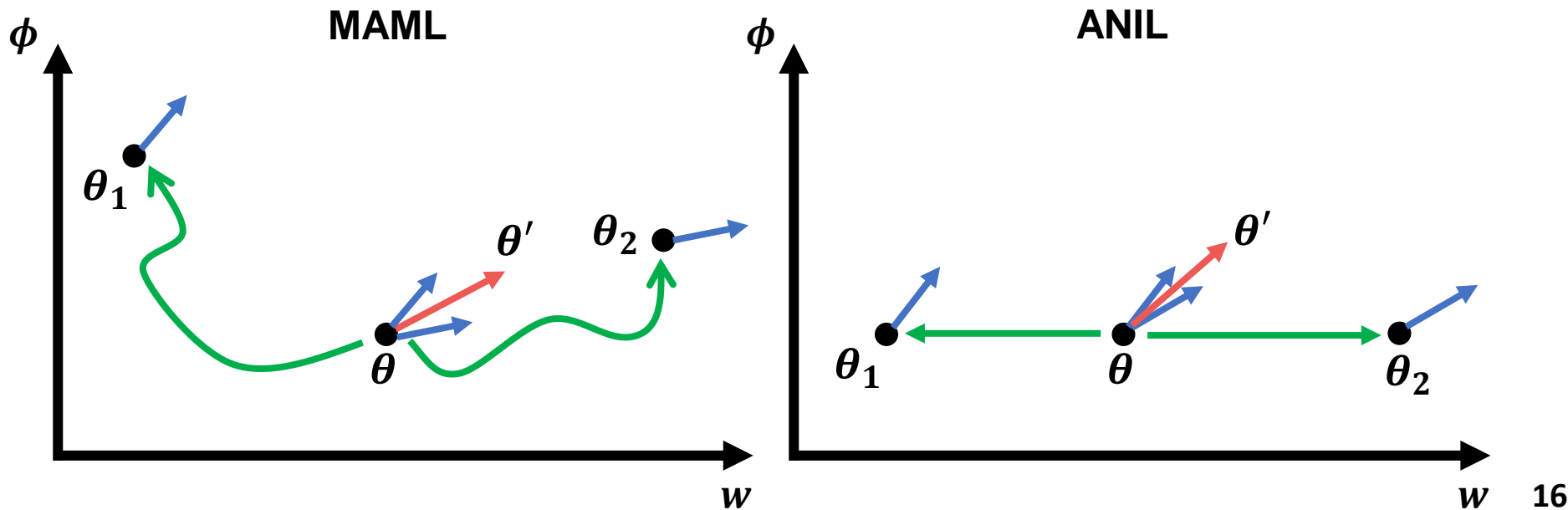
- We show that MAML is a noisy supervised contrastive learning algorithm.

Assumption

ANIL (Almost no inner loop)

Consider a model $\theta = \{\phi, w\}$, where ϕ is an encoder and w is a linear classifier.

ANIL states that the encoder ϕ is not updated during the inner loop.



Assumption

ANIL (Almost no inner loop)

Consider a model $\theta = \{\phi, w\}$, where ϕ is an encoder and w is a linear classifier.

ANIL states that the encoder ϕ is not updated during the inner loop.

The ANIL Assumption empirically sounds.

	Mini-ImageNet 5way-1shot	Mini-ImageNet 5way-5shot	Omniglot 20way-1shot	Omniglot 20way-5shot
MAML	46.9\pm0.2	63.1\pm0.4	93.7 \pm 0.7	96.4 \pm 0.1
ANIL	46.7 \pm 0.4	61.5 \pm 0.5	96.2\pm0.5	98.0\pm0.3

Main Derivation

Motivating example

Assumptions:

- ANIL.
- Linear classifier w is zeroed at the beginning.

Loss: Mean square error.

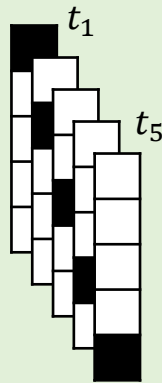
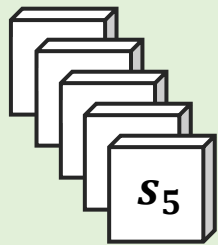
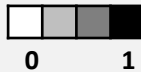
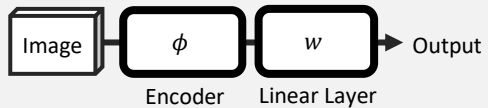
Condition: One inner loop update.

Setting

- 5-way: Each task contains 5 classes of images.
- 1-shot: Only one image per class in the support data.

Setting:
5-way 1-shot using MAML with one
inner-loop update under MSE loss.

Model:

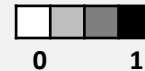
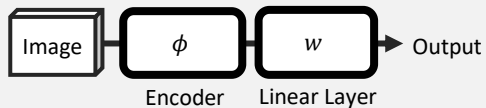


Inner Loop (1 step)

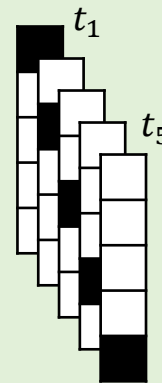
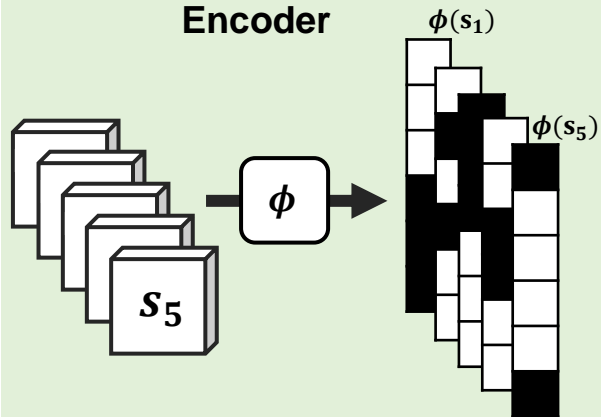
Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

Model:



Encoder

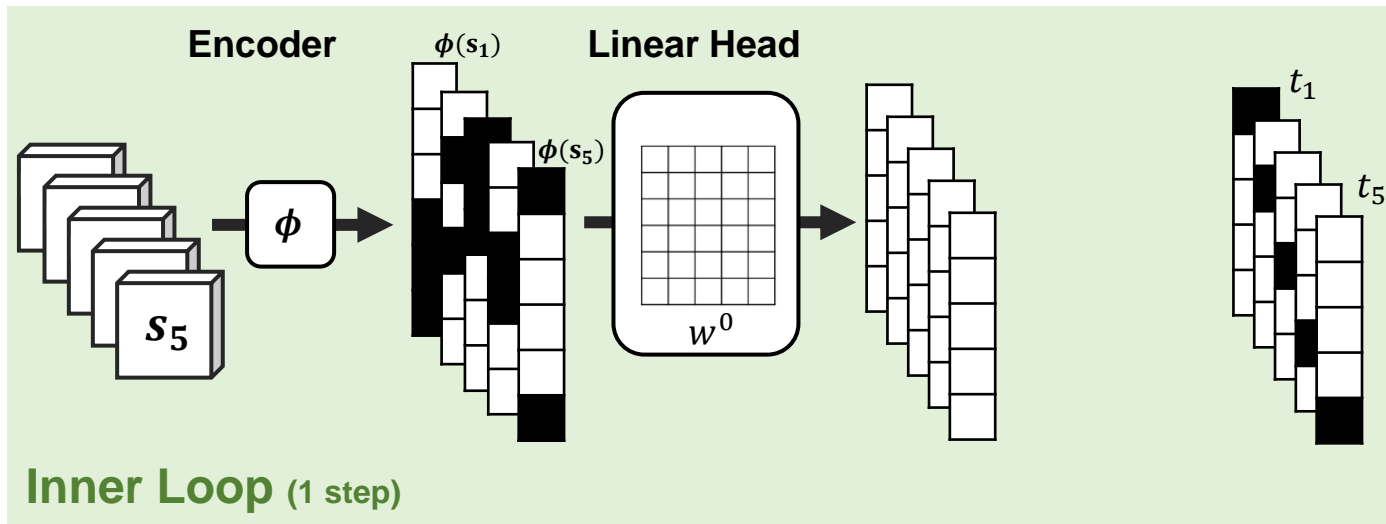
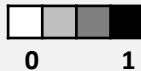
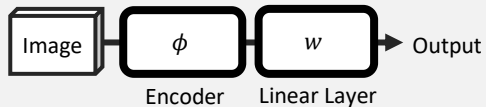


Inner Loop (1 step)

Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

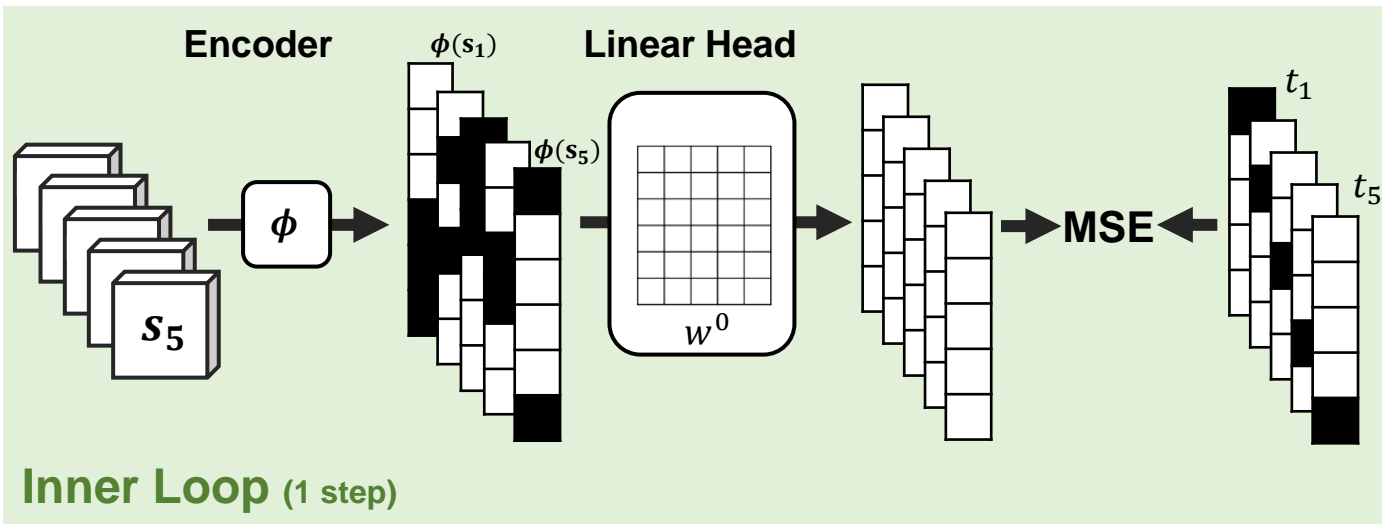
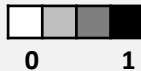
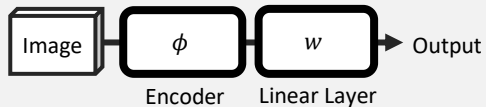
Model:



Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

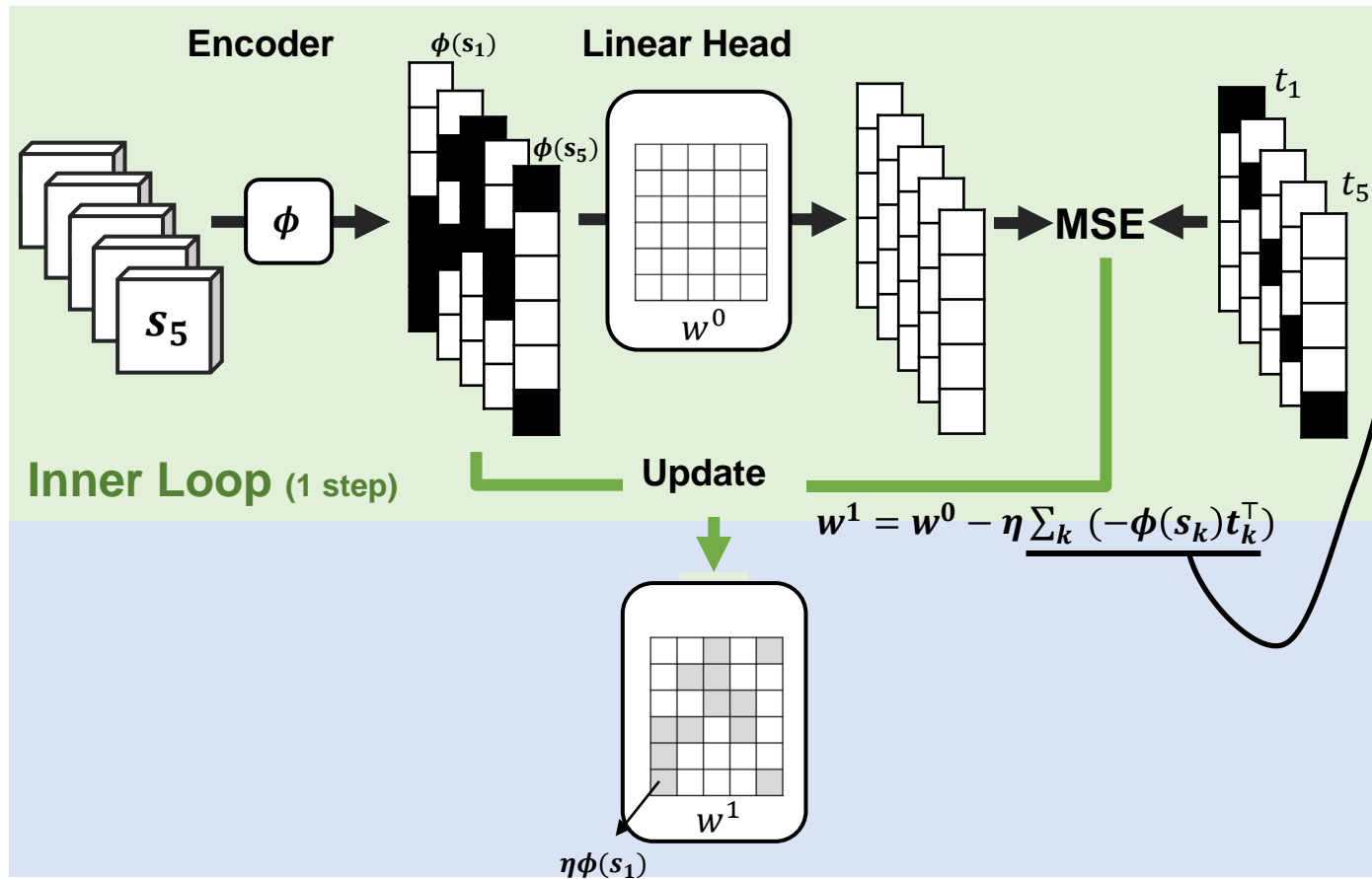
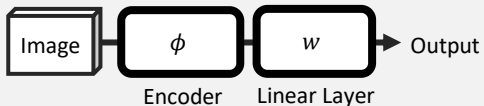
Model:



Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

Model:



$$\begin{bmatrix} \text{grid} \end{bmatrix} = \begin{bmatrix} \text{grid} \end{bmatrix} + \begin{bmatrix} \text{column} \end{bmatrix} \cdot \begin{bmatrix} \text{row} \end{bmatrix}$$

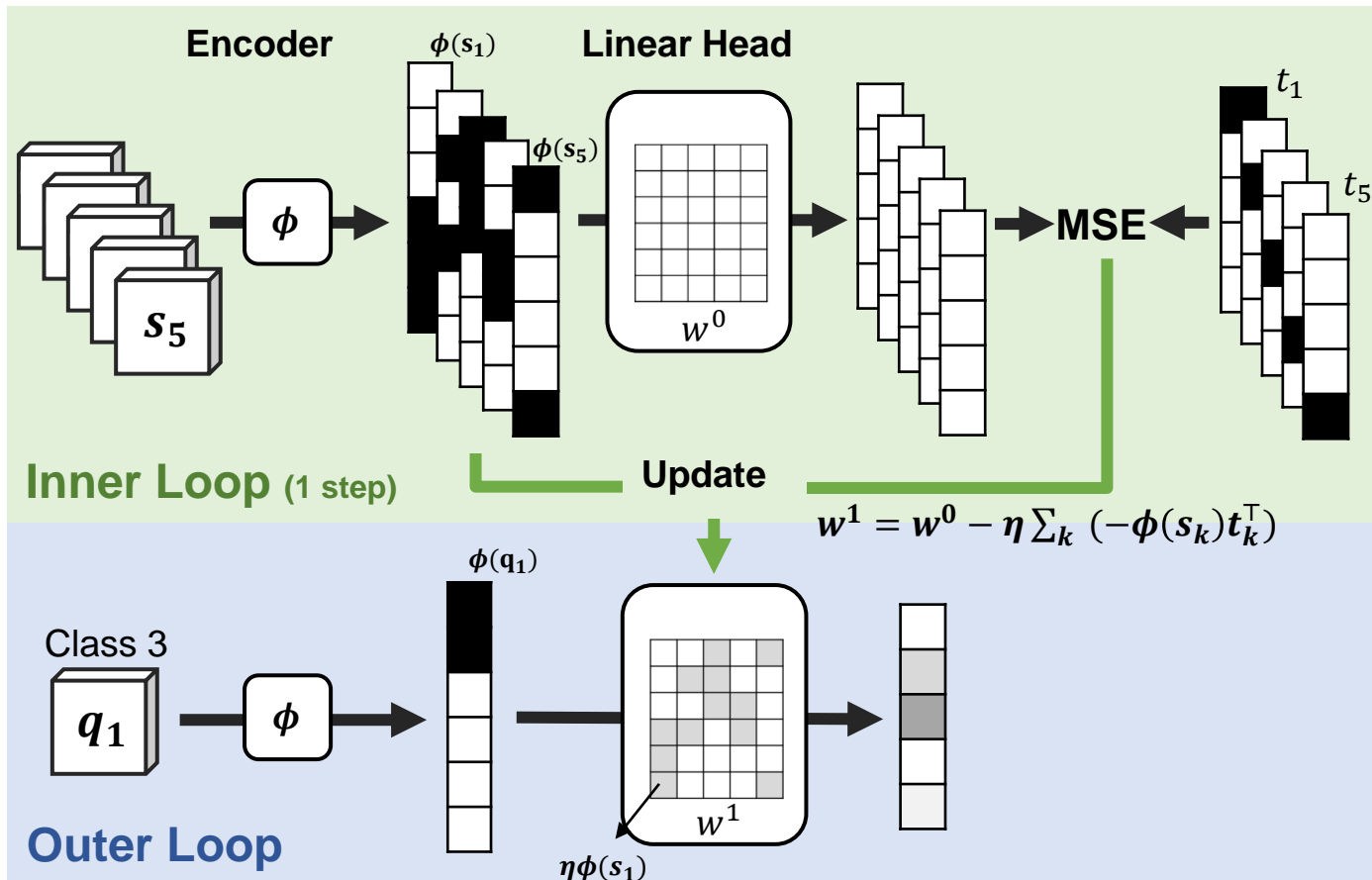
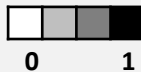
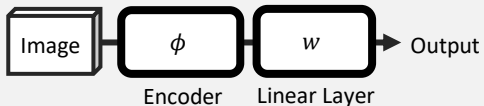
$\phi(s_1)$ t_1^T

$\phi(s_2)$ t_2^T

Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

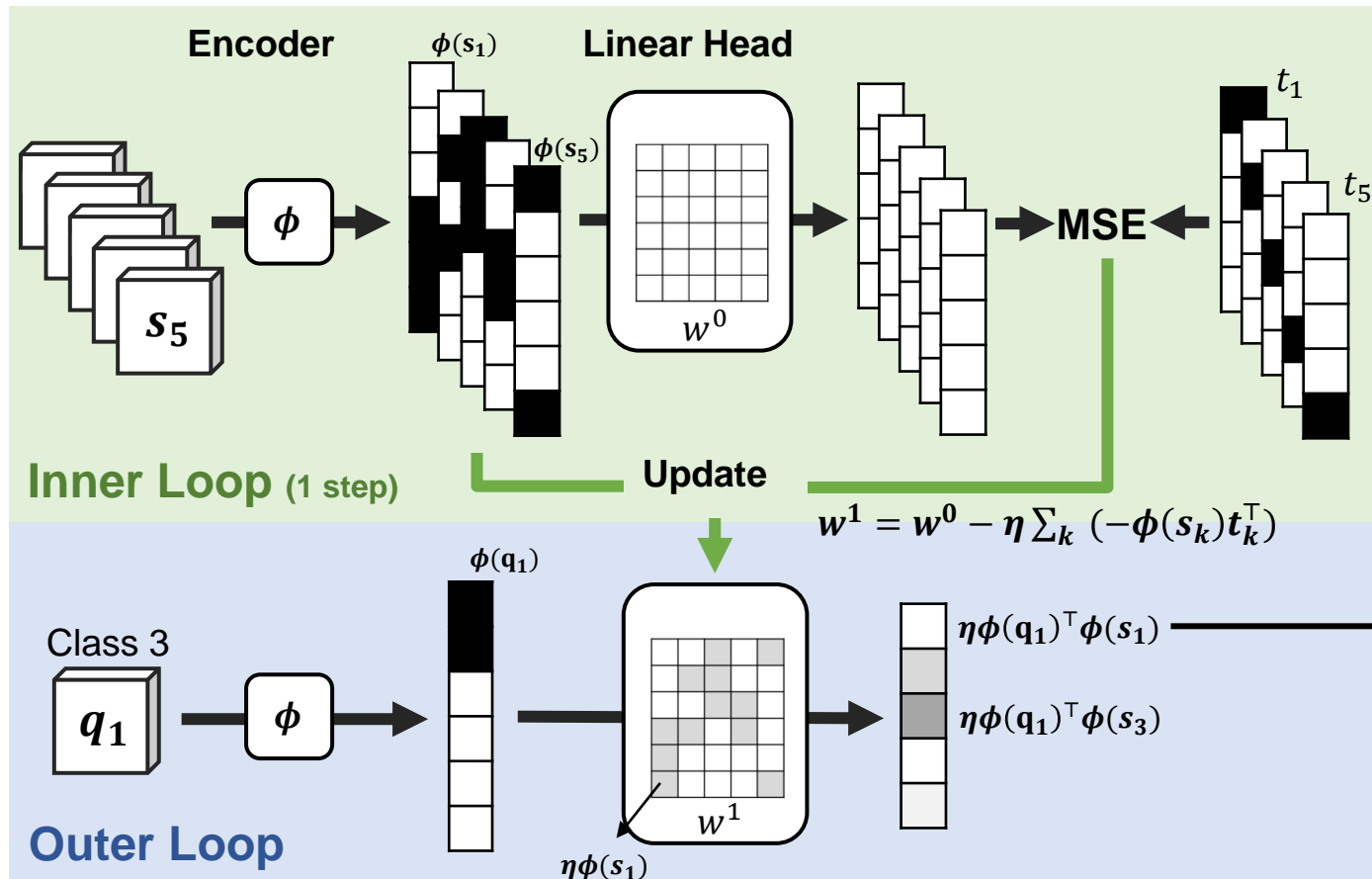
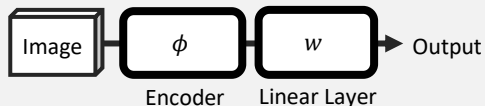
Model:



Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

Model:

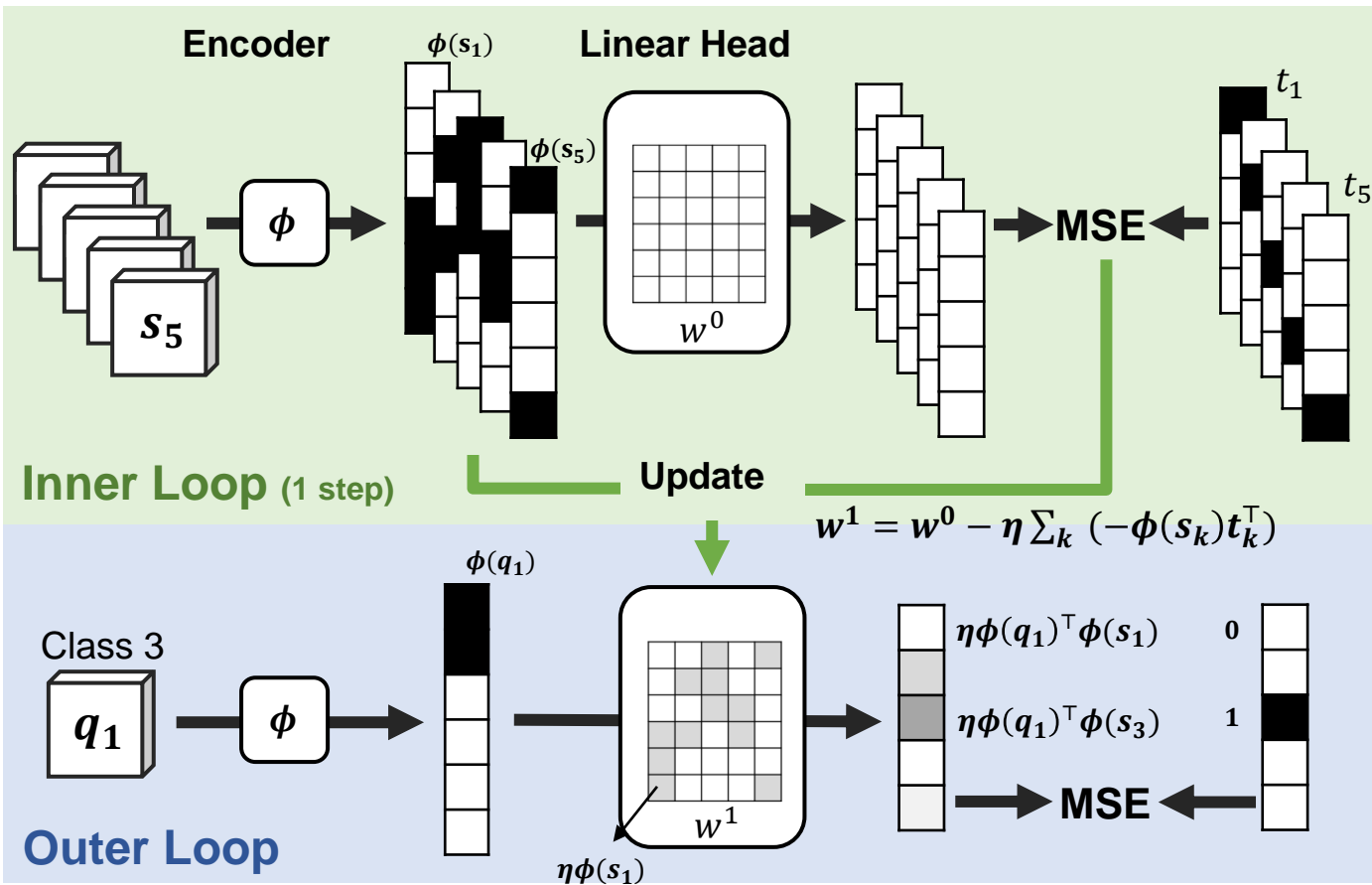
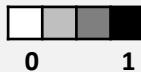
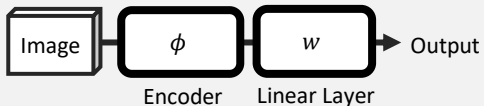


The inner product
between support feature
 s_1 and query feature q_1 .

Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

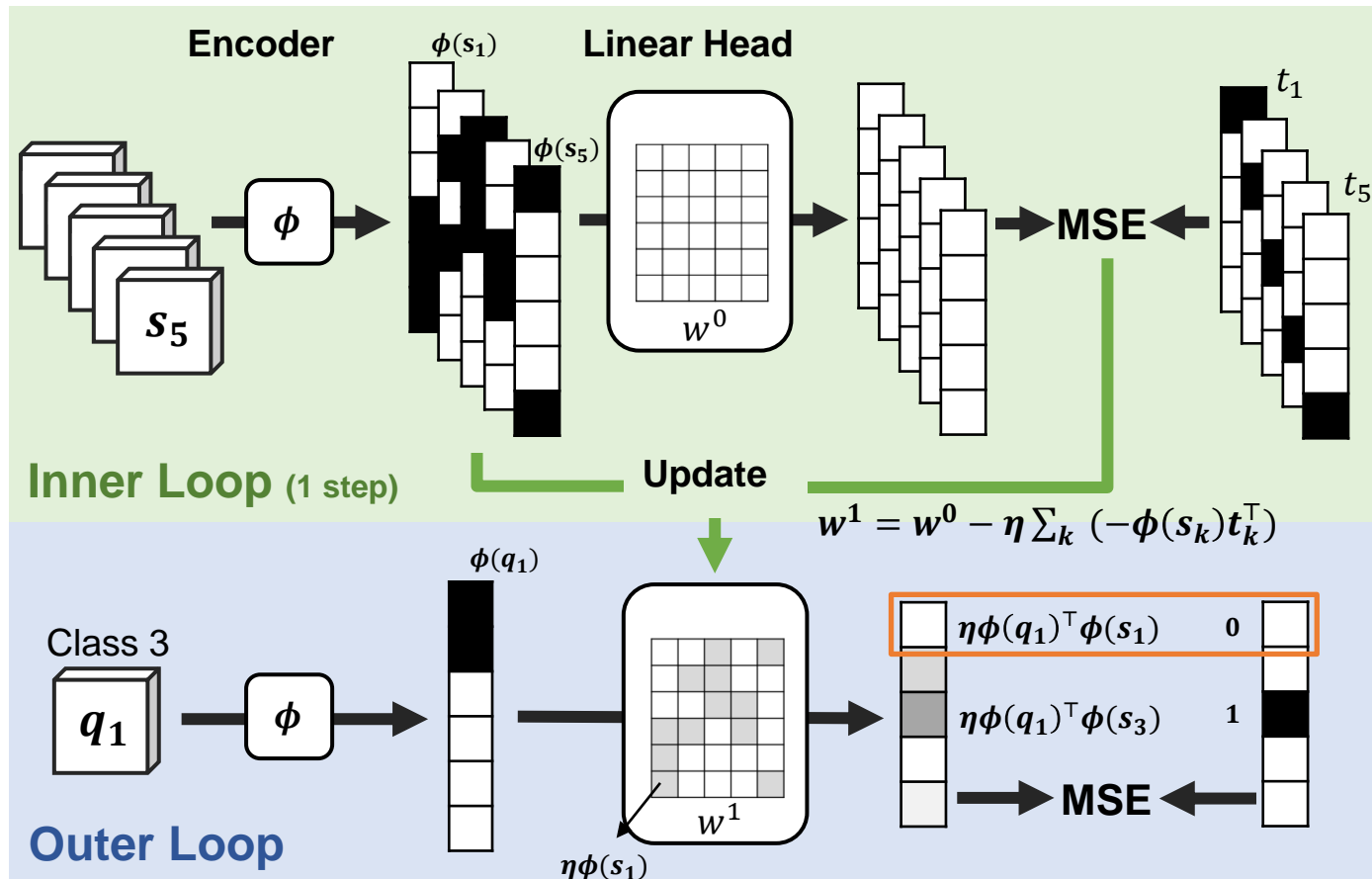
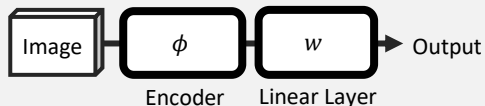
Model:



Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

Model:



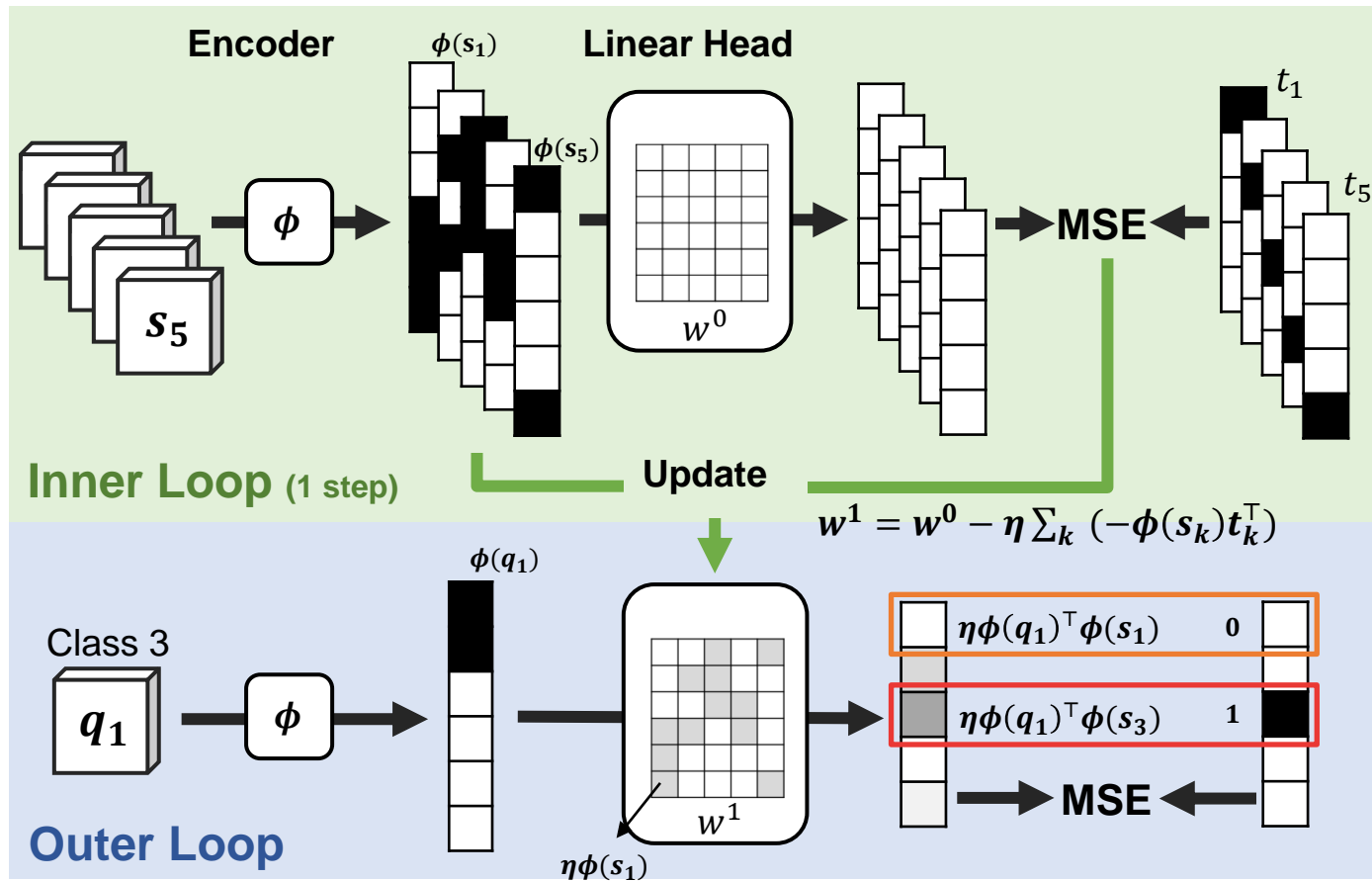
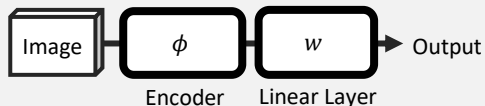
Negative sample

- q_1 and s_1 have different labels
- Their inner product of their features should be zero.

Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

Model:



Negative sample

- q_1 and s_1 have different labels
- Their inner product of their features should be zero.

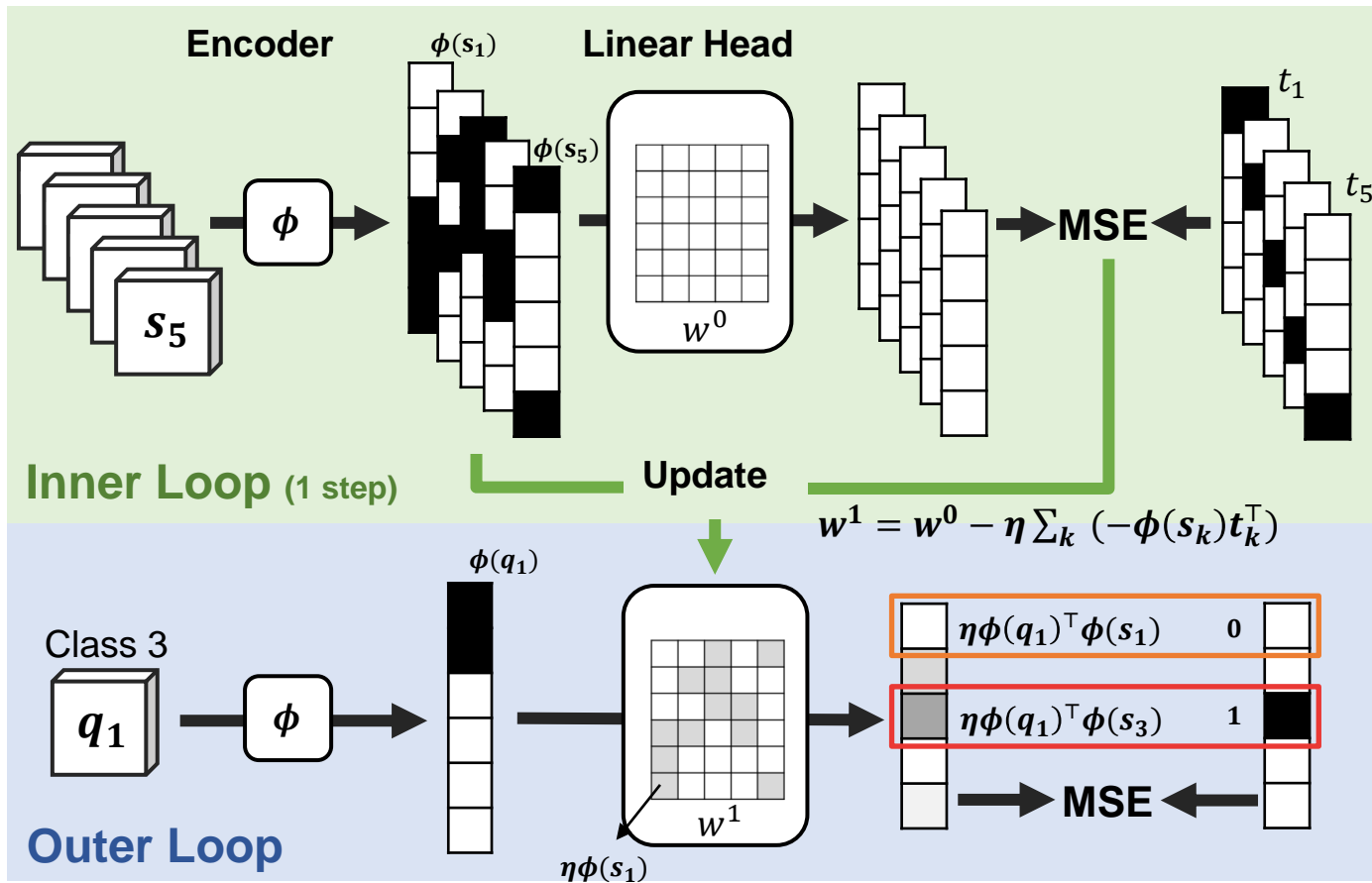
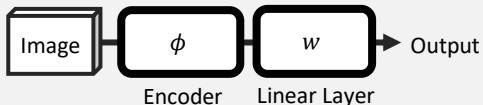
Positive sample

- q_1 and s_3 have same labels,
- Their inner product of their features should be one.

Setting:

5-way 1-shot using MAML with one
inner-loop update under MSE loss.

Model:



**Supervised
contrastive
learning**



Negative sample

- q_1 and s_1 have different labels
- Their inner product of their features should be zero.

Positive sample

- q_1 and s_3 have same labels,
- Their inner product of their features should be one.

Main Derivation

Main result

Consider support data $S = \{(s, t)\}$ and one query data (q, u) .

Under **ANIL assumption**, the **loss for the encoder** is :

- First-order MAML:

$$L = \sum_{i=1}^{N_{class}} \underbrace{(q_i - 1_{i=u}) \mathbf{w}_i^\top}_{\text{stop gradient}} \phi(q) + \eta \mathbf{E}_{(s,t) \sim S} \left[\underbrace{- \sum_{i=1}^{N_{class}} q_i s_i + s_u + q_t - 1_{t=u}}_{\text{stop gradient}} \phi(s)^\top \phi(q) \right]$$

- Meaning:
 - If q and s have sample label, then update ϕ s.t. $\phi(q)$ is closer to $\phi(s)$.
 - Otherwise, update ϕ s.t. $\phi(q)$ is closer to $\phi(s)$.

Main Derivation

Main result

Consider support data $S = \{(s, t)\}$ and one query data (q, u) .

Under **ANIL assumption**, the **loss for the encoder** is :

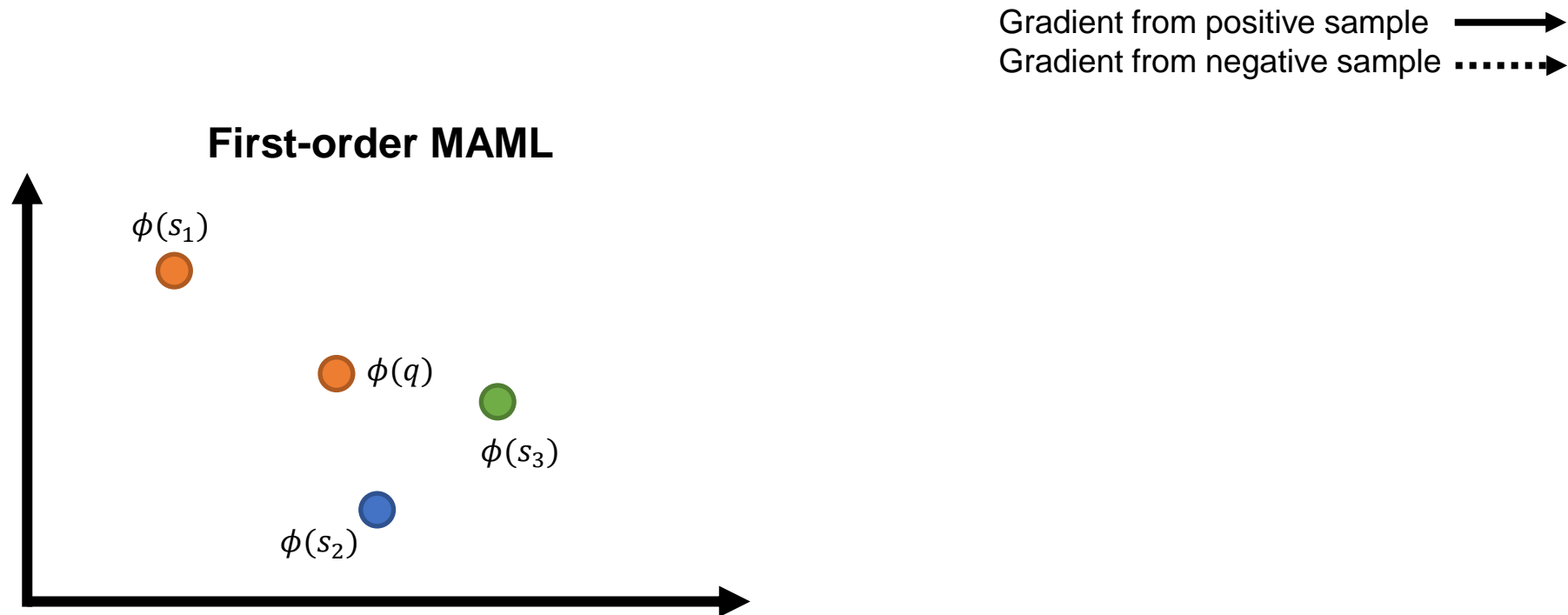
- First-order MAML:

$$L = \sum_{i=1}^{N_{class}} \underbrace{(q_i - 1_{i=u}) \mathbf{w}_i^\top}_{\text{stop gradient}} \phi(q) + \eta \mathbf{E}_{(s,t) \sim S} \underbrace{\left[- \sum_{i=1}^{N_{class}} q_i s_i + s_u + q_t - 1_{t=u} \right] \phi(s)^\top \phi(q)}_{\substack{\text{stop gradient} \\ \text{Contrastive coefficient} \quad \text{Inner product}}}$$

- Expected effect of the second term:
 - If q and s have sample label, then update ϕ s.t. $\phi(q)$ is closer to $\phi(s)$.
 - Otherwise, update ϕ s.t. $\phi(q)$ is closer to $\phi(s)$.

Main Derivation

Main result. Feature space illustration.



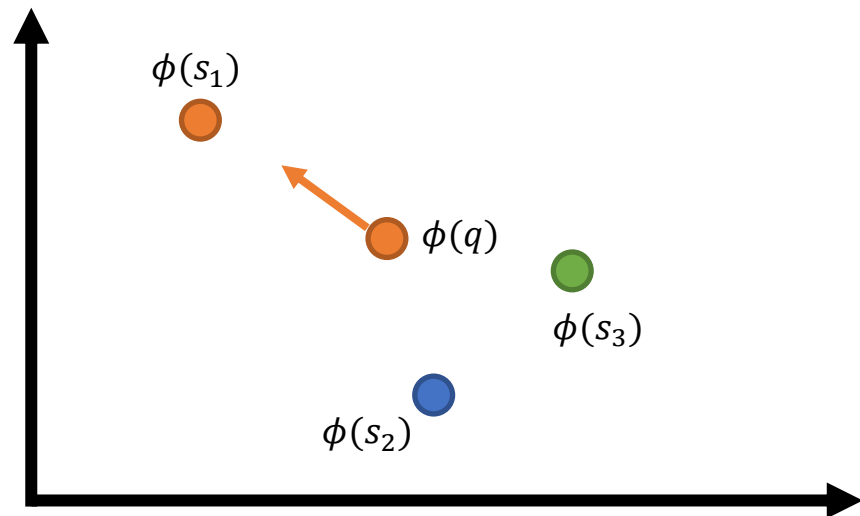
update ϕ such that $\phi(q)$ is closer/further to $\phi(s)$

Main Derivation

Main result. Feature space illustration.

Gradient from positive sample \longrightarrow
Gradient from negative sample $\cdots\cdots\longrightarrow$

First-order MAML



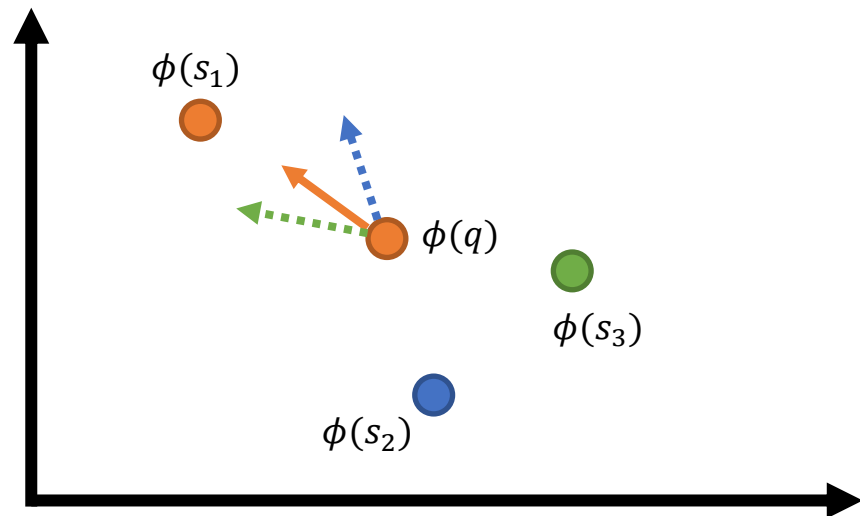
update ϕ such that $\phi(q)$ is closer/further to $\phi(s)$

Main Derivation

Main result. Feature space illustration.

Gradient from positive sample \longrightarrow
Gradient from negative sample $\cdots\cdots\longrightarrow$

First-order MAML



update ϕ such that $\phi(q)$ is closer/further to $\phi(s)$

Main Derivation

Main result

Consider support data $S = \{(s, t)\}$ and one query data (q, u) .

Under **ANIL assumption**, the **loss for the encoder** is :

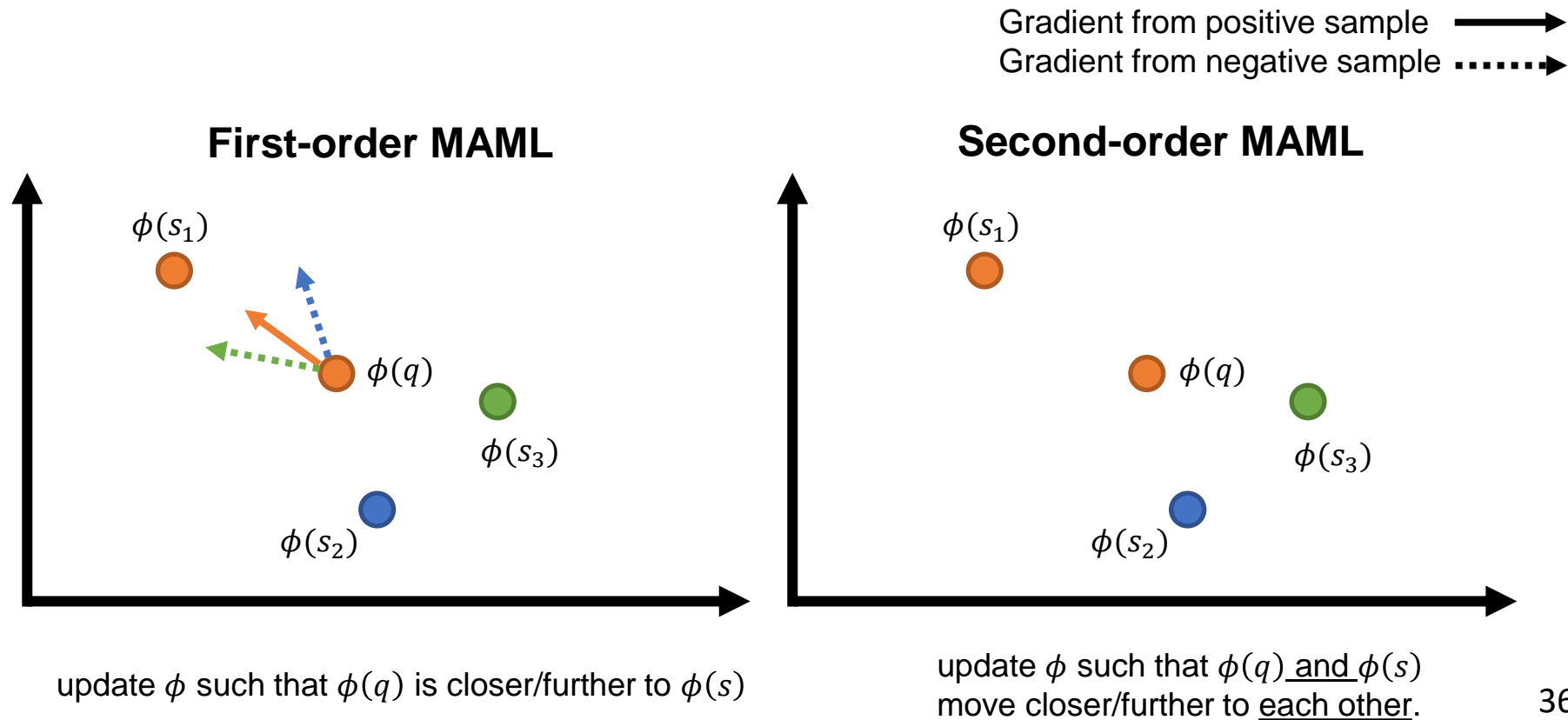
- Second-order MAML:

$$L = \sum_{i=1}^{N_{class}} \underbrace{(q_i - 1_{i=u}) \mathbf{w}_i^\top}_{\text{stop gradient}} \phi(q) + \eta \mathbf{E}_{(s,t) \sim S} \left[\underbrace{- \sum_{i=1}^{N_{class}} q_i s_i + s_u + q_t - 1_{t=u}}_{\text{Contrastive coefficient}} \underbrace{\phi(s)^\top \phi(q)}_{\text{Inner product}} \right]$$

- Expected effect of the second term:
 - If q and s have sample label, then update ϕ such that $\phi(q)$ and $\phi(s)$ move closer to each other.
 - Otherwise, update ϕ such that $\phi(q)$ and $\phi(s)$ move further to each other.

Main Derivation

Main result. Feature space illustration.

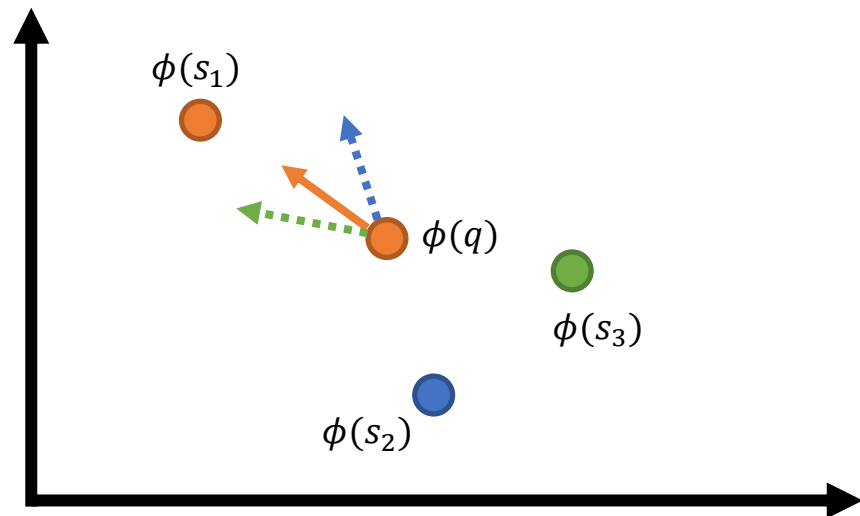


Main Derivation

Main result. Feature space illustration.

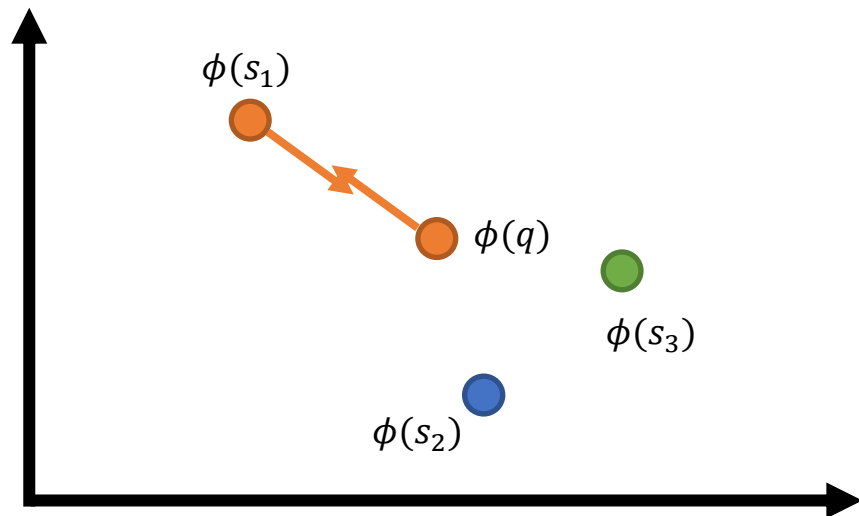
Gradient from positive sample \longrightarrow
Gradient from negative sample $\cdots\cdots\longrightarrow$

First-order MAML



update ϕ such that $\phi(q)$ is closer/further to $\phi(s)$

Second-order MAML



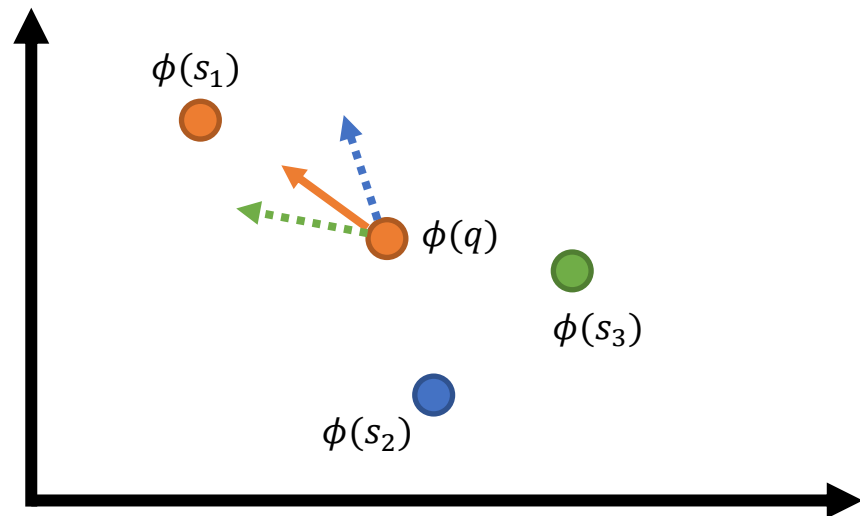
update ϕ such that $\phi(q)$ and $\phi(s)$
move closer/further to each other.

Main Derivation

Main result. Feature space illustration.

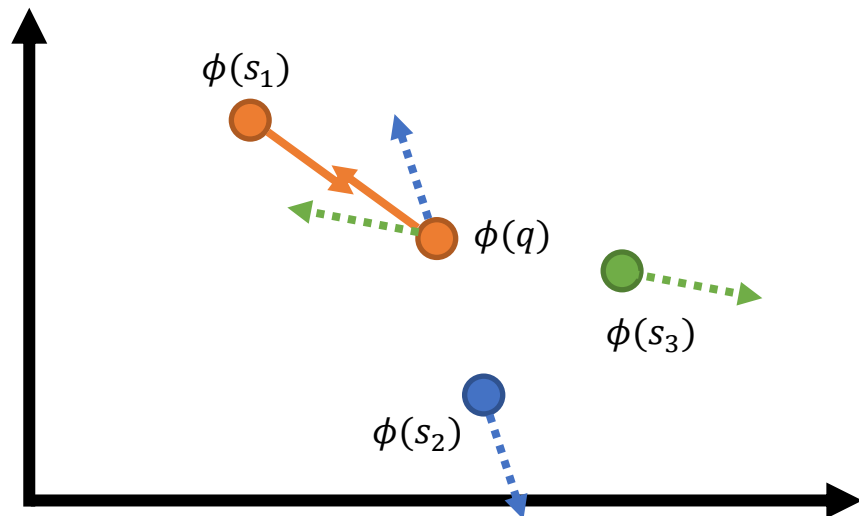
Gradient from positive sample \longrightarrow
Gradient from negative sample $\cdots\cdots\longrightarrow$

First-order MAML



update ϕ such that $\phi(q)$ is closer/further to $\phi(s)$

Second-order MAML



update ϕ such that $\phi(q)$ and $\phi(s)$
move closer/further to each other.

Main Derivation

Main result

Consider support data $S = \{(s, t)\}$ and one query data (q, u) .

Under **ANIL assumption**, the **loss for the encoder** is :

- First-order MAML:

$$L = \underbrace{\sum_{i=1}^{N_{class}} (\underbrace{q_i - 1_{i=u}}_{\text{stop gradient}}) \mathbf{w}_i^\top \phi(q)}_{\text{Random initialization}} + \eta \underbrace{\mathbf{E}_{(s,t) \sim S} \left[- \sum_{i=1}^{N_{class}} q_i s_i + s_u + q_t - 1_{t=u} \right] \phi(s)^\top \phi(q)}_{\text{Cross task interference}}$$

- Random initialization
- Cross task interference

- The coefficient is not always positive when s and q come from different classes

Theorem 1 *With the assumption of (a) no inner loop update of the encoder, FOMAML is a noisy SCL algorithm. With assumptions of (a) no inner loop update of the encoder and (b) a single inner-loop update, SOMAML is a noisy SCL algorithm.*

Main Derivation

Main result. Introducing the zeroing trick.

Consider support data $S = \{(s, t)\}$ and one query data (q, u) .

Under ANIL assumption and the zeroing trick, the loss for the encoder is :

- First-order MAML:

$$L = \eta \mathbf{E}_{(s,t) \sim S} \underbrace{(\mathbf{q}_t - \mathbf{t}_{t=u}) \phi(s)^\top}_{\text{stop gradient}} \phi(q)$$

- Second-order MAML:

$$L = \eta \mathbf{E}_{(s,t) \sim S} \underbrace{(\mathbf{q}_t - \mathbf{t}_{t=u}) \phi(s)^\top}_{\text{stop gradient}} \phi(q)$$

Corollary 1 *With mild assumptions of (a) no inner loop update of the encoder, (b) a single inner-loop update and (c) training with the zeroing trick (i.e., the linear layer is zeroed at the end of each outer loop), both FOMAML and SOMAML are SCL algorithms.*

Main Derivation

Main result. Introducing the zeroing trick.

Algorithm 1 Second-order MAML

```
1: while not done do
2:   Sample tasks  $\{T_1, T_2\}$ 
3:
4:   for  $n = 1, 2$  do
5:      $\{S_n, Q_n\} \leftarrow$  sample from  $T_n$ 
6:      $\theta_n = \theta$ 
7:     for  $i = 1, 2, \dots, N_{step}$  do
8:        $\theta_n \leftarrow \theta_n - \eta \nabla_{\theta_n} L(\theta_n, S_n)$ 
9:     end for
10:  end for
11:  Update  $\theta \leftarrow \theta - \rho \sum_{n=1}^{N_{batch}} \nabla_{\theta} L(\theta_n, Q_n)$ 
12:
13: end while
```

Algorithm 2 Second-order MAML with zeroing trick

```
1: while not done do
2:   Sample tasks  $\{T_1, T_2\}$ 
3:    $w = 0$ 
4:   for  $n = 1, 2$  do
5:      $\{S_n, Q_n\} \leftarrow$  sample from  $T_n$ 
6:      $\theta_n = \theta$ 
7:     for  $i = 1, 2, \dots, N_{step}$  do
8:        $\theta_n \leftarrow \theta_n - \eta \nabla_{\theta_n} L(\theta_n, S_n)$ 
9:     end for
10:  end for
11:  Update  $\theta \leftarrow \theta - \rho \sum_{n=1}^{N_{batch}} \nabla_{\theta} L(\theta_n, Q_n)$ 
12:    $w = 0$ 
13: end while
```

Results

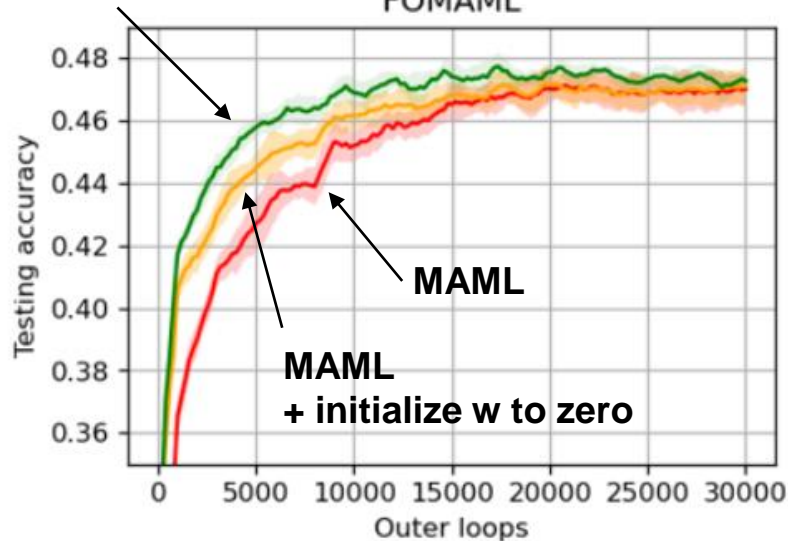
Using the zeroing trick improves performance.

Setting: MinImageNet.

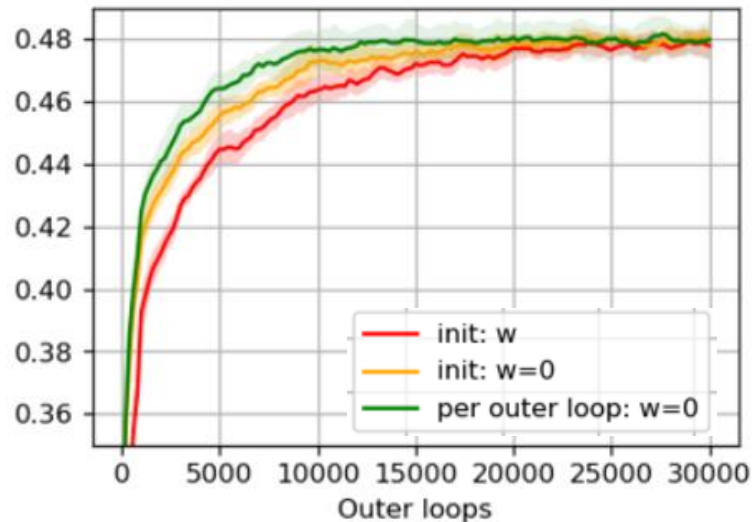
5-way 1-shot setting

MAML + zeroing trick

FOMAML



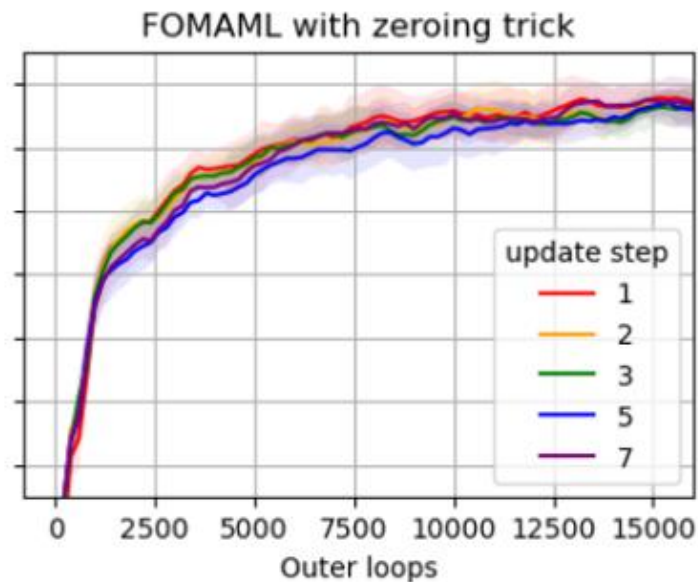
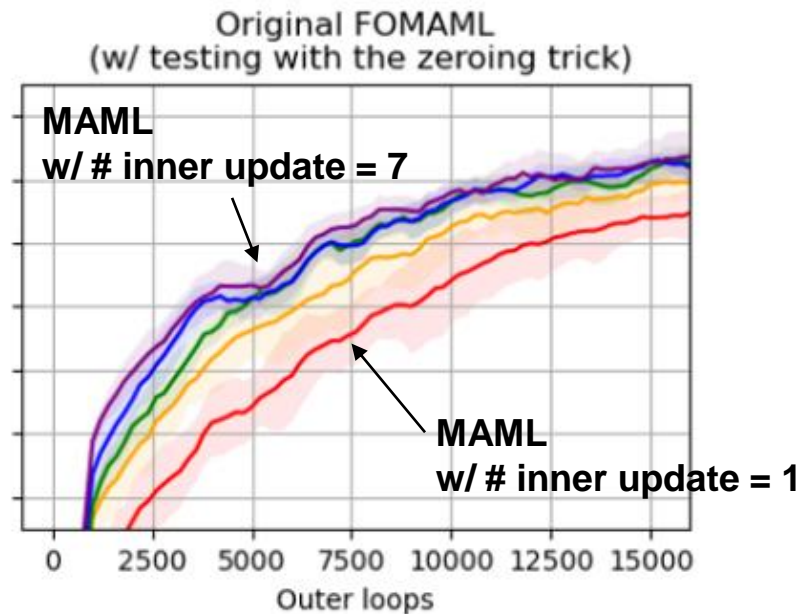
SOMAML



Results

With zeroing trick, # of inner loop steps no longer matters

Setting: MinImageNet 5-way 1-shot.

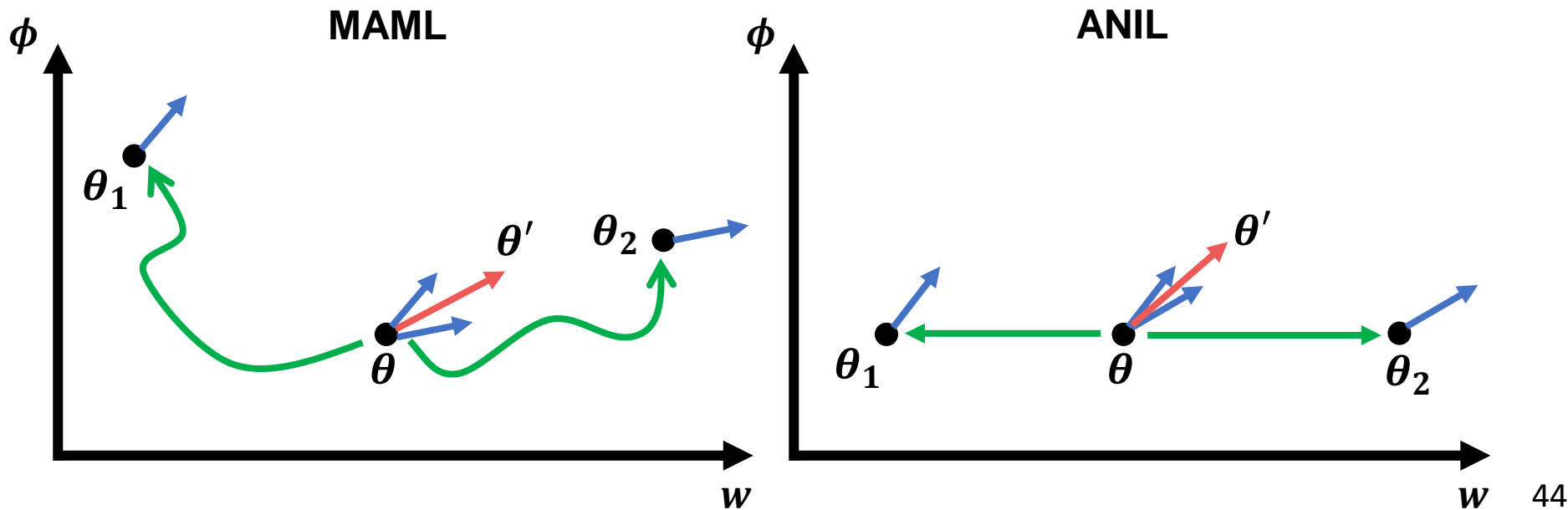


Wrap up

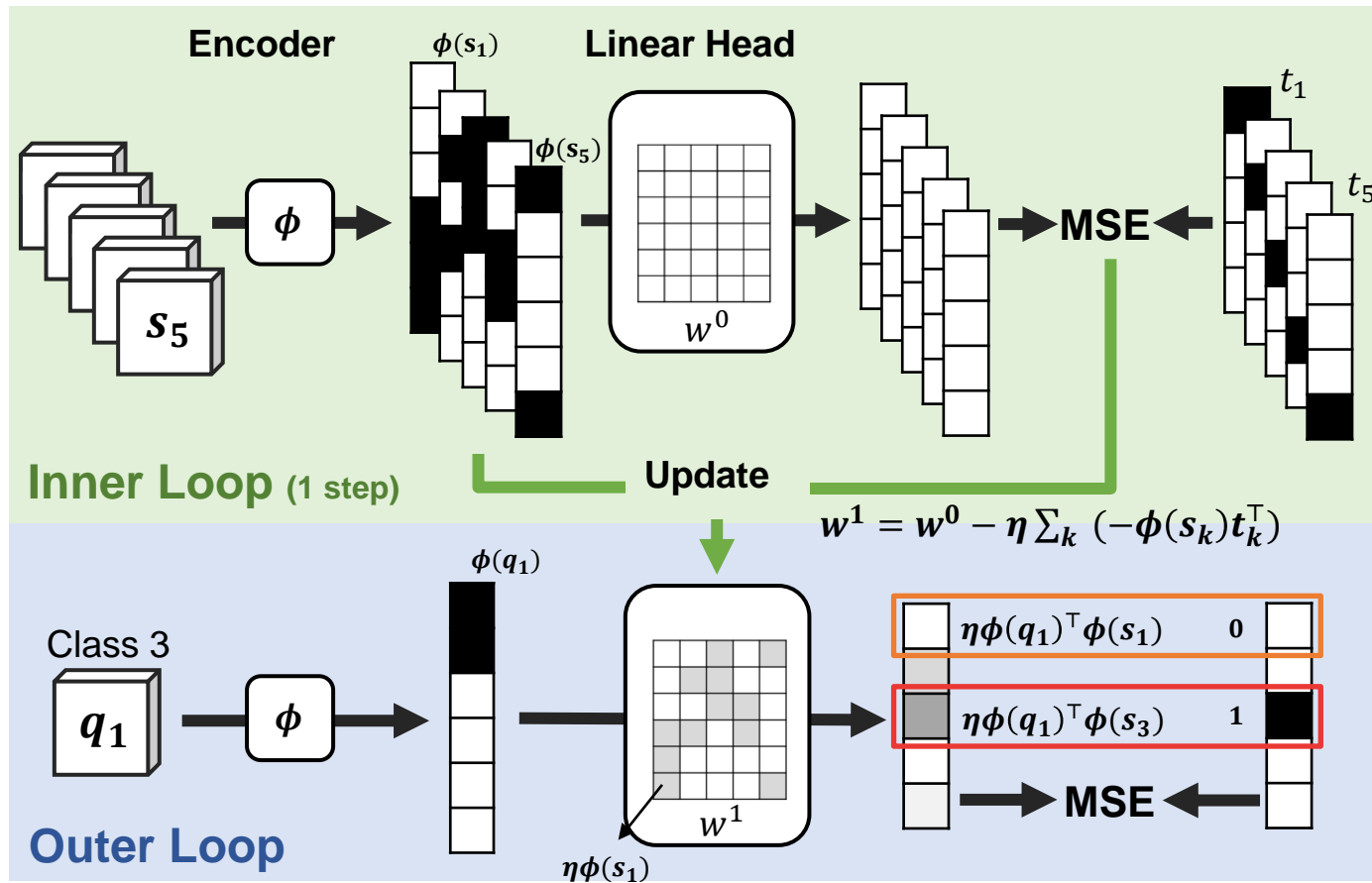
We use the ANIL assumption for derivation.

Consider a model $\theta = \{\phi, w\}$, where ϕ is an encoder and w is a linear classifier.

ANIL states that the encoder ϕ is not updated during the inner loop.



Wrap up



**Supervised
contrastive
learning**



Negative sample

- q_1 and s_1 have different labels
- Their inner product of their features should be zero.

Positive sample

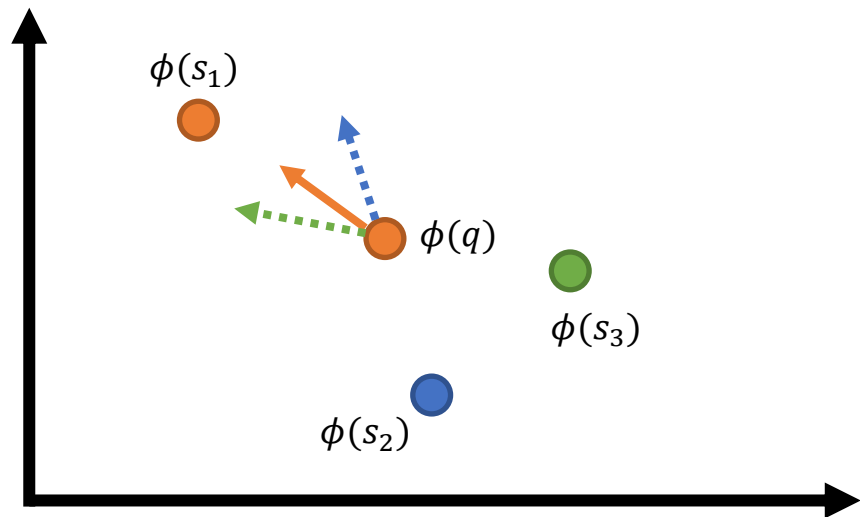
- q_1 and s_3 have same labels,
- Their inner product of their features should be one.

Wrap up

We show how FOMAML different from SOMAML.

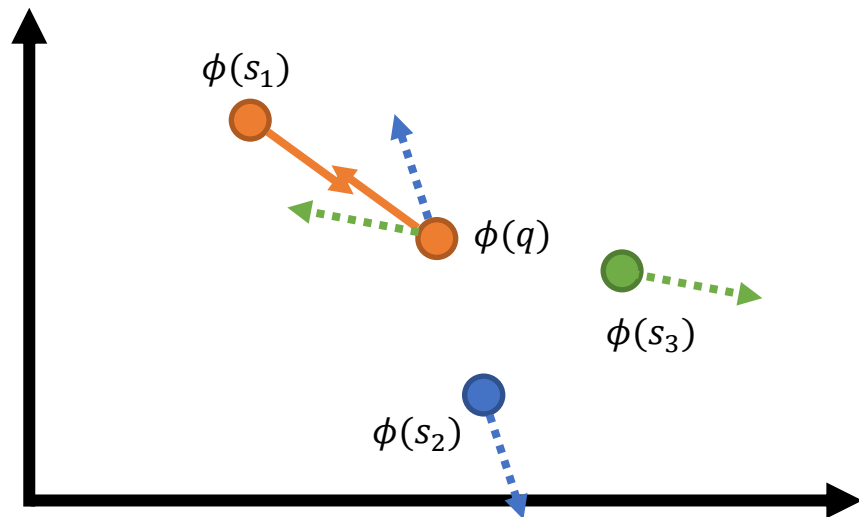
Gradient from positive sample \longrightarrow
Gradient from negative sample $\cdots\cdots\longrightarrow$

First-order MAML



update ϕ such that $\phi(q)$ is closer/further to $\phi(s)$

Second-order MAML



update ϕ such that $\phi(q)$ and $\phi(s)$
move closer/further to each other.

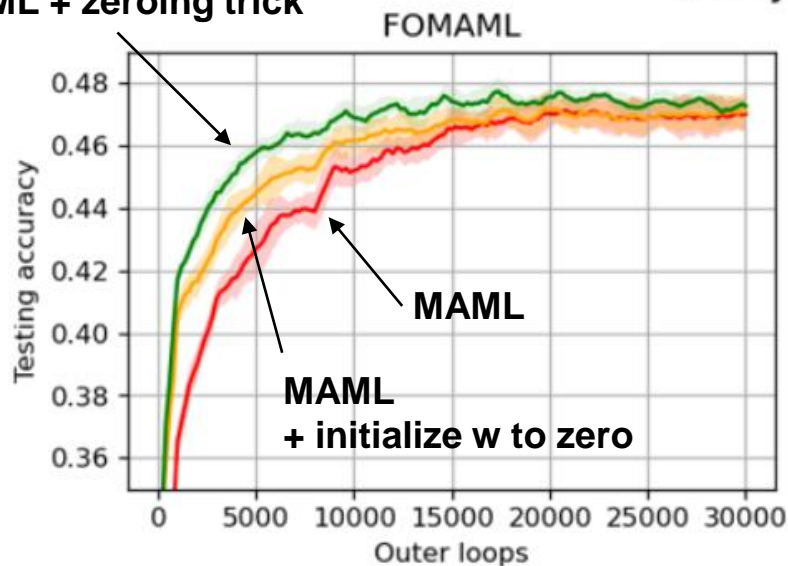
Wrap up

We show that the zeroing trick improves MAML.

Setting: MinImageNet.

5-way 1-shot setting

MAML + zeroing trick



SOMAML

