Meta Learning and Its Applications to Human Language Processing

Hung-yi Lee, Ngoc Thang Vu, Shang-Wen (Daniel) Li



Part I: Basic Idea of Meta Learning

All speakers

Hung-yi

Thang Vu

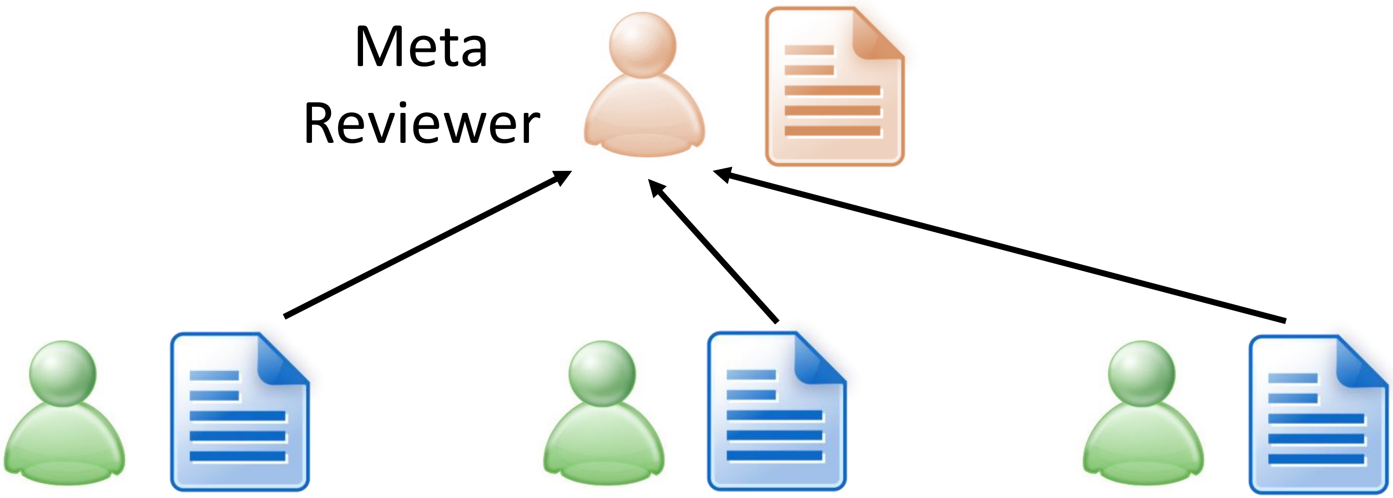
Part II: Applications

Why Meta Learning?

Hung-yi Lee (NTU)

- What does "meta" mean?

meta-X = X about X



<u>Meme</u>


Meta Learning
THEY ARE EVERYWHERE
meme-arsenal.ru
a random image with words
Is it a meme?

Meta Meme

Why Meta Learning?

Hung-yi Lee (NTU)

- What does "meta" mean?

  meta-X = X about X

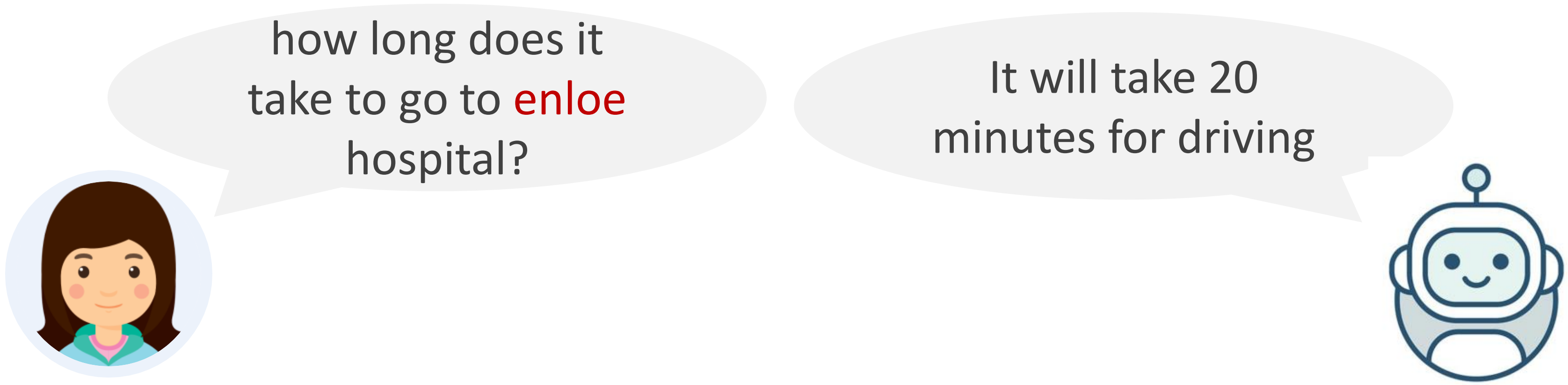
- Meta Learning is "learning about learning", or "learn to learn".
- How to reduce the requirement of annotated data for language and speech processing?
- Can machines find learning algorithms that only require limited labeled data?

Why Meta Learning?

- Thang Vu – A professor at the Institute for Natural Language Processing (IMS), University of Stuttgart, Germany
- My reasons:
  - Better understanding on results which I published during my PhD on *mutlilingual speech recognition, e.g. [1]*
  - A potential solution for low resource settings in speech and language processing applications

[1] Ngoc Thang Vu, Wojtek Breiter, Florian Metze, Tanja Schultz. **Initialization Schemes for Multilayer Perceptron Training and their Impact on ASR Performance using Multilingual Data**. Interspeech, 2012

Meta-learning for ConvAI



- ConvAI
- Impact on industry
  - many to-B, to-C, and to-D applications
- Language, domain, and task expansion
  - Meta-learning is a scalable solution for industry need

Machine Learning 101

Machine Learning

= Looking for a function

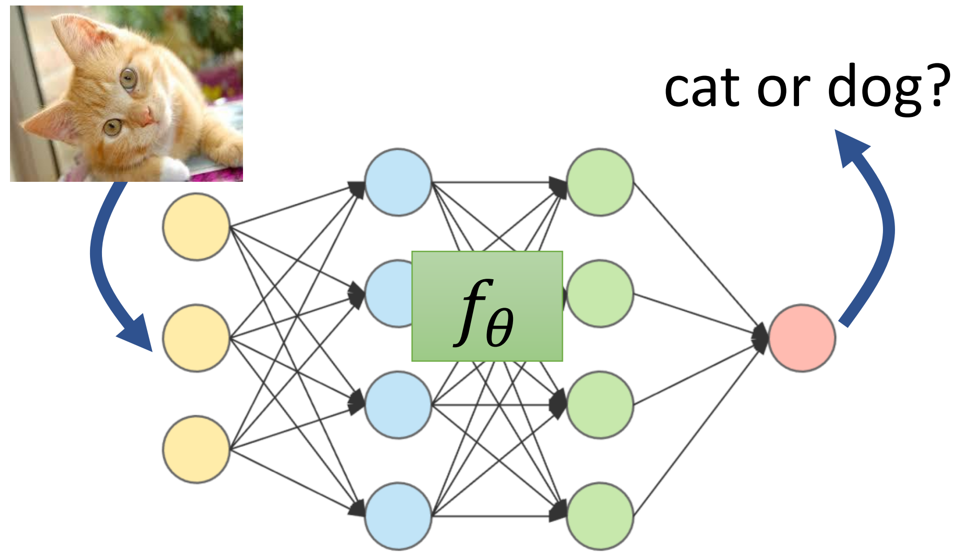
Step 1: What is learnable

Step 2: Define loss function

Step 3: Optimization

Dog-Cat Classification

$f($  $)$ = "cat"

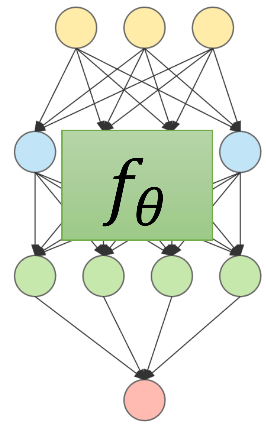


Weights and biases of neurons are learnable.

Using θ to represent the learnable parameters.

Machine Learning

Step 1: What is learnable

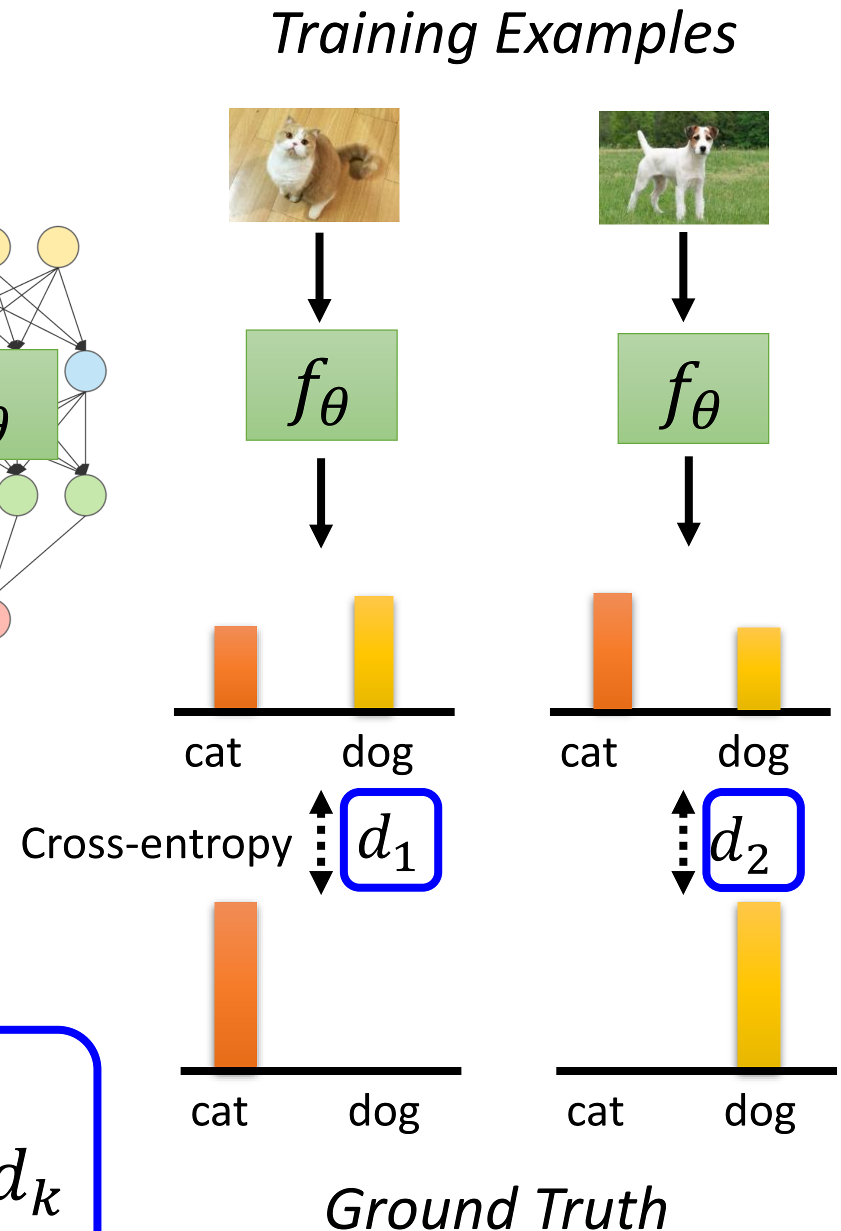


Step 2: Define loss function

$\boxed{l(\theta)}$

Step 3: Optimization

$$\boxed{l(\theta)} = \boxed{\sum_{k=1}^{K} d_k}$$



Machine Learning 101

Step 1: What is learnable

Step 2: Define loss function

Step 3: Optimization

loss: $l(\theta) = \sum_{k=1}^{K} d_k$ sum over training examples

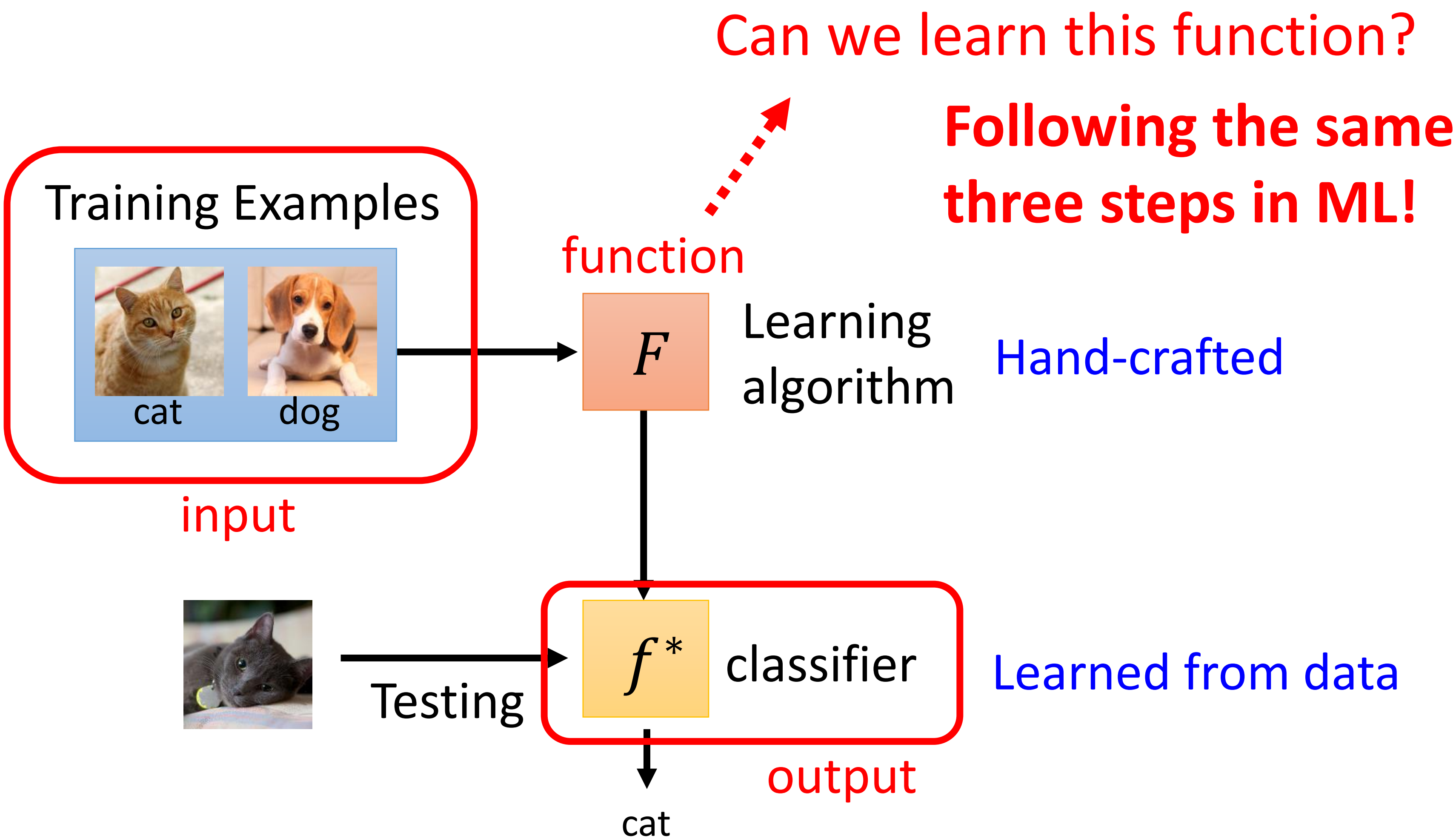
$\hat{\theta} = arg \min_{\theta} l(\theta)$

done by gradient descent

$f_{\hat{\theta}}$ is the function learned by learning algorithm from data

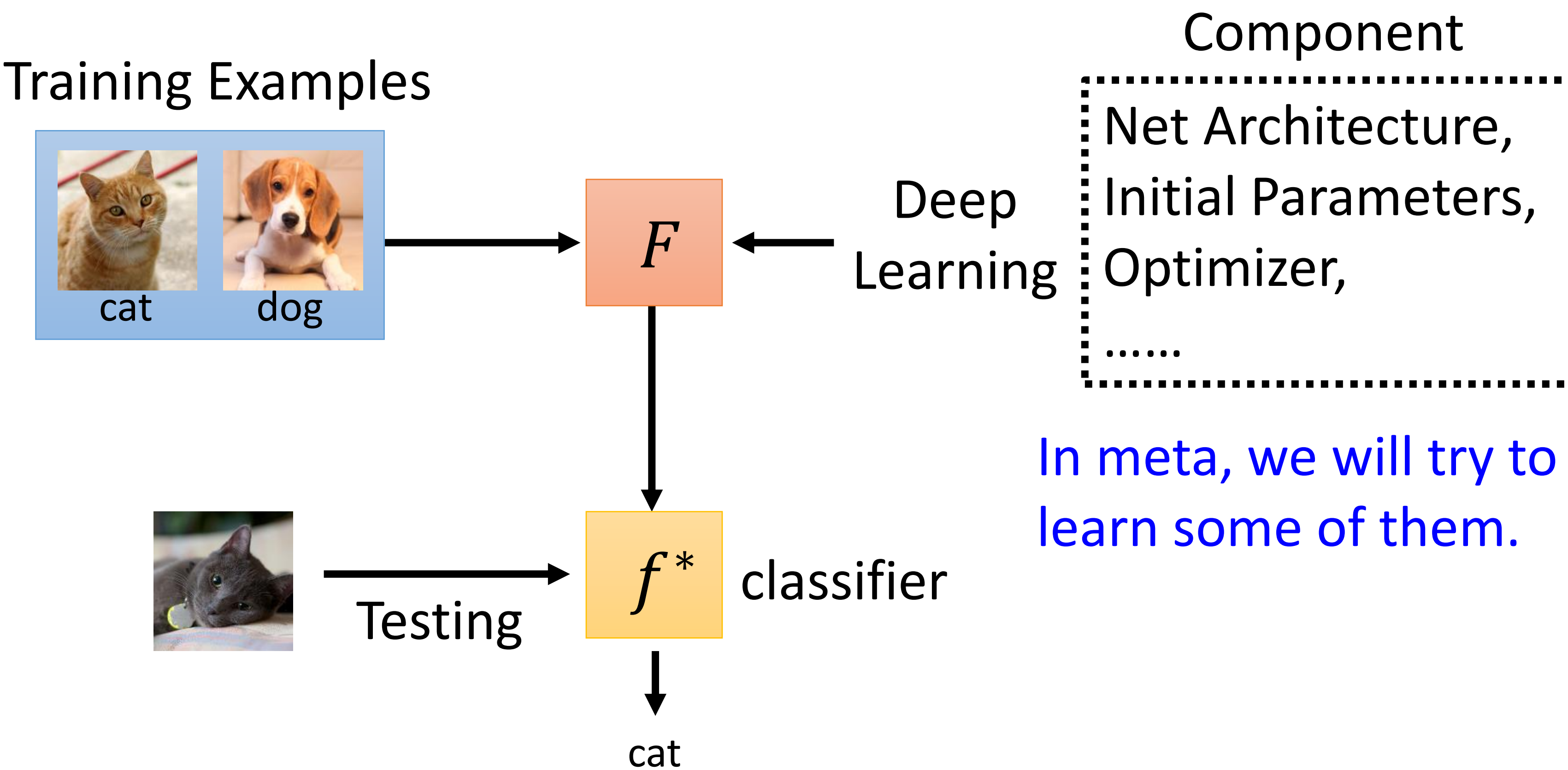
Introduction of Meta Learning

What is Meta Learning?



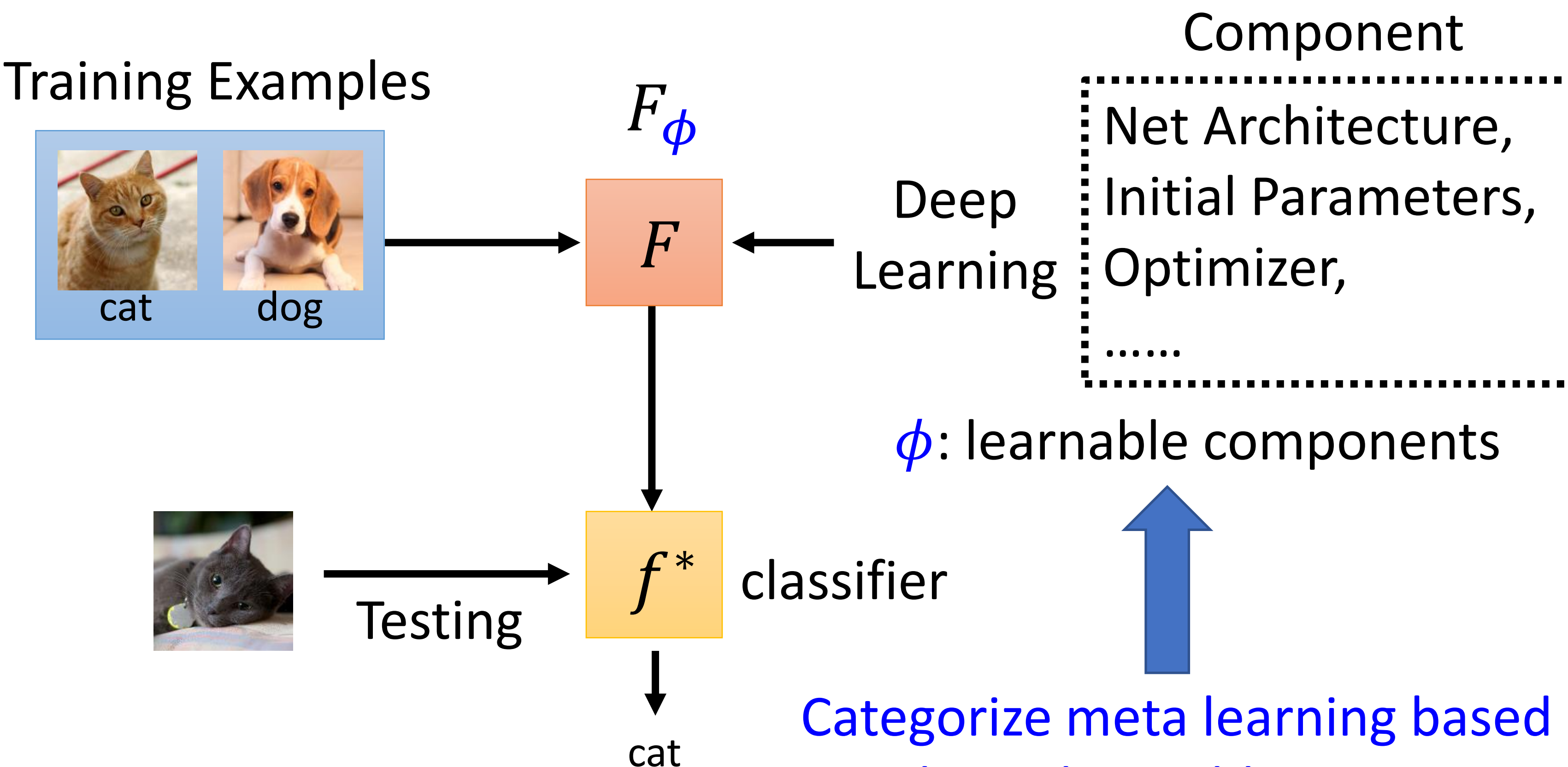
Meta Learning – Step 1

- What is ***learnable*** in a learning algorithm?



Meta Learning – Step 1

- What is ***learnable*** in a learning algorithm?



Categorize meta learning based on what is learnable

Meta Learning – Step 2

- Define ***loss function*** for ***learning algorithm*** F_{ϕ}

$$L(\phi)$$

$L(\phi)\downarrow$  $L(\phi)\uparrow$ 

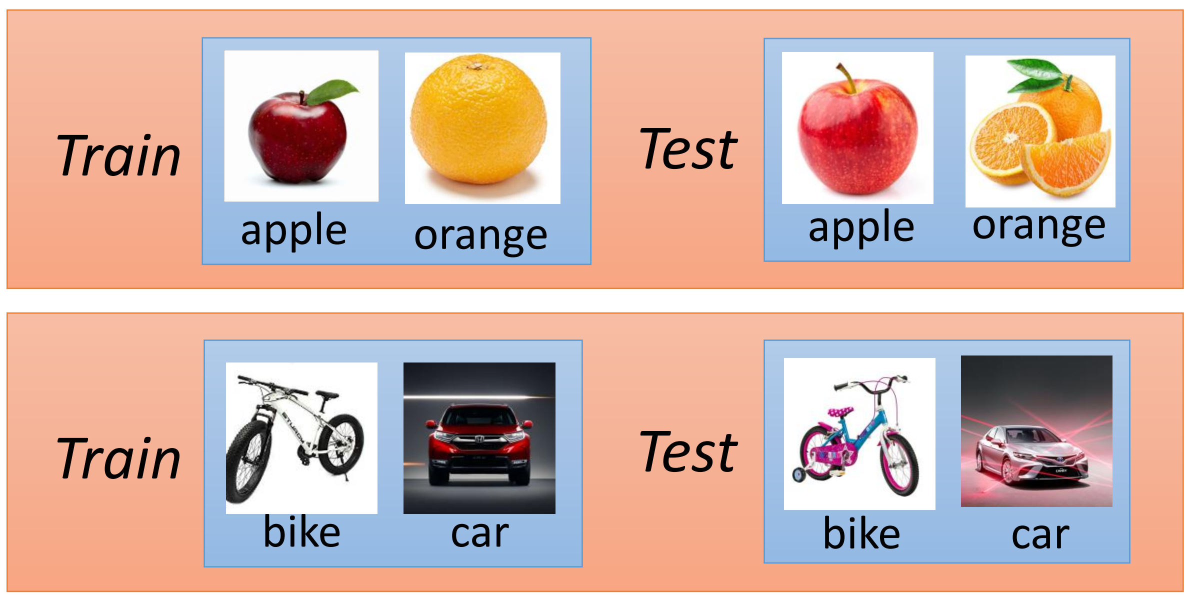
Training Tasks

Task 1

Apple & Orange

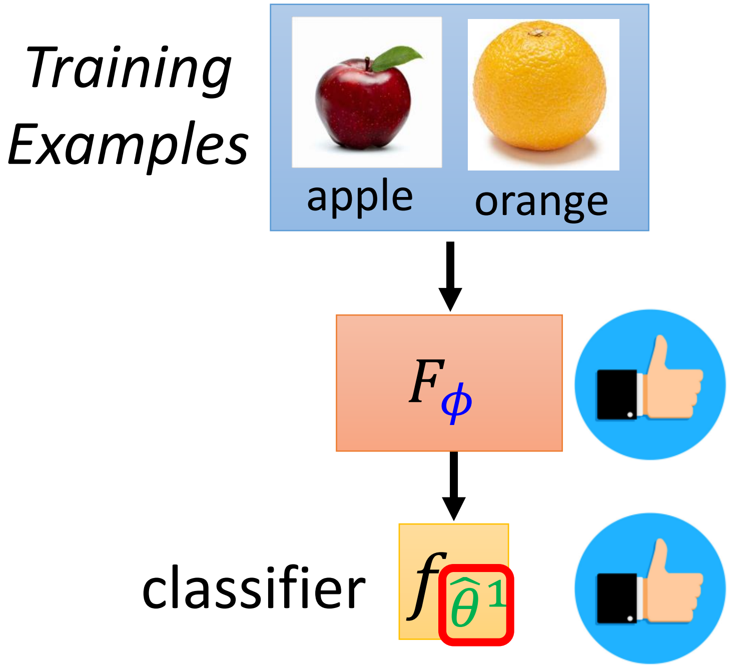
Task 2

Car & Bike



Meta Learning – Step 2

Task 1



How to define $L(\phi)$

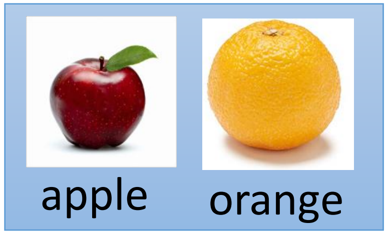
$L(\phi) \downarrow$

$\boxed{\hat{\theta}^1}$: parameters of the classifier learned by F_ϕ using the training examples of task 1

Meta Learning – Step 2

Task 1

Training Examples



F_{ϕ}

classifier $f_{\hat{\theta}^1}$

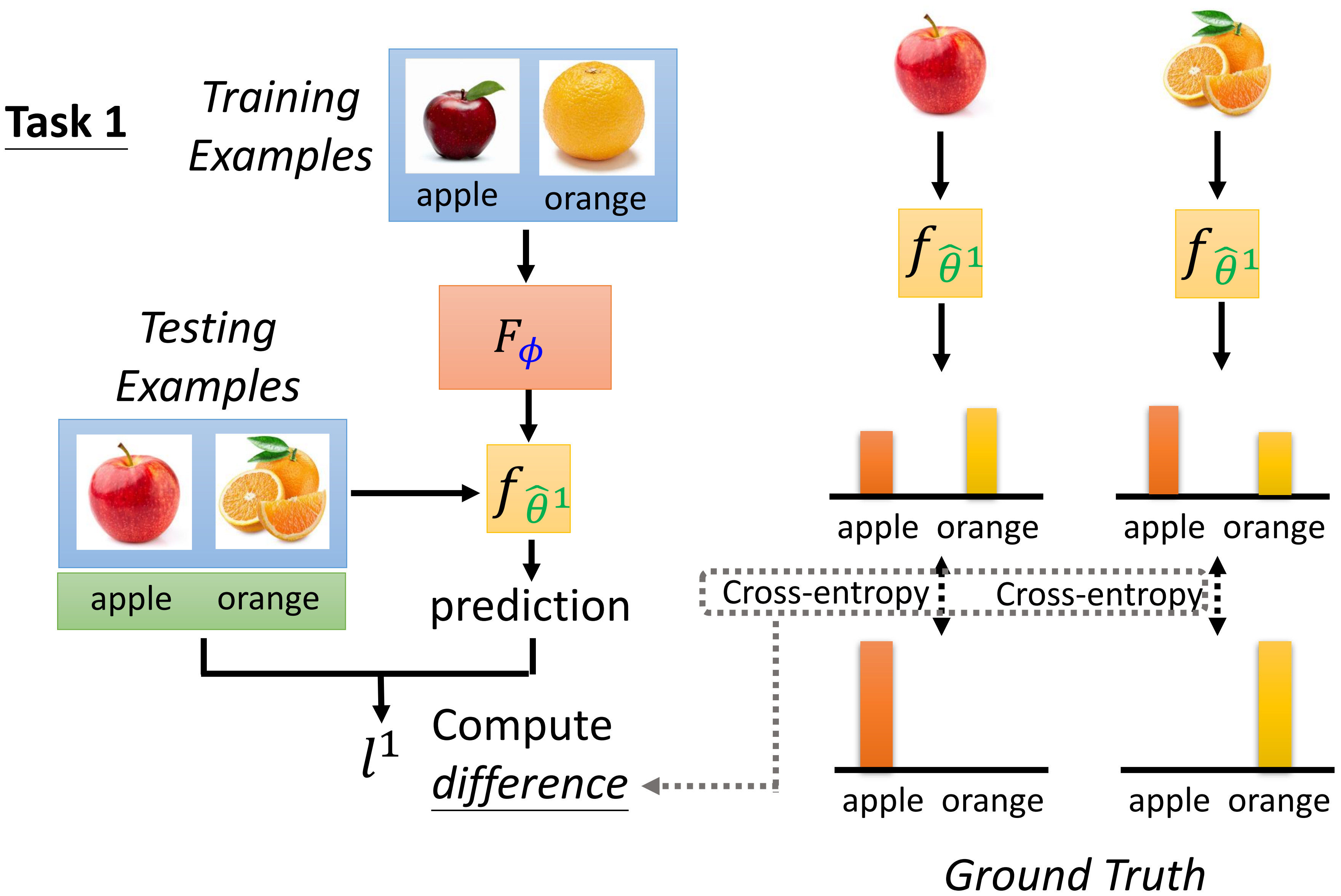
How can we know a classifier is good or bad?

Evaluate the classifier on testing set

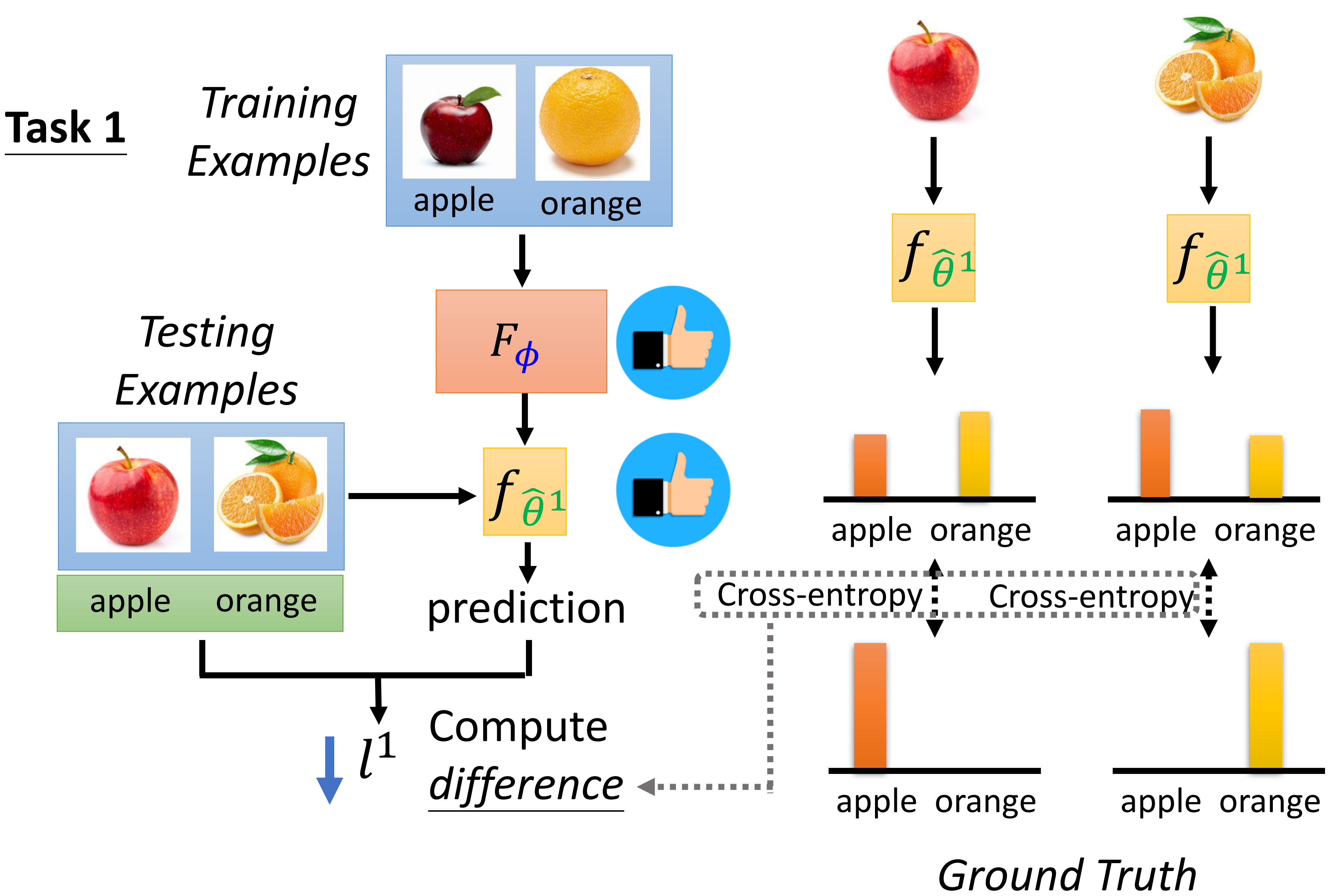
How to define $L(\phi)$

$L(\phi)$ ↑

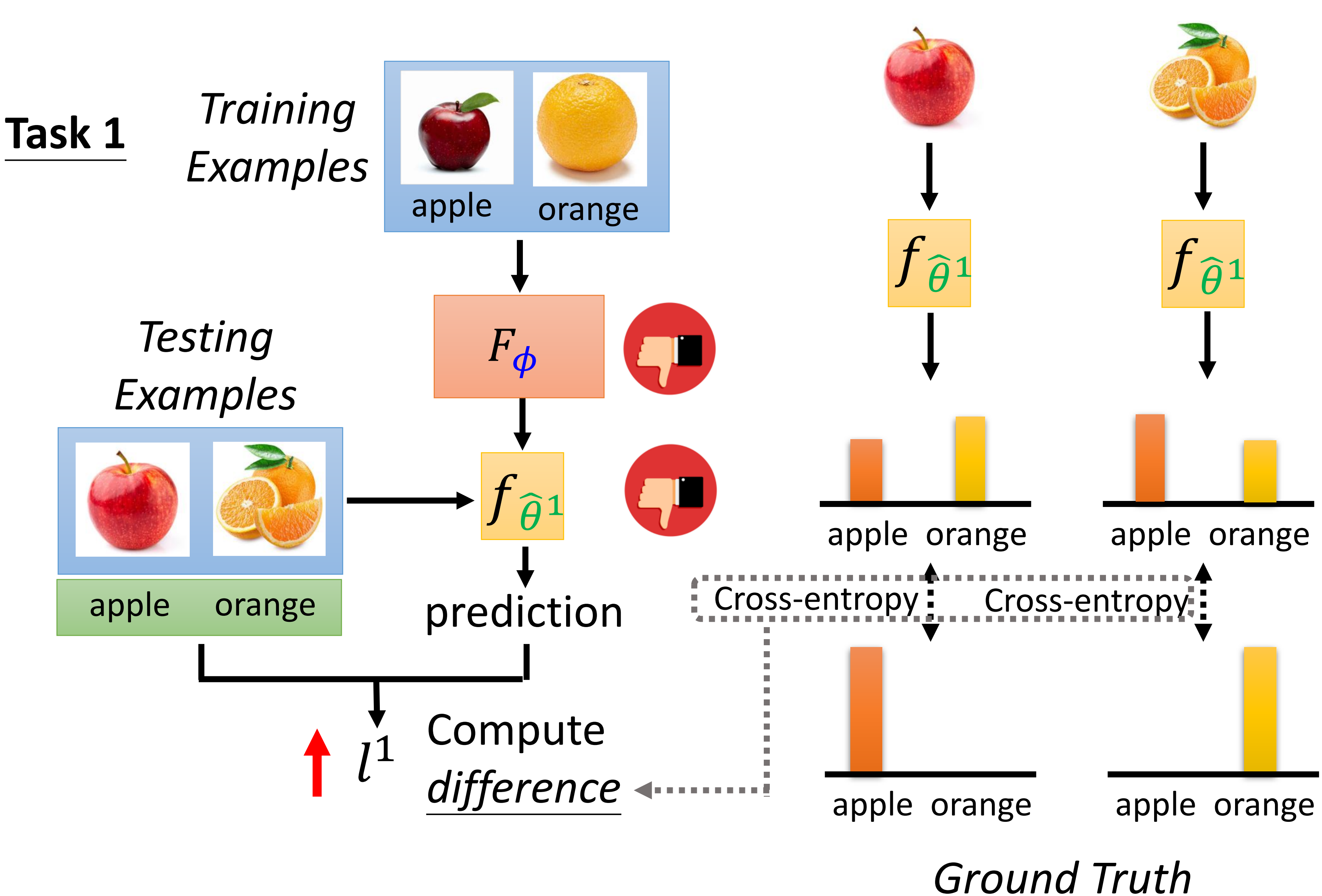
Meta Learning – Step 2



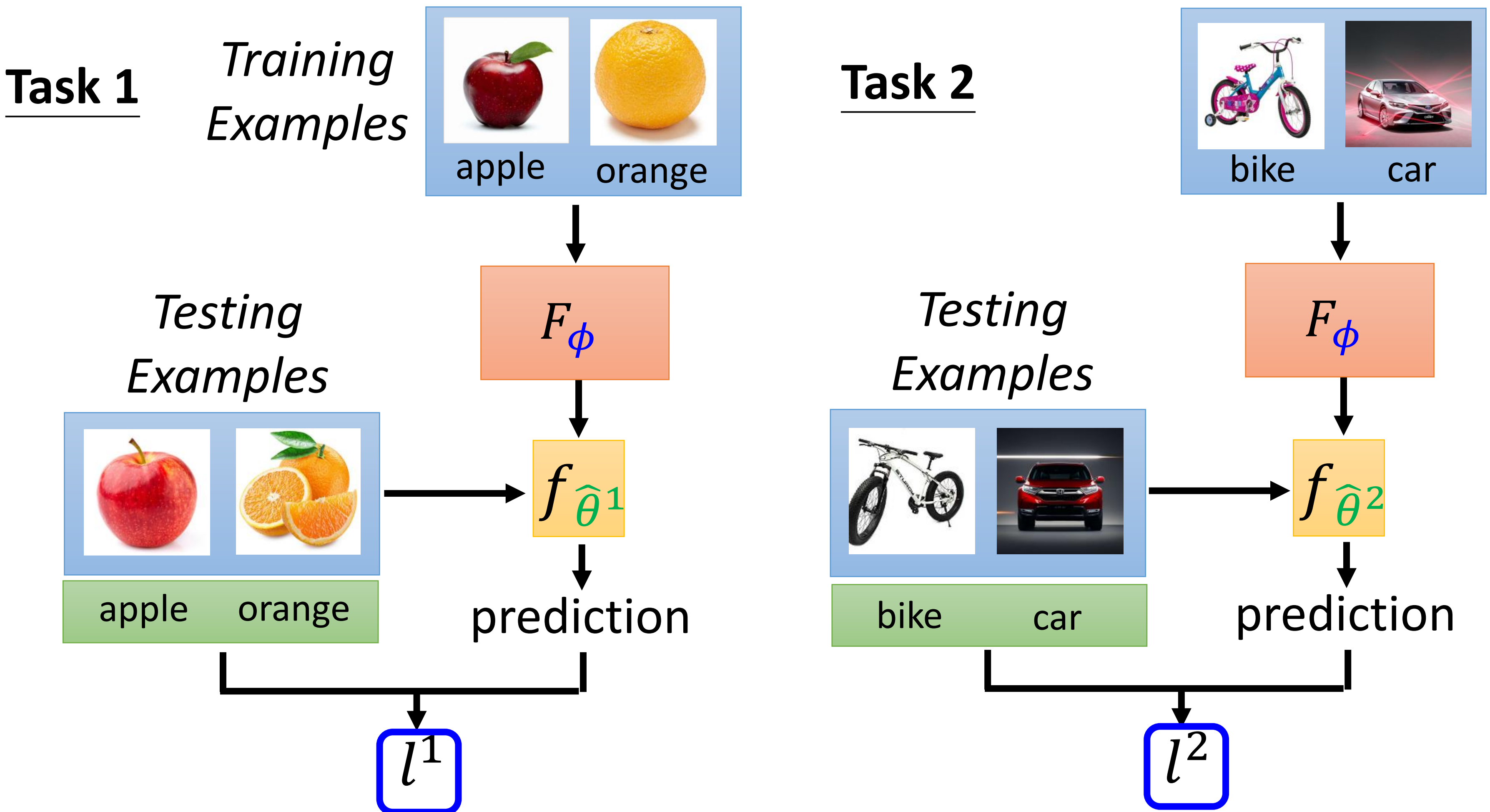
Meta Learning – Step 2



Meta Learning – Step 2

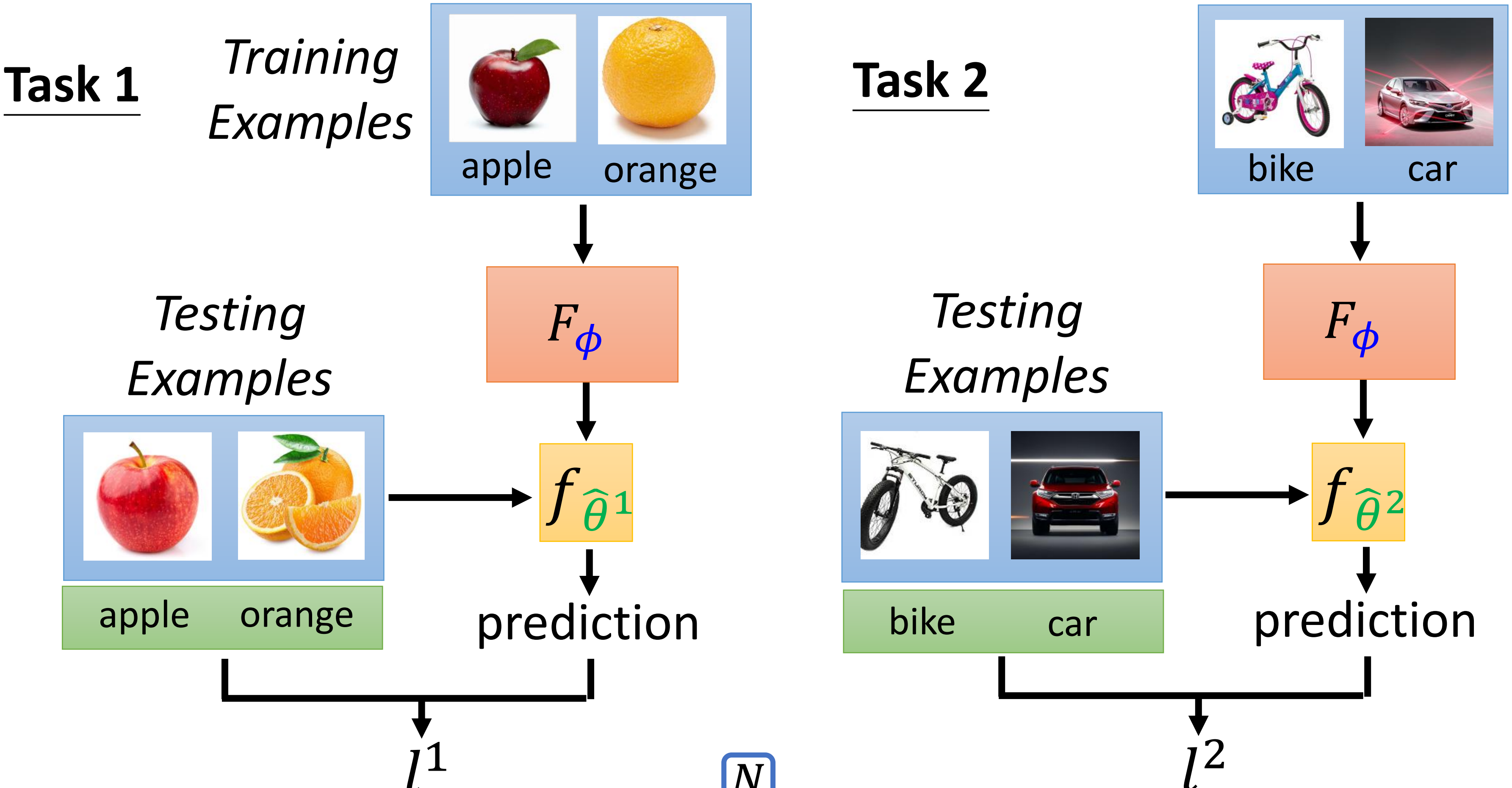


Meta Learning – Step 2



Total loss: $L(\phi) = \boxed{l^1 + l^2}$ (sum over all the training tasks)

Meta Learning – Step 2



Total loss: $L(\phi) = \sum_{n=1}^{\boxed{N}} l^n$ (N is the number of the training tasks)

Meta Learning – Step 2

Testing Examples

$f_{\hat{\theta}^1}$ $f_{\hat{\theta}^1}$

Task 1

In typical ML, you compute the loss based on training examples

In meta, you compute the loss based on testing examples

Hold on! You use testing examples during training???

apple orange

prediction

l^1 Compute *difference*

apple orange apple orange

Ground Truth

Meta Learning – Step 2

Testing Examples

$f_{\hat{\theta}^1}$ $f_{\hat{\theta}^1}$

Task 1

In typical ML, you compute the loss based on training examples

In meta, you compute the loss based on testing examples of training tasks.

apple orange

prediction

l^1

Compute *difference*

Meta Learning – Step 3

- Loss function for learning algorithm $L(\phi) = \sum_{n=1}^{N} l^n$
- Find ϕ that can minimize $L(\phi)$ $\hat{\phi} = arg \min_{\phi} L(\phi)$
- Using the optimization approach you know

If you know how to compute $\partial L(\phi) / \partial \phi$

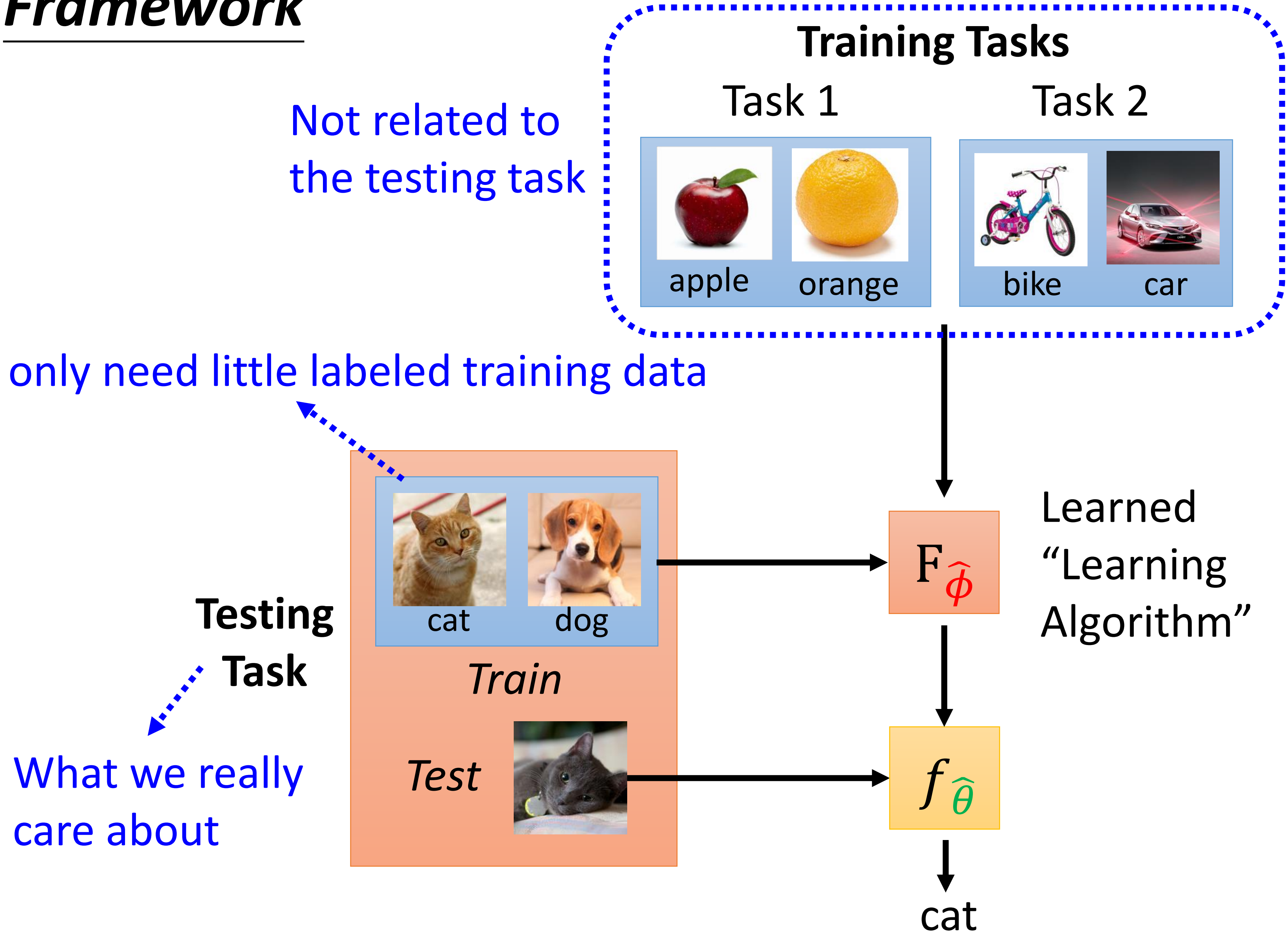
Gradient descent is your friend.

What if $L(\phi)$ is not differentiable?

Reinforcement Learning / Evolutionary Algorithm

Now we have a learned "learning algorithm" $F_{\hat{\phi}}$

Framework


Training Tasks
Task 1
Task 2
apple
orange
bike
car
Not related to the testing task
only need little labeled training data
Testing Task
What we really care about
cat
dog
Train
Test
$F_{\hat{\phi}}$
Learned "Learning Algorithm"
$f_{\hat{\theta}}$
cat

ML v.s. Meta

Goal

Machine Learning ≈ find a function f

Dog-Cat Classification

$f($  $) =$ "cat"

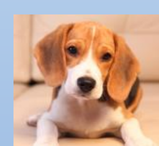
Meta Learning

≈ find a function F that finds a function f

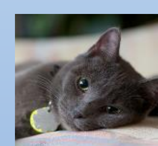
Learning Algorithm

$F($

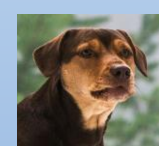
cat



dog



cat



dog

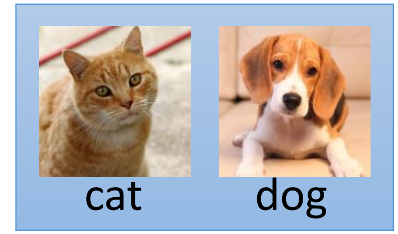
Training Examples

$) = f$

Training Data

Machine Learning

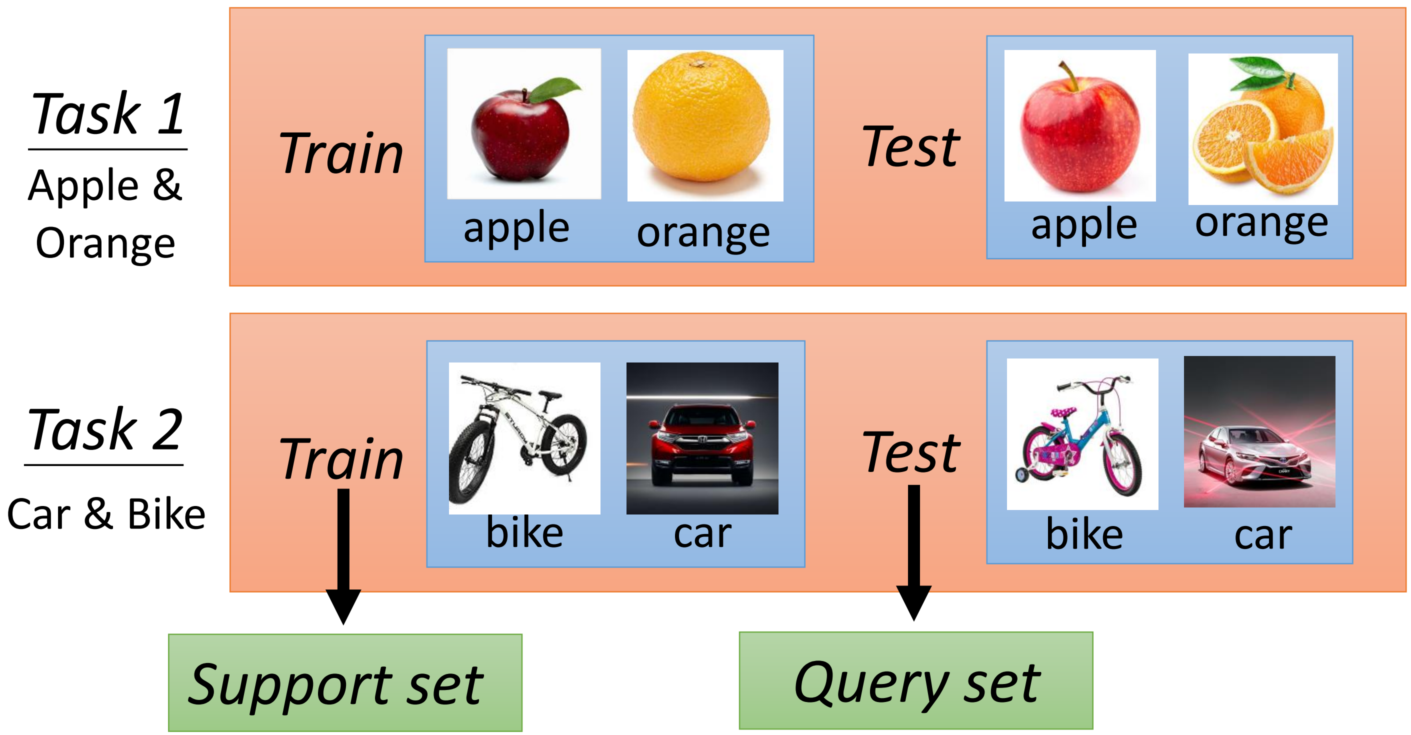
One task



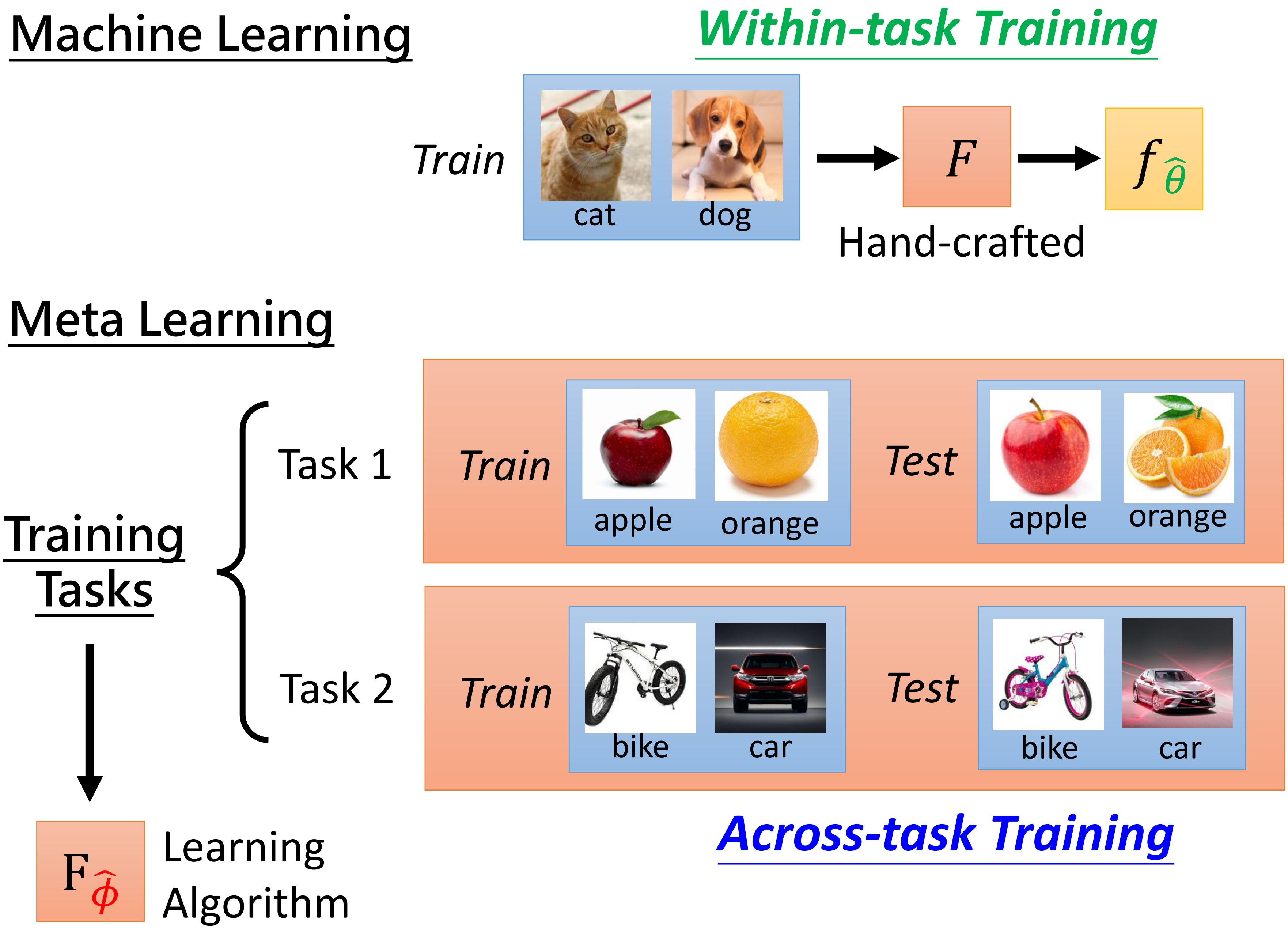
Train

Meta Learning

Training tasks

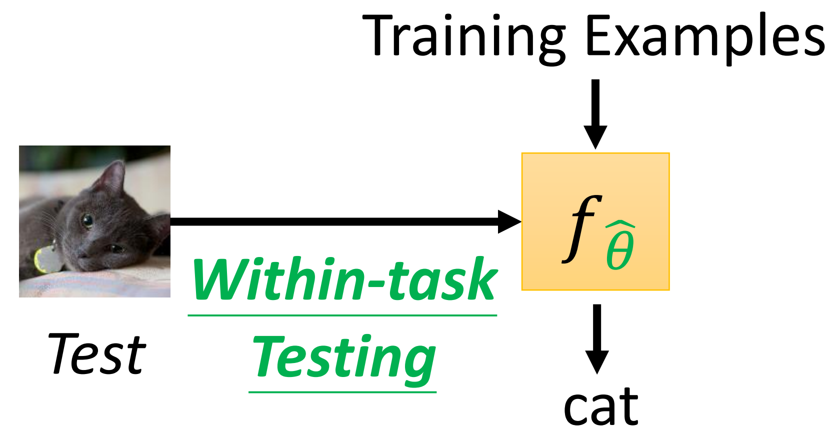


(in the literature of "*learning to compare*")

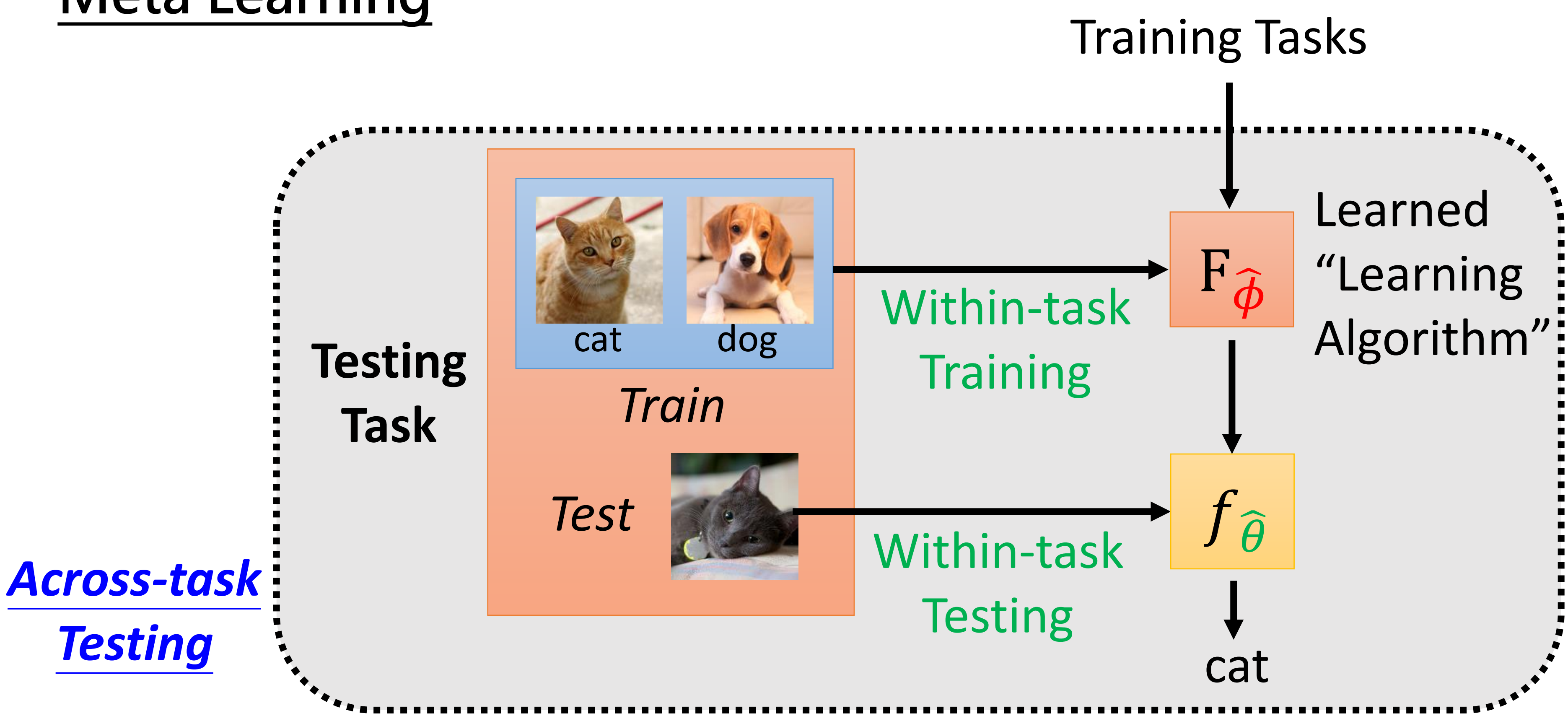


Machine Learning
Within-task Training
Train
cat
dog
F
$f_{\hat{\theta}}$
Hand-crafted
Meta Learning
Training Tasks
Task 1
Train
apple
orange
Test
apple
orange
Task 2
Train
bike
car
Test
bike
car
$F_{\hat{\phi}}$
Learning Algorithm
Across-task Training

Machine Learning



Meta Learning



Loss

Machine Learning

$$l(\theta) = \boxed{\sum_{k=1}^{K} d_k}$$

Sum over training examples in one task

Meta Learning

$$L(\phi) = \boxed{\sum_{n=1}^{N} \boxed{l^n}}$$

Sum over testing examples in one task

Sum over training tasks

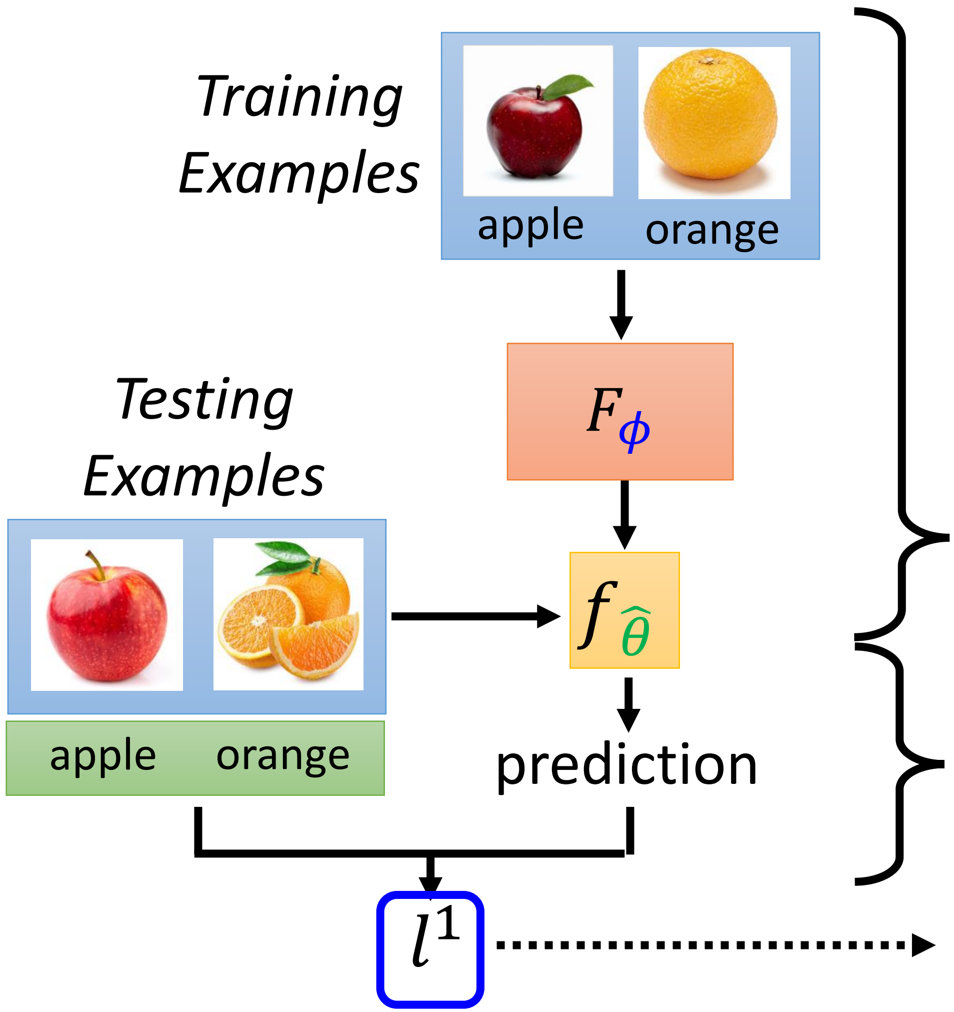
$$L(\phi) = \sum_{n=1}^{N} \boxed{l^n}$$

If your optimization method needs to compute $L(\phi)$

Outer Loop in *"Learning to initialize"*

Across-task training includes **within-task training and testing**

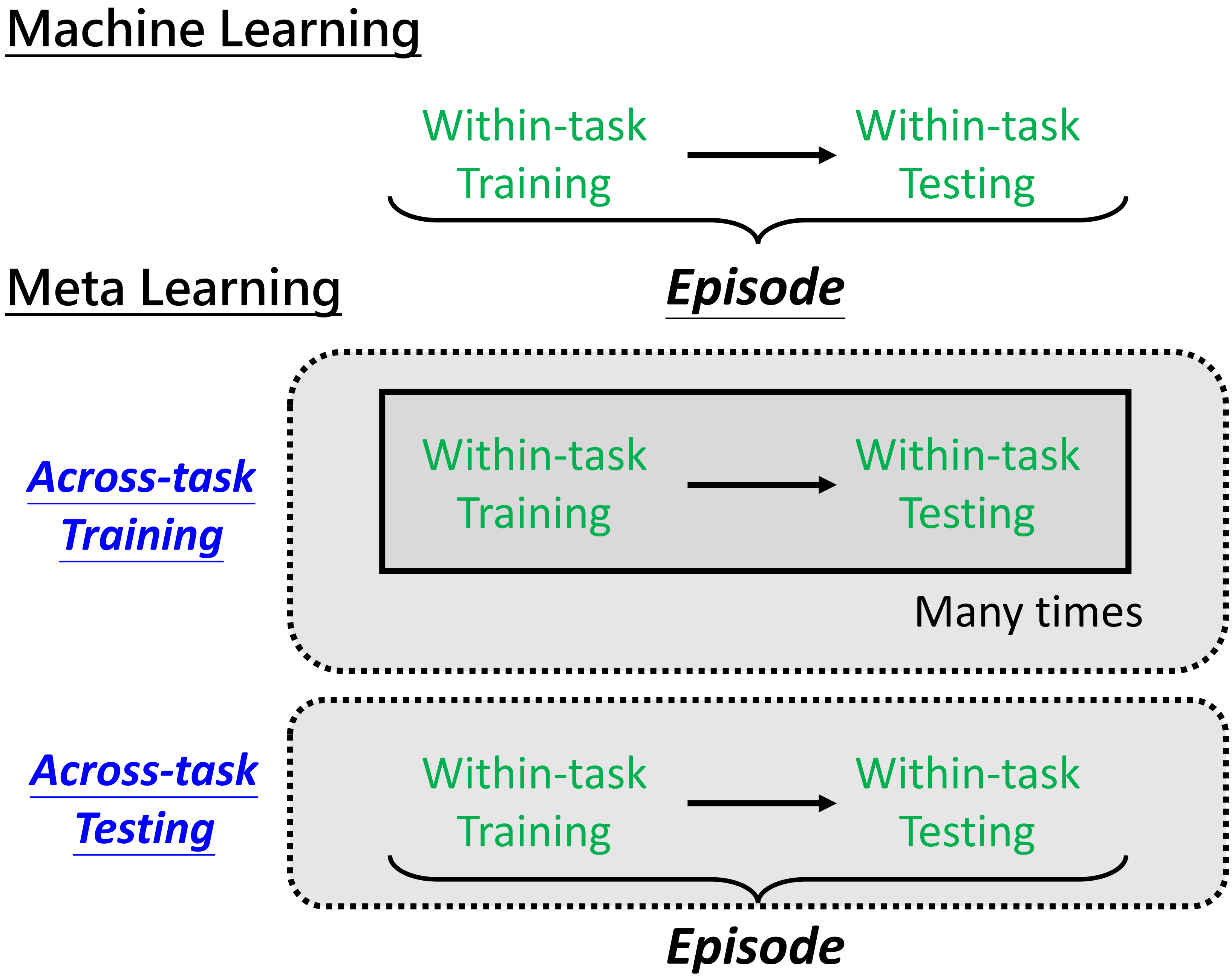
Inner Loop in *"Learning to initialize"*



Within-task Training

Within-task Testing

To compute the loss



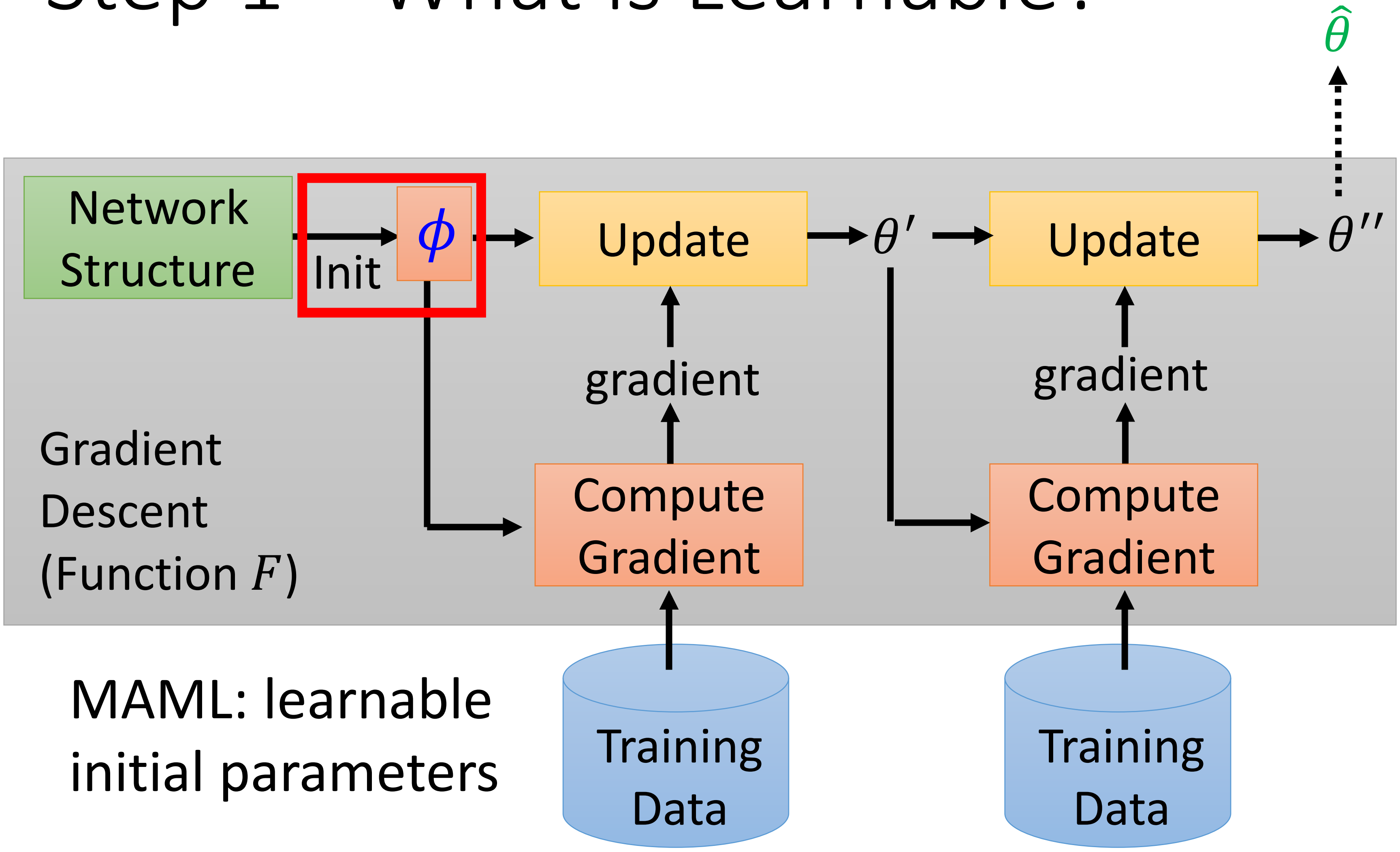
Machine Learning
Within-task Training
Within-task Testing
Episode
Meta Learning
Across-task Training
Within-task Training
Within-task Testing
Many times
Across-task Testing
Within-task Training
Within-task Testing
Episode

Learning to Initialize

Model-Agnostic Meta-Learning (MAML)

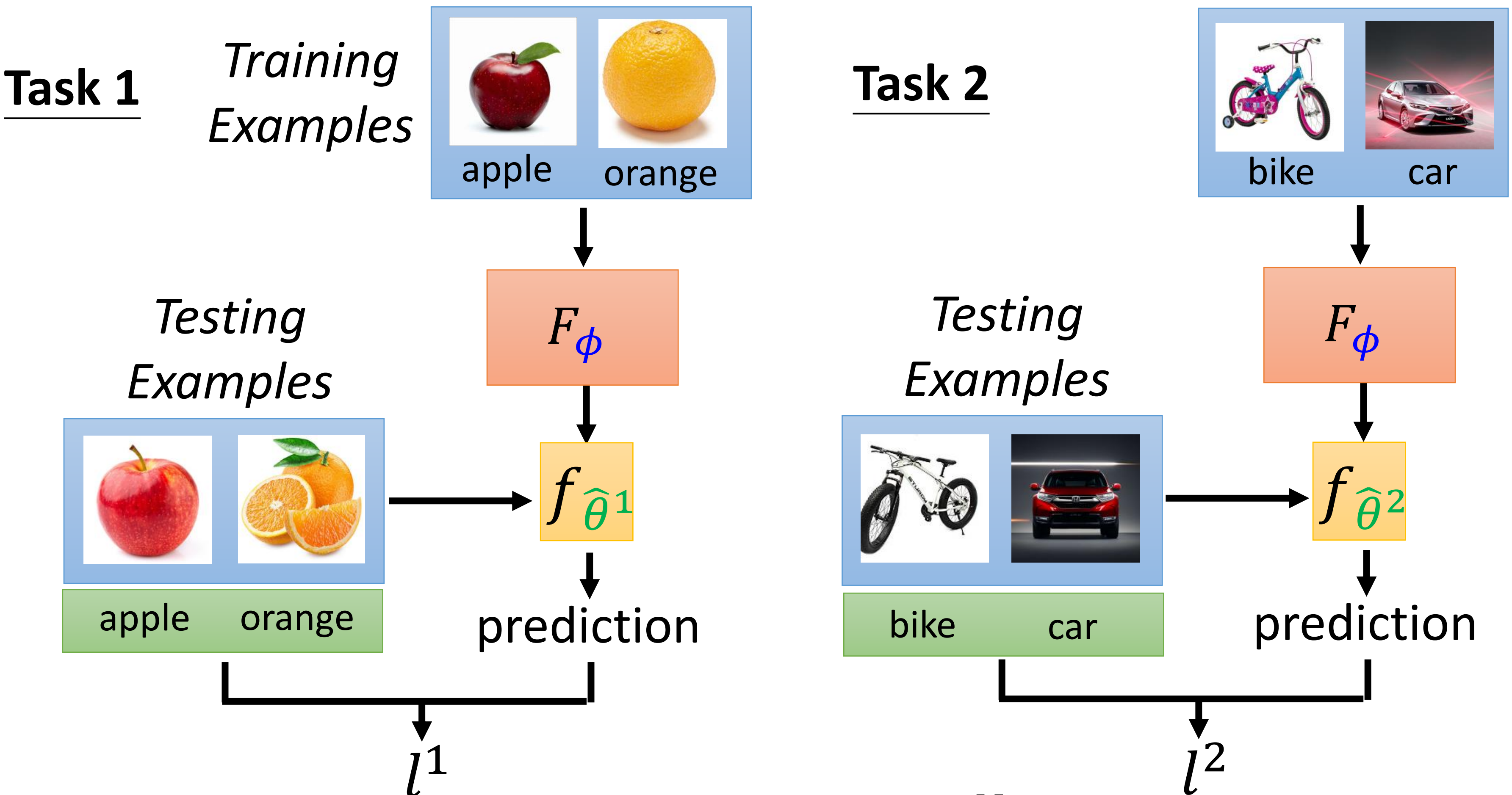
Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”, ICML, 2017

Step 1 – What is Learnable?



MAML: learnable initial parameters

Step 2 – Loss Function



Total loss: $L(\phi) = \sum_{n=1}^{N} l^n$

Step 3 – Optimization

$$L(\phi) = \sum_{n=1}^{N} l^n$$

$$\boxed{\phi \leftarrow \phi - \eta \nabla_\phi L(\phi)}$$

Across-task training
(outer loop in MAML)

$$\nabla_\phi L(\phi) = \nabla_\phi \sum_{n=1}^{N} l^n = \sum_{n=1}^{N} \nabla_\phi l^n$$

$$\nabla_\phi l = \begin{bmatrix} \partial l / \partial \phi_1 \\ \partial l / \partial \phi_2 \\ \vdots \\ \boxed{\partial l / \partial \phi_i} \\ \vdots \end{bmatrix}$$

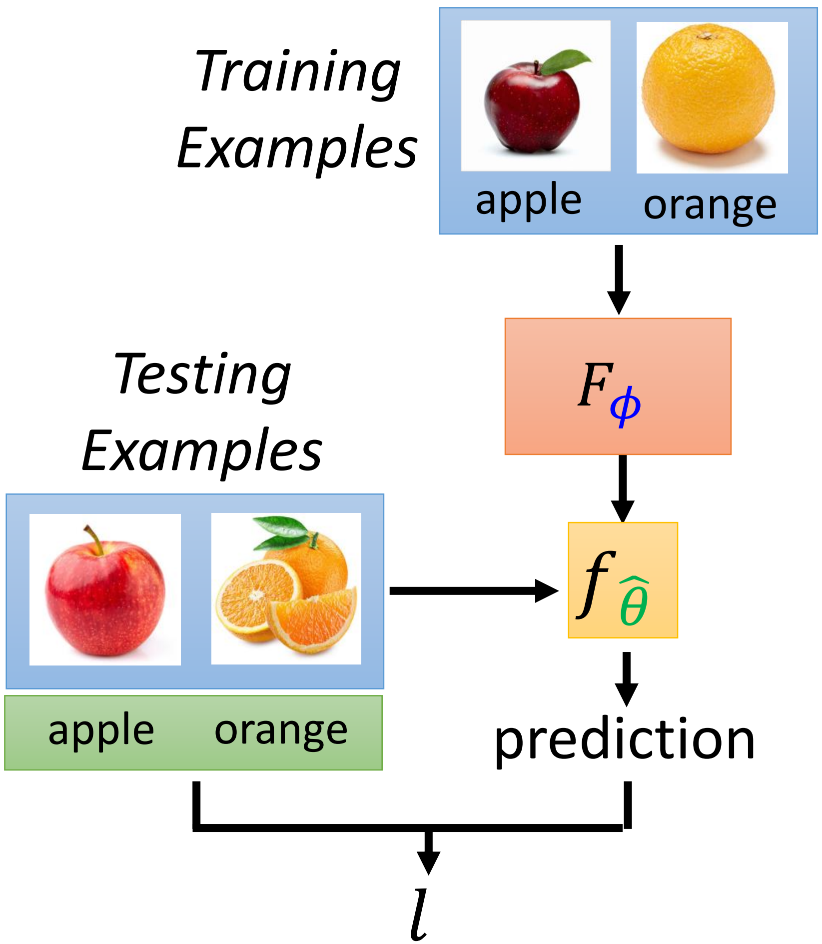
How to compute $\nabla_\phi l$

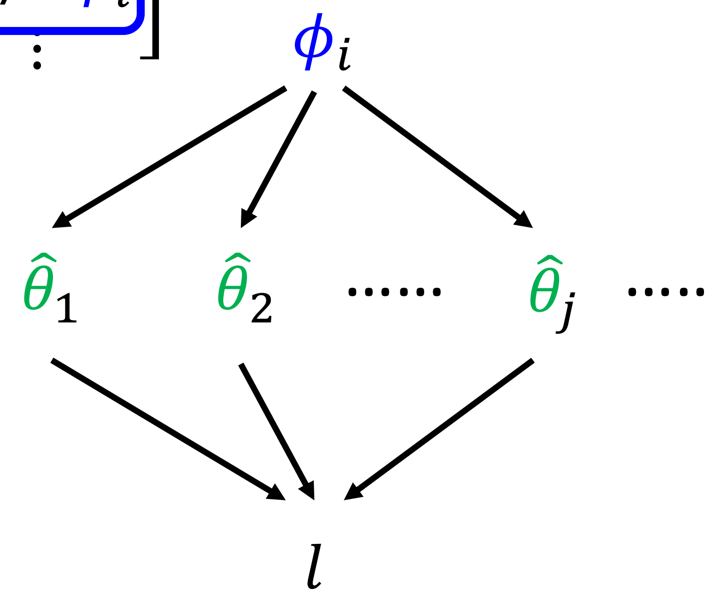
(^{n} is ignored here)

ϕ_i : the i-th parameter of ϕ

Step 3 – Optimization

$$\phi \leftarrow \phi - \eta \nabla_{\phi} L(\phi) \qquad \nabla_{\phi} l = \begin{bmatrix} \partial l / \partial \phi_1 \\ \partial l / \partial \phi_2 \\ \vdots \\ \boxed{\partial l / \partial \phi_i} \\ \vdots \end{bmatrix}$$





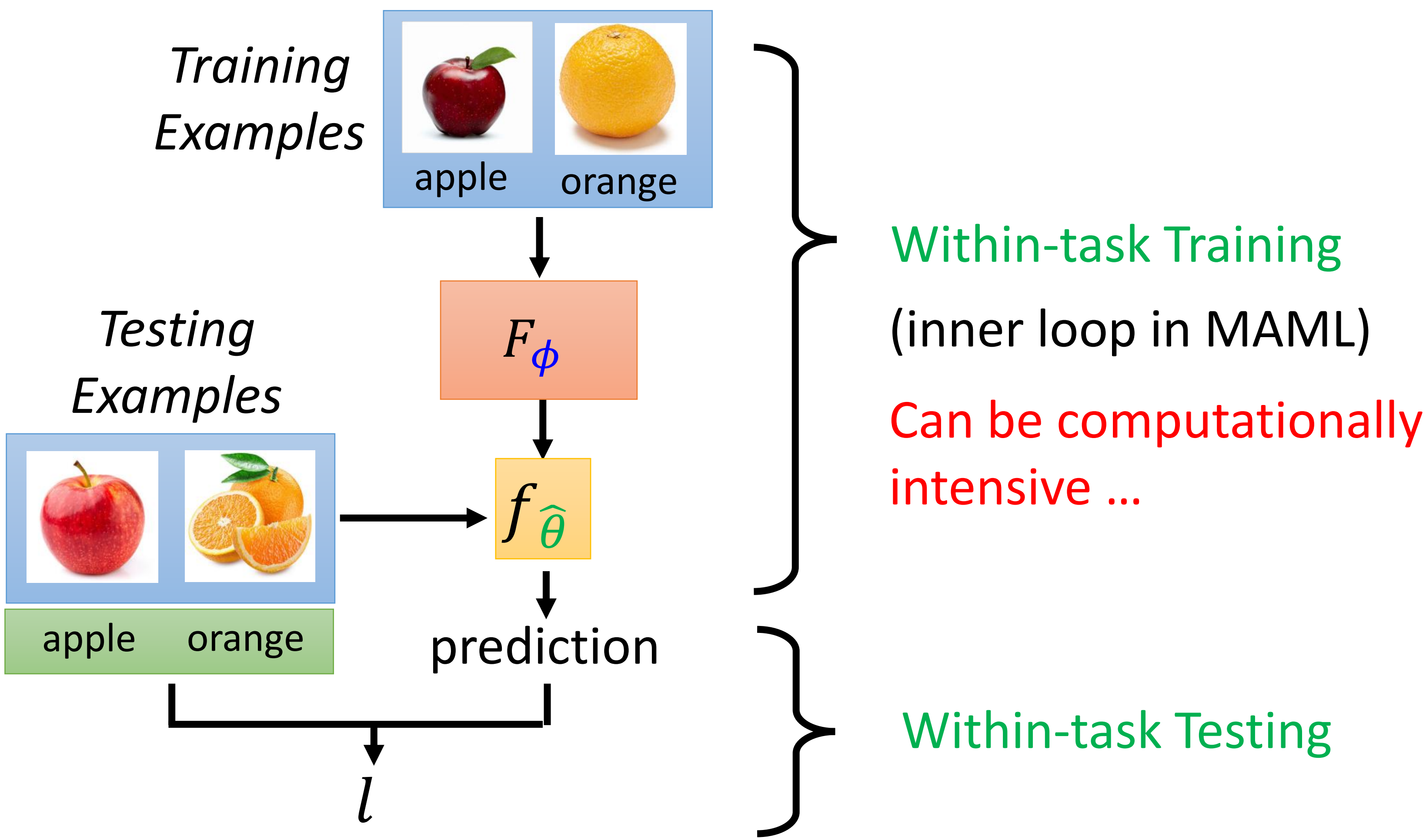
$$\frac{\partial l}{\partial \phi_i} = \sum_j \boxed{\frac{\partial l}{\partial \hat{\theta}_j}} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$

Sum over the parameters in $\hat{\theta}$

Step 3 – Optimization

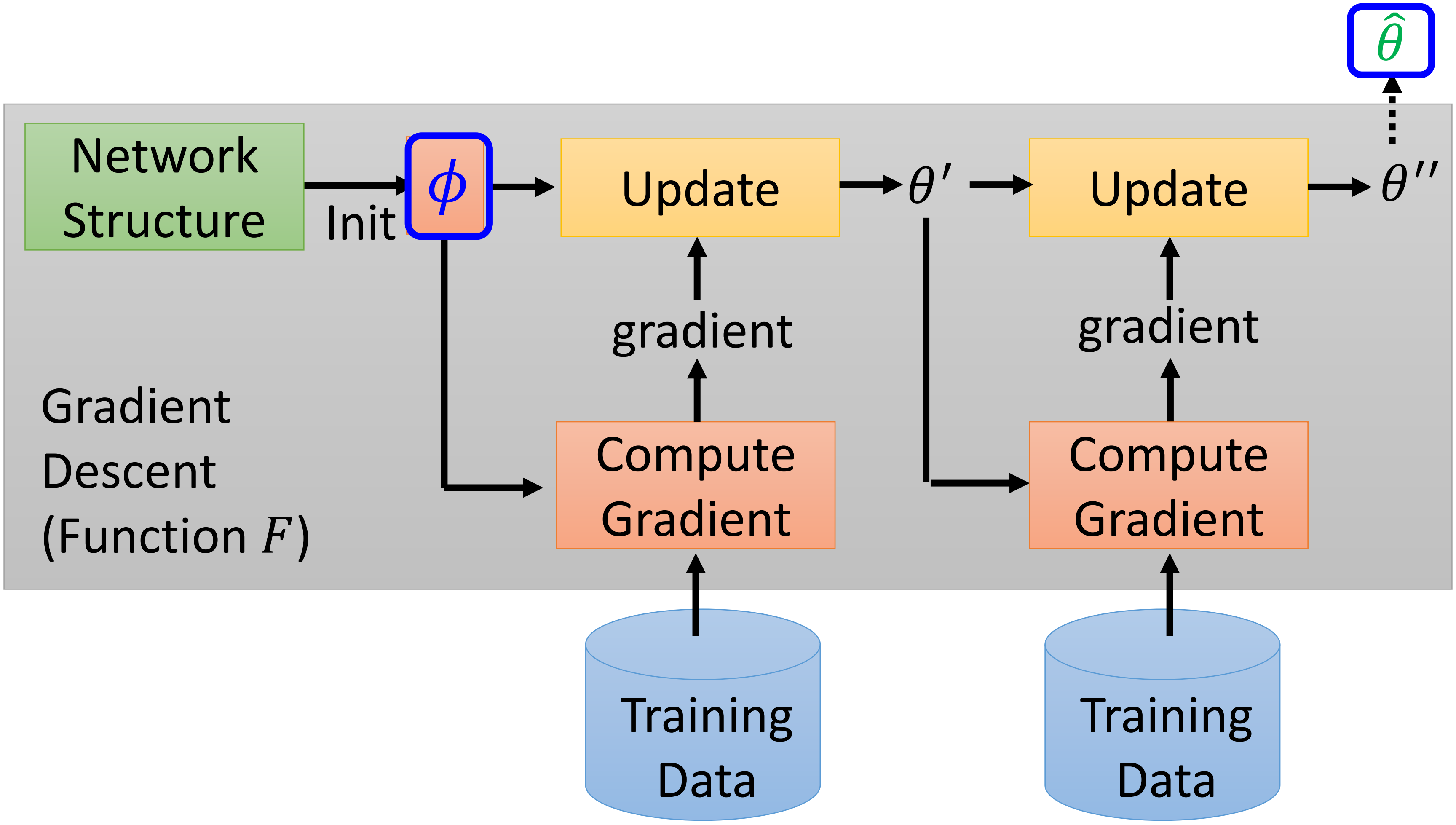
$$\phi \leftarrow \phi - \eta \nabla_{\phi} L(\phi)$$

$$\frac{\partial l}{\partial \phi_i} = \sum_j \boxed{\frac{\partial l}{\partial \hat{\theta}_j}} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$



Step 3 – Optimization

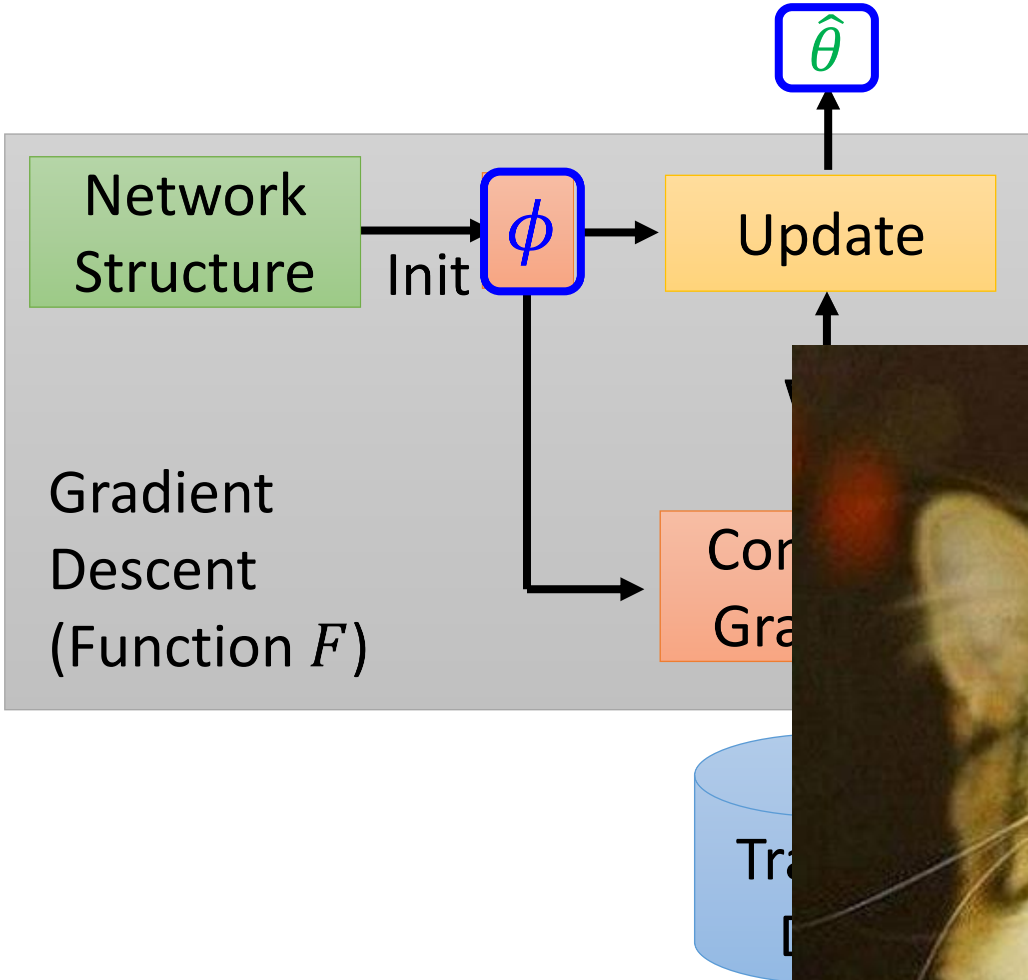
Can be computationally intensive …



Step 3 – Optimization

Here within-task training only uses **one step** ...

$$\hat{\theta} = \phi - \varepsilon \nabla_{\phi} l(\phi)$$



very different from usual gradient descent ...

Step 3 – Optimization

Here within-task training only uses **one step** …

$$\hat{\theta} = \phi - \varepsilon \nabla_{\phi} l(\phi)$$

- Fast … Fast … Fast …
- Good to train a model with one step. ☺
- In few-shot learning scenario, we have limited training examples in each task.

  → only update once to prevent overfitting
- Within-task training is different in across-task training and testing

In across-task **training**, within-task training updates once
In across-task **testing**, within-task training updates many times

Step 3 – Optimization

$$\frac{\partial l}{\partial \phi_i} = \sum_j \frac{\partial l}{\partial \hat{\theta}_j} \boxed{\frac{\partial \hat{\theta}_j}{\partial \phi_i}}$$

Here within-task training only uses **one step** …

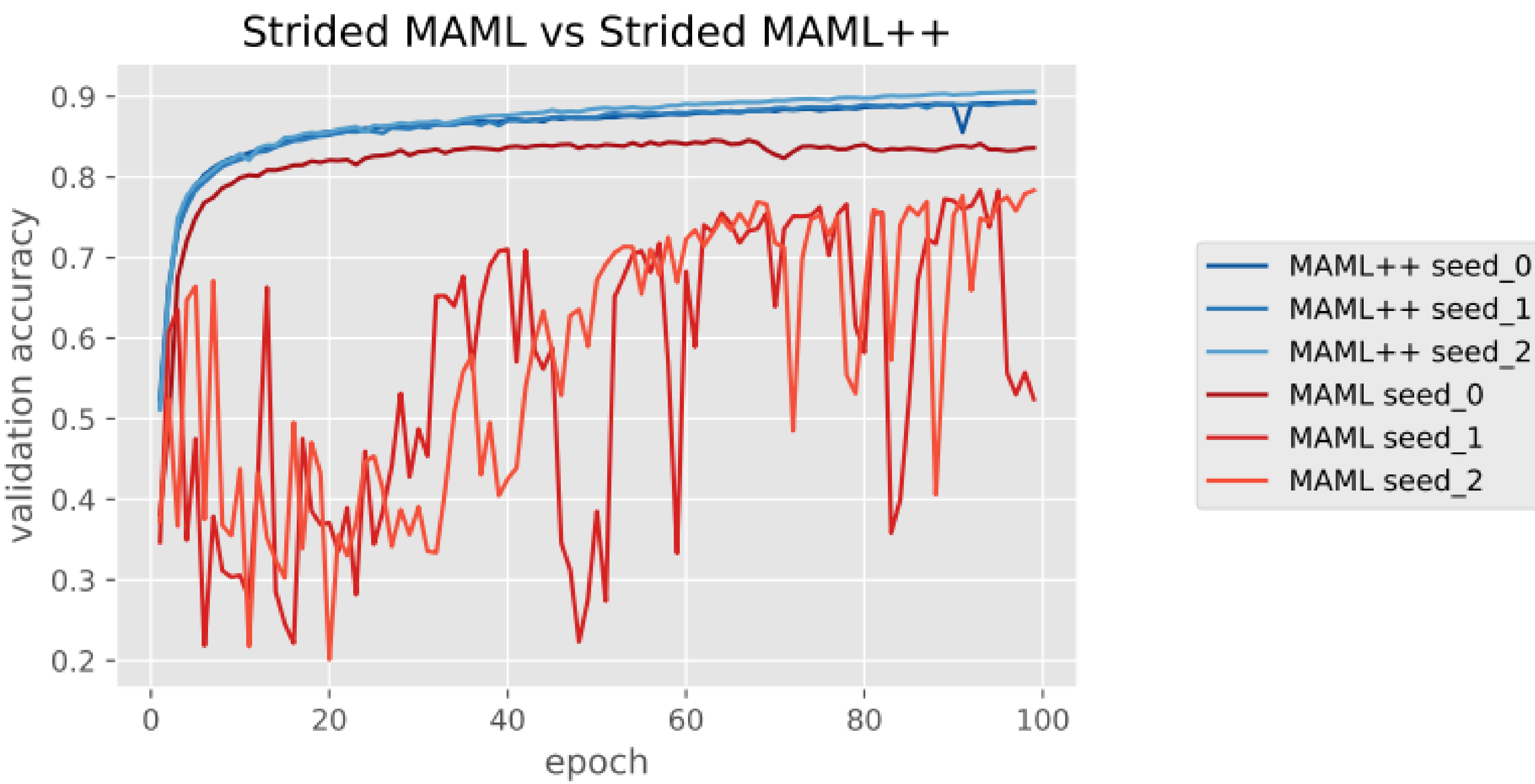
$$\hat{\theta} = \phi - \varepsilon \nabla_{\phi} l(\phi)$$

The first-order approximation can be further used

$$i \neq j: \quad \frac{\partial \hat{\theta}_j}{\partial \phi_i} = -\varepsilon \cancel{\frac{\partial l(\phi)}{\partial \phi_i \partial \phi_j}} \approx 0$$

$$i = j: \quad \frac{\partial \hat{\theta}_j}{\partial \phi_i} = 1 - \varepsilon \cancel{\frac{\partial l(\phi)}{\partial \phi_i \partial \phi_j}} \approx 1$$

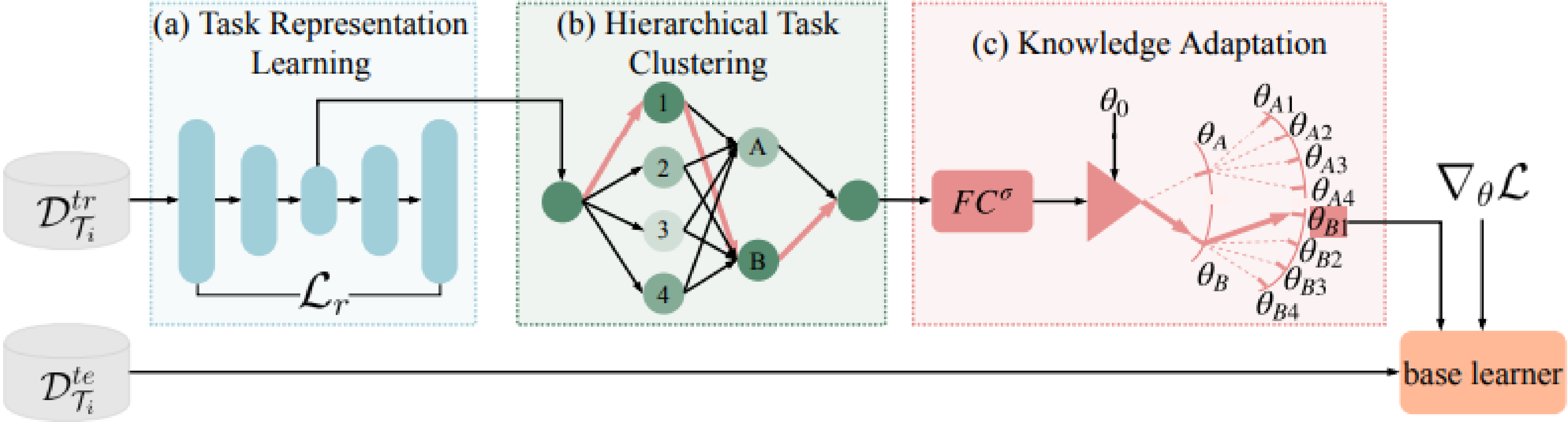
How to train your ~~Dragon~~ MAML



Antreas Antoniou, Harrison Edwards, Amos Storkey, How to train your MAML, ICLR, 2019

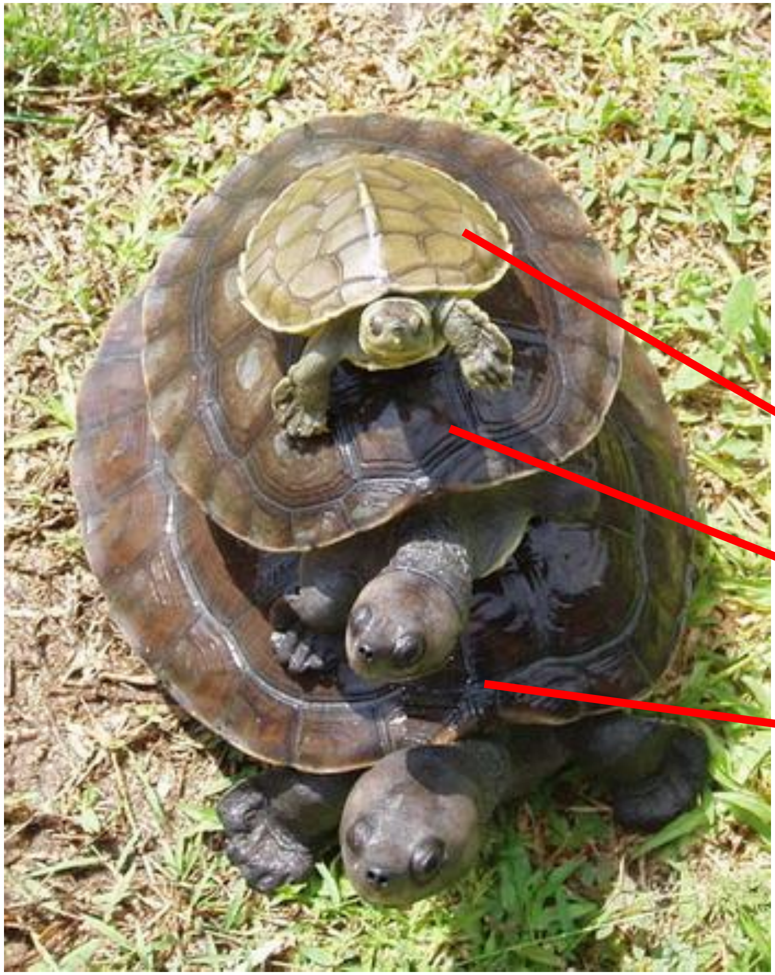
One “Initialization” fits all?

- Task conditioning: Give different tasks different initialization.



Huaxiu Yao, Ying Wei, Junzhou Huang, Zhenhui Li, Hierarchically Structured Meta-learning, ICML, 2019

Initialization of "Learn to initialize"



Turtles all the way down?

- MAML learns the initialization parameter ϕ by gradient descent
- What is the initialization parameter ϕ^0 for ϕ?

Gradient descent

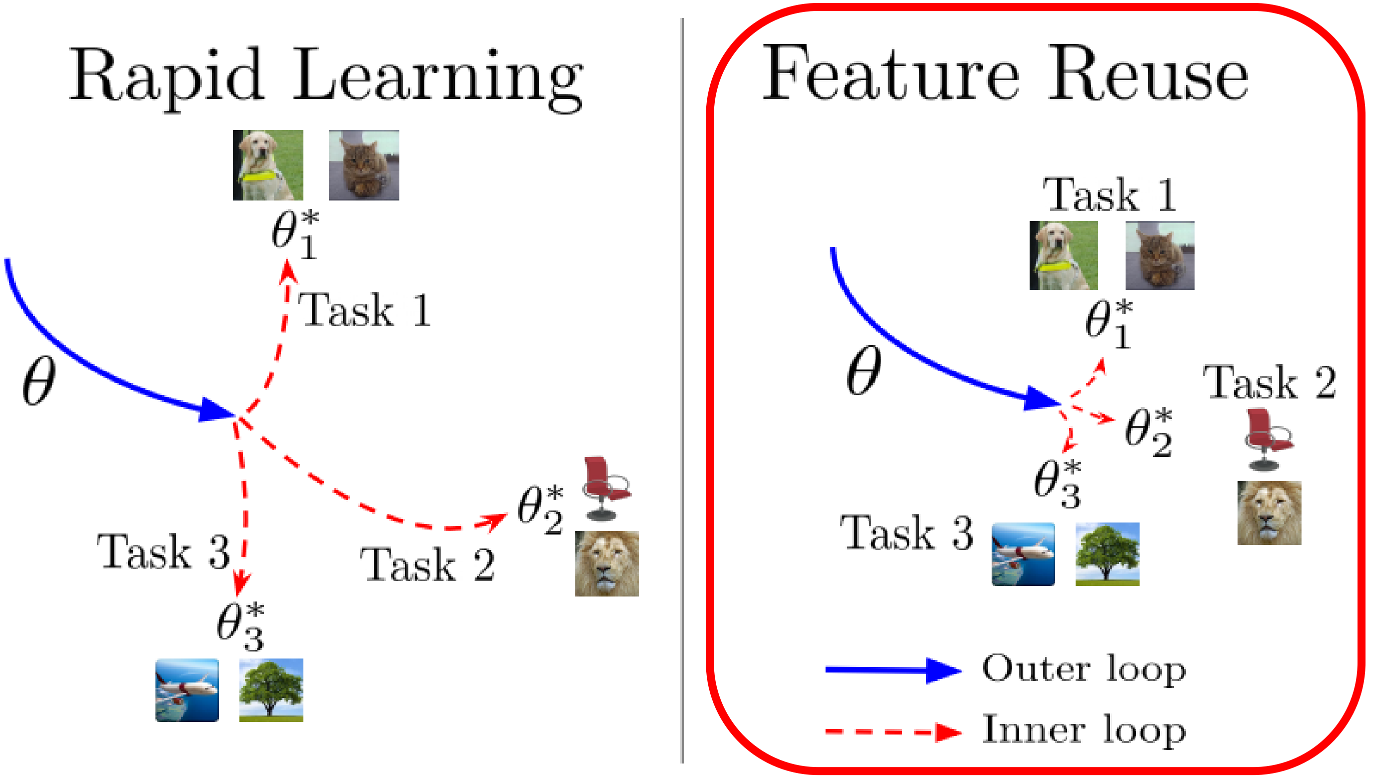
Learn to initialize

Learn to learn to initialize

⋮

MAML is good because ……

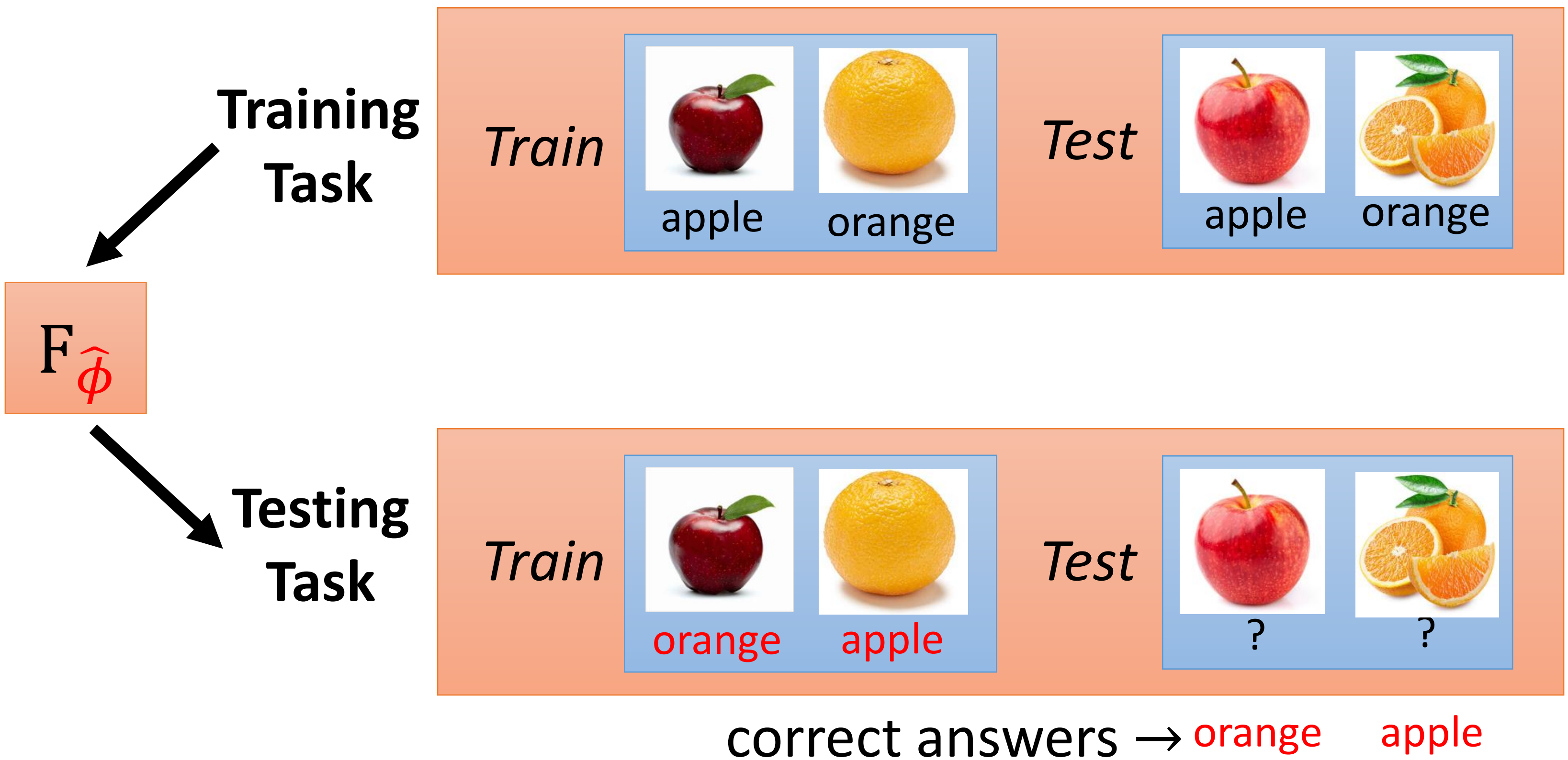
- Rapid Learning or Feature Reuse?



Aniruddh Raghu, Maithra Raghu, Samy Bengio, Oriol Vinyals, Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML, ICLR, 2020

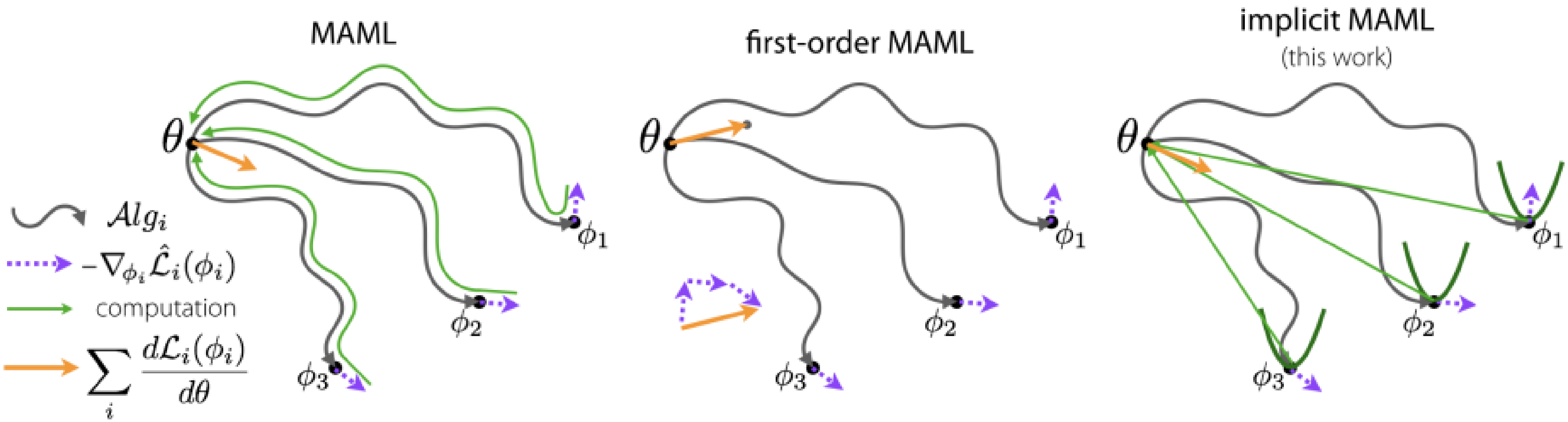
Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, Chelsea Finn, "Meta-Learning without Memorization", ICLR, 2020

Ignoring Training Examples?



For the models simply learn what apple & orange look like (not learn to learn), they cannot obtain correct answers.

If you want more update steps, but don't like intensive computation.



- **iMAML**: Aravind Rajeswaran, Chelsea Finn, Sham Kakade, Sergey Levine, Meta-Learning with Implicit Gradients, NeurIPS, 2019
- **Reptile:** Alex Nichol, Joshua Achiam, John Schulman, On First-Order Meta-Learning Algorithms, arXiv, 2018
- Pan Zhou, Xiao-Tong Yuan, Huan Xu, Yan Shuicheng, Jiashi FengEfficient Meta Learning via Minibatch Proximal Update, NeurIPS, 2019
- Luca Bertinetto, João F. Henriques, Philip H.S. Torr, Andrea Vedaldi, Meta Learning with Differentiable Closed-Form Solver, ICLR, 2019

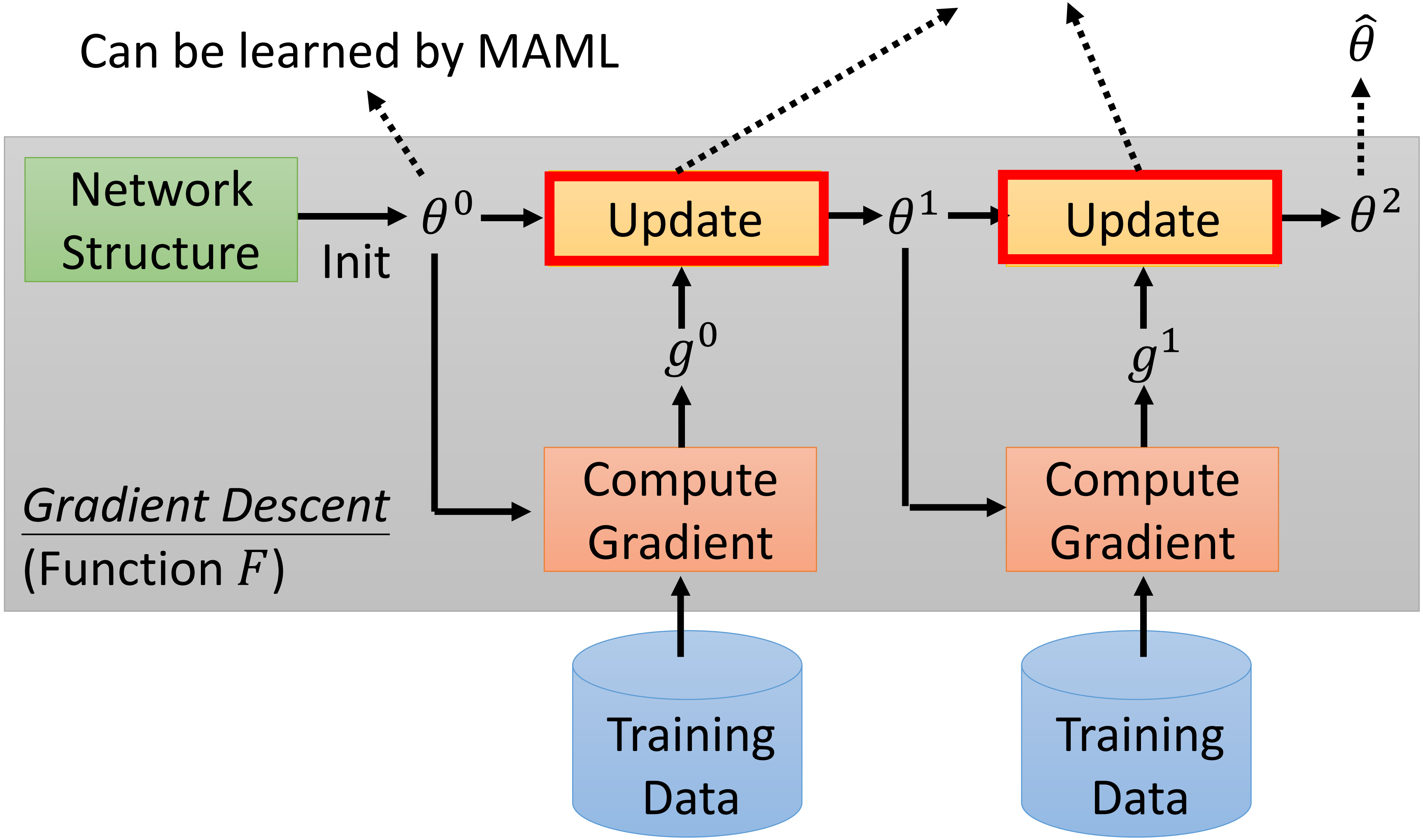
More Approaches

Optimizer

Basis form: $\theta^{t+1} \leftarrow \theta^{t} - \lambda g^{t}$

Adagrad, RMSprop, NAG, Adam ……

Is the optimizer learnable?

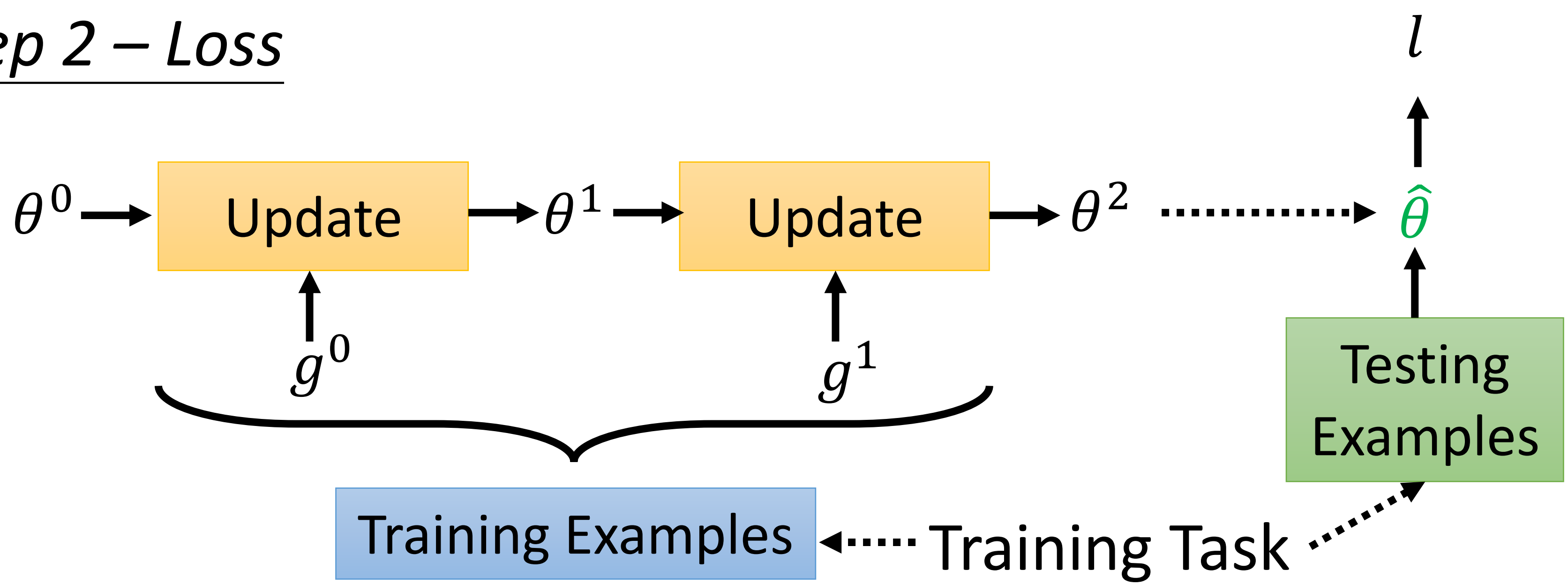


Learning Optimizer

Step 1 – What is learnable?

$$\text{Update} = \theta^{t} \leftarrow \underbrace{\phi'}_{\text{Weight Decay}} \odot \theta^{t-1} + \underbrace{\phi''}_{\text{Dynamic learning rate}} \odot \overbrace{-g^{t-1}}^{\text{gradient}}$$

Step 2 – Loss



Learning Optimizer

Step 3 – Optimization

forget gate

input gate

gradient

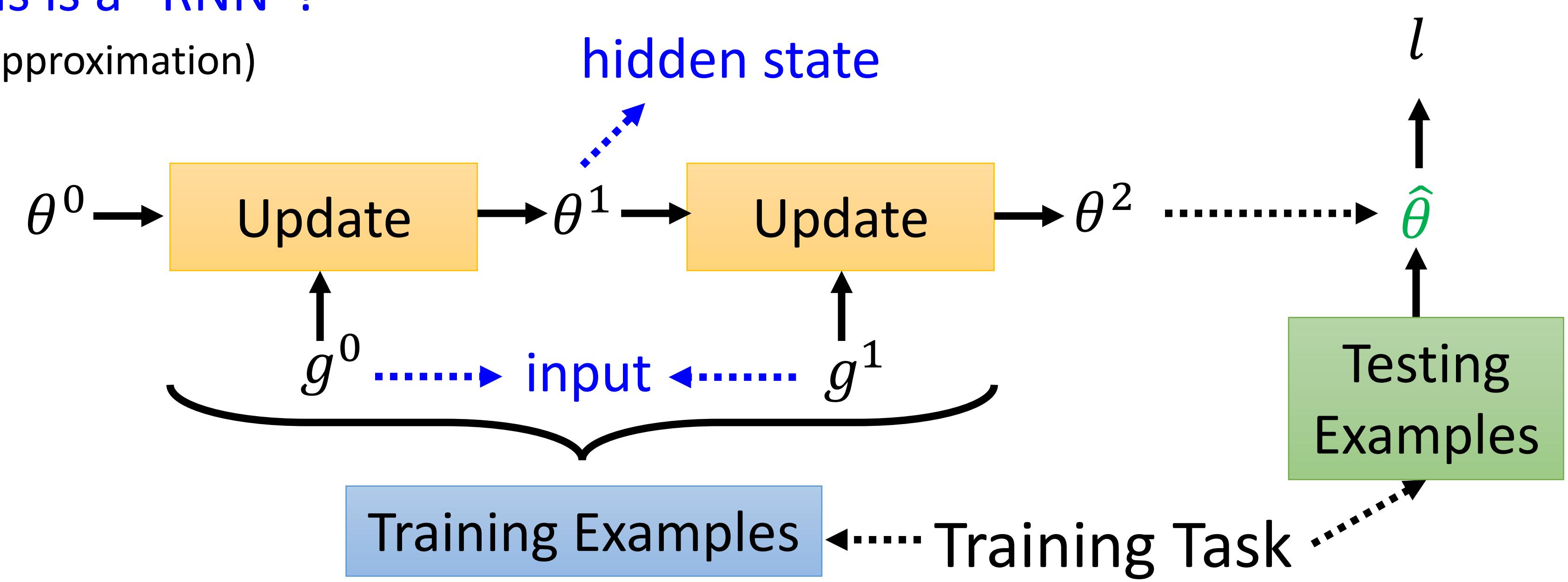
$$\text{Update} = \theta^{t} \leftarrow \phi' \odot \theta^{t-1} + \phi'' \odot -g^{t-1}$$

Weight Decay

Dynamic learning rate

This is a "RNN"!

(approximation)



Sachin Ravi, et al., Optimization as a Model for Few-Shot Learning, ICLR, 2017

Optimizer

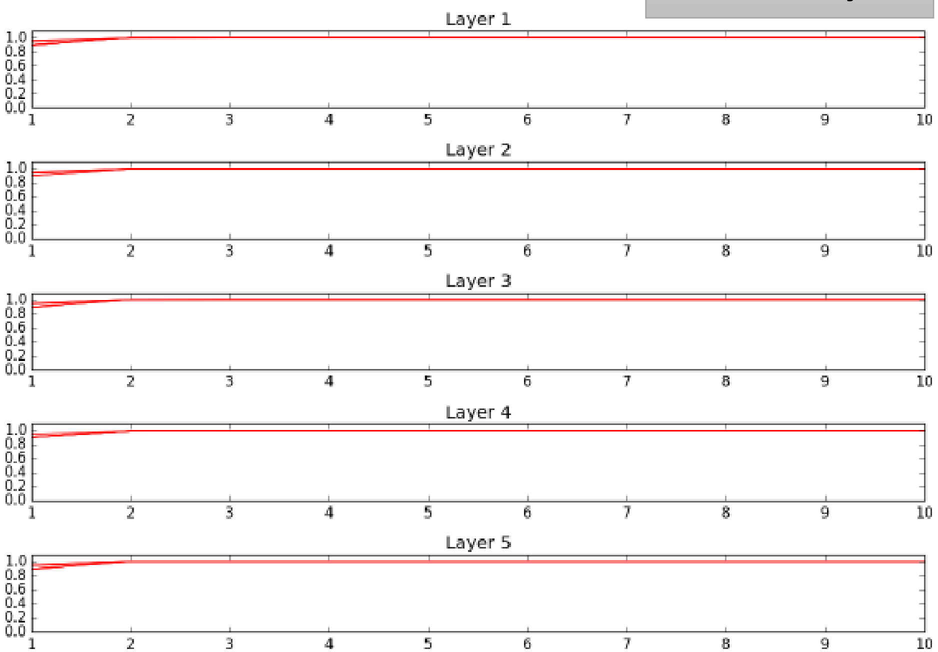
forget gate — input gate

gradient

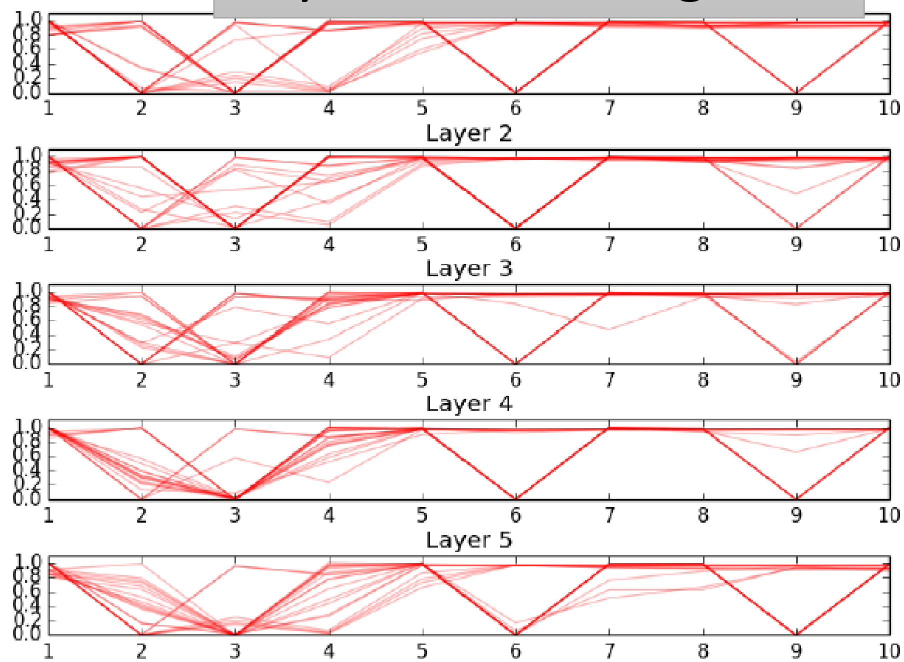
$$\boxed{\text{Update}} = \boxed{\theta^{t}} \leftarrow \phi' \odot \boxed{\theta^{t-1}} + \phi'' \odot \boxed{-g^{t-1}}$$

Weight Decay

Dynamic learning rate



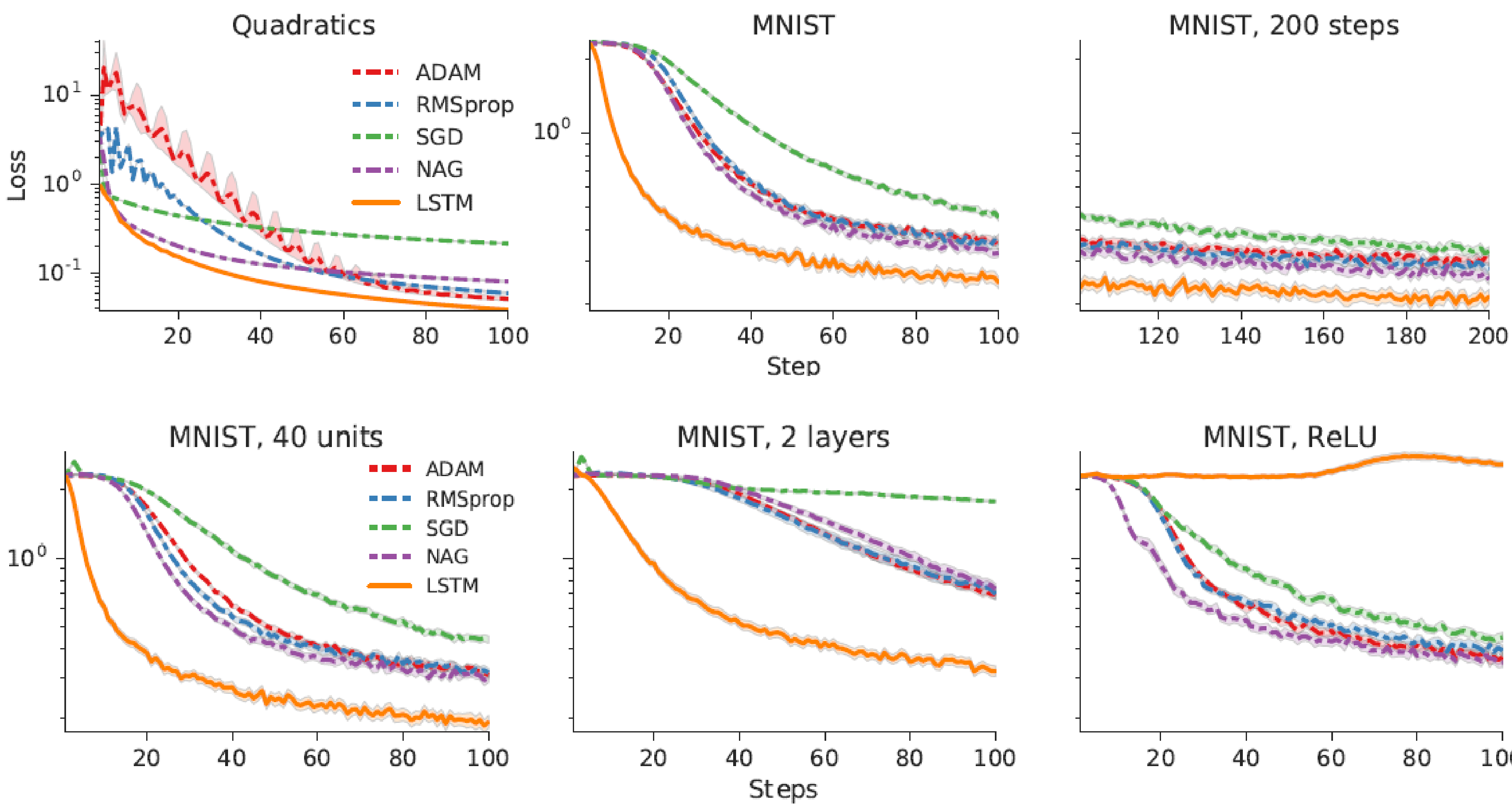
(a) Forget gate values for 1-shot meta-learner



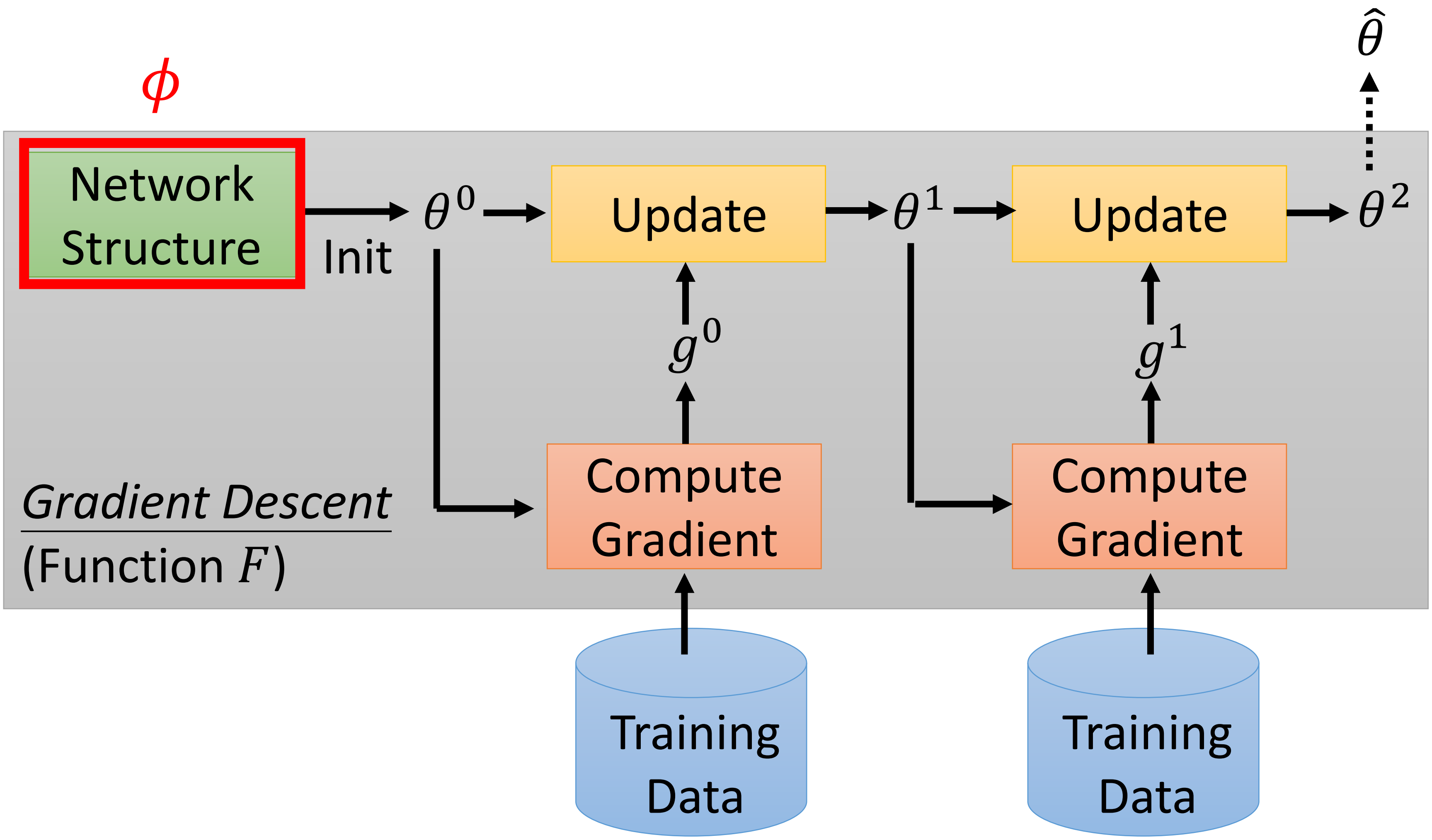
(b) Input gate values for 1-shot meta-learner

Optimizer

Marcin Andrychowicz, et al., Learning to learn by gradient descent by gradient descent, NIPS, 2016



Network Architecture Search (NAS)



Network Architecture Search (NAS)

$$\hat{\phi} = arg \min_{\phi} L(\phi) \qquad \nabla_{\phi} L(\phi) = ?$$

ϕ → Network Architecture

- Reinforcement Learning
  - Barret Zoph, et al., Neural Architecture Search with Reinforcement Learning, ICLR 2017
  - Barret Zoph, et al., Learning Transferable Architectures for Scalable Image Recognition, CVPR, 2018
  - Hieu Pham, et al., Efficient Neural Architecture Search via Parameter Sharing, ICML, 2018

An agent uses a set of actions to determine the network architecture.

ϕ: the agent's parameters

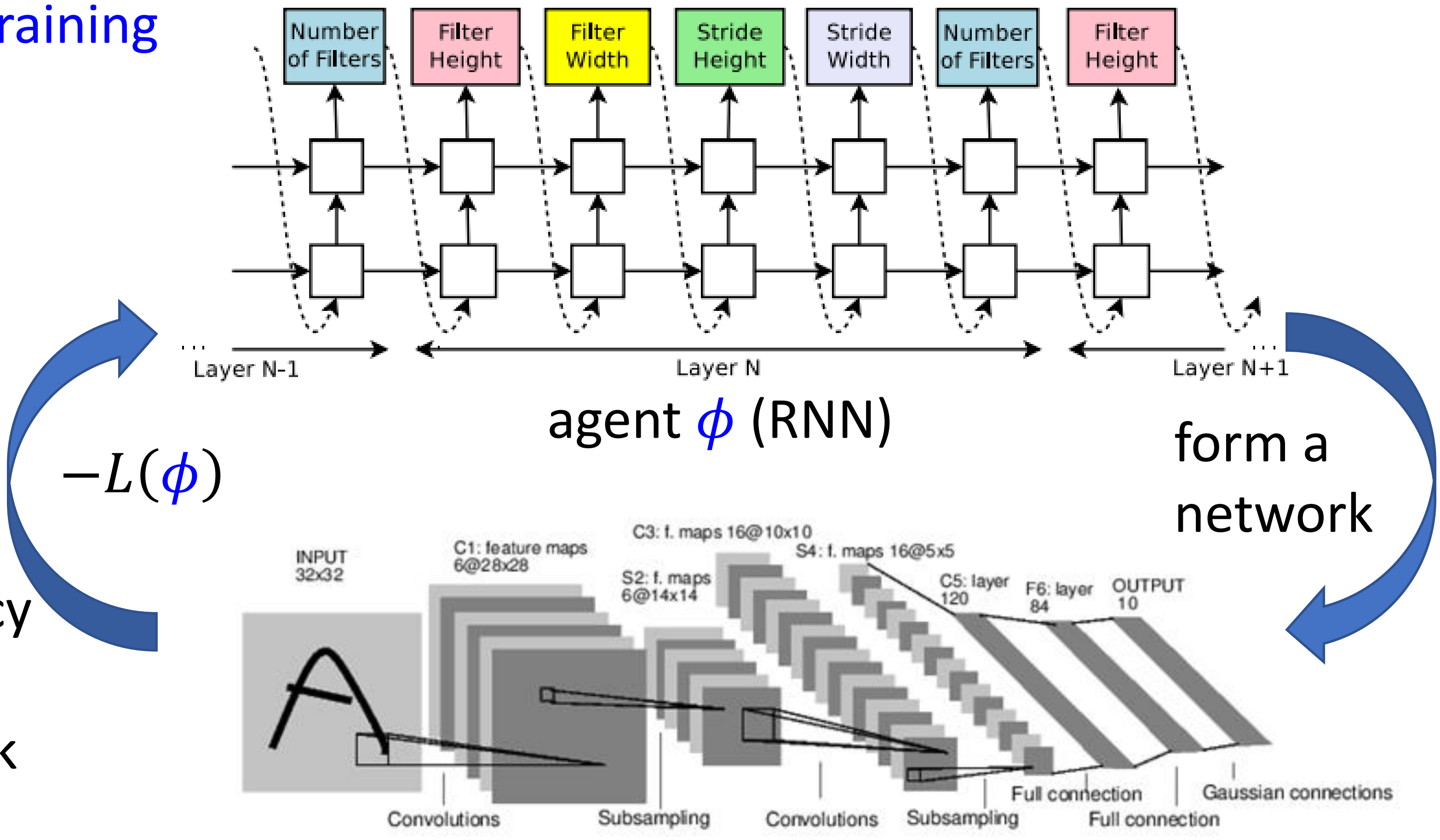
$-L(\phi)$

Reward to be maximized

Network Architecture Search (NAS)

Across-task Training

Update ϕ to maximize reward $-L(\phi)$



A Full Convolutional Neural Network (LeNet)

Accuracy of the network

Train the network

Within-task Training

Network Architecture Search (NAS)

$$\hat{\phi} = arg\min_{\phi} L(\phi) \qquad \nabla_{\phi} L(\phi) = ?$$

Network Architecture

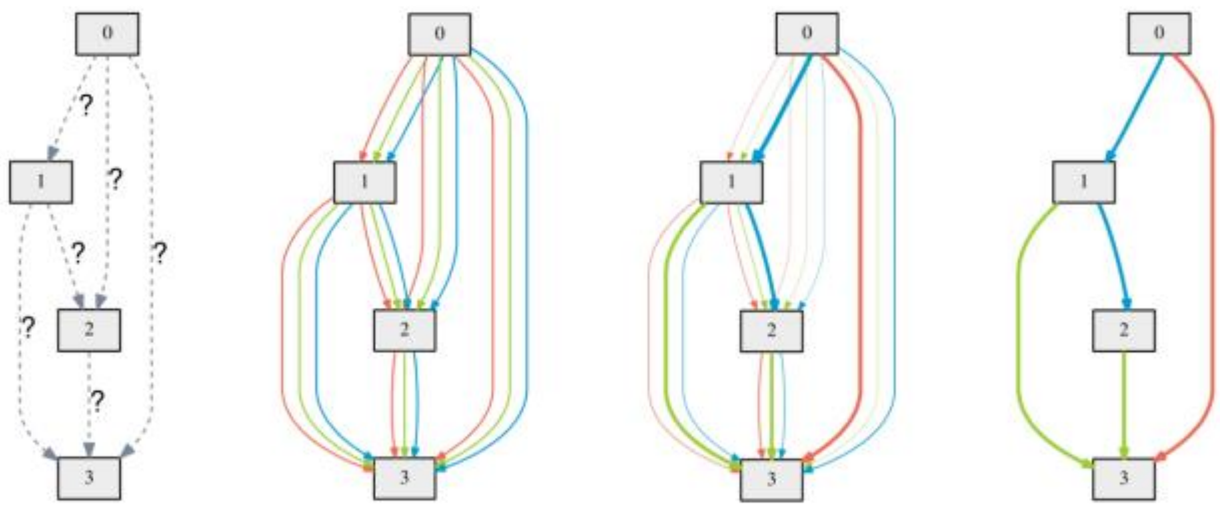
- Reinforcement Learning
  - Barret Zoph, et al., Neural Architecture Search with Reinforcement Learning, ICLR 2017
  - Barret Zoph, et al., Learning Transferable Architectures for Scalable Image Recognition, CVPR, 2018
  - Hieu Pham, et al., Efficient Neural Architecture Search via Parameter Sharing, ICML, 2018
- Evolution Algorithm
  - Esteban Real, et al., Large-Scale Evolution of Image Classifiers, ICML 2017
  - Esteban Real, et al., Regularized Evolution for Image Classifier Architecture Search, AAAI, 2019
  - Hanxiao Liu, et al., Hierarchical Representations for Efficient Architecture Search, ICLR, 2018

Network Architecture Search (NAS)

$$\hat{\phi} = \arg\min_{\phi} L(\phi) \qquad \nabla_{\phi} L(\phi) = ?$$

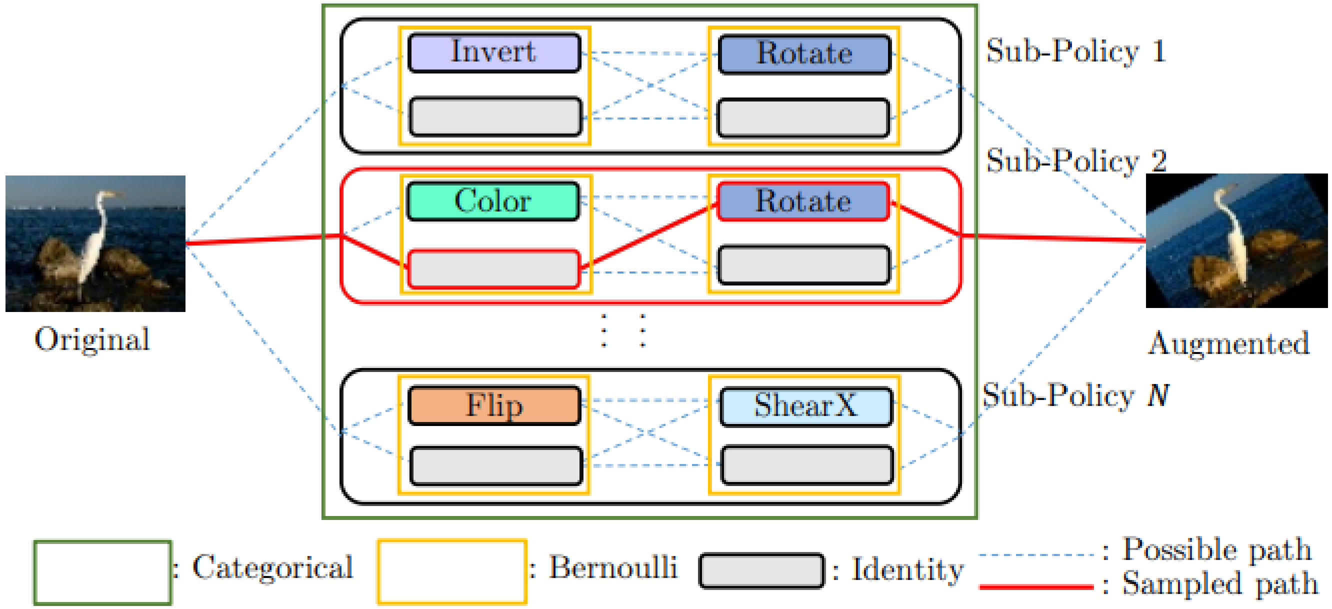
Network Architecture

- DARTS   Hanxiao Liu, et al., DARTS: Differentiable Architecture Search, ICLR, 2019



(a) (b) (c) (d)

Data Augmentation



Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M. Robertson, Yongxin Yang, DADA: Differentiable Automatic Data Augmentation, ECCV, 2020

Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, Xi Chen, Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules, ICML, 2019

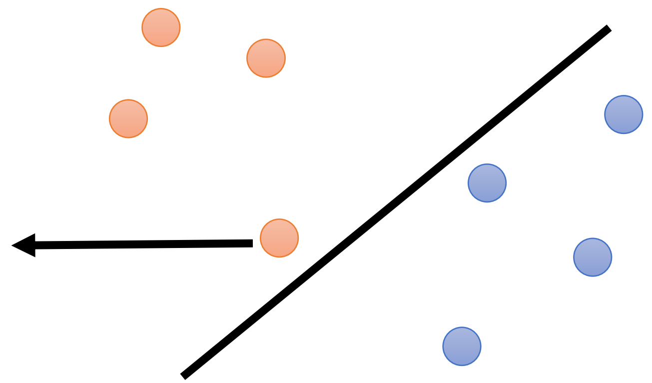
Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, Quoc V. Le, AutoAugment: Learning Augmentation Policies from Data, CVPR, 2019

Sample Reweighting

- Give different samples different weights

Larger weights (focus on tough examples)?

Smaller weights (the labels are noisy)?

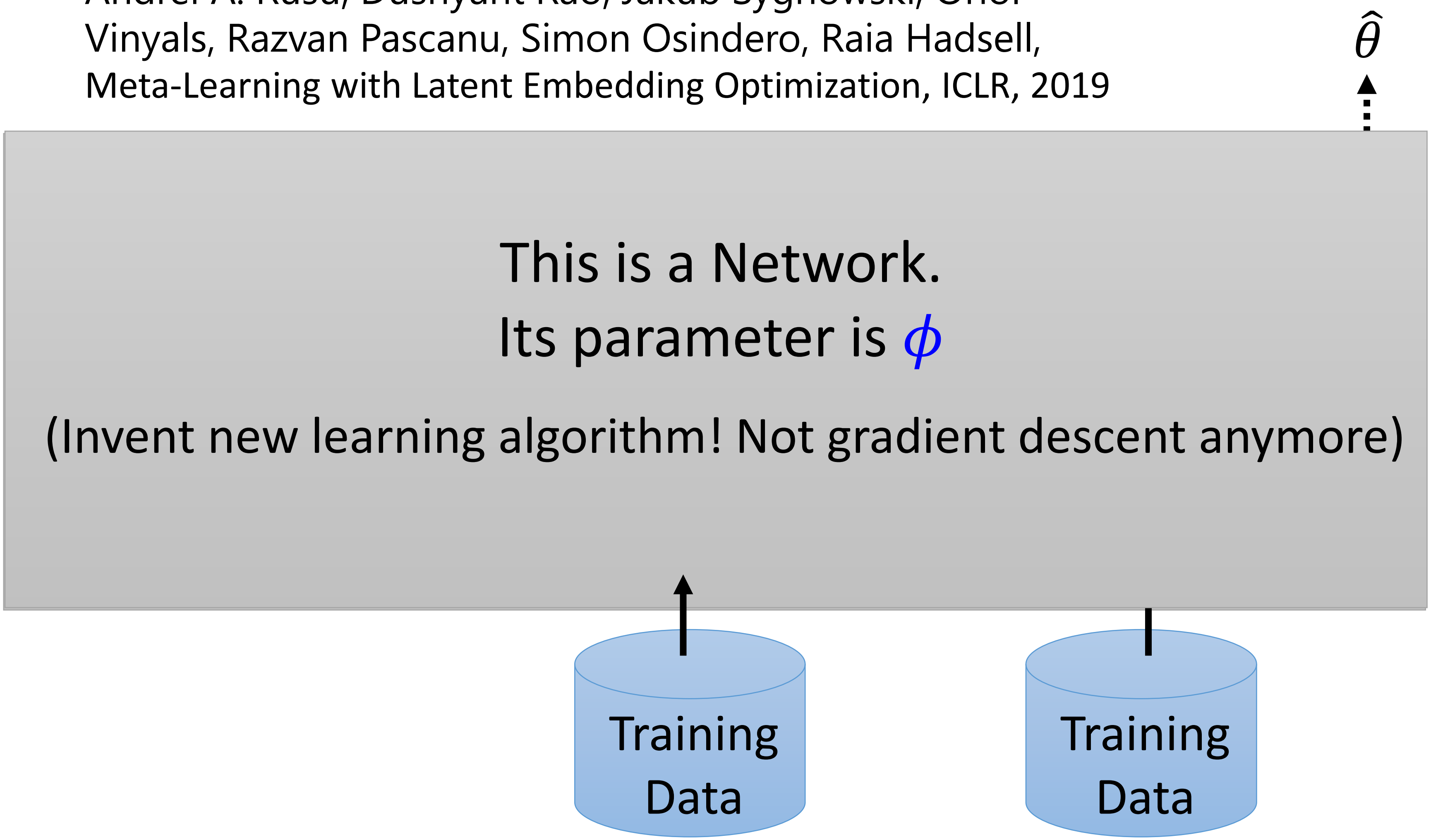


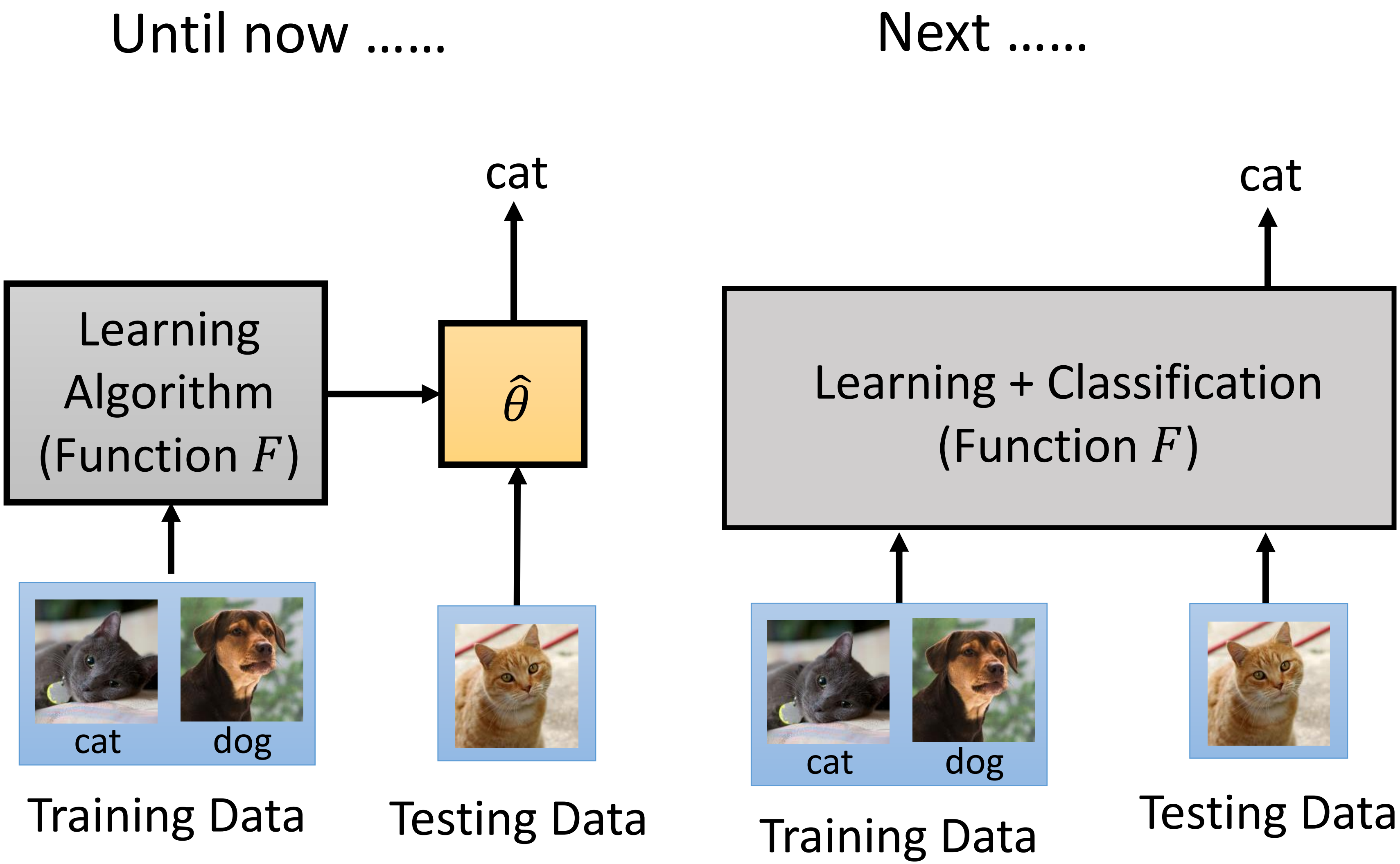
Sample Weighting Strategies → Learnable ϕ

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, Deyu Meng, Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting, NeurIPS, 2019
Mengye Ren, Wenyuan Zeng, Bin Yang, Raquel Urtasun, Learning to Reweight Examples for Robust Deep Learning, ICML, 2018

Learning as a Network?

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, Raia Hadsell, Meta-Learning with Latent Embedding Optimization, ICLR, 2019



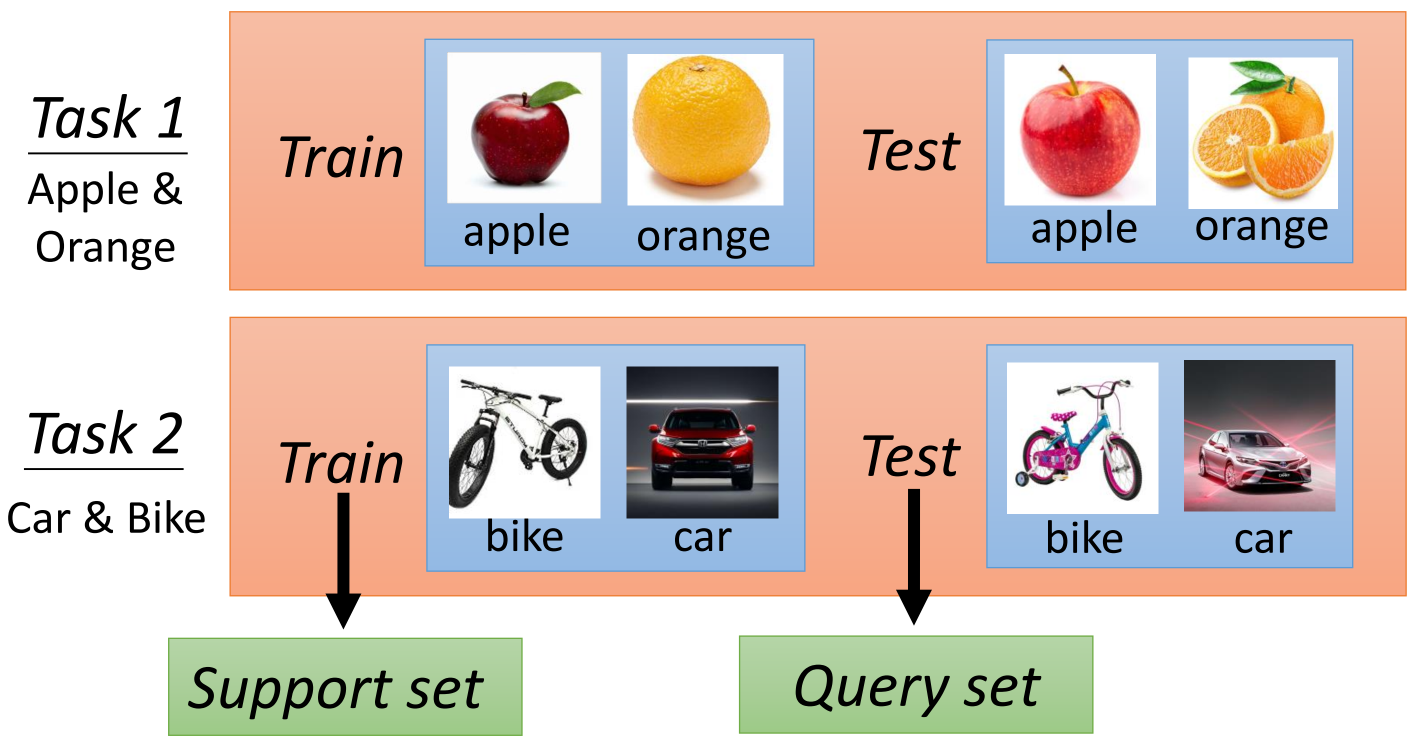

Until now ......
Next ......
cat
Learning
Algorithm
(Function F)
$\hat{\theta}$
cat
dog
Training Data
Testing Data
cat
Learning + Classification
(Function F)
cat
dog
Training Data
Testing Data

Learning to Compare

Training

<u>Meta Learning</u>

Training tasks

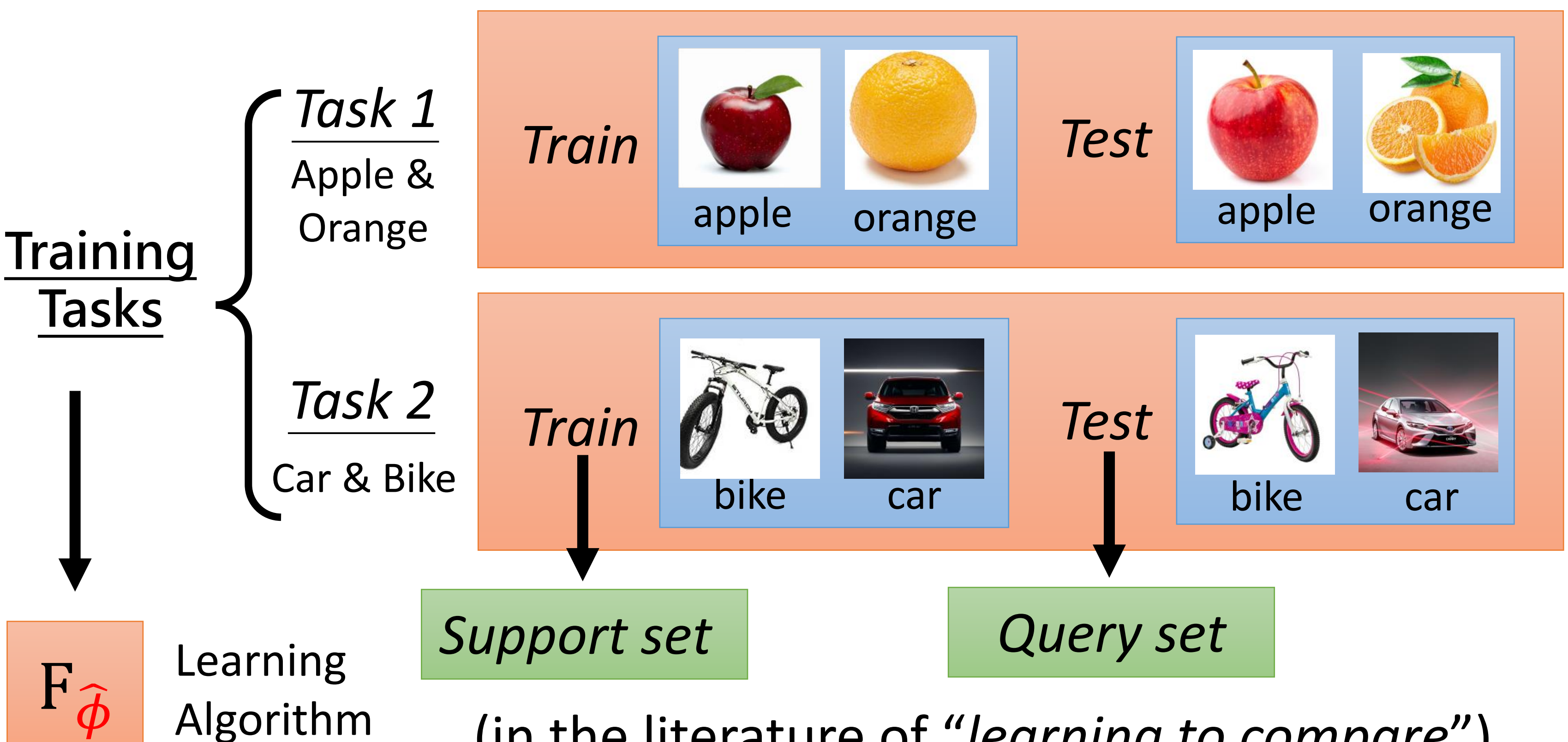


(in the literature of "*learning to compare*")

Training

Meta Learning

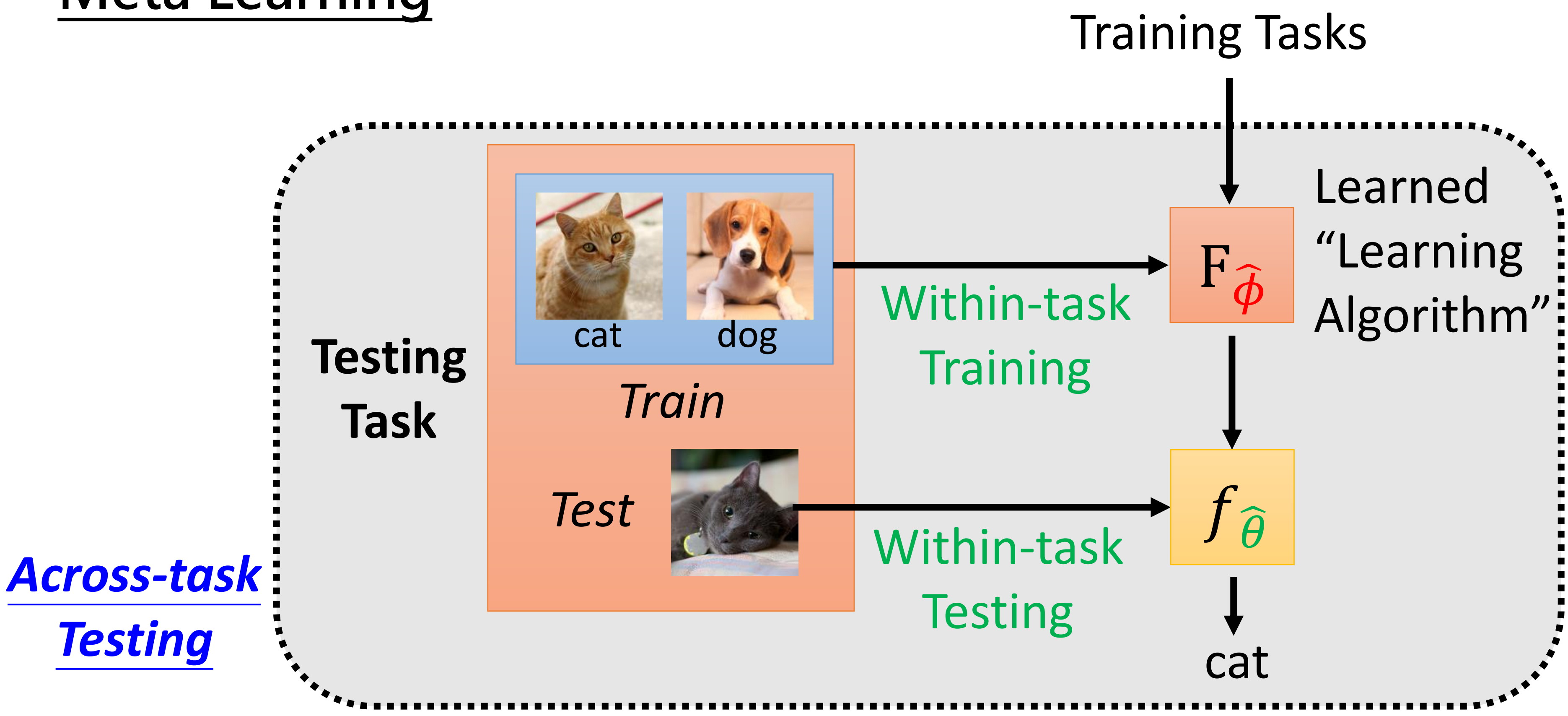
Training tasks



(in the literature of "*learning to compare*")

Testing

Meta Learning

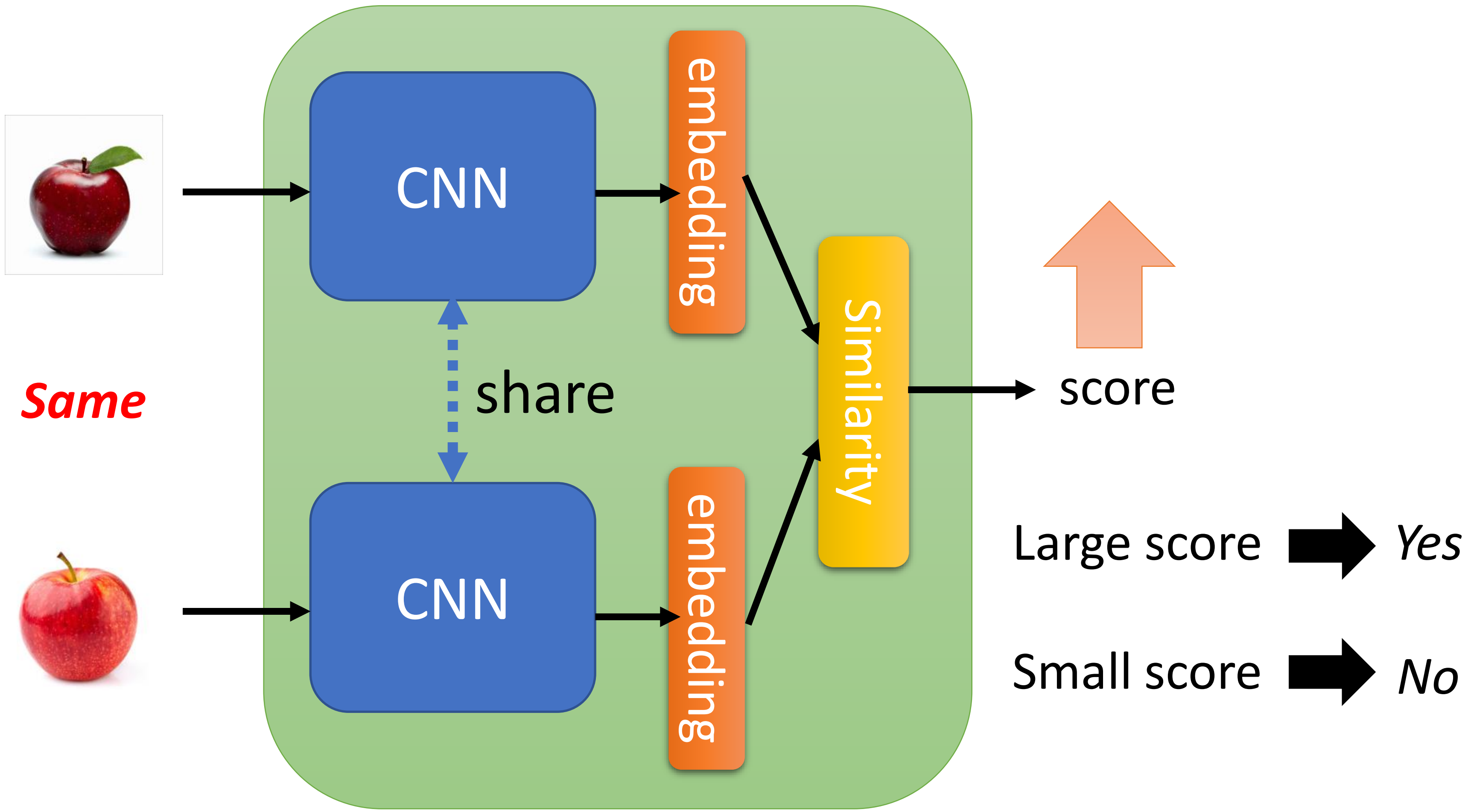


Learning to Compare

- What is the learned *learning algorithm* in this case?
- Think about ***non parametric models*** such as **k-nearest neighbors**
  - All training data are stored ➡ no learning needed
  - Performance depends **on the distance/similarity metrics**
- 'Learning to compare' algorithms
  - learn such models
  - do not have the within-task training
  - make the metrics *trainable* across tasks

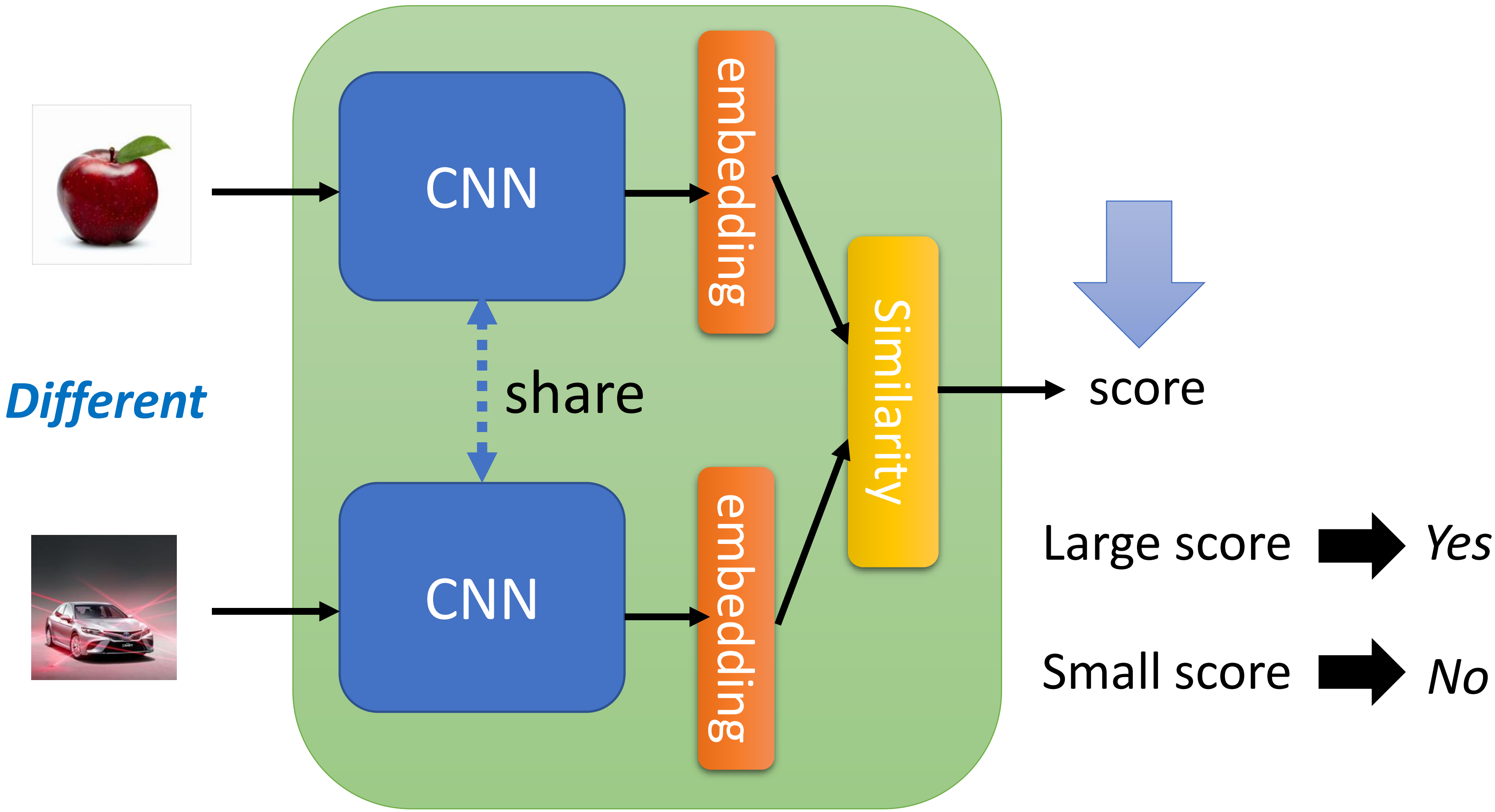
First Example: Siamese Network

Koch, Zemel, Salakhutdinov, 2015



First Example: Siamese Network

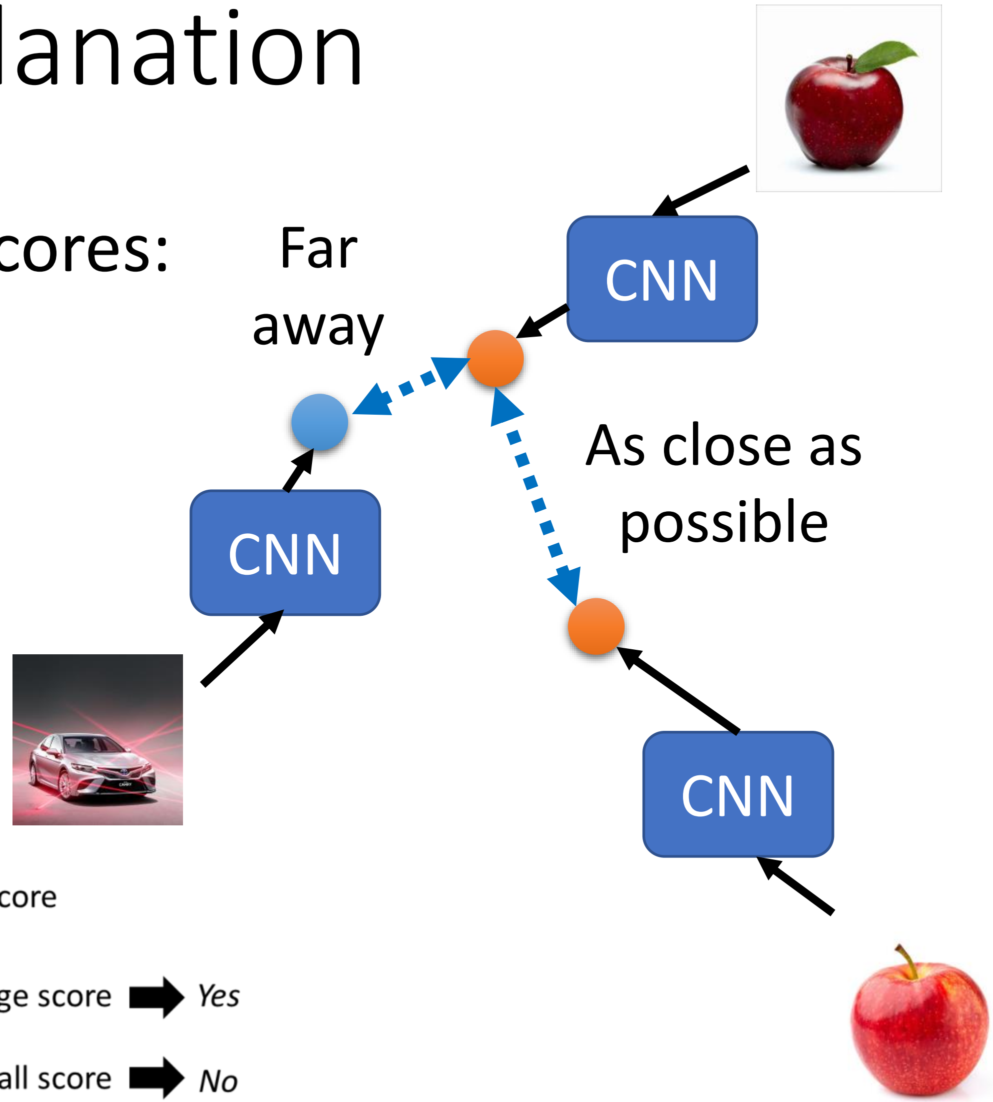
Koch, Zemel, Salakhutdinov, 2015

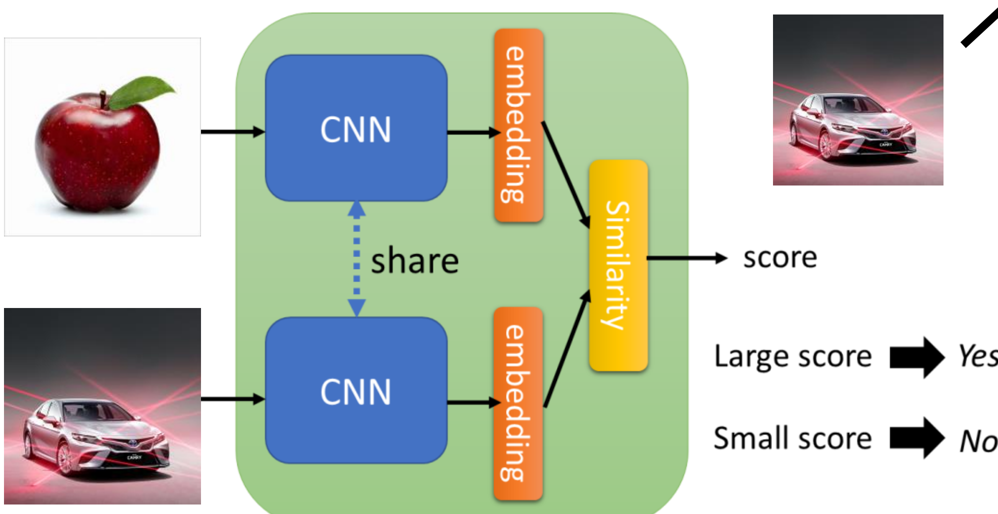


Siamese Network
- Intuitive Explanation

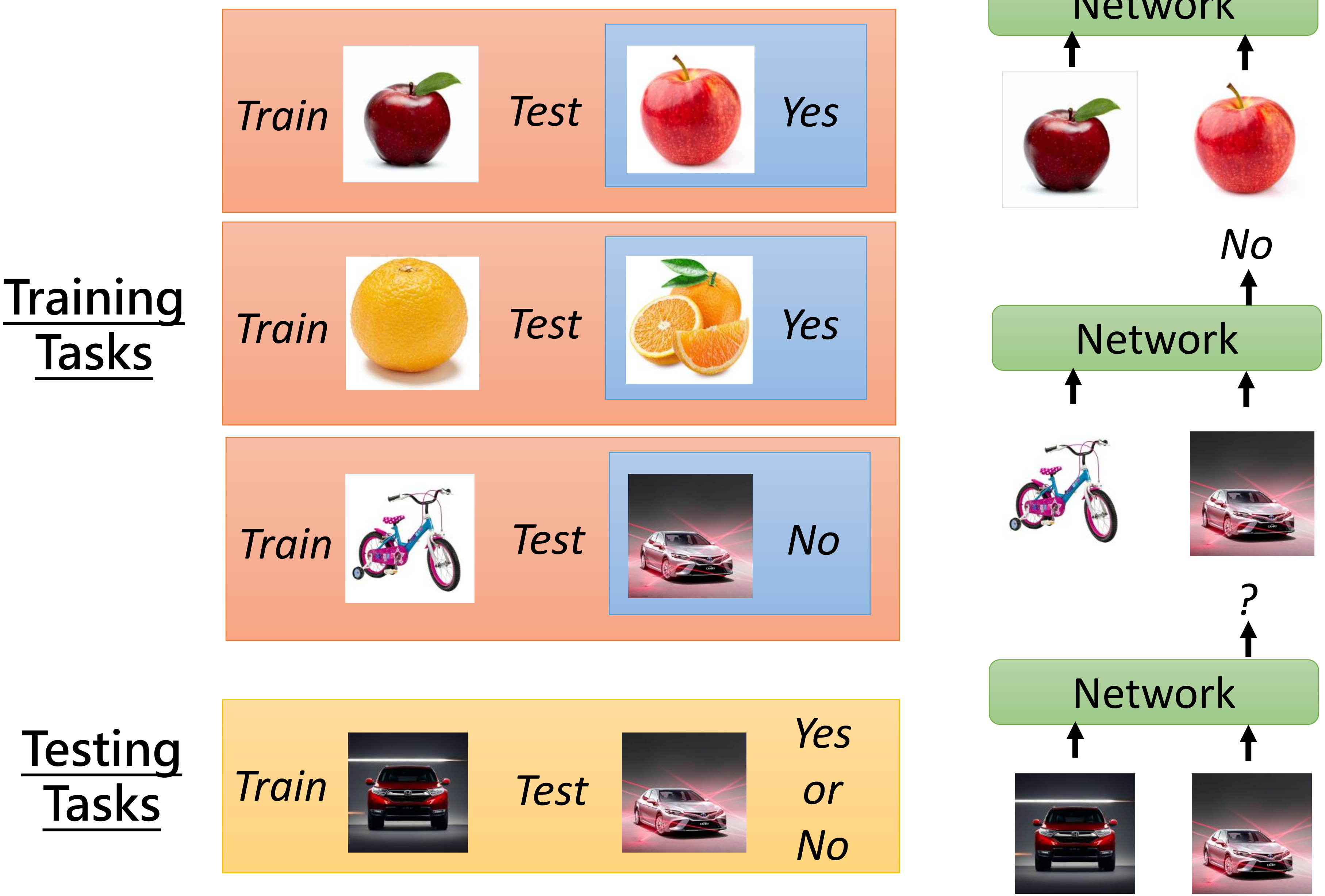
Learning the similarity scores:

- Convolutional NN
- Similarity functions



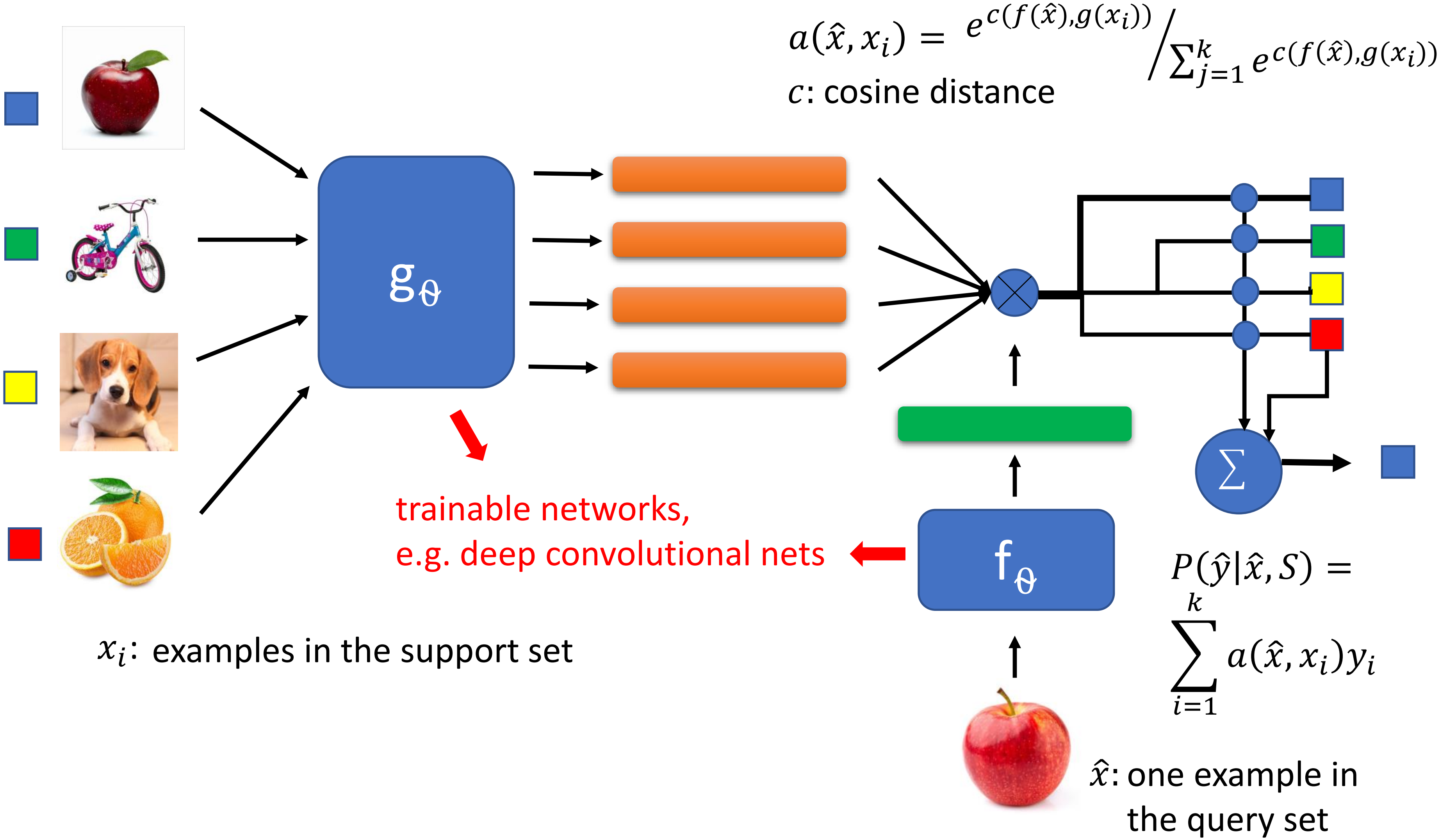


Frame It as a Meta Learning Setting



Matching Network

Vinyals, Blundell, Lillicrap, Kavukcupglu, Wierstra, 2017



Prototypical Network

Snell, Swersky, Zemel, 2017

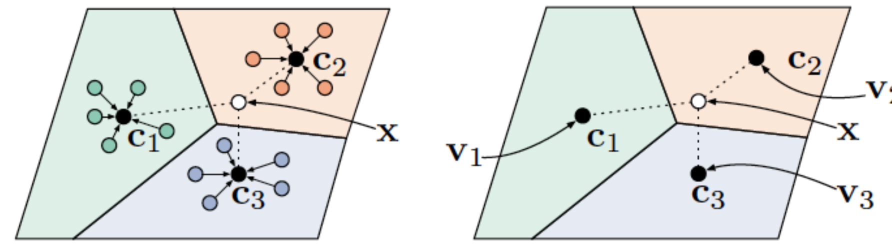
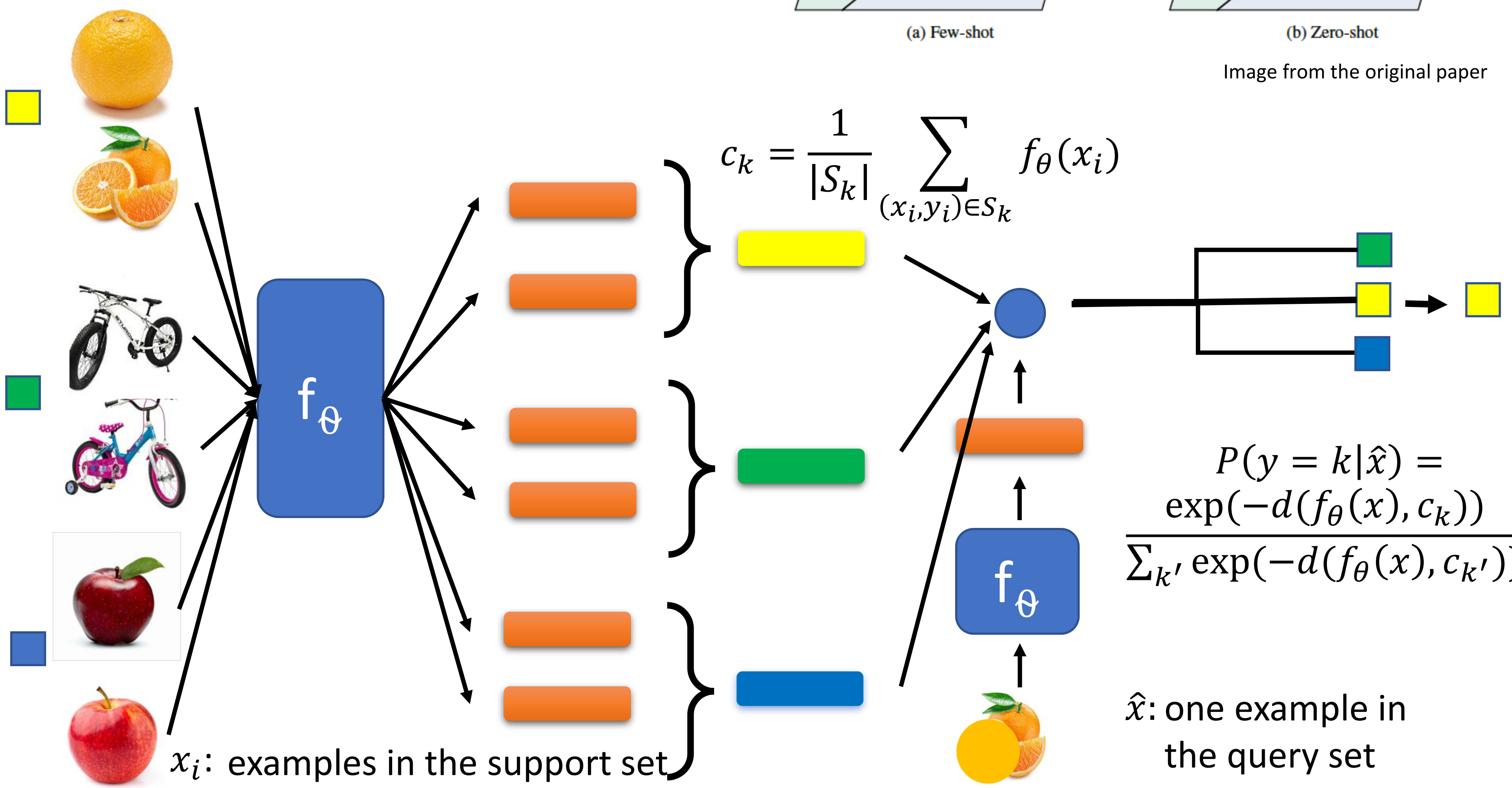


Image from the original paper



$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\theta(x_i)$$

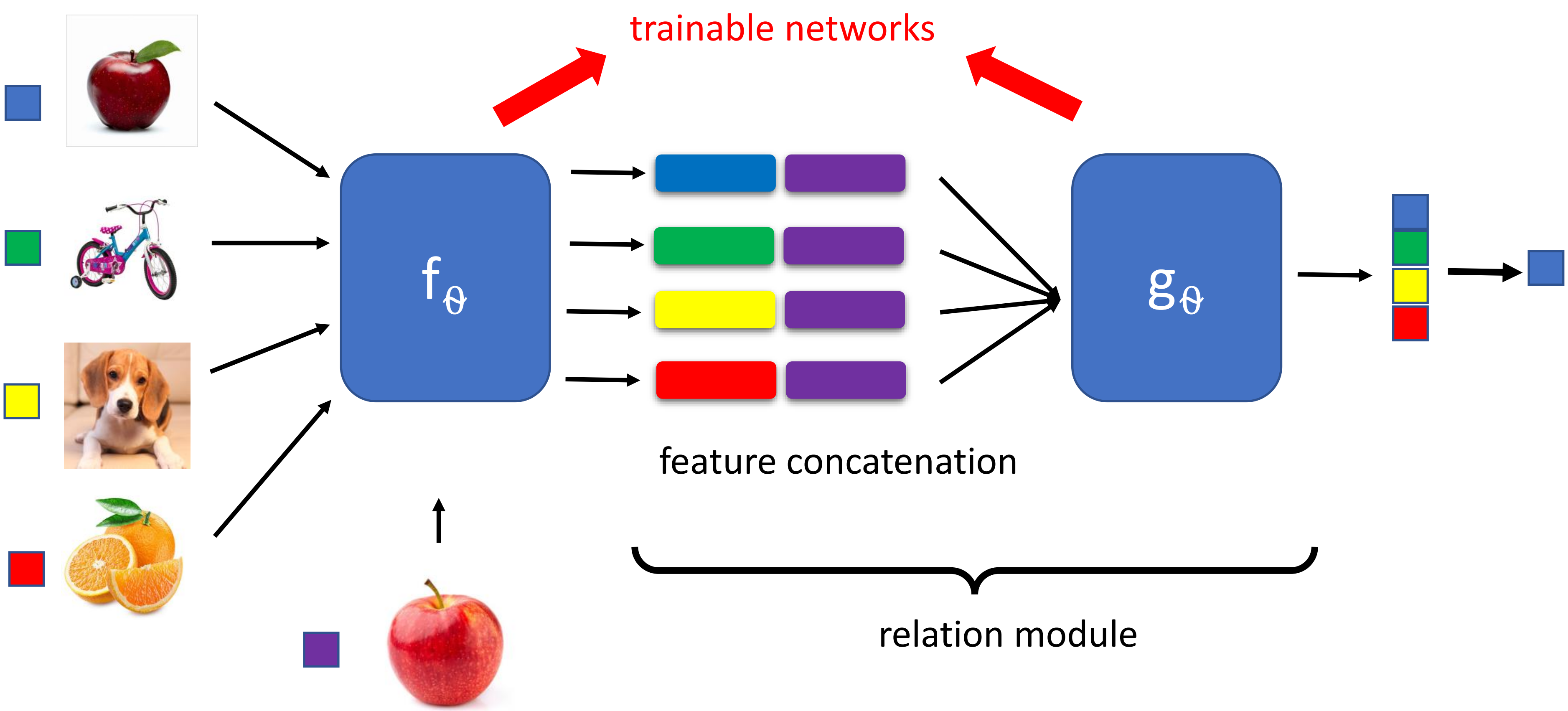
$$P(y = k|\hat{x}) = \frac{\exp(-d(f_\theta(x), c_k))}{\sum_{k'} \exp(-d(f_\theta(x), c_{k'}))}$$

x_i: examples in the support set

$\hat{x}$: one example in the query set

Relation Network

Sung, Yang, Zhang, Xiang, Torr, Hospedales, 2018



Meta Learning vs. Multi-task Learning vs. Transfer Learning

Meta Learning vs. Multi-task Learning

- Both use training data from many different tasks but have different objectives
- Meta learning aims at improving the accuracies of future tasks while multi-task learning optimizes the accuracies on all existing tasks
- The more tasks, the better the meta model, while multi-task learning methods might have problems with a large number of tasks

Meta Learning vs. Transfer Learning

- The goals are similar: improving accuracies on future new tasks
- While meta learning focuses on improving the training algorithms for future tasks, transfer learning aims at re-using knowledge learnt from previous tasks
- Meta learning assumes the same distribution between training tasks and testing tasks while transfer learning does not assume it between previous tasks and future tasks

Part I: Basic Idea of Meta Learning

- Opening → All speakers
- Starting from Machine learning
- Introduction of Meta Learning
- Learning to Initialize
- More Meta Learning Approaches

Hung-yi

- Learning to Compare
- Meta learning v.s. Other Methods

Thang Vu

Part II: Applications

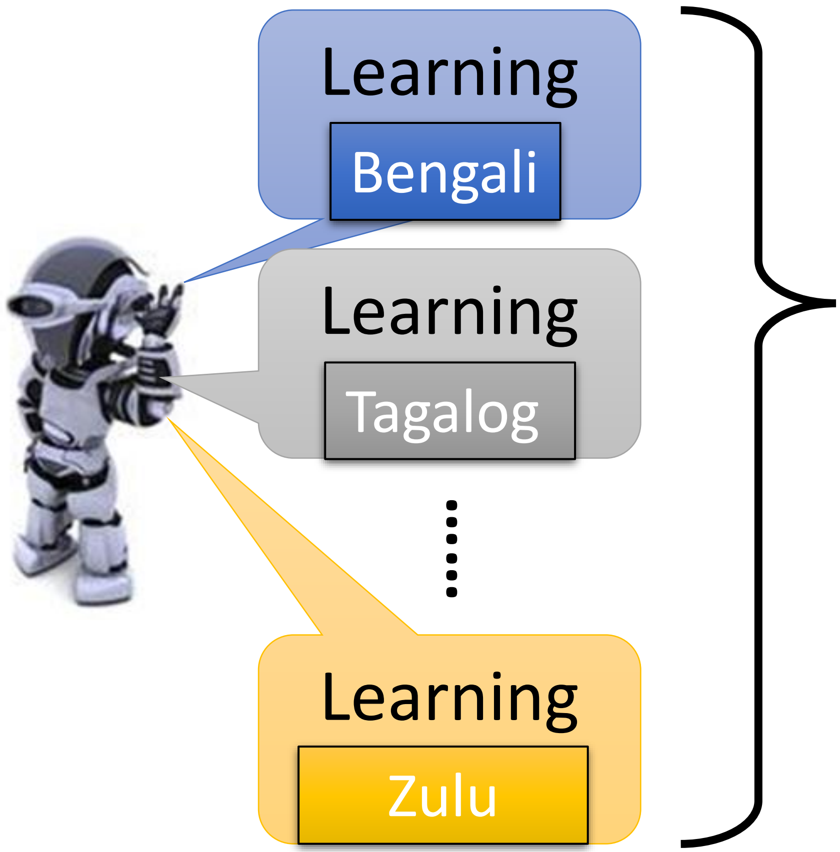
- Applications to Speech → Hung-yi Lee
- Applications to Conversational AI → Daniel
- Applications to Natural Language Processing
- Concluding Remarks

Thang Vu

Speech Recognition

Fast Adapt to Unseen Languages

Training Tasks



Testing Tasks

- There are more than 7000 languages.
- Fast adapt to the languages lack of labeled data

Fast Adapt to Unseen Languages

Learning to Initialize (MAML)

Jui-Yang Hsu, Yuan-Jui Chen, Hung-yi Lee, META LEARNING FOR END-TO-END LOW-RESOURCE SPEECH RECOGNITION, ICASSP, 2020

- Data from the IARPA BABEL project
  - Training tasks: Bengali, Tagalog, Zulu, Turkish, Lithuanian, Guarani
  - Testing tasks: Vietnamese, Swahili, Tamil, Kurmanji
- ASR model: CTC

CER (%)	Vietnamese	Swahili	Tamil	Kurmanji
Rand Init	71.8	47.5	69.9	64.3
Pre-train	59.7	48.8	65.6	62.6
MAML	50.1	42.9	58.9	57.6

Fast Adapt to Unseen Languages

Learning to Initialize (MAML)

Jui-Yang Hsu, Yuan-Jui Chen, Hung-yi Lee, META LEARNING FOR END-TO-END LOW-RESOURCE SPEECH RECOGNITION, ICASSP, 2020

- Data from the IARPA BABEL project
  - Training tasks: Bengali, Tagalog, Zulu, Turkish, Lithuanian, Guarani
  - Testing tasks: Vietnamese, Swahili, Tamil, Kurmanji
- ASR model: CTC

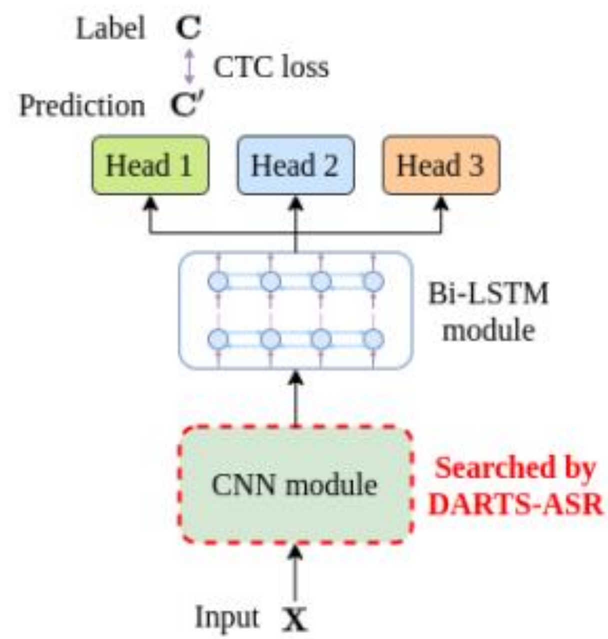
CER (%)	Vietnamese	Swahili	Tamil	Kurmanji
Rand Init	71.8	47.5	69.9	64.3
Pre-train	59.7	48.8	65.6	62.6
MAML	50.1	42.9	58.9	57.6

Fast Adapt to Unseen Languages

Network Architecture Search (DARTS)

Yi-Chen Chen, Jui-Yang Hsu, Cheng-Kuang Lee, Hung-yi Lee, "DARTS-ASR: Differentiable Architecture Search for Multilingual Speech Recognition and Adaptation", INTERSPEECH, 2020

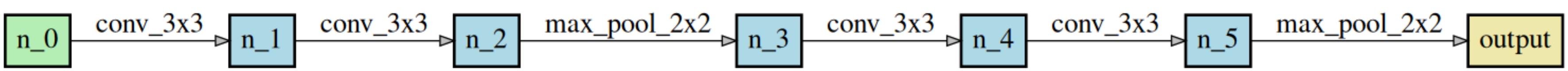
- Data from the IARPA BABEL project
    - ***Monolingual***: Training and testing tasks are the same language
        - oracle setting in meta learning
    - ***Multilingual***: Training and testing tasks are different languages
- ASR model: CTC



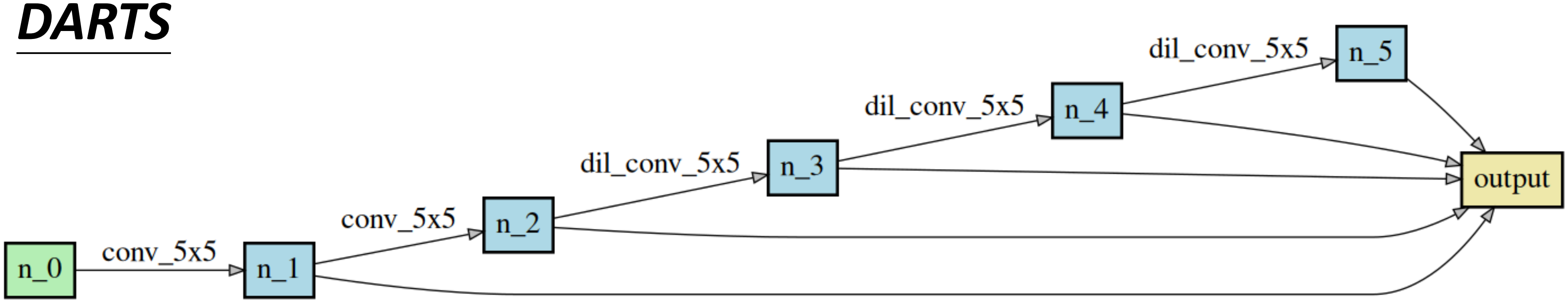
CER (%)	Vietnamese	Swahili	Tamil	Kurmanji
VGG-small	45.3	36.3	55.7	54.5
VGG-large	43.2	36.1	55.0	55.1
DARTS-ASR	40.9	32.3	45.9	53.5

(the network architecture can also be fine-tuned in the testing task)

VGG (not only useful in image, but speech related applications)

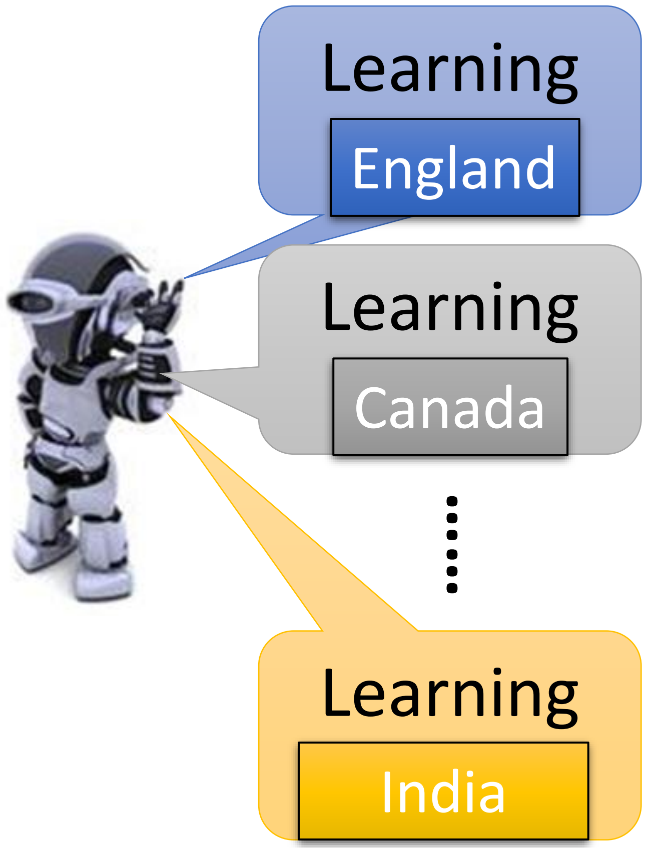


DARTS



Fast Adapt to Unseen Accents

Training Tasks



Testing Tasks

English accent lack of labeled data

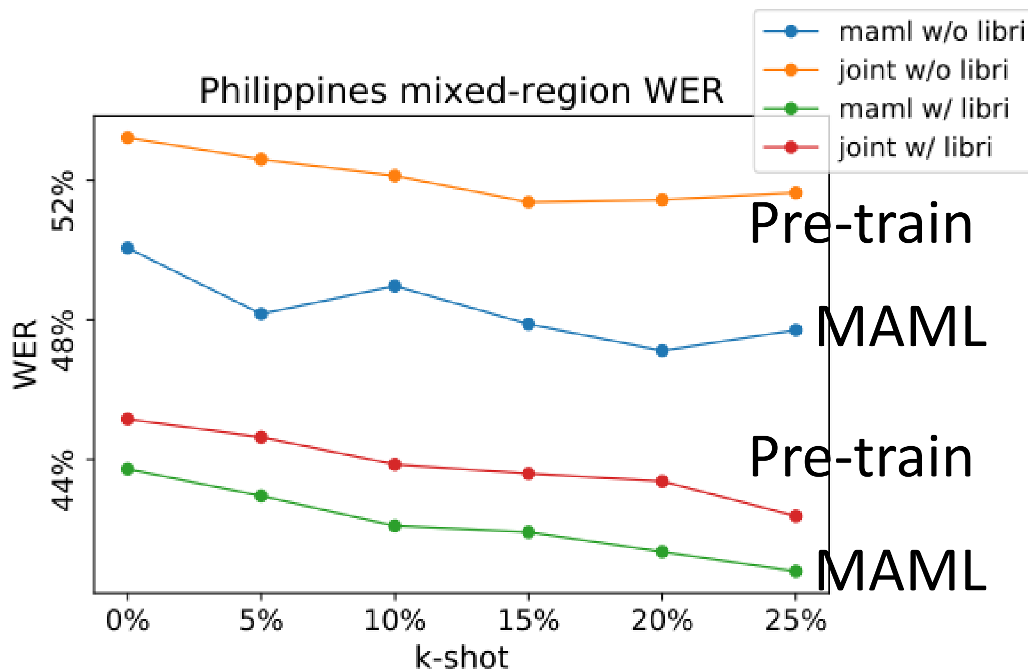
Different English accents

Fast Adapt to Unseen Accents

Learning to Initialize (MAML)

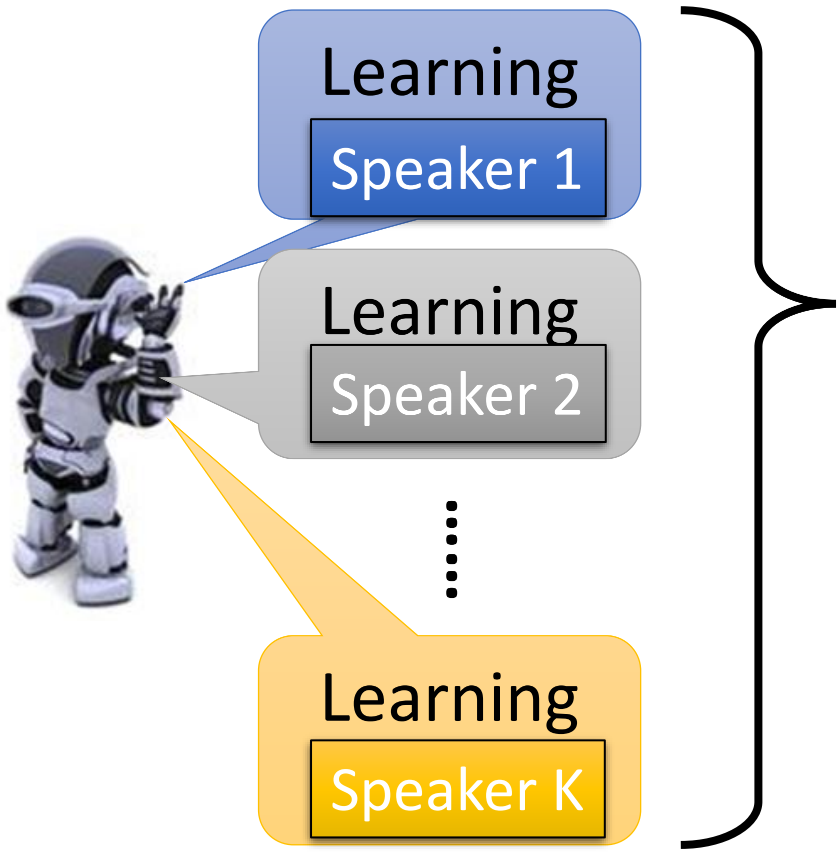
Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, Peng Xu, Pascale Fung, Learning Fast Adaptation on Cross-Accented Speech Recognition, INTERSPEECH, 2020

- Data from CommonVoice Dataset
- ASR model: Seq2seq

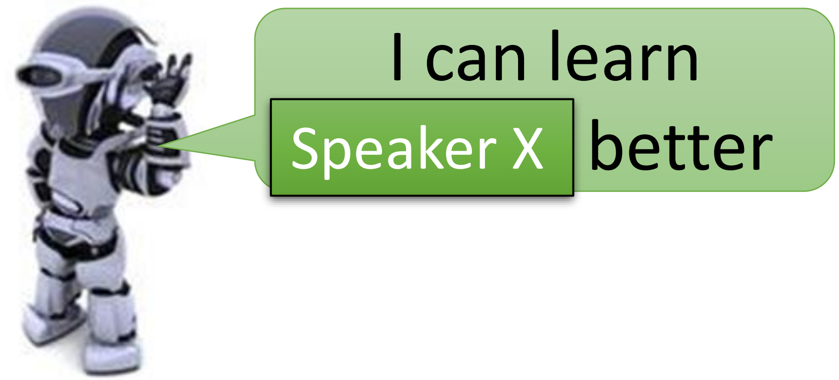


Fast Adapt to Unseen Speakers

Training Tasks



Testing Tasks



Speaker Adaptive Training?

Yes. New approaches for speaker adaptive training.

Fast Adapt to Unseen Speakers

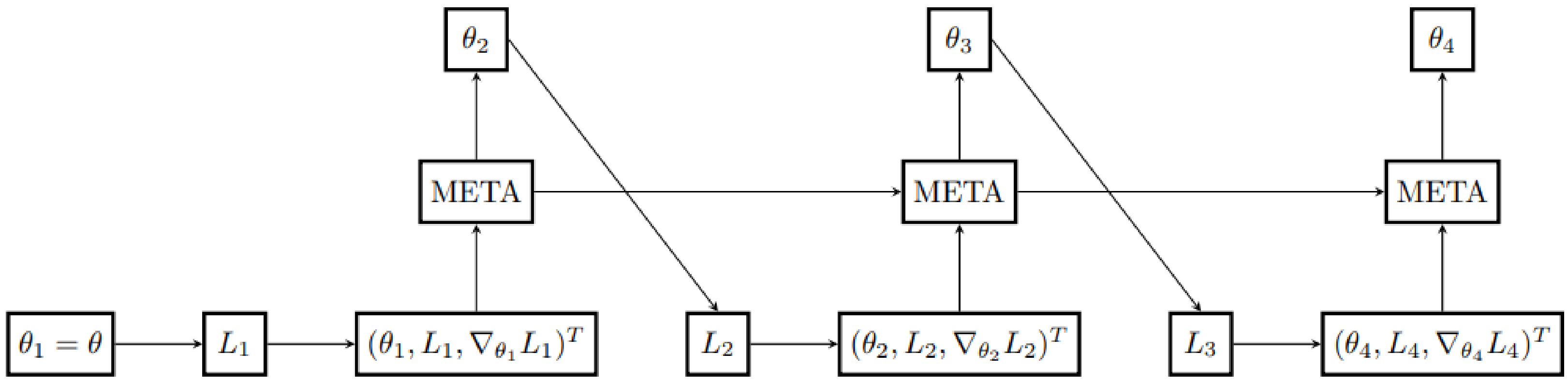
- Learning to Initialize (MAML)

  Ondřej Klejch, Joachim Fainberg, Peter Bell, Steve Renals, Speaker Adaptive Training using Model Agnostic Meta-Learning, ASRU, 2019

  Huaxin Wu, Genshun Wan, Jia Pan, Speaker Code Based Speaker Adaptive Training Using Model Agnostic Meta-learning, INTERSPEECH, 2020

- Learning Optimizer (optimizer as RNN)

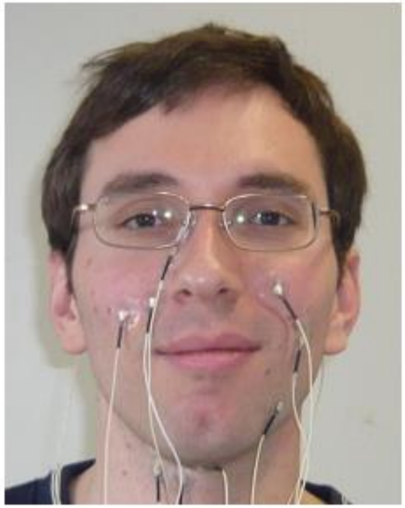
  Ondřej Klejch, Joachim Fainberg, Peter Bell, Learning to adapt: a meta-learning approach for speaker adaptation, INTERSPEECH, 2018

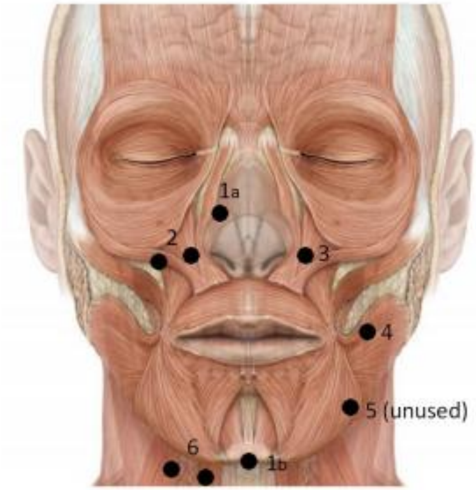


More ……

Krsto Proroković, et al., Adaptation of an EMG-Based Speech Recognizer via Meta-Learning, GlobalSIP, 2019

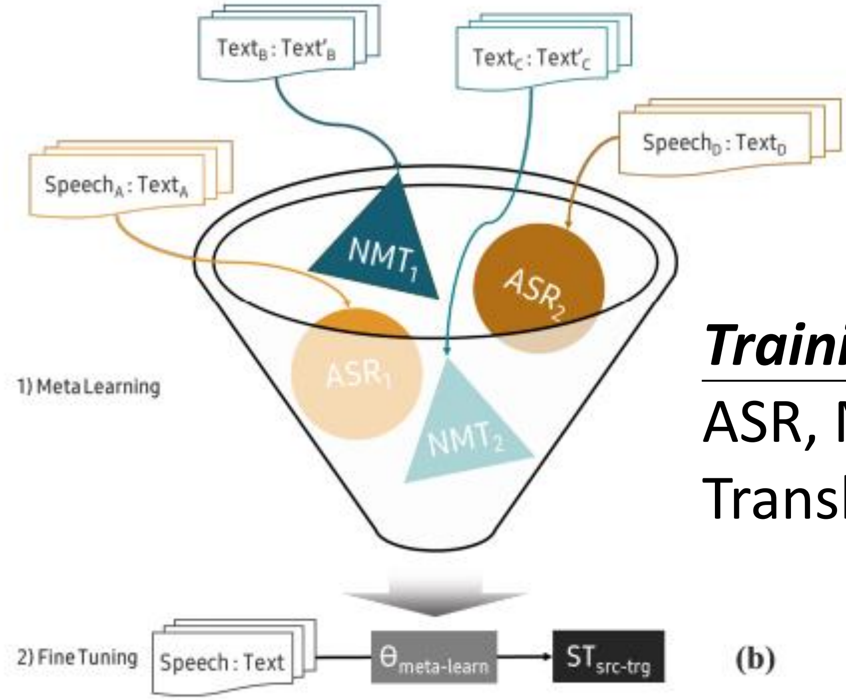
- Speech recognition from EMG signal of facial muscles





- Session = recording between putting and removing the electrodes
- Adaptation is needed for each session
- Using learning to initialize (MAML)

More ……



Speech Translation

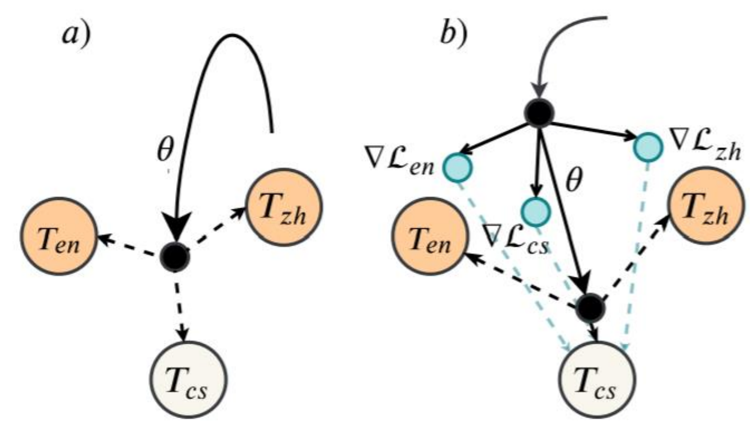
Sathish Indurthi, et al., Data Efficient Direct Speech-to-Text Translation with Modality Agnostic Meta-Learning, ICASSP 2020

Training Task: ASR, Machine Translation

Testing Task: Speech Translation

Code Switching

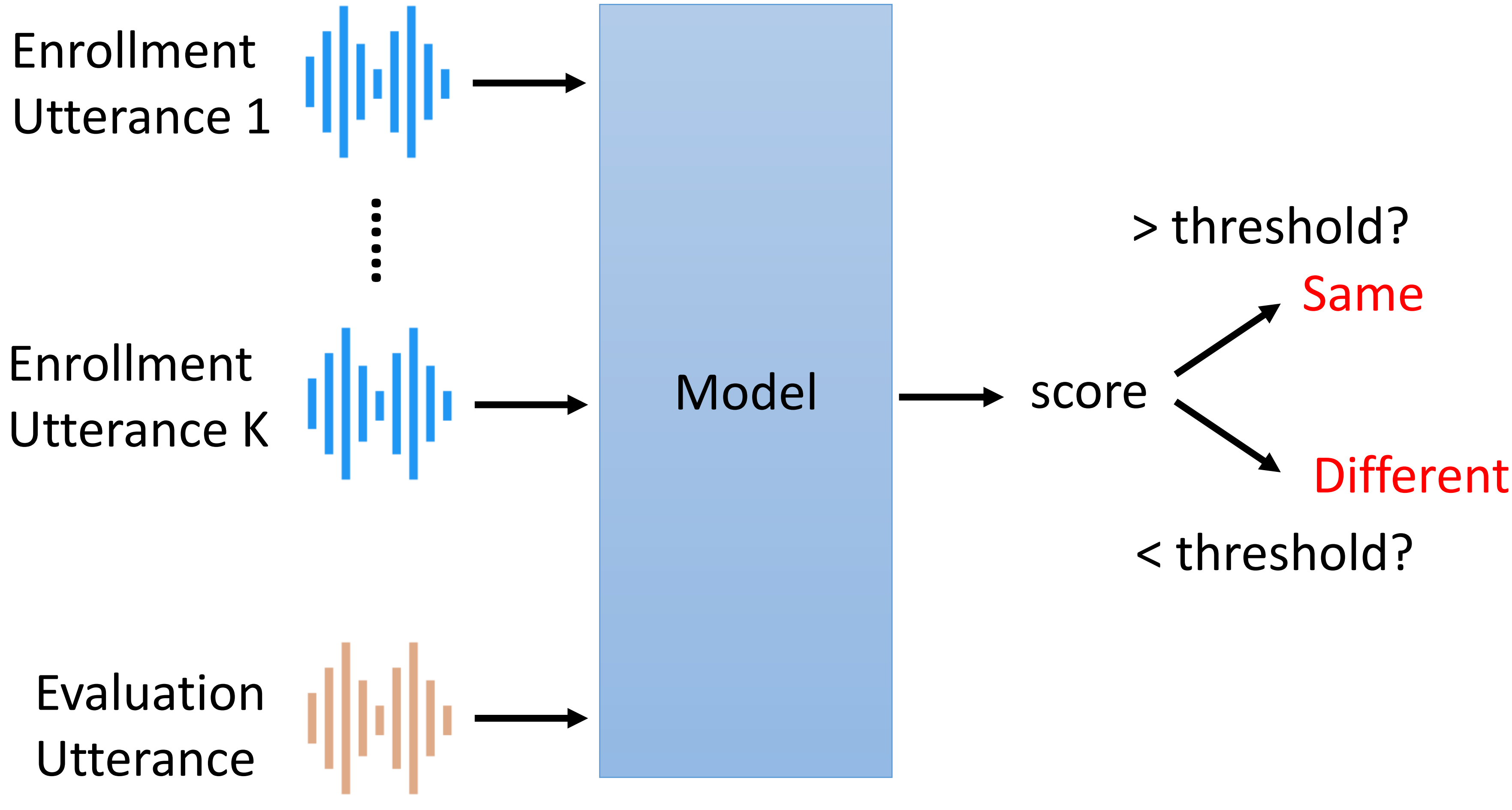
Genta Indra Winata, Samuel Cahyawijaya, Zhaojiang Lin, Zihan Liu, Peng Xu, Pascale Fung, Meta-Transfer Learning for Code-Switched Speech Recognition, ACL, 2020



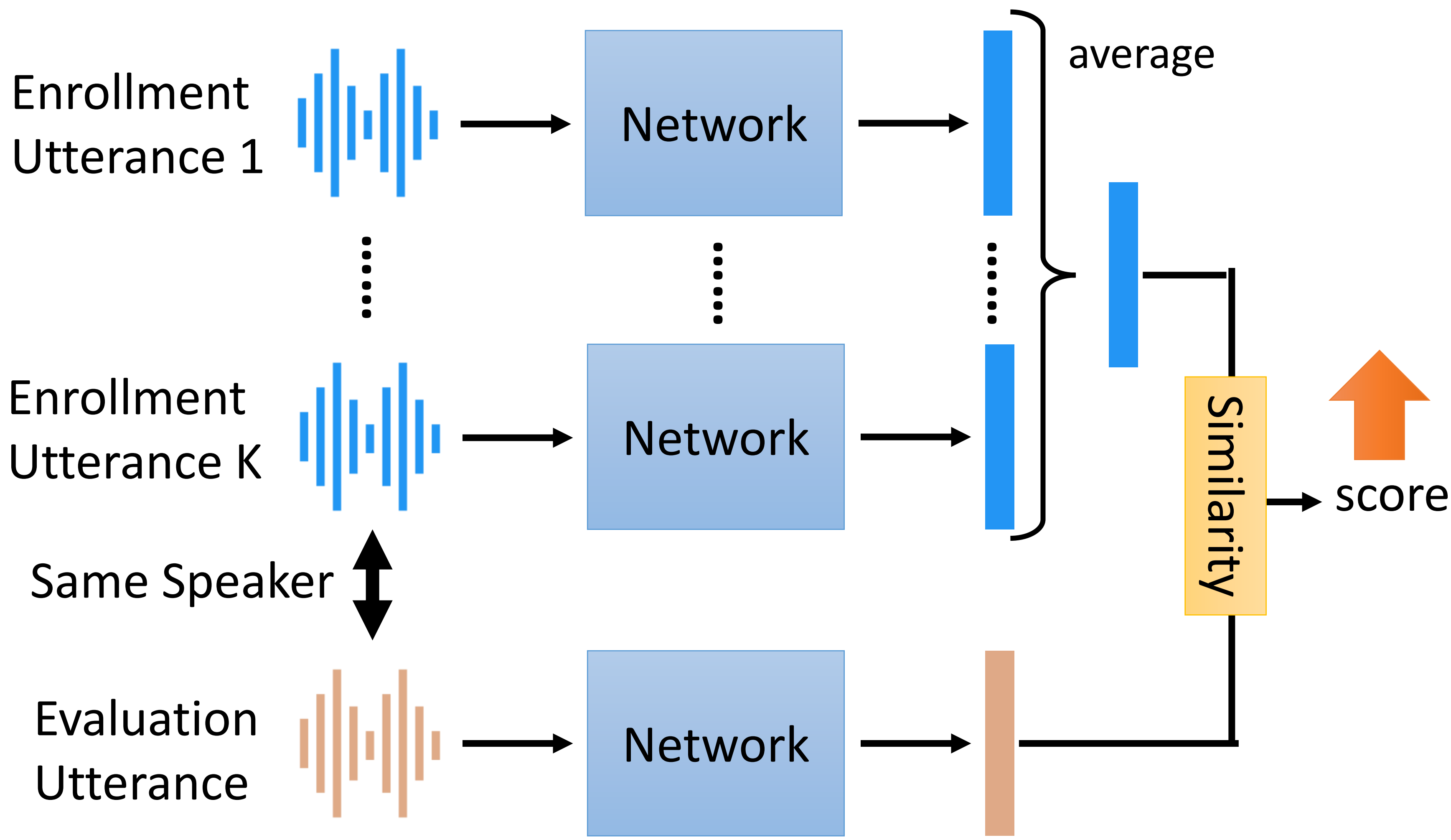
Speaker Verification

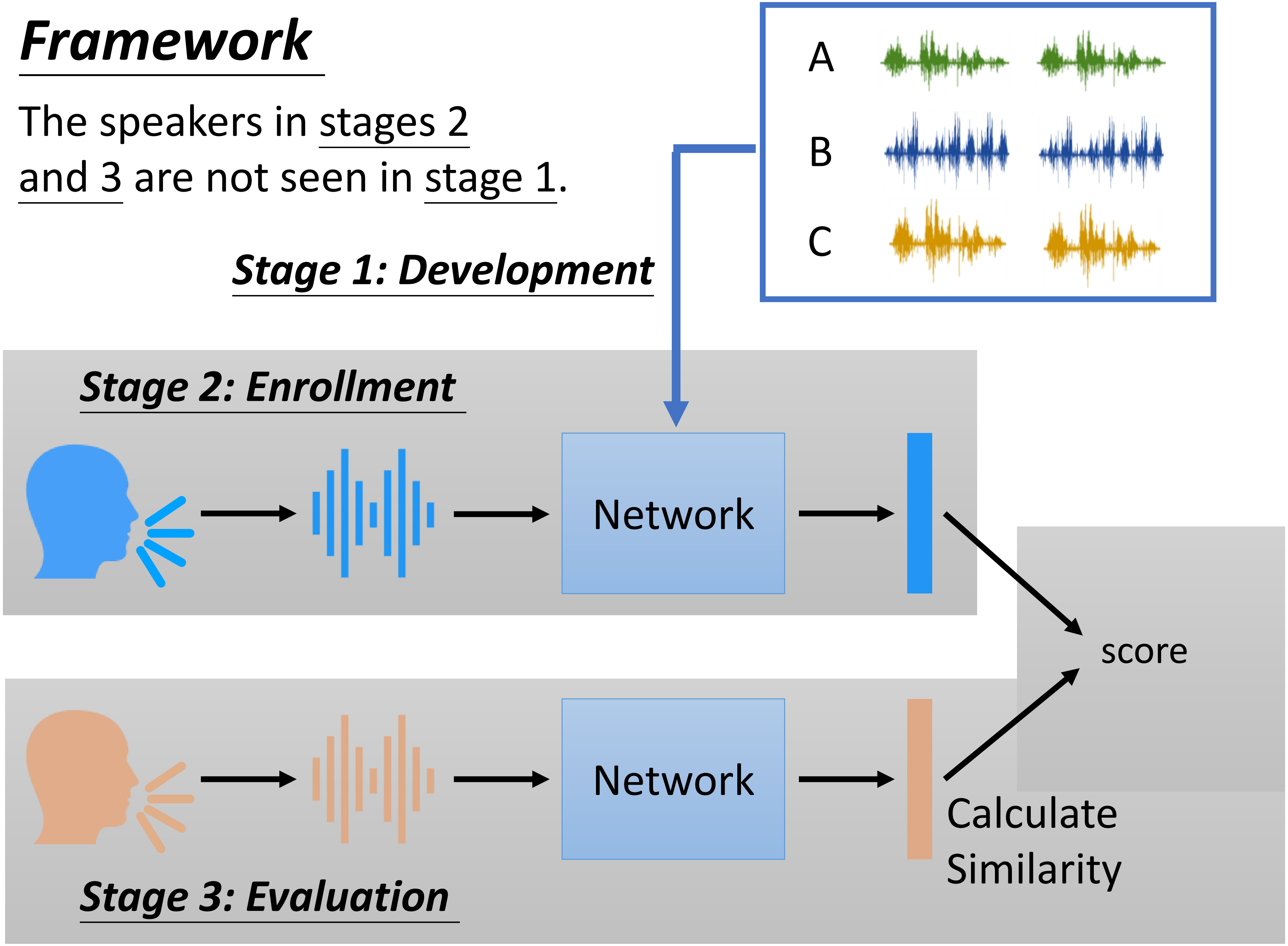
(Sound Event Detection, Keyword Spotting)

Speaker Verification



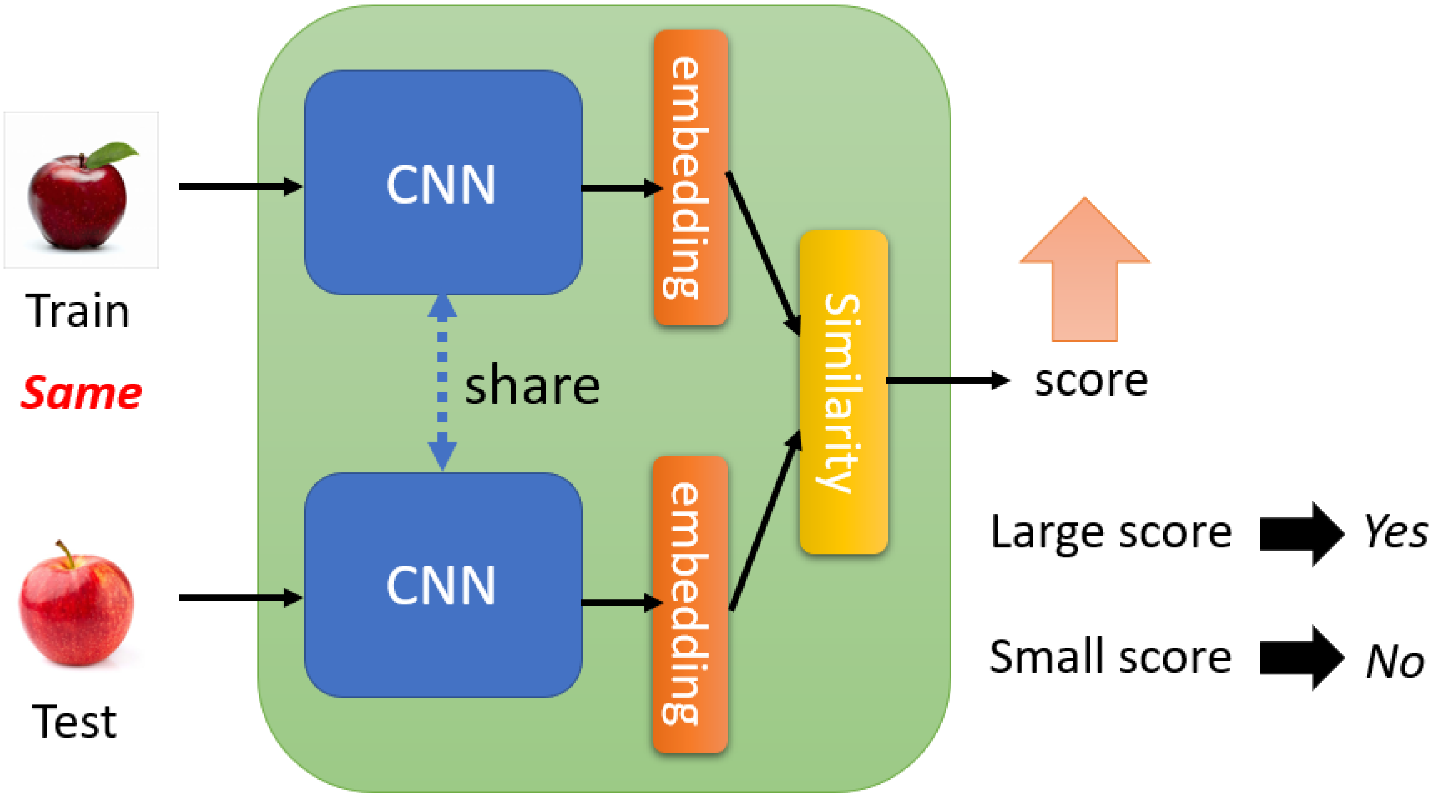
Speaker Verification




Framework
The speakers in stages 2 and 3 are not seen in stage 1.
Stage 1: Development
A
B
C
Stage 2: Enrollment
Network
score
Network
Calculate Similarity
Stage 3: Evaluation

First Example: Siamese Network

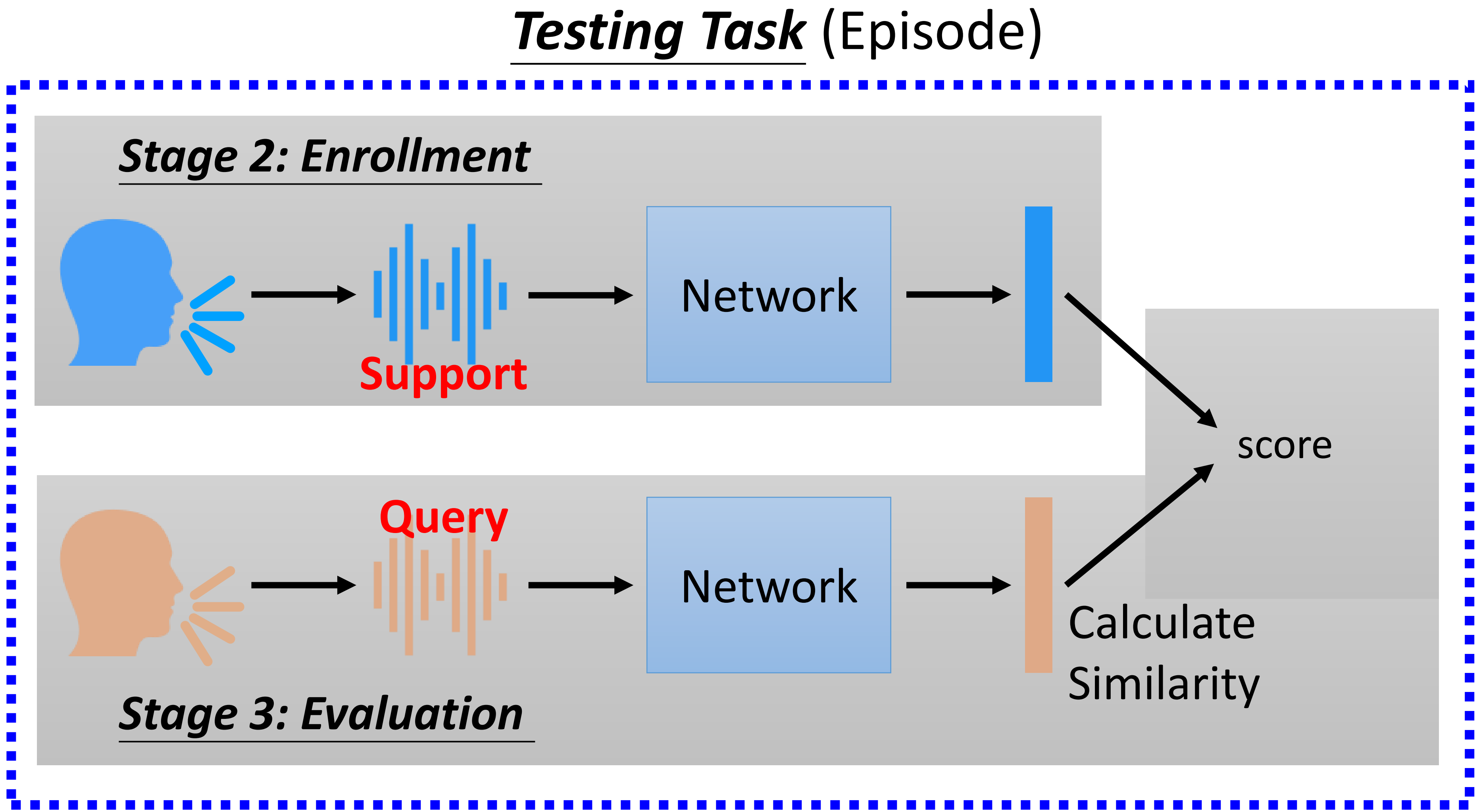
Koch, Zemel, Salakhutdinov, 2015

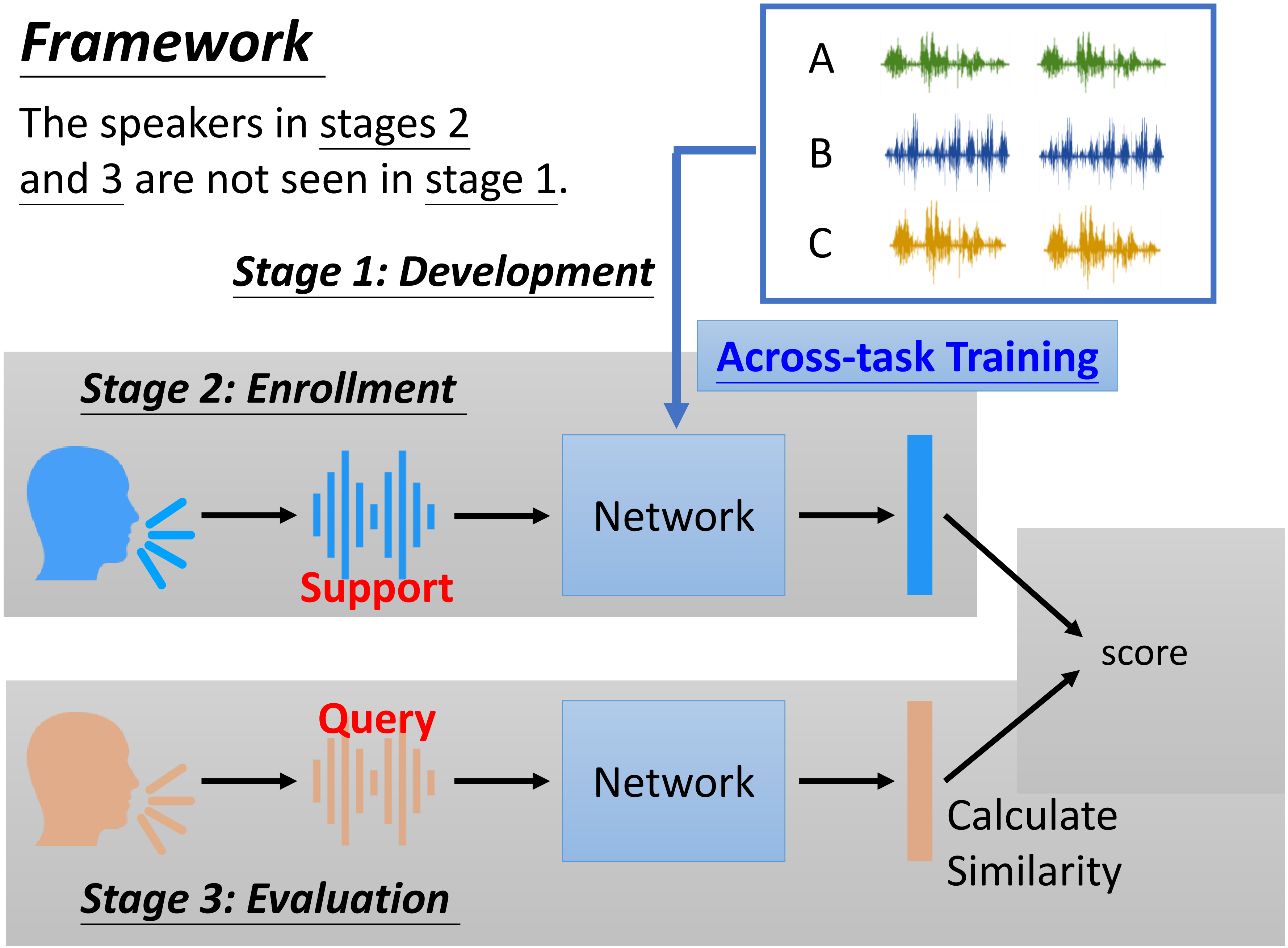


This is "*Learning to Compare*"!

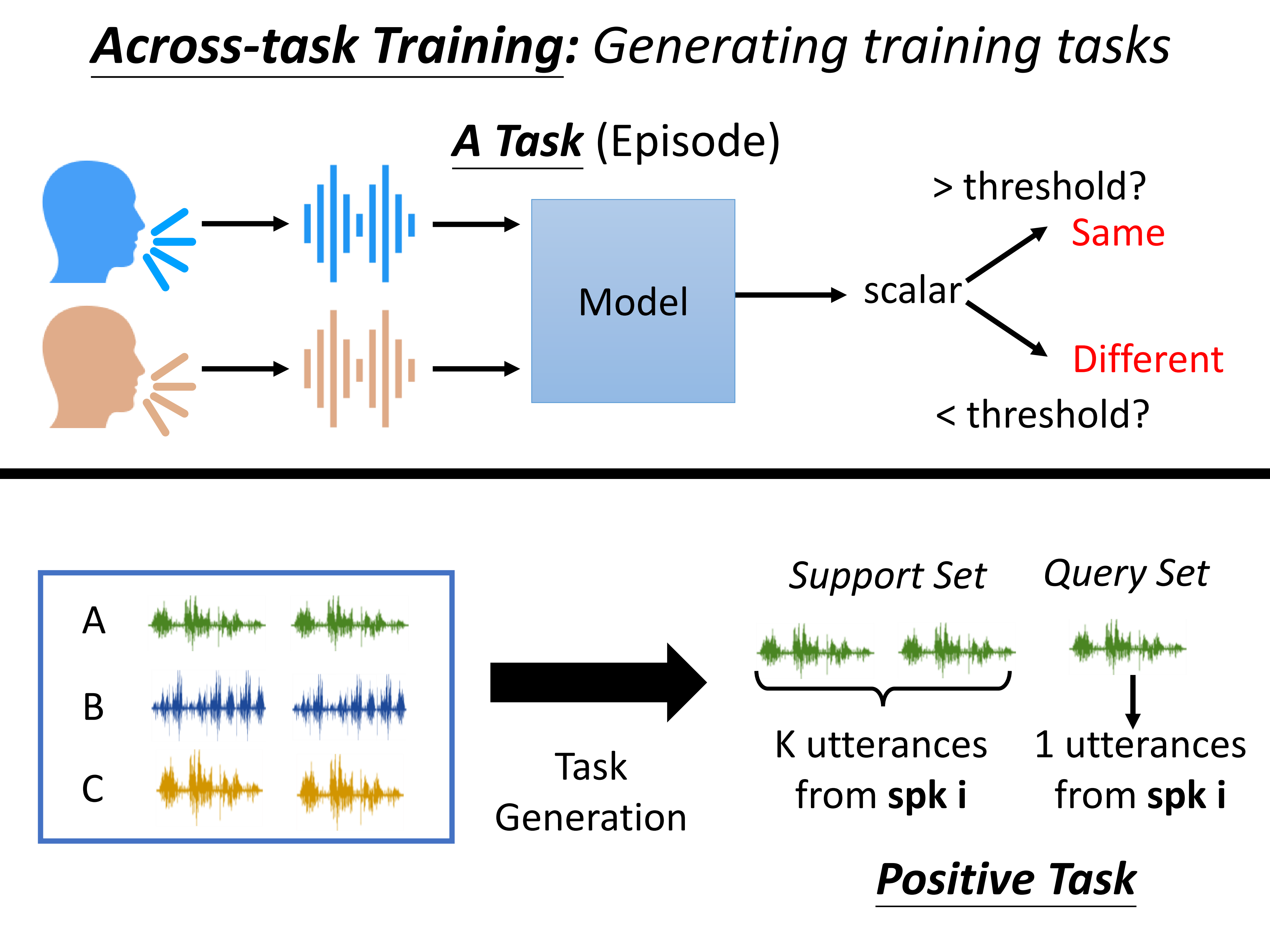
Framework

The speakers in stages 2 and 3 are not seen in stage 1.

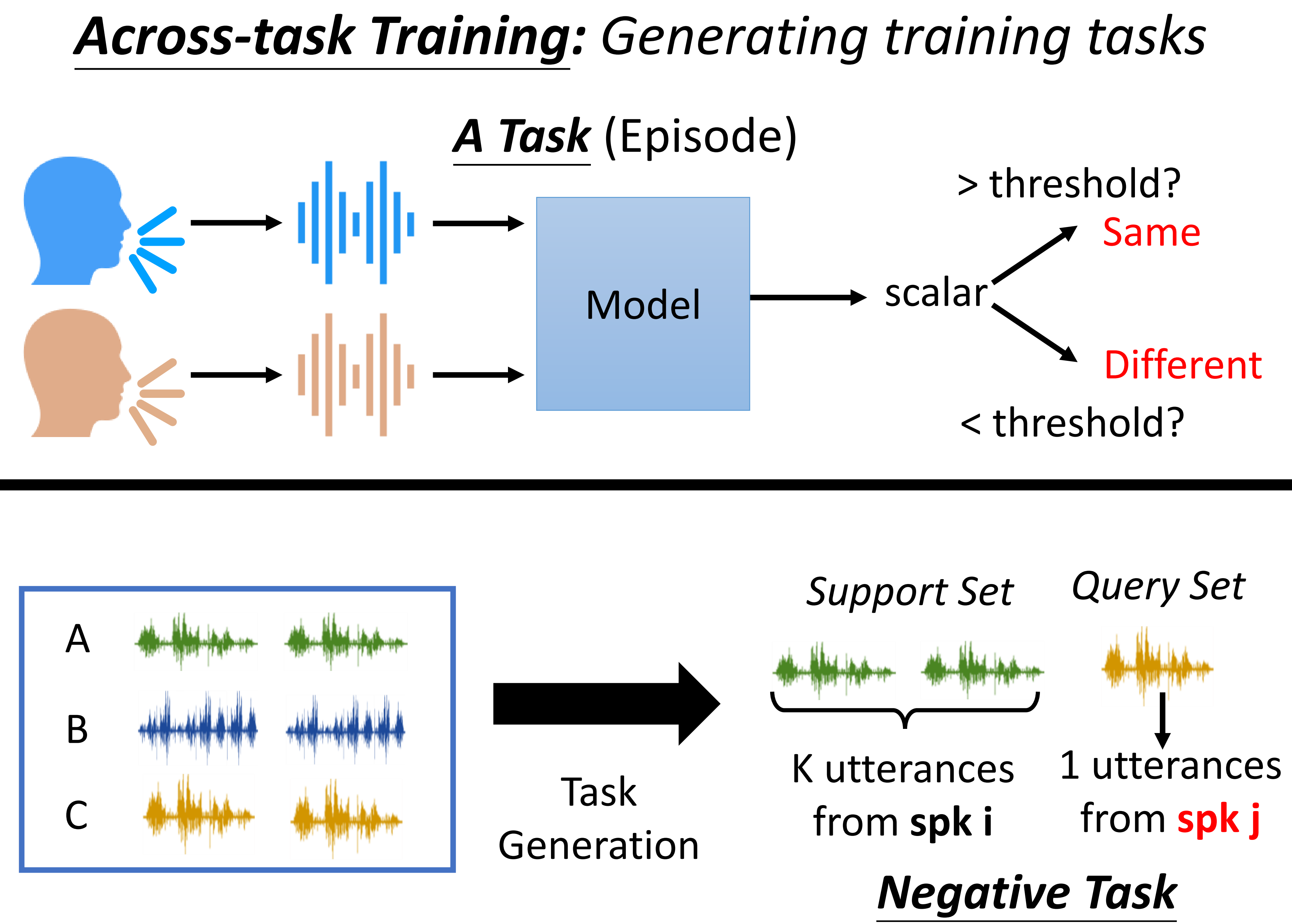




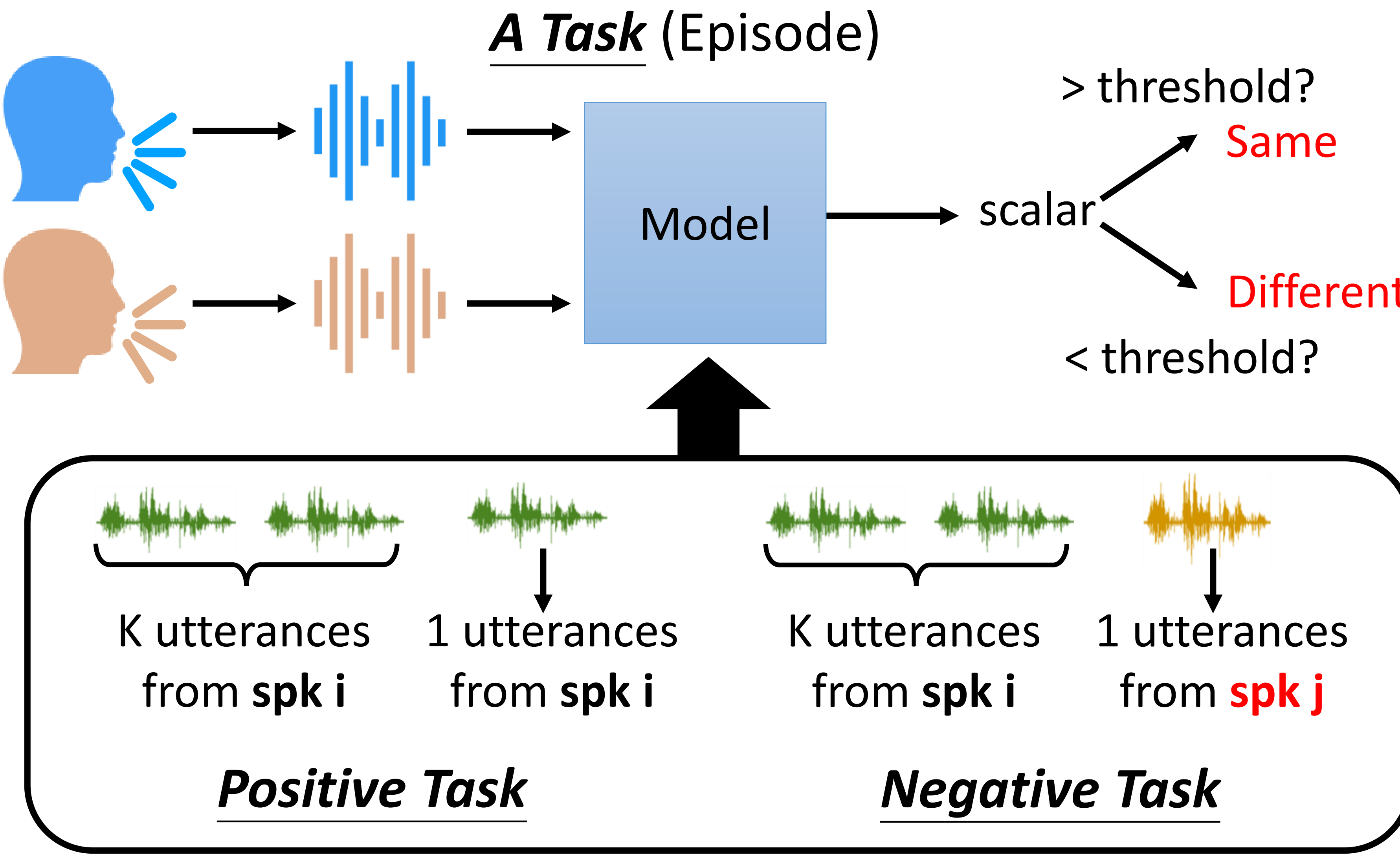
Framework
The speakers in stages 2 and 3 are not seen in stage 1.
Stage 1: Development
A
B
C
Across-task Training
Stage 2: Enrollment
Support
Network
score
Query
Network
Calculate Similarity
Stage 3: Evaluation



Across-task Training: Generating training tasks
A Task (Episode)
Model
scalar
> threshold?
Same
Different
< threshold?
A
B
C
Task Generation
Support Set
Query Set
K utterances from spk i
1 utterances from spk i
Positive Task


Across-task Training: Generating training tasks
A Task (Episode)
Model
scalar
> threshold?
Same
Different
< threshold?
A
B
C
Task
Generation
Support Set
Query Set
K utterances
from spk i
1 utterances
from spk j
Negative Task

Across-task Training: *Generating training tasks*

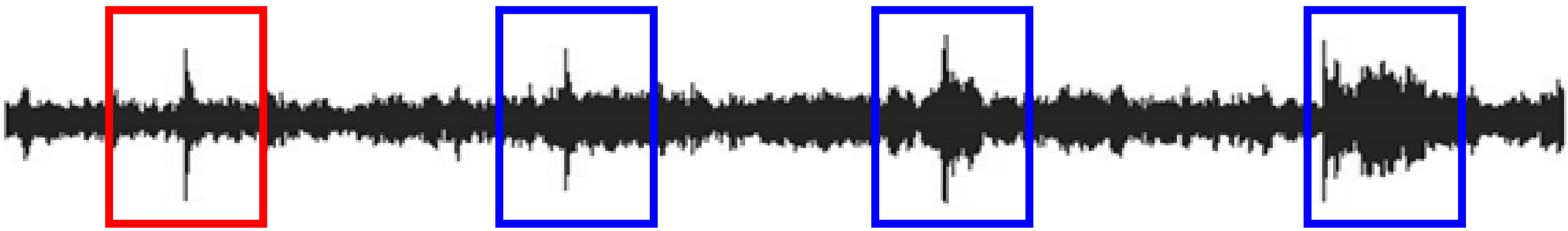


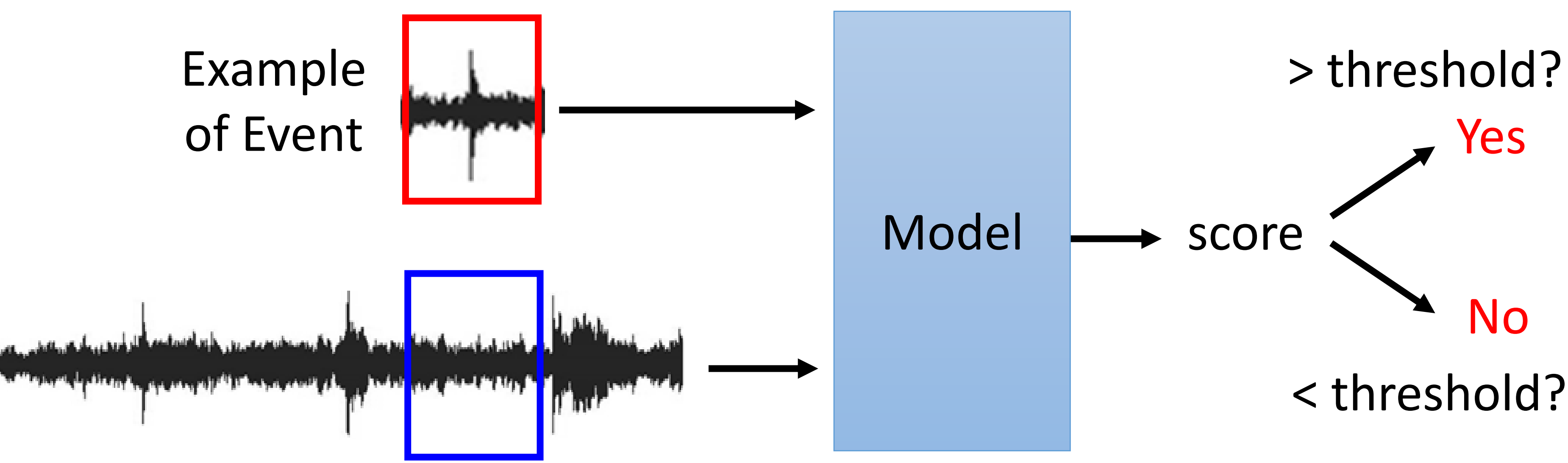
Also refer to generalized end-to-end (GE2E)

Li Wan, et al., Generalized End-to-End Loss for Speaker Verification, ICASSP, 2018

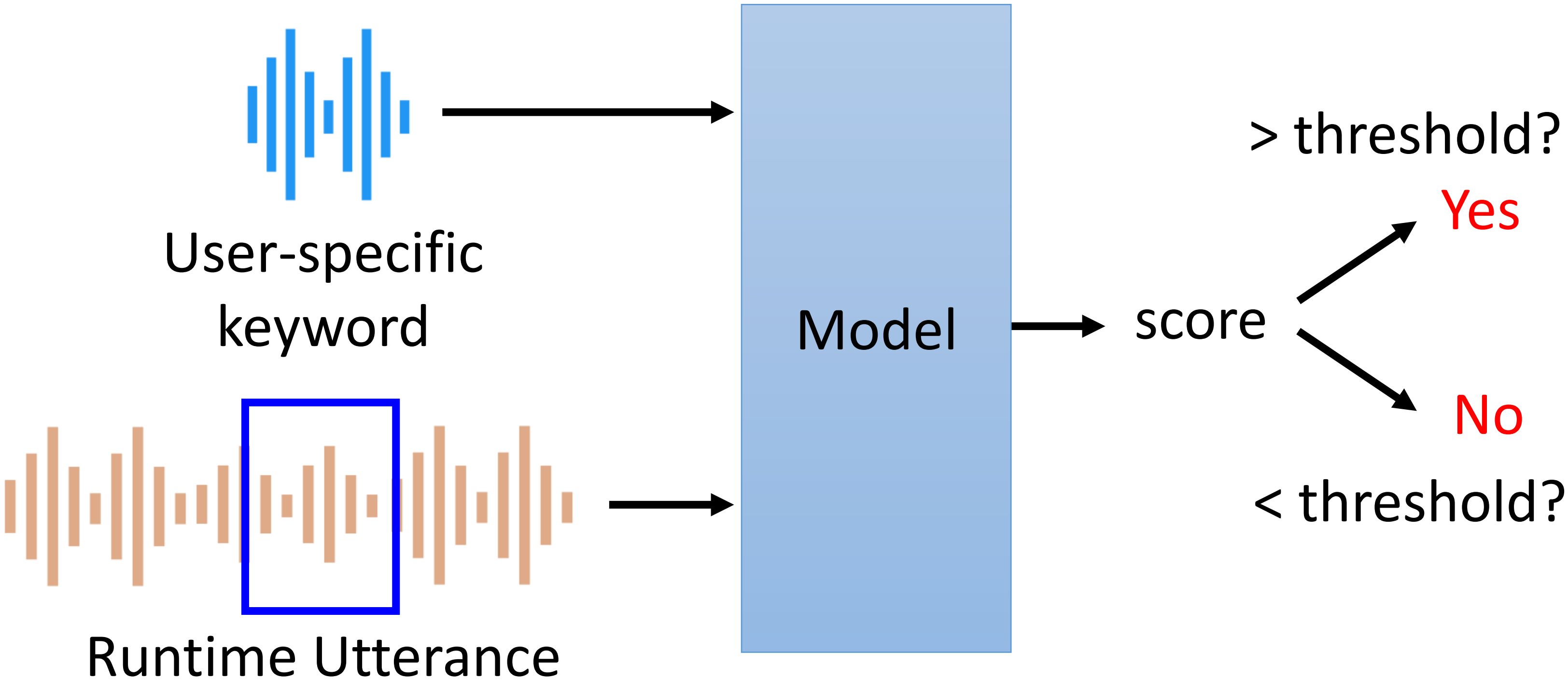
Sound Event Detection

Find this kind of sound event

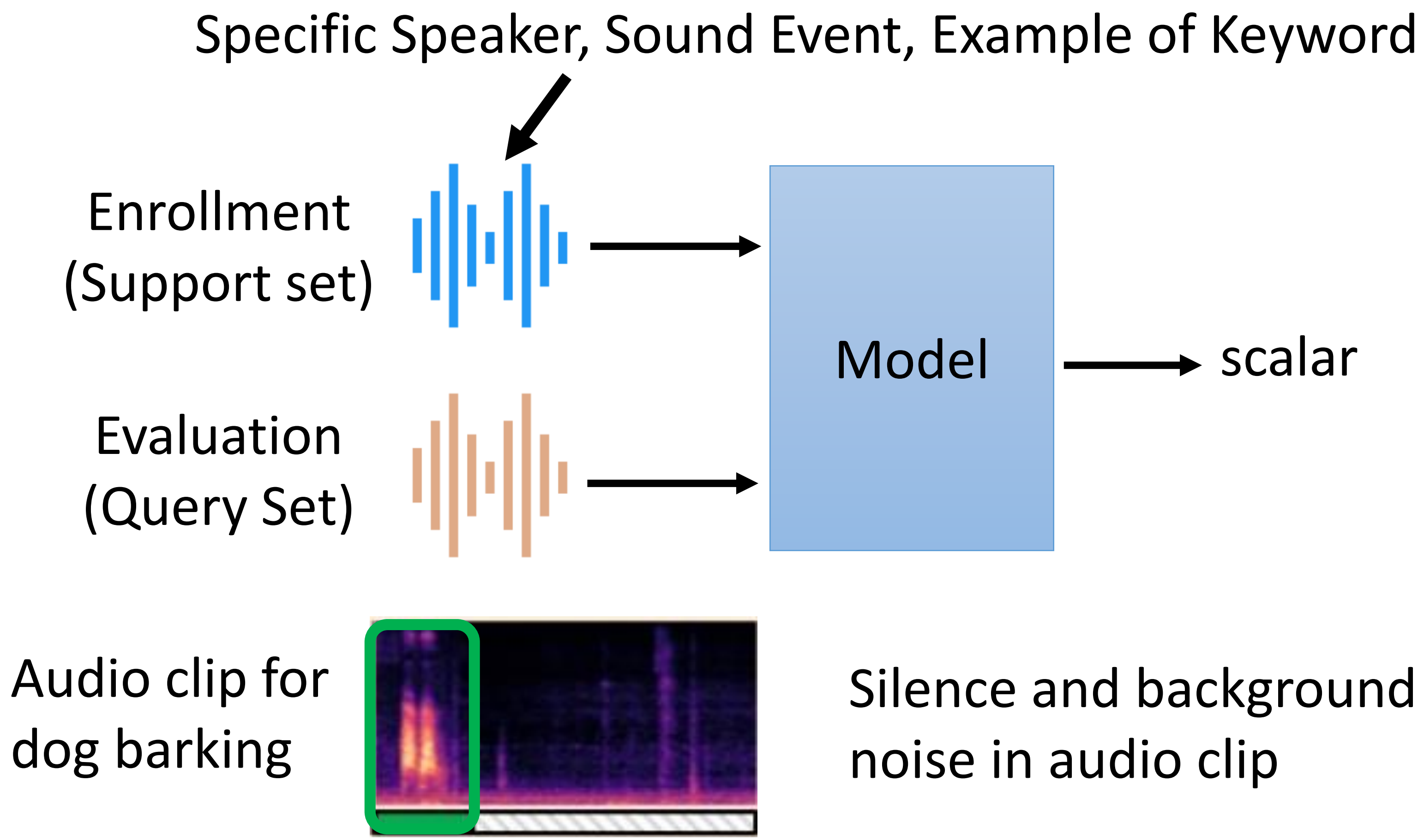




Query-by-Example Keyword Spotting



Learning to Compare in Audio

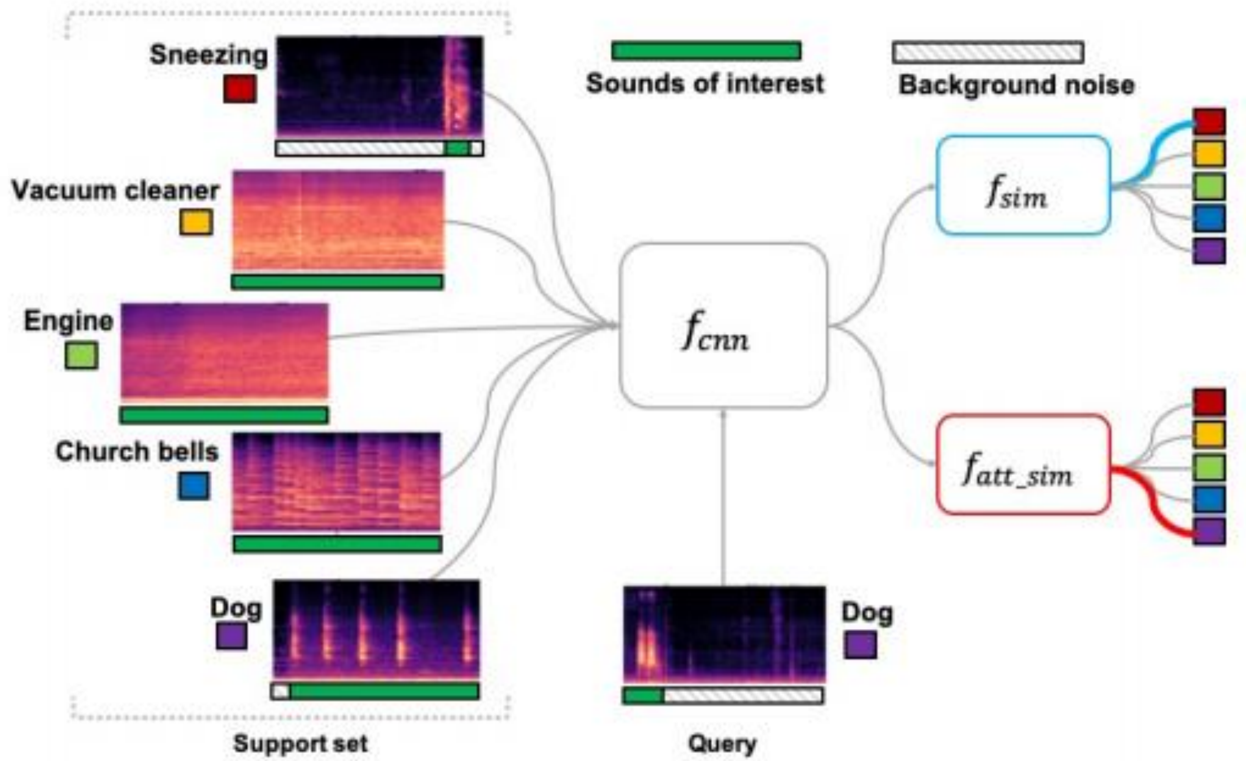


“*Attention*” is used to gather useful information in audio clips

Source of image: https://arxiv.org/pdf/1812.01269.pdf

Attentional Similarity

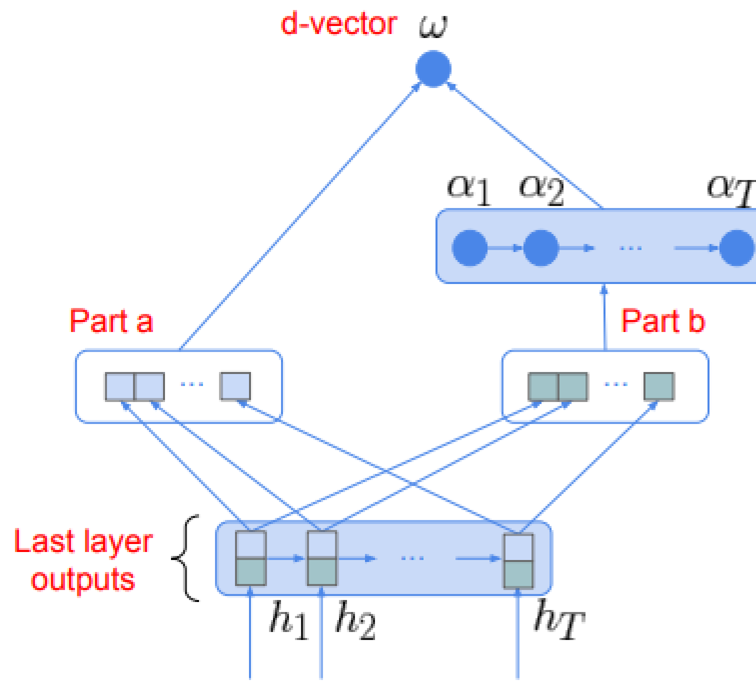
- Sound Event



Szu-Yu Chou, Kai-Hsiang Cheng, Jyh-Shing Roger Jang, Yi-Hsuan Yang, Learning to match transient sound events using attentional similarity for few-shot sound recognition, ICASSP, 2019

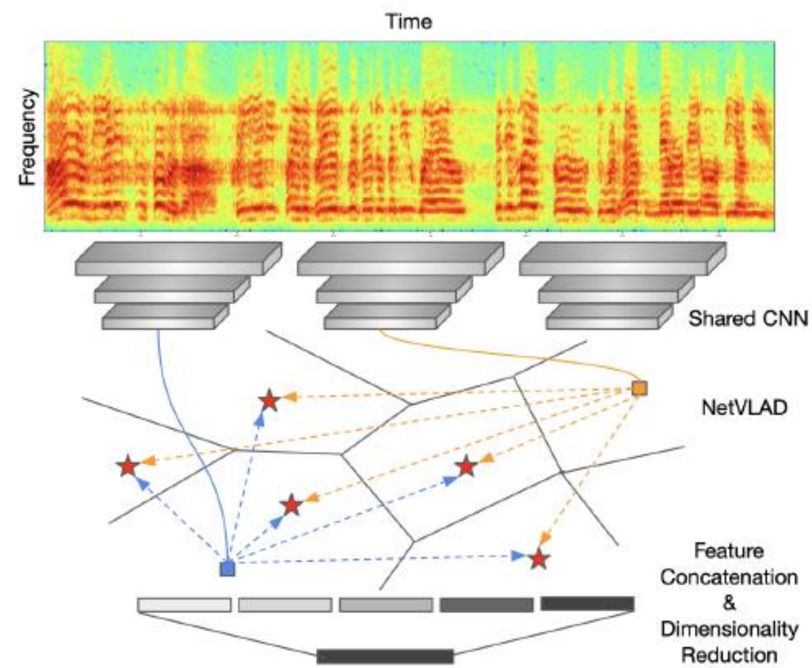
Attention Mechanism

Chowdhury, et al., Attention-Based Models for Text-Dependent Speaker Verification, ICASSP, 2018



NetVLAD

Xie et al., Utterance-level Aggregation For Speaker Recognition In The Wild, ICASSP, 2019



VLAD = Vector of Locally Aggregated Descriptors

More Speaker Verification

- More Learning to compare
  - Georg Heigold, et al., End-to-End Text-Dependent Speaker Verification, ICASSP, 2016
  - Li Wan, et al., Generalized End-to-End Loss for Speaker Verification, ICASSP, 2018
  - Jixuan Wang, et al., Centroid-based deep metric learning for speaker recognition, ICASSP, 2019
  - Tom Ko, , et al., Prototypical Networks for Small Footprint Text-Independent Speaker Verification, ICASSP, 2020
  - Seong Min Kye, et al., Meta-Learning for Short Utterance Speaker Recognition with Imbalance Length Pairs, INTERSPEECH 2020
  - Joon Son Chung, et al., In defence of metric learning for speaker recognition, INTERSPEECH 2020

More Speaker Verification

- Also architecture search
  - Shaojin Ding, et al., AutoSpeech: Neural Architecture Search for Speaker Recognition, INTERSPEECH, 2019
- Learn to initialize
  - Jiawen Kang, Ruiqi Liu, Lantian Li, Dong Wang, Thomas Fang Zheng, Domain-Invariant Speaker Vector Projection by Model-Agnostic Meta-Learning, INTERSPEECH, 2020

More Sound Event Detection

- More Learning to compare
  - Pranay Manocha, et al., Content-based Representations of audio using Siamese neural networks, ICASSP, 2018
  - Kazuki Shimada, et al., Metric Learning with Background Noise Class for Few-shot Detection of Rare Sound Events, ICASSP 2020
  - Yu Wang, et al., Few-Shot Sound Event Detection, ICASSP, 2020
  - Bowen Shi, Ming Sun, Krishna C. Puvvada, Chieh-Chi Kao, Spyros Matsoukas, Chao Wang, Few-Shot Acoustic Event Detection Via Meta Learning, ICASSP 2020
- Also architecture search
  - Jixiang Li, et al., Neural Architecture Search on Acoustic Scene Classification, INTERSPEECH, 2020

Keyword Spotting / Intent Detection

- Learn to compare
  - Ashish Mittal, Samarth Bharadwaj, Shreya Khare, Saneem Chemmengath, Karthik Sankaranarayanan, Brian Kingsbury, Representation based meta-learning for few-shot spoken intent recognition, INTERSPEECH, 2020
- Learning to initialize
  - Yangbin Chen, et al., An Investigation of Few-Shot Learning in Spoken Term Classification, INTERSPEECH, 2020
- Also architecture search
  - Tom Véniat, et al., Stochastic Adaptive Neural Architecture Search for Keyword Spotting, ICASSP, 2019
  - Hanna Mazzawi, et al., Improving Keyword Spotting and Language Identification via Neural Architecture Search at Scale, INTERSPEECH, 2019
  - Bo Zhang, et al., UTOKWS: KEYWORD SPOTTING WITH DIFFERENTIABLE ARCHITECTURE SEARCH, arXiv, 2020
  - Tong Mo, et al., Neural Architecture Search For Keyword Spotting, INTERSPEECH, 2020

Emotion Recognition

- Sample weighting
  - Takuya Fujioka, Takeshi Homma, Kenji Nagamatsu, META-LEARNING FOR SPEECH EMOTION RECOGNITION CONSIDERING AMBIGUITY OF EMOTION LABELS, INTERSPEECH 2020
- Learn to initialize
  - Ruichu Cai, Kaibin Guo, Boyan Xu, Xiaoyan Yang, Zhenjie Zhang, Meta Multi-task Learning for Speech Emotion Recognition, INTERSPEECH 2020

Application to ConvAI

- Intent classification
- Named-entity recognition (NER), slot tagging
- Dialog state tracking
- Question answering
- Dialog generation

Application to ConvAI

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price

Which city are you looking for?

I am thinking of Boston

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

I would like to celebrate my birthday

Sorry, could you be more specific?

Which hotel provides lounge for party

Hotel XYZ has that

Application to ConvAI

Intent classification

How are you today?

Phatic

Pretty good. How can I help you?

I want a Hotel of moderate price

BookHotel

Which city are you looking for?

I am thinking of Boston

BookHotel

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

I would like to celebrate my birthday

Out of Domain

Sorry, could you be more specific?

Which hotel provides lounge for party

InquireAmenity

Hotel XYZ has that

Application to ConvAI

NER, Slot tagging

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price

PriceRange

Which city are you looking for?

I am thinking of Boston

City

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

I would like to celebrate my birthday

Sorry, could you be more specific?

Which hotel provides lounge for party

Hotel XYZ has that

Application to ConvAI

Dialog state tracking

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price → (BookHotel, PriceRange = Moderate)

Which city are you looking for?

I am thinking of Boston → (BookHotel, PriceRange = Moderate)
(BookHotel, City = Boston)

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

I would like to celebrate my birthday

Sorry, c ... ific?

Which hotel provides lounge for party → (BookHotel, PriceRange = Moderate)
(BookHotel, City = Boston)
(Inquire, Amenity = lounge for party)

Hotel XYZ has that

Application to ConvAI

Question Answering

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price

Which city are you looking for?

I am thinking of Boston

+

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

I would like to celebrate my birthday

Sorry, could you be more specific?

Context

Which hotel provides lounge for party

Question

Hotel XYZ has that

Answer

Application to ConvAI

Dialog generation

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price

Which city are you looking for?

\+

Context

I am thinking of Boston

Query

Response

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

Intent classification

- Zero-shot OOD detection and few-shot ID classification
- OOD-resistant Prototypical Network

Intent Label	Example
Help_List	List what you can help me with.
	Watson, I need your help
Schedule_Appointment	Can you book a cleaning with my dentist for me?
	Can you schedule my dentist's appointment?
End_Meeting	You can end the meeting now
	Meeting is over
...	...
OOD utterances	My birthday is coming!
	blah blah...

Tan et al. "Out-of-Domain Detection for Low-Resource Text Classification Tasks" EMNLP 2019

Intent classification

- Across task training

Task 1

Sample Intent *in, out,* and 4 *others* from D^{across_train}

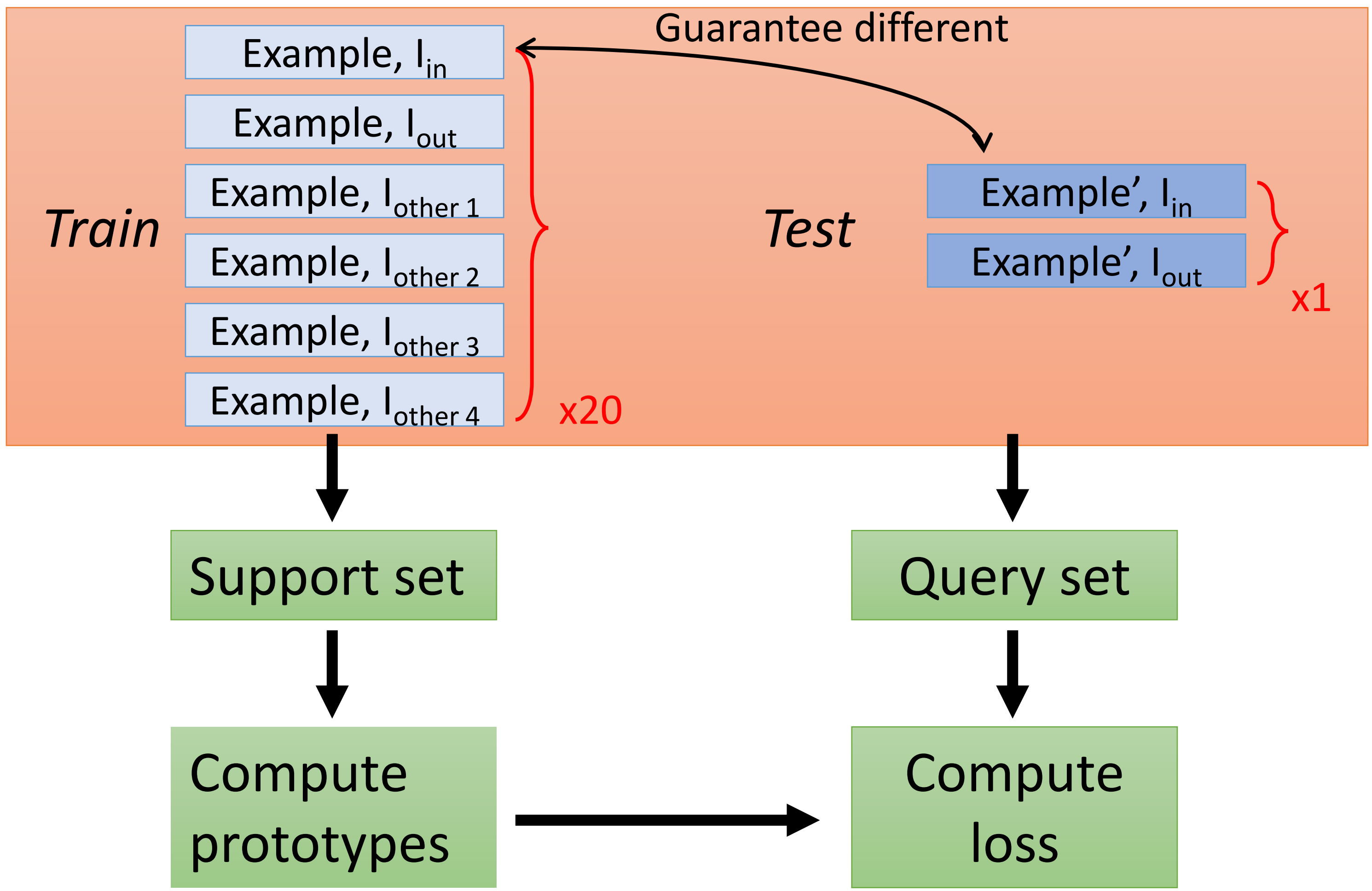
5-way (in + 4)

Task 2

Sample Intent *in, out,* and 4 *others* from D^{across_train}

…

Task N

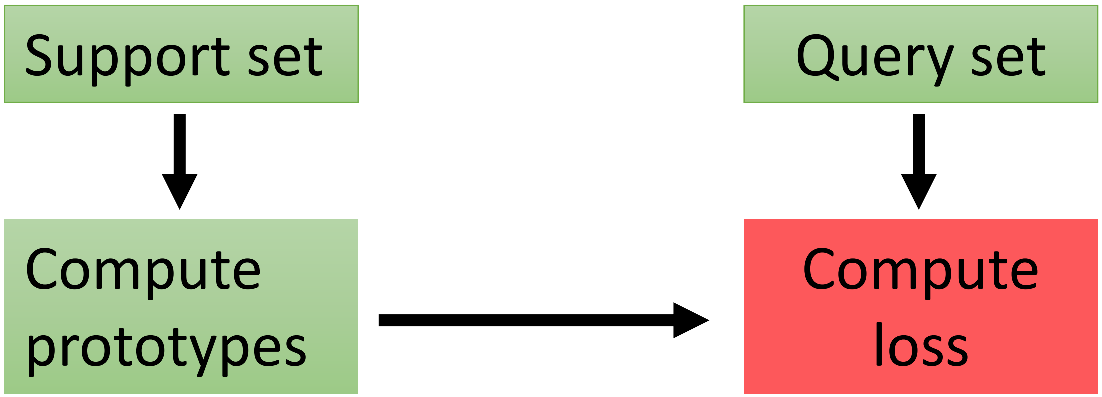


Intent classification

$$\mathcal{L}_{in} = -\log \frac{\exp \alpha F(x_i^{in}, S_{l_i}^{in})}{\sum_{l'} \exp \alpha F(x_i^{in}, S_{l'}^{in})}$$

$$\mathcal{L}_{ood} = max[0, \max_l (F(x_j^{out}, S_l^{in}) - \mathcal{M}_1)]$$

$$\mathcal{L}_{gt} = max[0, \mathcal{M}_2 - F(x_i^{in}, S_{l_i}^{in}))]$$



Intent classification

- Dataset: Amazon review, conversation
- Metrics: intent classification error rate at EER

	Amazon review	Conversation
LSTM AutoEnc.	38.6	79.5
CNN	42.8	77.6
O-Proto (proposed)	**29.1**	40.8

- Meta-learning (Proto) > Supervised learning (CNN / / LSTM)

NER

x I am thinking of Boston
y O O O O B_{City}

x Which hotel provides lounge for party
y O O O $B_{Amenity}$ $I_{Amenity}$ $I_{Amenity}$

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \mathcal{S}) = \frac{1}{Z}\exp(\text{TRANS}(\boldsymbol{y}) + \lambda \cdot \text{EMIT}(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{S})),$$

$$Z = \sum_{\boldsymbol{y}' \in \boldsymbol{Y}} \exp(\text{TRANS}(\boldsymbol{y}') + \lambda \cdot \text{EMIT}(\boldsymbol{y}', \boldsymbol{x}, \boldsymbol{S})),$$

$$\text{TRANS}(\boldsymbol{y}) = \textstyle\sum_{i=1}^{n} f_T(y_{i-1}, y_i) \quad f_T(y_{i-1}, y_i) = p(y_i \mid y_{i-1}).$$

Hou et al. "Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network" ACL 2020

NER

- Abstract labels: O, sB, dB, sI and dI

current

O	B-time	B-weather	I-time	I-weather	
0.6	0.4	0.4	0	0	O
0.3	0.1	0.1	0.5	0	B-time
0.3	0.1	0.1	0	0.5	B-weather
0.6	0.1	0.1	0.2	0	I-time
0.6	0.1	0.1	0	0.2	I-weather

previous

→

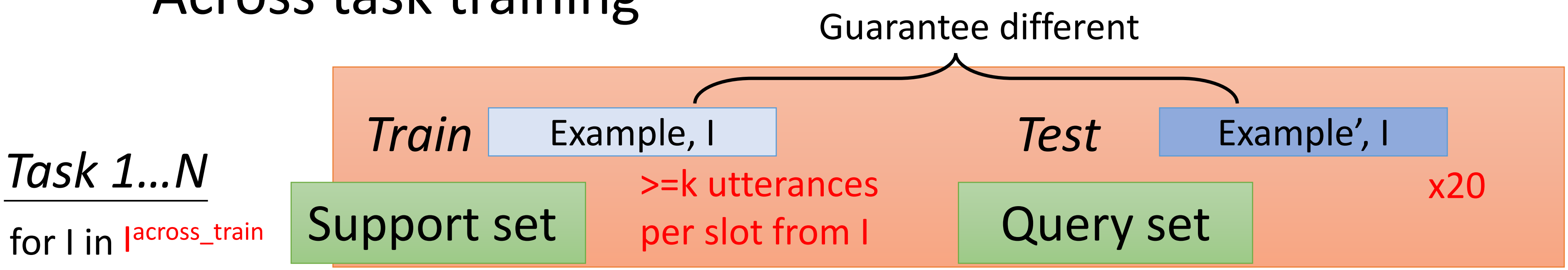
current

	O	sB	dB	sI	dI
O	0.6	0.4	\	0	\
B	0.3	0.1	0.1	0.5	0
I	0.6	0.1	0.1	0.2	0

previous

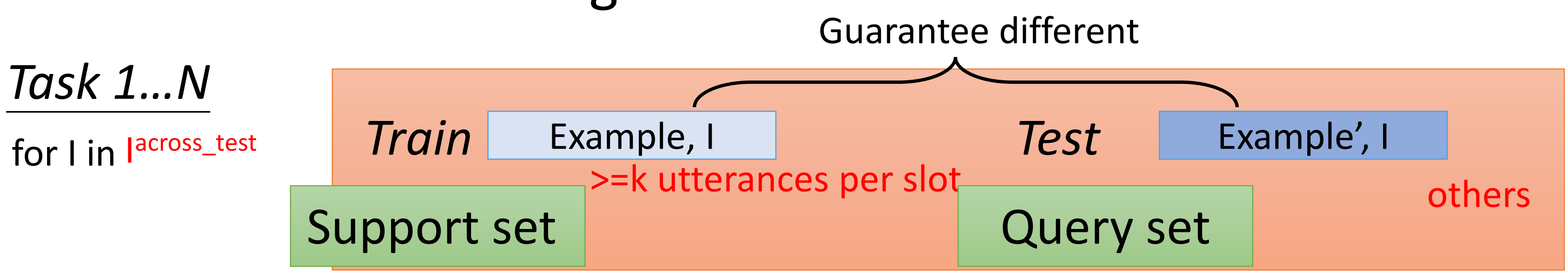
NER

- Across task training



compute transition over tasks

- Across task testing



compute emissions

Transition prob from across-task training

NER

- SNIPS
  - 5 for across-task training, 1 for dev, 1 for across-task test
  - 5-shot

	We	Mu	Pl	Bo	Se	Re	Cr	Avg
BiLSTM	25.2	39.8	46.1	74.6	53.5	40.4	25.1	43.5
SimBERT	53.5	54.1	42.8	75.5	57.1	55.3	32.4	53.0
TransferBERT	59.4	42.0	46.1	20.7	28.2	67.8	58.6	46.1
Meta-learning	67.8	56.0	46.0	72.2	73.6	60.2	66.9	63.2
Meta-learning + transition	**74.7**	**56.7**	**52.2**	**78.8**	**80.6**	**69.6**	**67.5**	**68.6**

- Meta-learning > transfer learning > no transfer
- Transition helps

Cross-lingual NER

- Goal: source -> target
- Previously: annotation projection
- Meta-learning for language-independent features
  - On top of multilingual BERT
  - Further improvement

Wu et al. “Enhanced Meta-Learning for Cross-lingual Named Entity Recognition with Minimal Resources ” AAAI 2020

Cross-lingual NER

- Across task training

Guarantee different

Train Examples x_j *Test* x_i

Task 1…N

for x_i in $D^S = \{x_i\}_{i=1}^N$

Support set

top-k similar examples from D^s

Query set

MAML – model: M

- Across task testing

Guarantee different

Task 1…N

for x_i in D^T

Train Examples xj *Test* x_i

Support set

top-k similar examples from D^s

Query set

Cross-lingual NER

- English-en / German-de / Spanish-es / Dutch-nl, (CoNLL); French-fr, Europeana News; Chinese-zh , MSRA
- Results (F1-scores)

	es	nl	de	fr	zh	Avg
Ni, Dinu, and Florian (2017)	65.1	65.4	58.5	-	-	-
Mayhew, Tsai, and Roth (2017)	66.0	66.5	59.1	-	-	-
Xie et al. (2018)	72.4	71.3	57.8	-	-	-
Multilingual BERT	74.6	79.6	70.8	50.9	76.4	70.5
Meta-cross-lingual NER	**76.8**	**80.4**	**73.2**	**55.3**	**77.9**	**72.7**

- Meta-learning > multilingual BERT > annotation projection

Dialog state tracking

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price

Which city are you looking for?

Dialog context

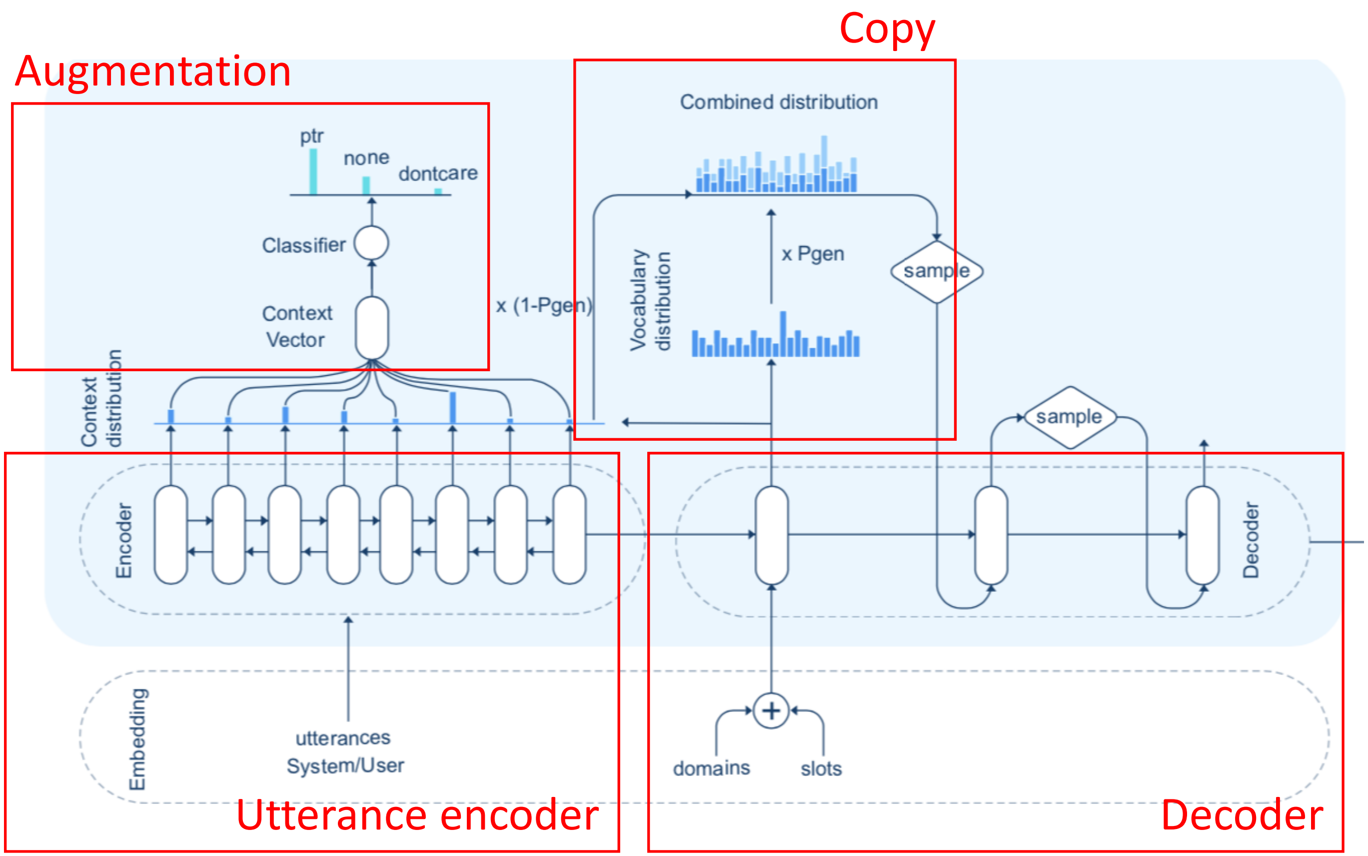
I am thinking of Boston

Current utterance

(BookHotel, PriceRange = Moderate)
(BookHotel, City = Boston)

Dialog frame

Dialog state tracking



Dialog state tracking

Algorithm 2 MAML algorithm

Input: D_d^{train}; D_d^{valid}; α; β.
Output: Trained model M with MAML algorithm.

```
while not done do
    for each domain d do
        Select a batch of size from D_d^train and D_d^valid to get D_d^t and D_d^v;
        Pre-update model with gradient descent:
        M'_d ← M − α∇_M L_d(M, D_d^t)
        Compute L_d(M'_d, D_d^v) using D_d^v;
    end for
    Update the current model M:
    M ← M − β∇_M Σ_d L_d(M'_d, D_d^v)
end while
return meta-learned model M;
```

Support set: D_d^t

Query set: D_d^v

Pre-update: $M'_d \leftarrow M - \alpha \nabla_M L_d(M, D_d^t)$

Update meta-learn model M: $M \leftarrow M - \beta \nabla_M \sum_d L_d(M'_d, D_d^v)$

Huang et al. “Meta-Reinforced Multi-Domain State Generator for Dialogue Systems” ACL 2020

Dialog state tracking

- MultiWOZ
  - Restaurant, hotel, train (source)
  - Taxi / attrachtion (new)
- Results

New Domain (proportion)	Initialization	Joint Acc.	Slot Acc.
Taxi (1%)	Public BERT	60.6	73.3
	Supervised-learning	59.0	78.7
	Meta-learning	**64.4**	**83.2**
Attraction (1%)	Public BERT	27.9	63.4
	Supervised-learning	29.1	62.2
	Meta-learning	**43.1**	**74.3**

- Meta-learning > Supervised with source domain >= BERT (no pre-training with matched tasks)

Question answering

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price

Which city are you looking for?

I am thinking of Boston

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

\+

I would like to celebrate my birthday

Sorry, could you be more specific?

Context

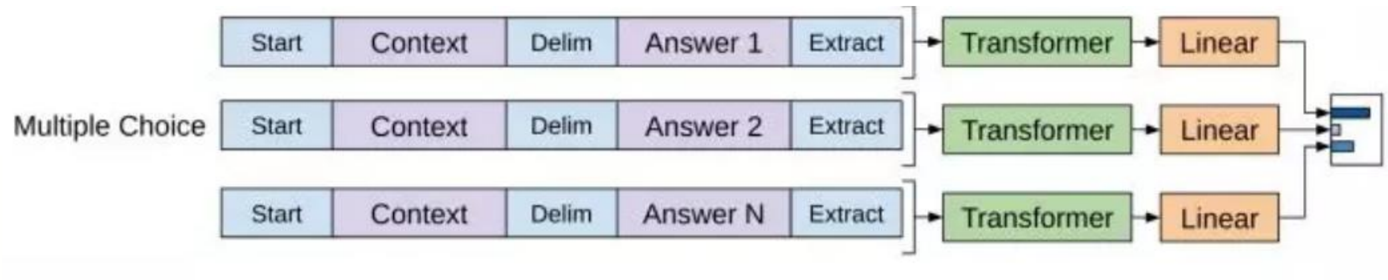
Which hotel provides lounge for party

Question

Hotel XYZ has that

Answer

Question answering



Yan et al. “Multi-source Meta Transfer for Low Resource Multiple-Choice Question Answering” ACL 2020

Question answering

Input: Task distribution over source $p^s(\tau)$, data distribution over target $p^t(\tau)$, backbone model $f(\theta)$, learning rates in MMT α, β, λ.

Output: Optimized parameters θ.

Initialize θ from backbone model;
while *not done* **do**
  for *all source S* **do**
    Sample batch of tasks $\tau_i^s \sim p^s(\tau)$;
    for *all τ_i^s* **do**
      Evaluate $\nabla_\theta L_{\tau_i^s}(f(\theta))$ with respect to K examples;
      Compute gradient for fast adaption: $\theta' = \theta - \alpha \nabla_\theta L_{\tau_i^s}(f(\theta))$;
    end
    Meta model update: $\theta = \theta - \beta \nabla_\theta \sum_{\tau_i^s \sim p^s(\tau)} L_{\tau_i^s}(f(\theta'))$;
    Get batch of data $\tau_i^t \sim p^t(\tau)$;
    for *all τ_i^t* **do**
      Evaluate $\nabla_\theta L_{\tau_i^t}(f(\theta))$ with respect to K examples;
      Gradient for target fine-tuning: $\theta = \theta - \beta \nabla_\theta L_{\tau_i^t}(f(\theta))$;
    end
  end
end

Perform typical MAML on each source

Within task training with support set

Compute loss with query set and θ’

Within task training with target domains

Question answering

Get all batches of data $\tau_i^t \sim p^t(\tau)$;

for *all* τ_i^t **do**

  Evaluate $\nabla_\theta L_{\tau_i^t}(f(\theta))$ with respect to batch size;

  Gradient for meta transfer learning: $\theta = \theta - \lambda \nabla_\theta L_{\tau_i^t}(f(\theta))$;

end

Adaptation

Question answering

Name	DREAM	RACE	MCTEST	SemEval	SWAG
Type	Dialogue	Exam	Story	Narrative Text	Scenario Text
Ages	15+	12-18	7+	-	-
Generator	Expert	Expert	Crowd.	Crowd.	AF./Crowd.
Level	High School/College	High/Middle School	Children	Unlimited	Unlimited
Choices	3	4	4	2	4
Samples	6,444	27,933	660	2,119	92,221
Questions	10,197	97,687	2,640	13,939	113,557

- Source: RACE + SWAG, target: McTEST

Method	Target labels	Acc.
Supervised-learning (BERT-base)	Yes	68.0
IMC [1]	Yes	76.6
Transfer-learning (BERT-base)	No	79.2
Meta-learning (BERT-base)	**No**	**81.6**

- Meta-learning > previous SOTA and simple transfer learning

Question answering

- Target: DREAM/McTEST/SemEval, source: rest 4

Method	DREAM	MCTEST	SemEval
GPT+Strategies [2]	-	81.9	89.5
IMC [1]	-	76.6	-
XLNet [3]	72.0	-	-
Supervised-learning (BERT-base)	61.6	68.0	87.5
Supervised-learning (RoBERTa)	84.4	87.3	94.0
Meta-learning (BERT-base)	68.9	82.0	88.9
Meta-learning (RoBERTa)	**85.6**	**88.8**	**94.2**

- Meta-learning > previous SOTA and supervised learning

Dialog generation

How are you today?

Pretty good. How can I help you?

I want a Hotel of moderate price

Which city are you looking for?

I am thinking of Boston

Sure, there are 5 hotels in Boston moderately priced. Is there any specific amenity you are looking for?

+

Context

Query

Response

Dialog generation

- Problem is complicated
- Data collection is costly
- Domain adaptation
- Meta-learning

Qian et al. “Domain Adaptive Dialog Generation via Meta Learning ” ACL 2019

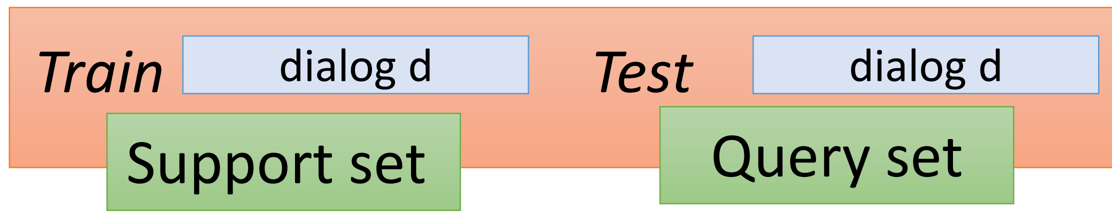
Dialog generation

- MAML – model: M
- Across task training

Task 1…N

for domain D in source domains

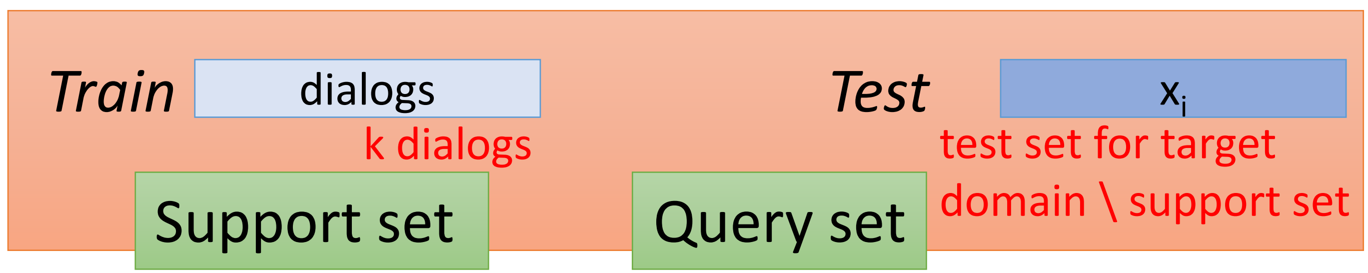
sample 1 dialog *d* from D



- Across task testing

Task 1…N

sample k dialogs from target domains



$$h = \text{Encoder}(B_{t-1}, R_{t-1}, U_t)$$
$$B_t = \text{BspanDecoder}(h)$$
$$R_t = \text{ResponseDecoder}(h, B_t, m_t)$$

Dialog generation

- SimDial
  - Source: *restaurant*, *weather* and *bus*
  - Target: *movie*
  - (within-task) train, valid, test: 900, 100, 500 dialogs
  - Across-task training: train of sources
  - Across-task testing:
    - 9 (1%) dialogs from train of target -> support set
    - Test of target -> query set

Method	Entity F1
Transfer learning	64.0
ZSDG	52.6
Meta-learning	66.2

Personalized Dialog generation

Persona 2
I am an artist I have four children I recently got a cat I enjoy walking for exercise I love watching Game of Thrones

[PERSON 1:] Hi
[PERSON 2:] Hello ! How are you today ?
[PERSON 1:] I am good thank you , how are you.
[PERSON 2:] Great, thanks ! My children and I were just about to watch Game of Thrones.
[PERSON 1:] Nice ! How old are your children?
[PERSON 2:] I have four that range in age from 10 to 21. You?
[PERSON 1:] I do not have children at the moment.
[PERSON 2:] That just means you get to keep all the popcorn for yourself.
[PERSON 1:] And Cheetos at the moment!
[PERSON 2:] Good choice. Do you watch Game of Thrones?
[PERSON 1:] No, I do not have much time for TV.
[PERSON 2:] I usually spend my time painting: but, I love the show.

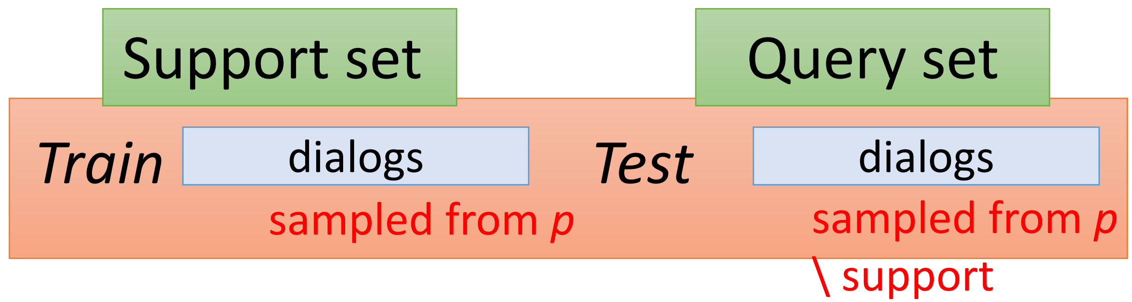
Given dialog context, persona, responses from counterpart (1), generate response (2) consistent with persona

Personalized Dialog generation

- MAML – model: M
- Across task training

Task 1…N

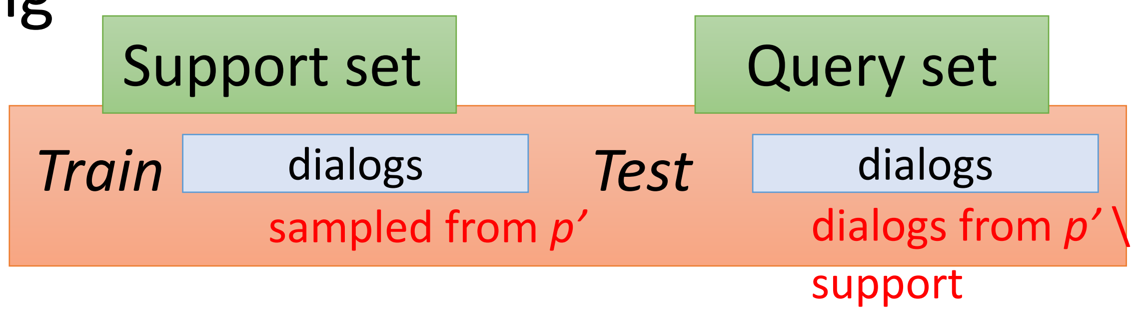
for each persona *p* (or speaker) in training set



- Across task testing

Task 1…N

for persona *p'* in test set



Madotto et al. "Personalizing Dialogue Agents via Meta-Learning " ACL 2019

Personalized Dialog generation

- Persona-chat dataset

	Coherence	Fluency	Consistency
Human	0.33	3.43	0.23
Supervised	-0.03	-	-
Supervised+Persona	0.07	3.05	0.01
Fine-tuning+Persona	0.00	3.10	0.04
Meta-learning+Persona	**0.20**	**3.19**	**0.20**

- Meta-learning > fine-tuning > no fine-tuning
- Using Persona is better

Conclusion

- Meta-learning in ConvAI
  - transferring knowledge across user intents, domains, and languages
  - Scalable solution for expansion

Meta Learning for NLP

Meta Learning in NLP

- Text classification
- Relation classification
  – a special type of text classification
- Sequence to sequence related tasks such as machine translation and semantic parsing
- Knowledge graphs related tasks

Text Classification

- Problem settings:

Input: a piece of text, e.g. sentences → NLP System → Output: a label

- Some examples in a sentiment analysis task:
  - This movie is great → positive
  - There are many meaningless points in the documentary film → negative
- Modern NLP methods:
  - Leverage word embeddings (word → vectors)
  - Input text → matrices

Diverse Few-Shot Text Classification with Multiple Metrics

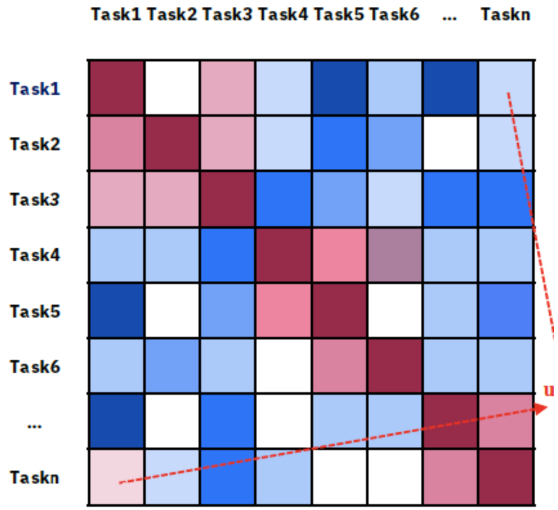
- Argued that in previous work, low variants among tasks → not realistic
  In a more realistic setting, tasks are diverse
- Key ideas and take-home messages:
  - Based on metrics based methods
  - Two steps: 1) tasks clustering; 2) metrics-based
  - Extend meta learning that allows combining multiple metrics depending on different task clusters

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, Bowen Zhou, Diverse Few-Shot Text Classification with Multiple Metrics, ACL 2018

Diverse Few-Shot Text Classification with Multiple Metrics

Image from the original paper

- How to cluster tasks:
  - Create a transfer performance matrix
  - Apply scores filtering and matrix completion
  - Apply spectral clustering



Cross-task transfer performance matrix

- How to combine decisions:
  - Linearly combine decisions from different task clusters
  - Linear coefficients are adaptable parameters

$$p(y|x) = \sum_k \alpha_k P(y|x; f_k).$$

Investigating Meta-Learning for Low-Resource NLU Tasks

- Applied MAML and Reptile on a bunch of tasks in GLUE dataset (four high resource tasks for training and four low resource tasks for testing)
- Key take-home messages:
  - Reptile worked best, outperformed other methods
  - Meta learning frameworks outperformed state-of-the-art systems trained with a single task and multi-task learning systems
  - It is more effective with less training data

Zi-Yi Dou, Keyi Yu, Antonios Anastasopoulos, Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks, EMNLP 2019

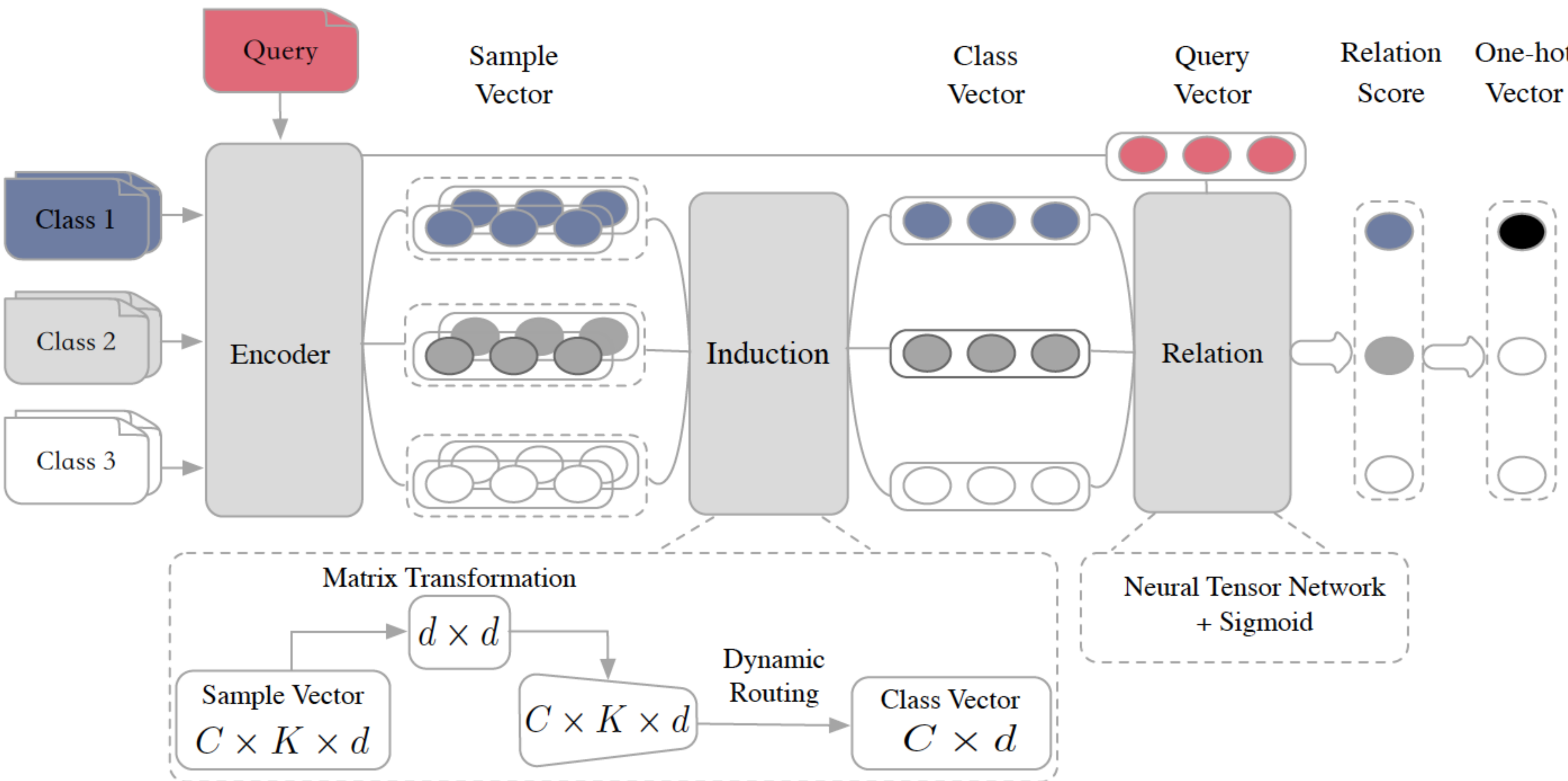
Induction Networks for Few-Shot Text Classification

- Key ideas and take-home messages
  - Leverage dynamic routing algorithms (proposed in capsule network – Sabour et al 2017) to improve the generalization of the class representation
  - Leverage the Neural Tensor Network (Socher et al 2013) to compute the relation scores between queries and class vectors
  - Both steps are important and their combination works best

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, Jian Sun, Induction Networks for Few-Shot Text Classification, EMNLP, 2019

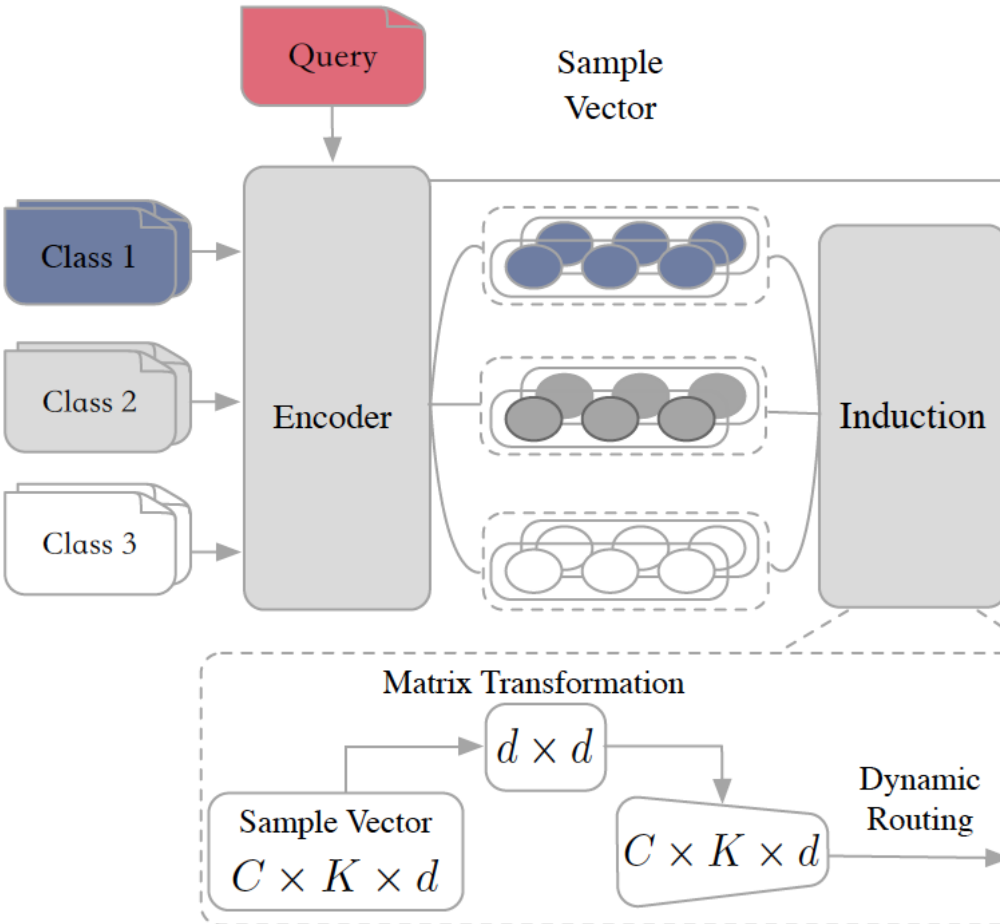
Induction Networks for Few-Shot Text Classification

Image from the original paper



Induction Networks for Few-Shot Text Classification

Image from the original paper



Algorithm 2 Dynamic Routing Induction

Require: sample vector e_{ij}^s in support set S and initialize the logits of coupling coefficients $b_{ij} = 0$

Ensure: class vector c_i

1: for all samples $j = 1, ..., K$ in class i:
2: $\quad \hat{e}_{ij}^s = squash(W_s e_{ij}^s + b_s)$
3: **for** $iter$ iterations **do**
4: $\quad d_i = softmax(b_i)$
5: $\quad \hat{c}_i = \sum_j d_{ij} \cdot \hat{e}_{ij}^s$
6: $\quad c_i = squash(\hat{c}_i)$
7: $\quad$ for all samples $j = 1, ..., K$ in class i:
8: $\quad\quad b_{ij} = b_{ij} + \hat{e}_{ij}^s \cdot c_i$
9: **end for**
10: **Return** c_i

Induction Networks for Few-Shot Text Classification

Image from the original paper

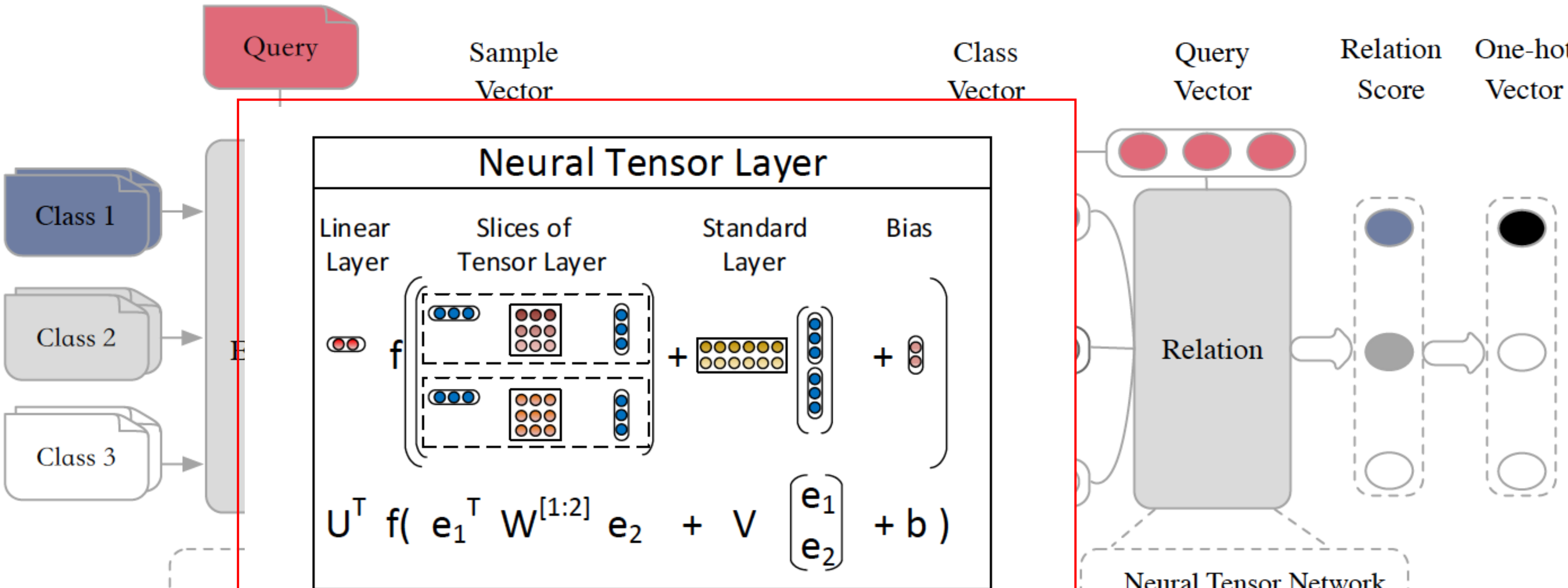


Figure 2: Visualization of the Neural Tensor Network. Each dashed box represents one slice of the tensor, in this case there are $k = 2$ slices.

Socher et al 2013

Hierarchical Attention Prototypical Networks for Few-Shot Text Classification

- Key ideas and take-home messages
  - Based on the prototypical network
  - Hierarchical attention architecture
    - Word level – attention over words to obtain the sentence representation
    - Instance level - attention over instances in the support set to form the prototypes
    - Feature level – as proposed in Gao et al AAAI 2019 – to improve the distance function

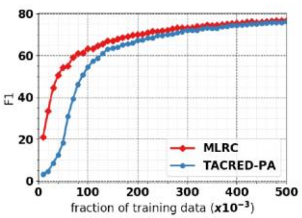
Shengli Sun, Qingfeng Sun, Kevin Zhou, Tengchao Lv, Hierarchical Attention Prototypical Networks for Few-Shot Text Classification, EMNLP 2019

Relation Classification

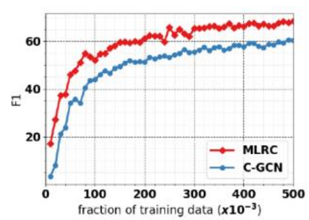
- As a special case of text classification
- Given a sentence with two marked entities, an NLP system should predict the semantic relation between these two entities
- Examples:
  - The Queen Consort [Jetsun Pema] gave birth to a son on 5 February 2016 , [Jigme Namgyel Wangchuck].
  - Relation: *mother*

Model-Agnostic Meta-Learning for Relation Classification with Limited Supervision

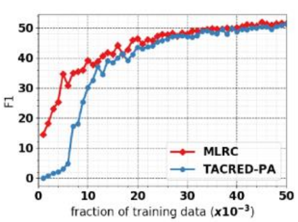
- Key ideas and take-home messages
  - Applied MAML for relation classification tasks
  - Utilized two strong models and applied MAML objectives
  - MAML worked very well on their setups



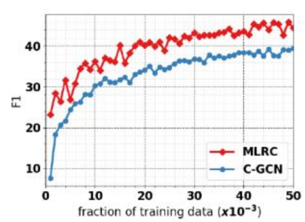
(a)



(a)



(b)



(b)

Abiola Obamuyide, Andreas Vlachos, Model-Agnostic Meta-Learning for Relation Classification with Limited Supervision, ACL 2019

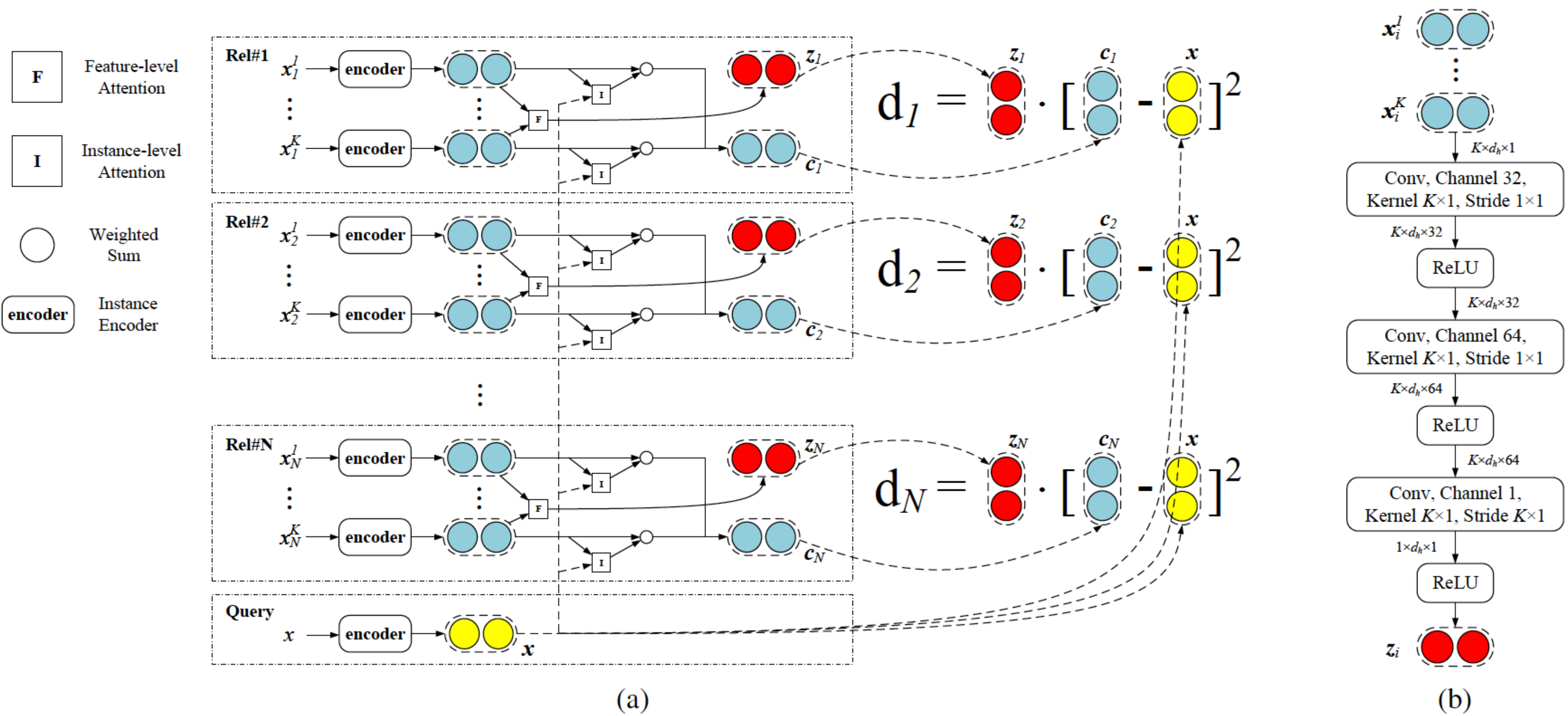
Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification

- Key ideas and take-home messages
  - Special design for corrupted text inputs
  - Based on prototypical network
  - Novel method to compute the matching scores based on attention mechanism
  - Hybrid attention:
    - Instance level attention: improves robustness against noisy instances
    - Feature level attention: improves the distance function

Tianyu Gao, Xu Han, Zhiyuan Liu, Maosong Sun, Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification, AAAI 2019

Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification

Image from the original paper



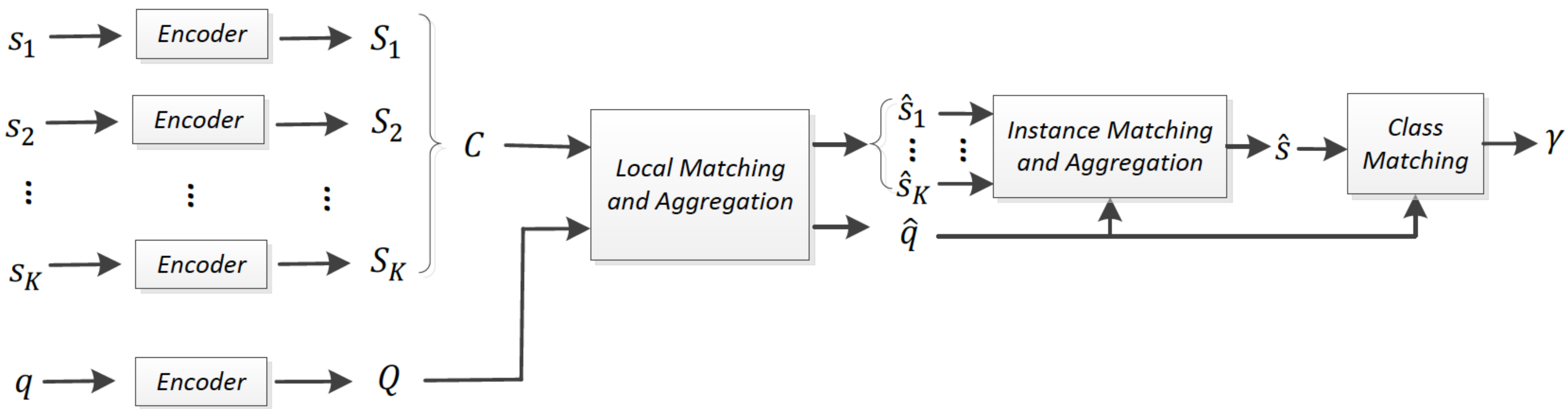
Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification

- Key ideas and take-home messages
  - Based on matching networks
  - Extend them to multi-level matching and aggregation
    - Local matching
    - Instances matching
    - Class matching

Zhi-Xiu Ye, Zhen-Hua Ling, Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification, ACL 2019

Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification

Image from the original paper



1) **Encoder:** use a CNN that convert a sentence and the positions of two entities to matrices
2) **Local matching**: use attention method to collect matching information between support instances and the query instance, then use max-pooling and average pooling to convert them to representation vectors for all the support instances and the query instance
3) **Instance matching**: use attention method to compute the prototype
4) **Class matching**: trainable matching scores between the query instance and prototypes

Seq2seq Tasks

- Problem settings:
  - Input: sequence of symbols, e.g. words in one language
  - Output: sequence of symbols, e.g. words in another language
- Examples:
  - Machine translation
    - How are you? → Wie geht es Dir?
  - Semantic parsing
    - Input text → SQL command

Meta-Learning for Low-Resource Neural Machine Translation

- Key ideas and take-home messages:
  - Apply MAML for multilingual neural translation
  - Propose to use the universal lexical representation to handle the problem of mismatch input and output
  - Utilize 18 translation pairs as training tasks to train the meta learner
  - Fine-tune the meta learnt models for five other translation pairs

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, Victor O.K. Li, Meta-Learning for Low-Resource Neural Machine Translation, EMNLP, 2018

Natural Language to Structured Query Generation via Meta-Learning

- Key ideas and take-home messages
  - Map a natural language question to a SQL query
  - Artificially generate **pseudo tasks** by sampling a batch of training data as a support set and one example as query
    - Design a *relevance function* to find similar examples
    - Relevance function is task dependent
    - E.g. in this paper, the relevance function depends on 1) the predicted SQL type of the input and 2) the input length
  - Apply MAML to train the meta learner

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, Xiaodong He, Natural Language to Structured Query Generation via Meta-Learning, NAACL 2018

Coupling Retrieval and Meta-Learning for Context-Dependent Semantic Parsing

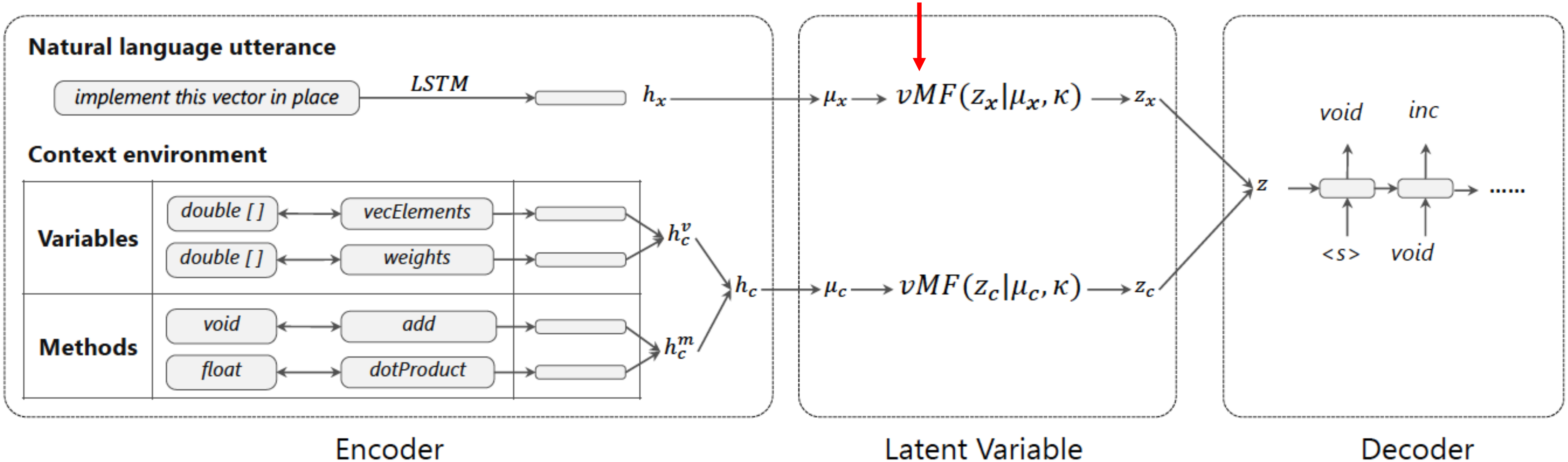
- Key ideas and take-home messages
  - Given a natural language, generate a source code conditioned on the class environment
  - Similar setup as previous paper
  - Introduce a *context aware retriever* to dynamically collect examples from the training as supporting evidences
  - Apply MAML to train the meta learner

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, Jian Yin, Coupling Retrieval and Meta-Learning for Context-Dependent Semantic Parsing, ACL, 2019

Coupling Retrieval and Meta-Learning for Context-Dependent Semantic Parsing

Von Mises-Fischer distribution

Image from the original paper

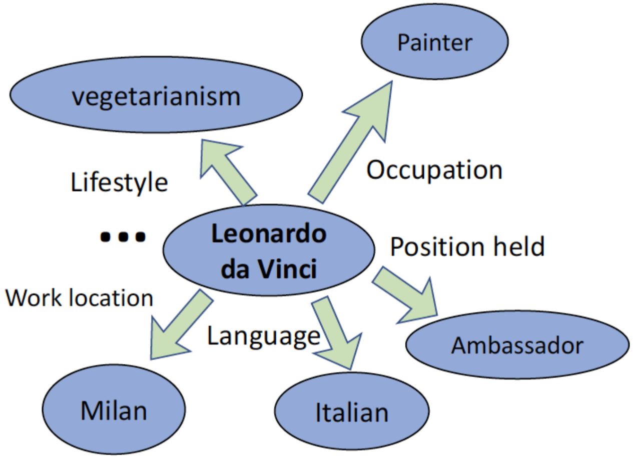


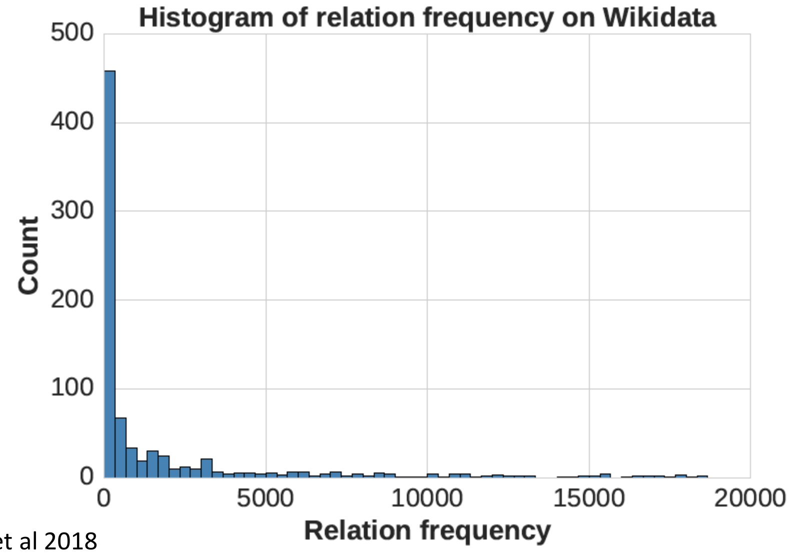
The retriever finds top-K nearest examples based on the following distance:

$$
\begin{aligned}
distance &= KL(p(z|x,c)||p(z|x',c')) \\
&= KL(p(z_x|x)||p(z_x|x')) \\
&\quad + KL(p(z_c|c)||p(z_c|c'))
\end{aligned}
$$

Knowledge Graphs

- Problem settings:
  - Knowledge graph, i.e. a collection of triples (h, r, t)
    h: head entity; r: relation; t: tail entity
  - Tasks could be either (h, ?r?, t) or (h, r, ?t?)
- Examples:





Figures from Xiong et al 2018

Knowledge Graphs

Training	
Task #1 (CountryCapital)	
Support	(China, CountryCapital, Beijing)
Query	(France, CountryCapital, Paris)
Task #2 (CEOof)	
Support	(Satya Nadella, CEOof, Microsoft)
Query	(Jack Dorsey, CEOof, Twitter)
Testing	
Task #1 (OfficialLanguage)	
Support	(Japan, OfficialLanguage, Japanese)
Query	(Spain, OfficialLanguage, Spanish)

The goal of the link prediction task is to look for the tail entity

Examples from Chen et al EMNLP 2019

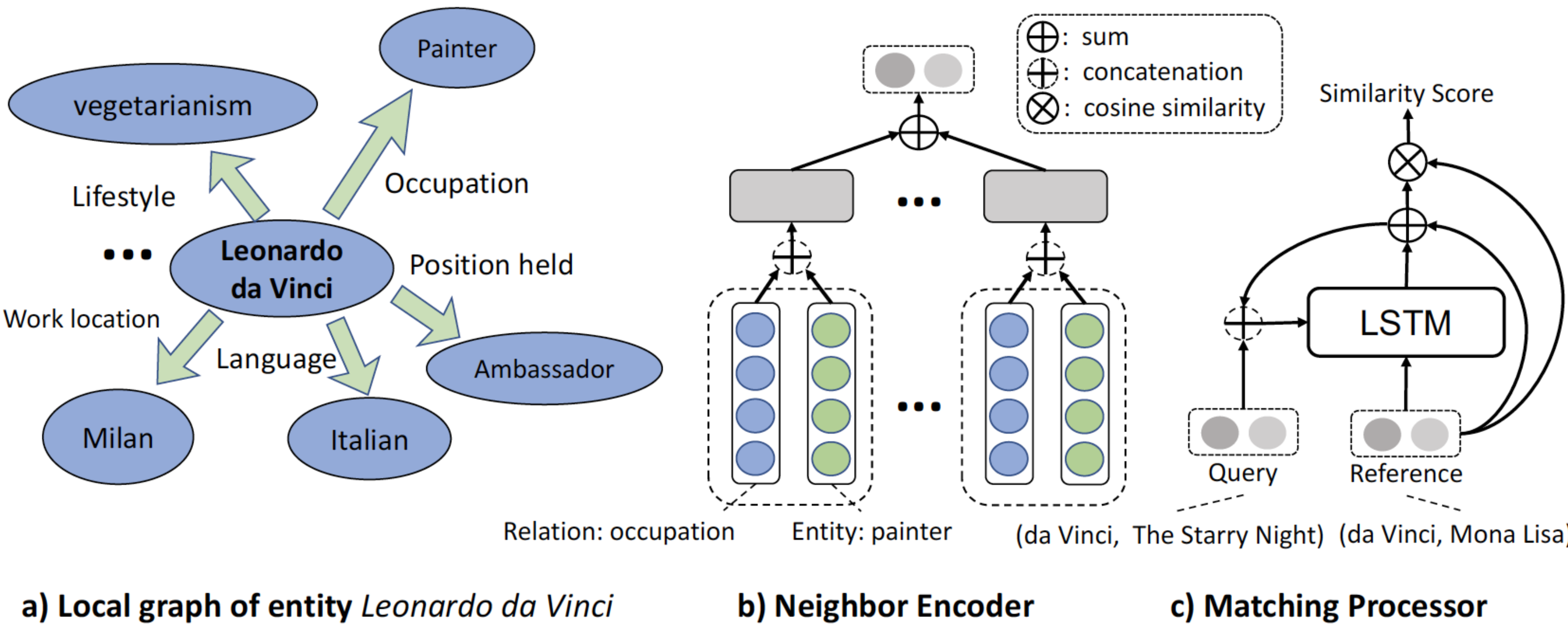
One-Shot Relational Learning for Knowledge Graphs

- (h, r, ?t?) - a ranking problem, i.e. search for the right *t* in a candidate pool C
- Key ideas and take-home messages:
  - Embedding function:
    - Entity embeddings and neighbor encoders
  - Matching scores:
    - Matching processor to compute similarity scores
  - Could be seen as applying matching network on tail entity ranking task

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, William Yang Wang, One-Shot Relational Learning for Knowledge Graphs, EMNLP 2018

One-Shot Relational Learning for Knowledge Graphs

Image from the original paper



Tackling Long-Tailed Relations and Uncommon Entities in Knowledge Graph Completion

- (h, r, ?t) with the same setup as Xiong et al.
- Key ideas and take-home messages:
  - Take into account uncommon entities, i.e. entities that appear only several times or absent
  - Apply matching network on tail entity ranking task
  - The novelty lies on the integration of text descriptions of entities and relations to compute the triple representation

Zihao Wang, Kwun Ping Lai, Piji Li, Lidong Bing, Wai Lam, Tackling Long-Tailed Relations and Uncommon Entities in Knowledge Graph Completion, EMNLP 2019

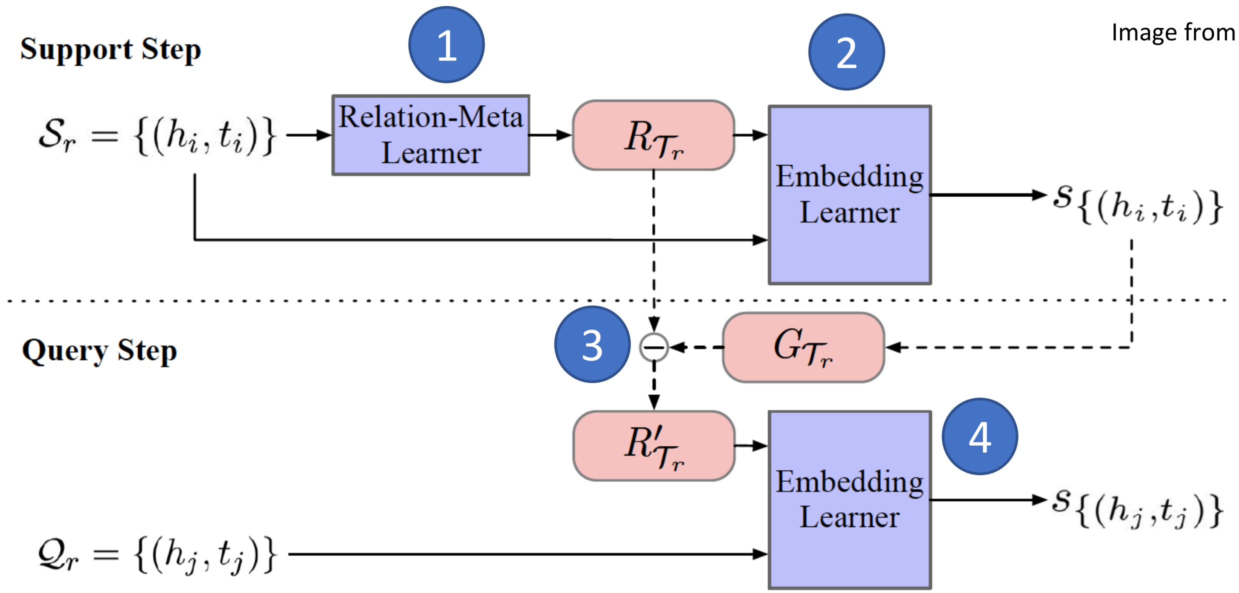
Relational Learning for Few-Shot Link Prediction in Knowledge Graphs

- (h, r, ?t) with the same setup as Xiong et al
- Key ideas and take-home messages:
  - Relation-Meta Learner: learn the relation embeddings between head and tail entities
  - Embedding Learner: evaluate the true value of entity pairs under specific relations and evaluate the within-task training loss
  - Outperformed Xiong et al paper

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, Huajun Chen, Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs, EMNLP 2019

Relational Learning for Few-Shot Link Prediction in Knowledge Graphs

Image from the original paper



1) **Relation-Meta Learner:** based on a multilayer perceptron that takes head and tail entity as inputs and outputs a vector presenting the relation from head and tail entity
2) **Embedding Learner:** outputs the L2 norm of the vector: *head + meta relation – tail*
3) **In the within-task training step**: update the meta relation vector with the meta gradient
4) **In the querying step**: use the embedding learner to output the score as mention in 2) but with an updated meta relation vector

Summary

- Meta learning framework
- Two concrete examples:
    - Learning to initialize
    - Learning to compare
- Applications of Meta learning:
    - Speech
    - Conversational AI
    - Natural language processing

Thanks for your attention!