Please find the latest version below.

https://speech.ee.ntu.edu.tw/~hylee/Meta_Tutorial.pdf

Pre-recorded video for ACL 2021

https://drive.google.com/drive/folders/1D663btPPMyWfAuOZCmV76_kC7ZjOuOmY?usp=sharing

Meta Learning and Its Applications to Natural Language Processing

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

Part I: Basic Idea of Meta Learning

break

Part II: Applications to Human Language Processing

break

Part III: Advanced Topics

Meta learning = Learn to learn

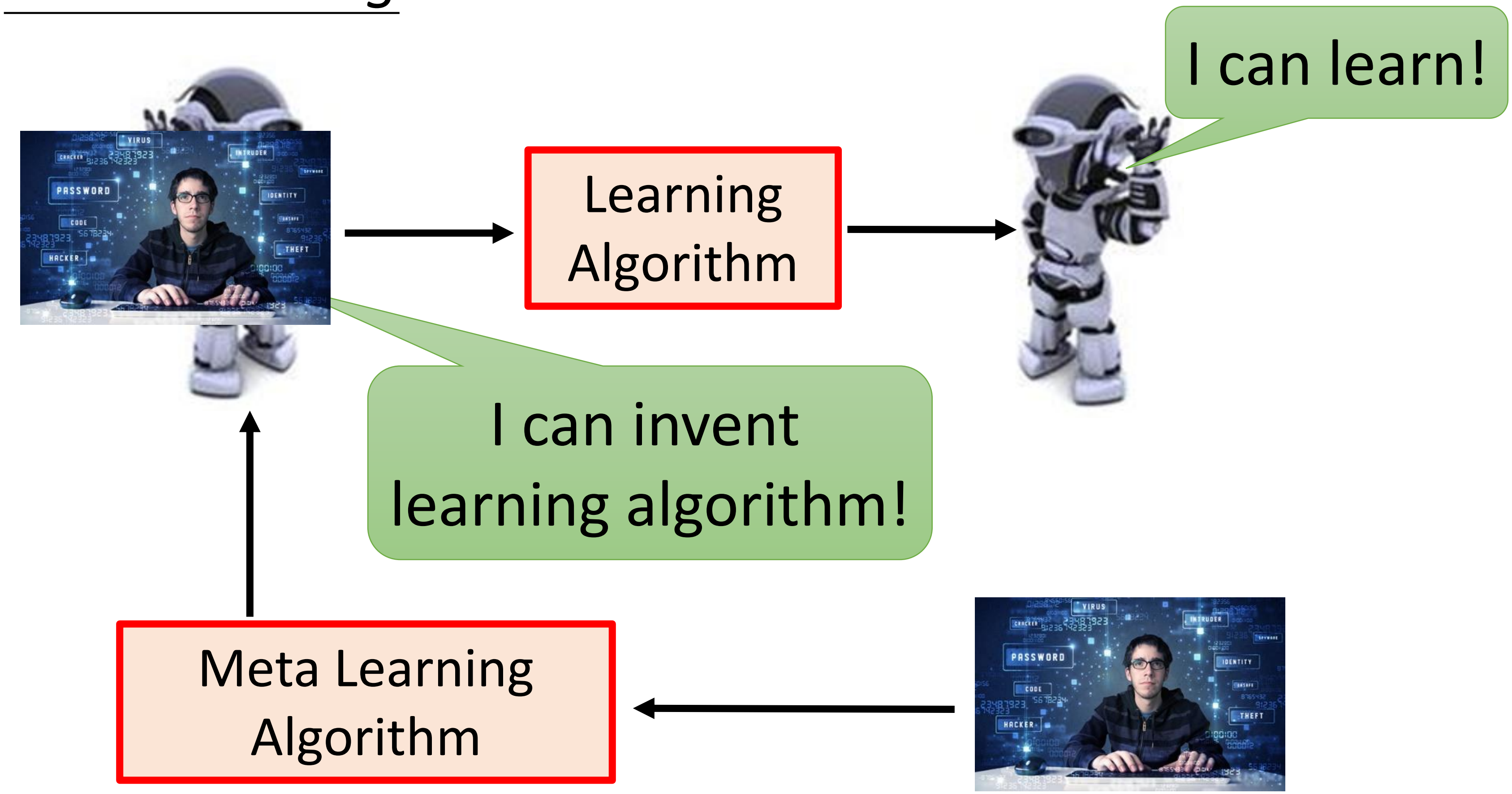
Typical Machine Learning

Learning Algorithm

I can learn!

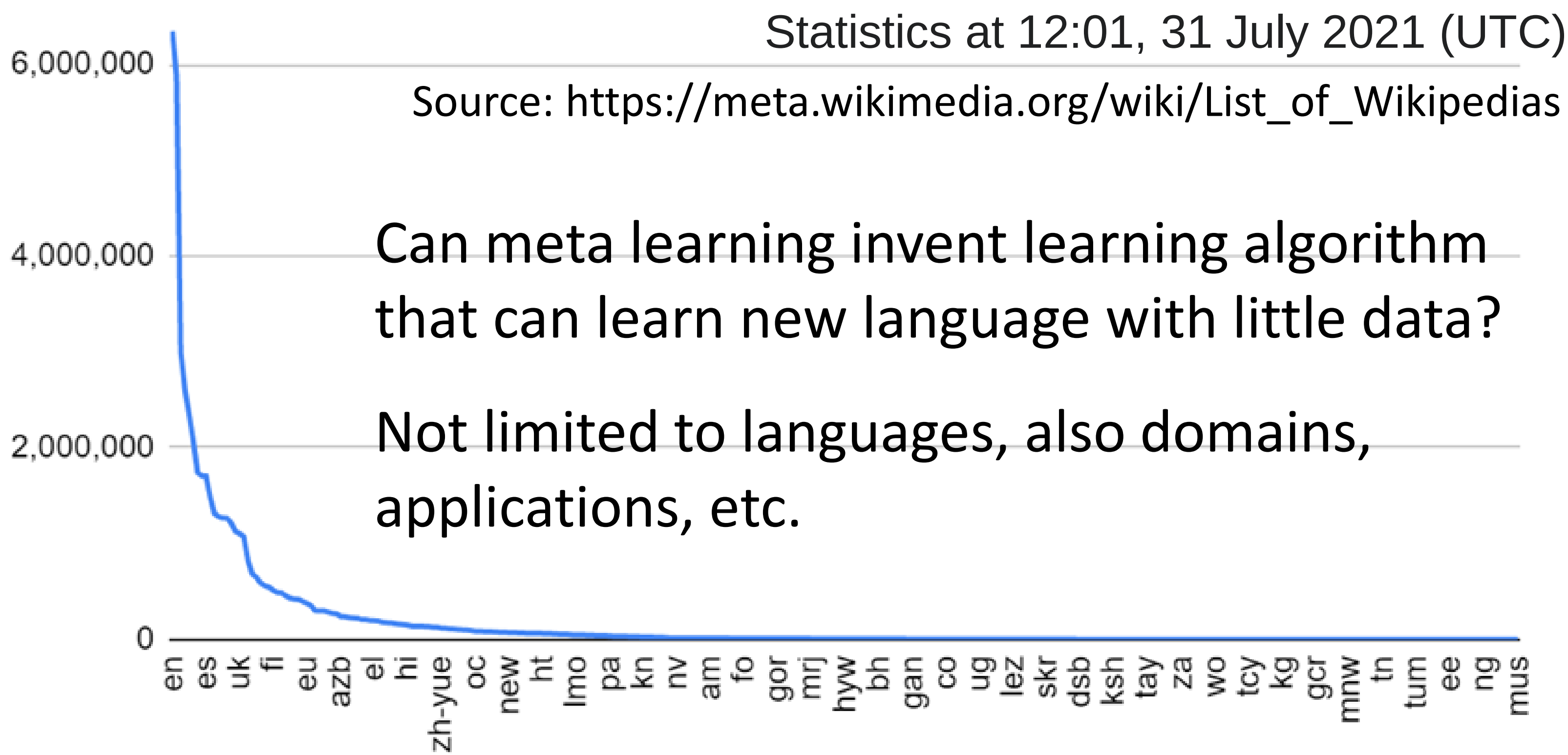
Meta learning = Learn to learn

Meta Learning



Why Meta Learning?

- Because human designed learning algorithms are not always efficient.
- E.g., typical deep learning needs a large amount of data, but we usually lack data in human language processing.



Source: https://meta.wikimedia.org/wiki/List_of_Wikipedias

Can meta learning invent learning algorithm that can learn new language with little data?

Not limited to languages, also domains, applications, etc.

321 languages

Why this tutorial?

	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

The table is online.

https://jeffeuxmartin.github.io/meta-learning-hlp/

Part I: Basic Idea of Meta Learning

- Starting from Machine learning
- Introduction of Meta Learning
- Learning to Initialize
- More Meta Learning Approaches
- Learning to Compare
- Meta learning vs. Other Methods

Part II: Applications to Human Language Processing

Part III: Advanced Topics

Part I:
Basic Idea of Meta Learning

Machine Learning 101

Machine Learning

= Looking for a function

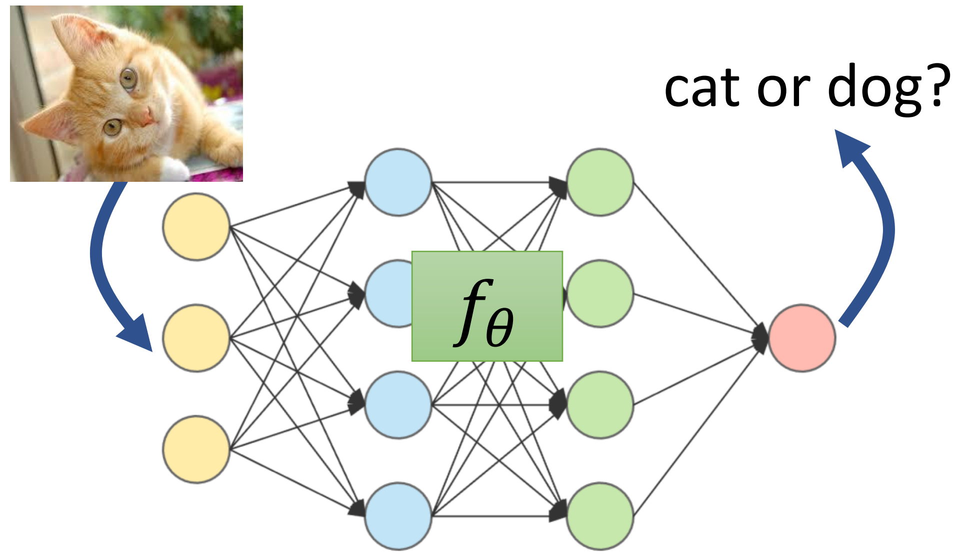
Step 1: What is learnable

Step 2: Define loss function

Step 3: Optimization

Dog-Cat Classification

$f($  $) =$ "cat"



Weights and biases of neurons are learnable.

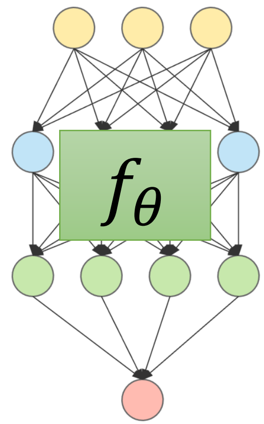
Using θ to represent the learnable parameters.

Machine Learning

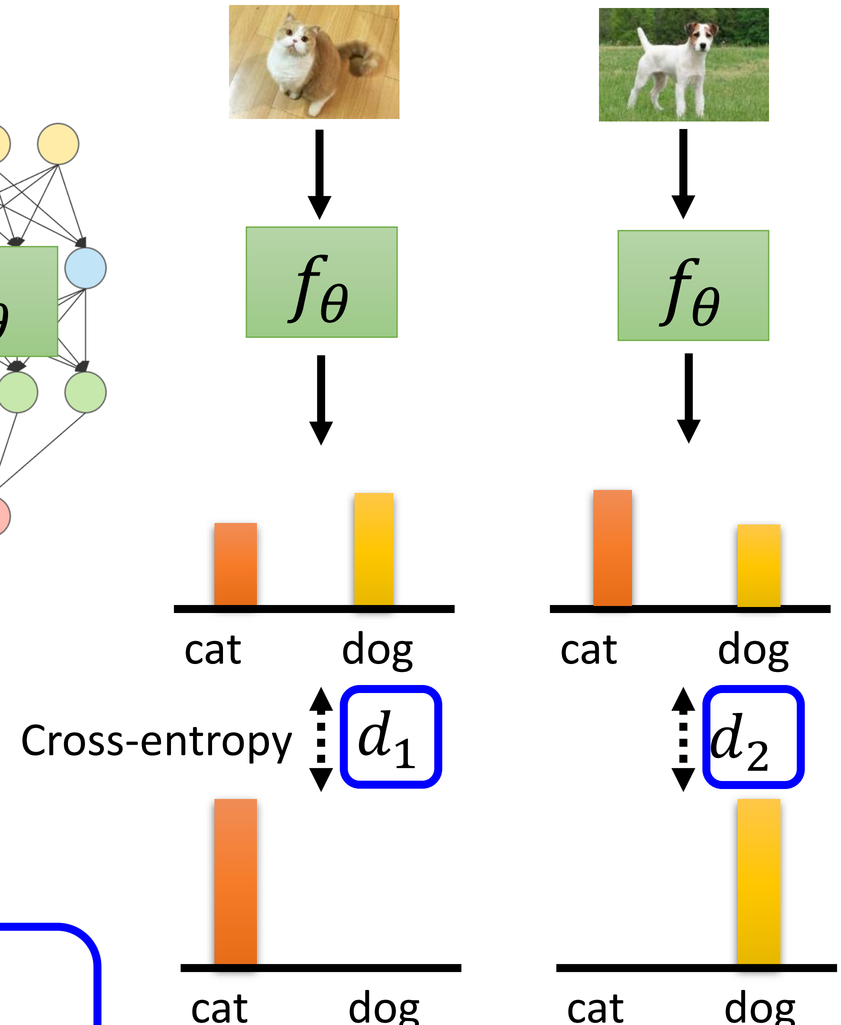
Step 1: What is learnable

Step 2: Define loss function $l(\theta)$

Step 3: Optimization



Training Examples



Ground Truth

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

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

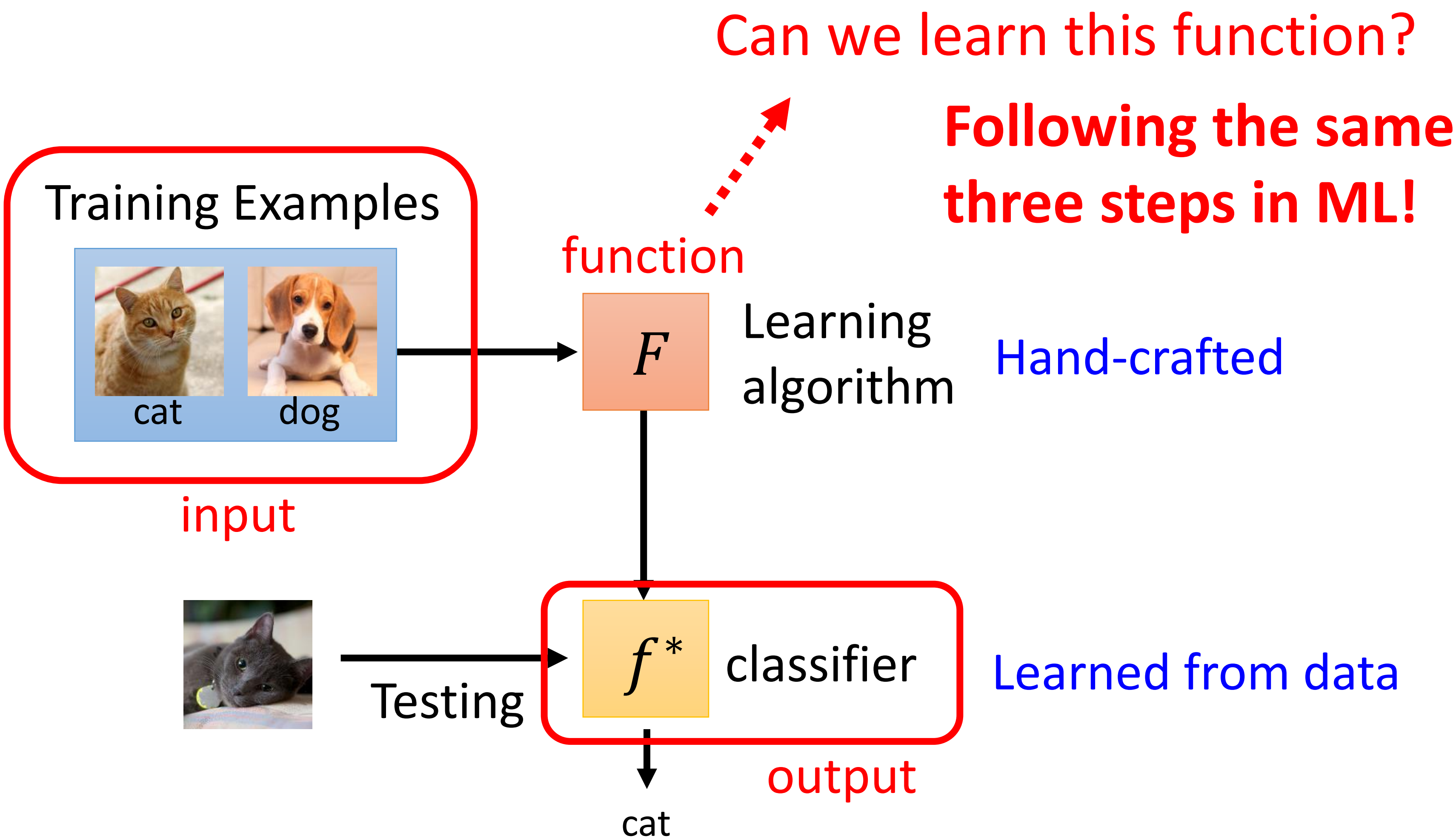
$\hat{\theta} = arg \min_{\theta} l(\theta)$

done by gradient descent

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

Introduction of Meta Learning

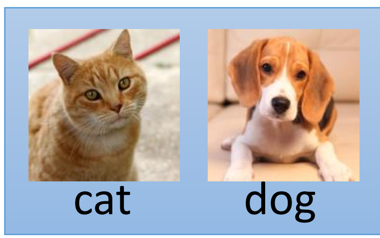
What is Meta Learning?



Meta Learning – Step 1

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

Training Examples



F

Deep Learning

Component

Net Architecture,
Initial Parameters,
Optimizer,
……

Testing

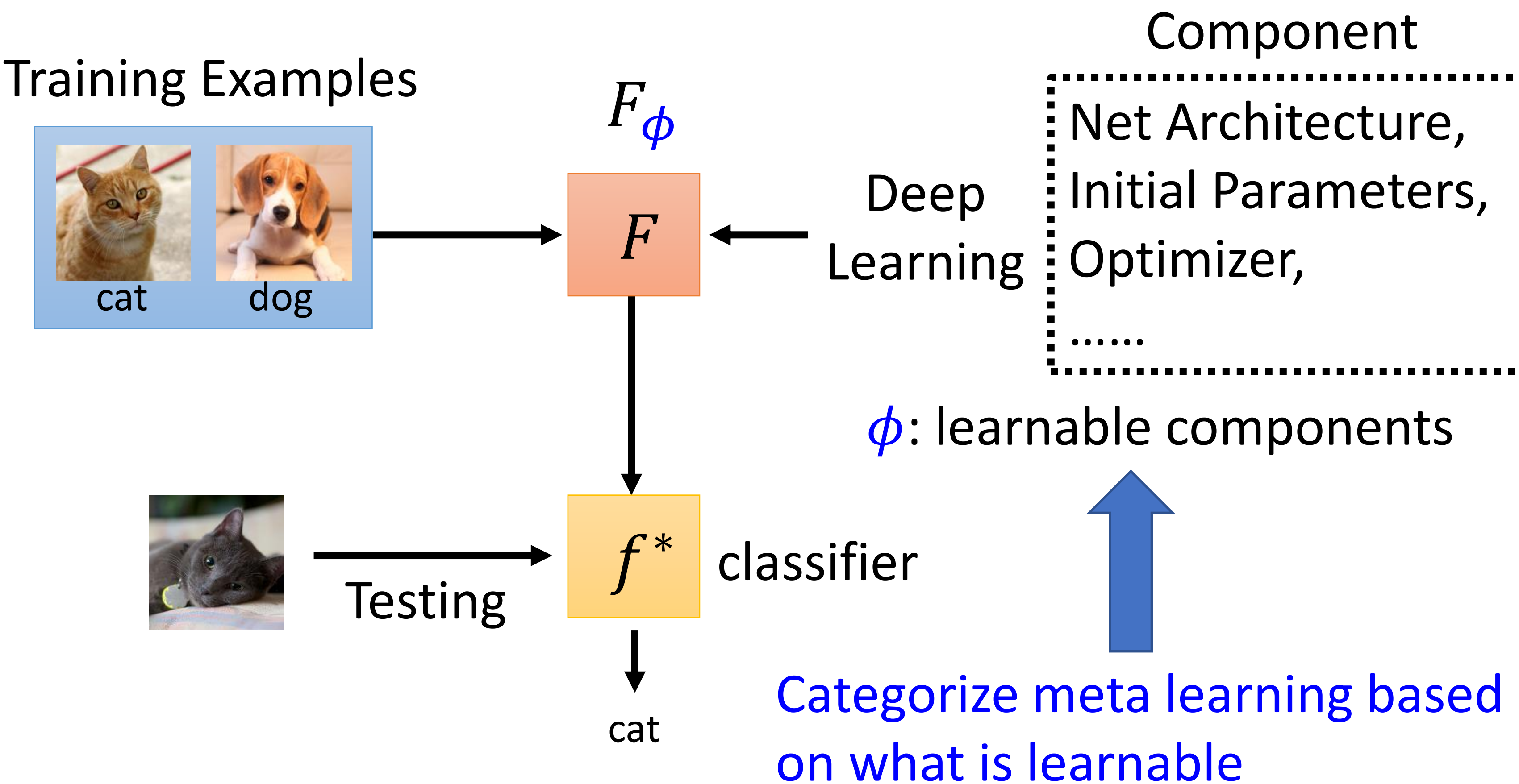
f^* classifier

cat

In meta, we will try to learn some of them.

Meta Learning – Step 1

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

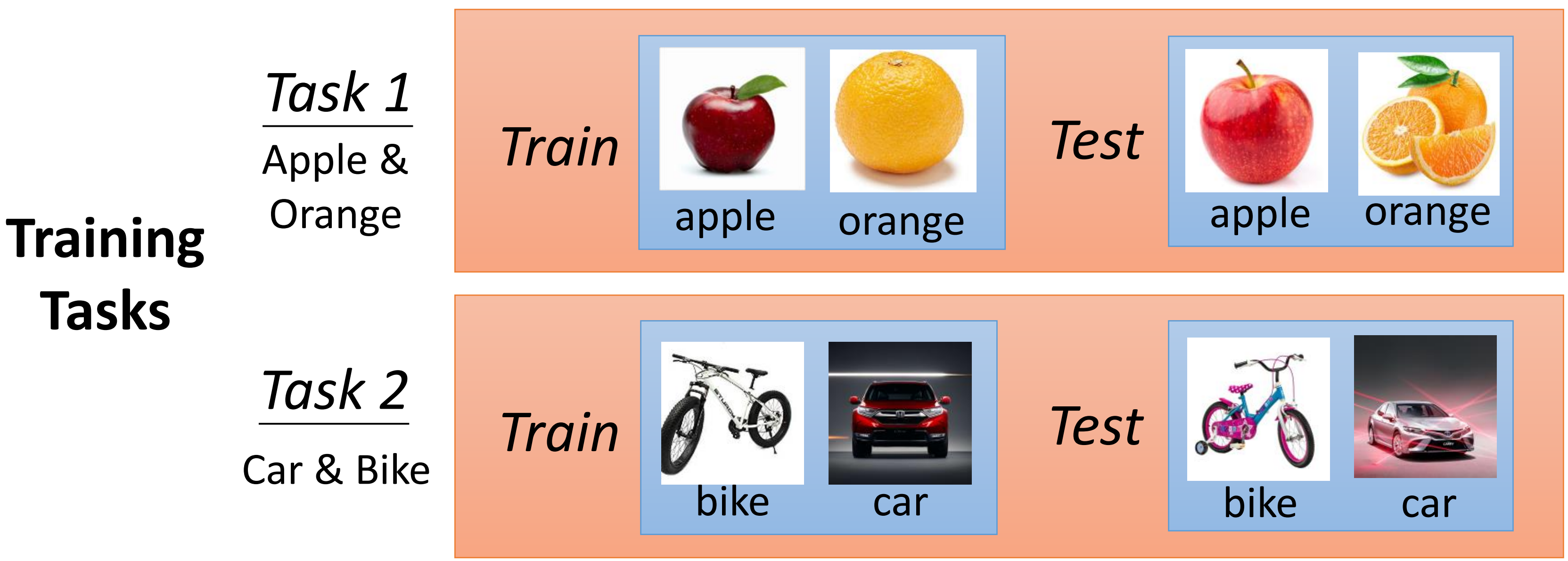


Meta Learning – Step 2

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

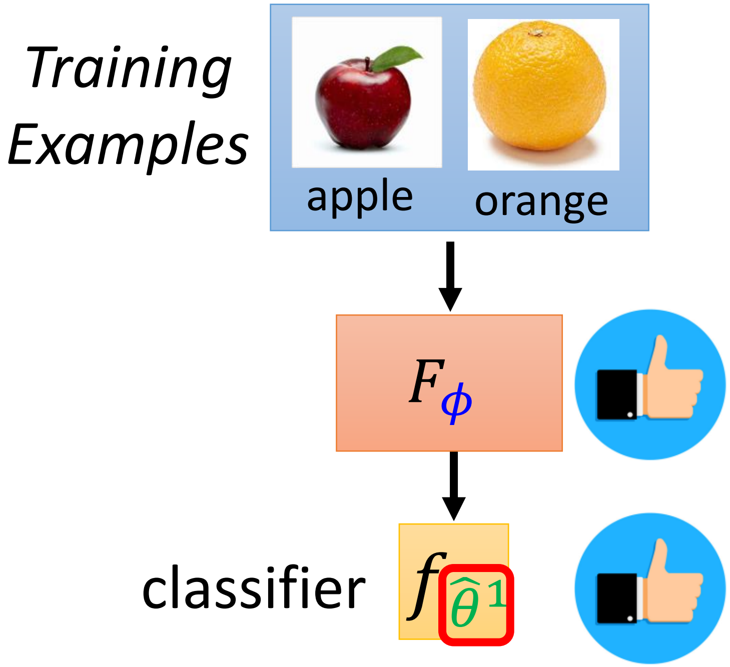
$$L(\phi)$$

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



Meta Learning – Step 2

Task 1



How to define $L(\phi)$

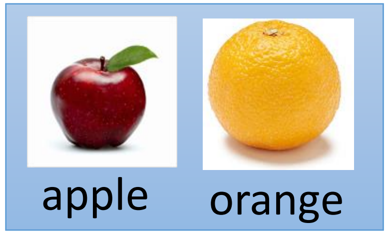
$L(\phi)$ ↓

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

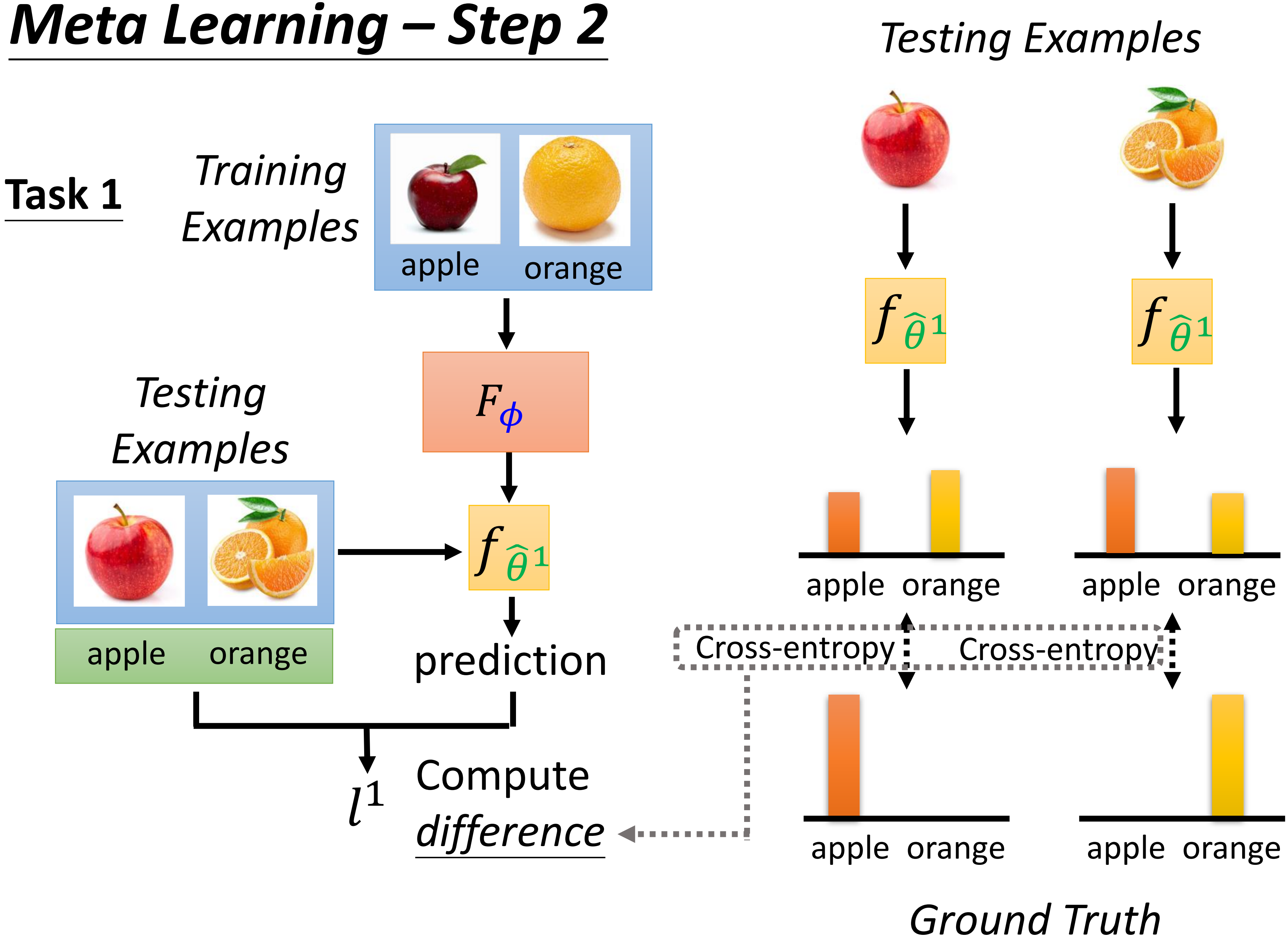
How to define $L(\phi)$

$L(\phi)$ ↑

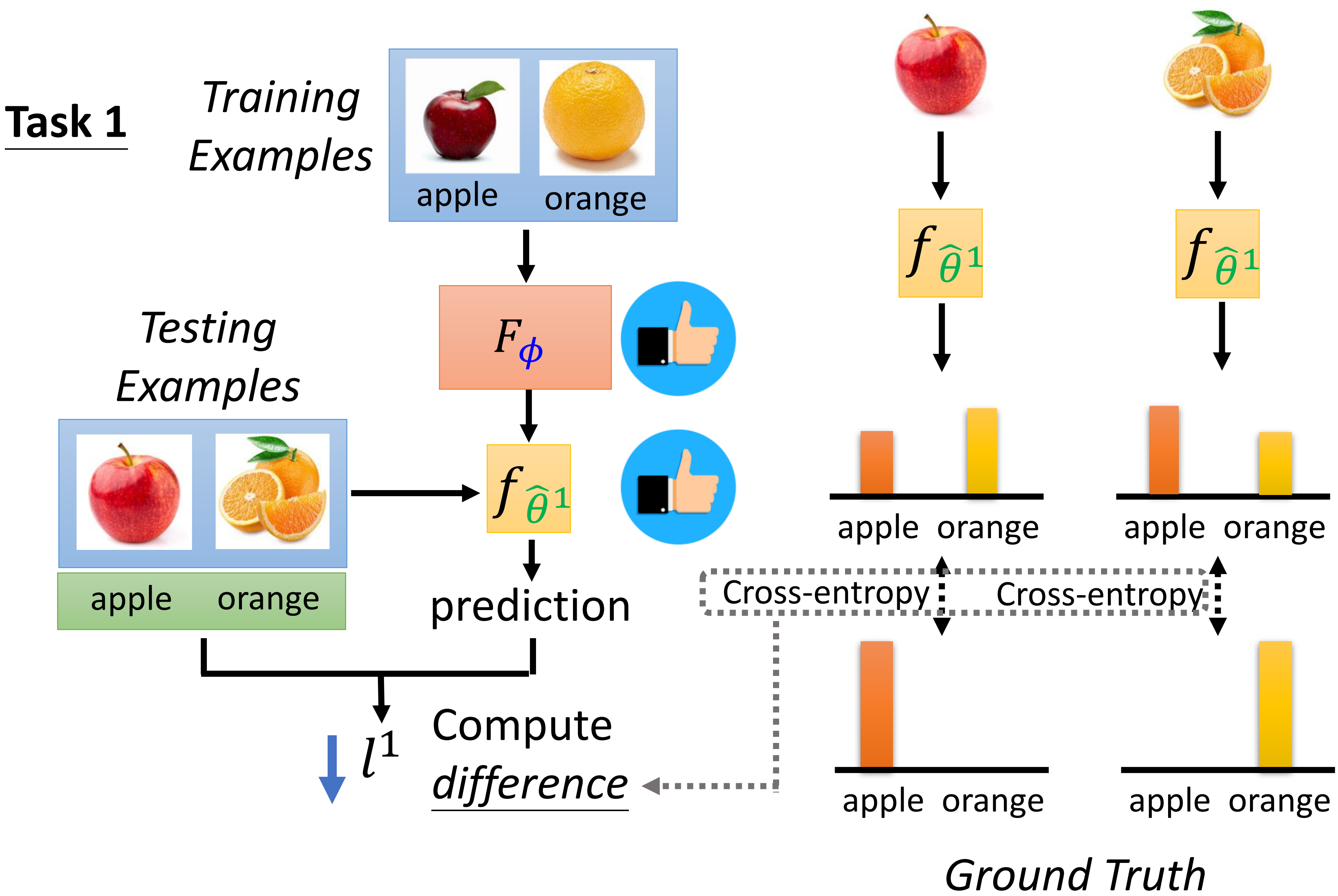
How can we know a classifier is good or bad?

Evaluate the classifier on testing set

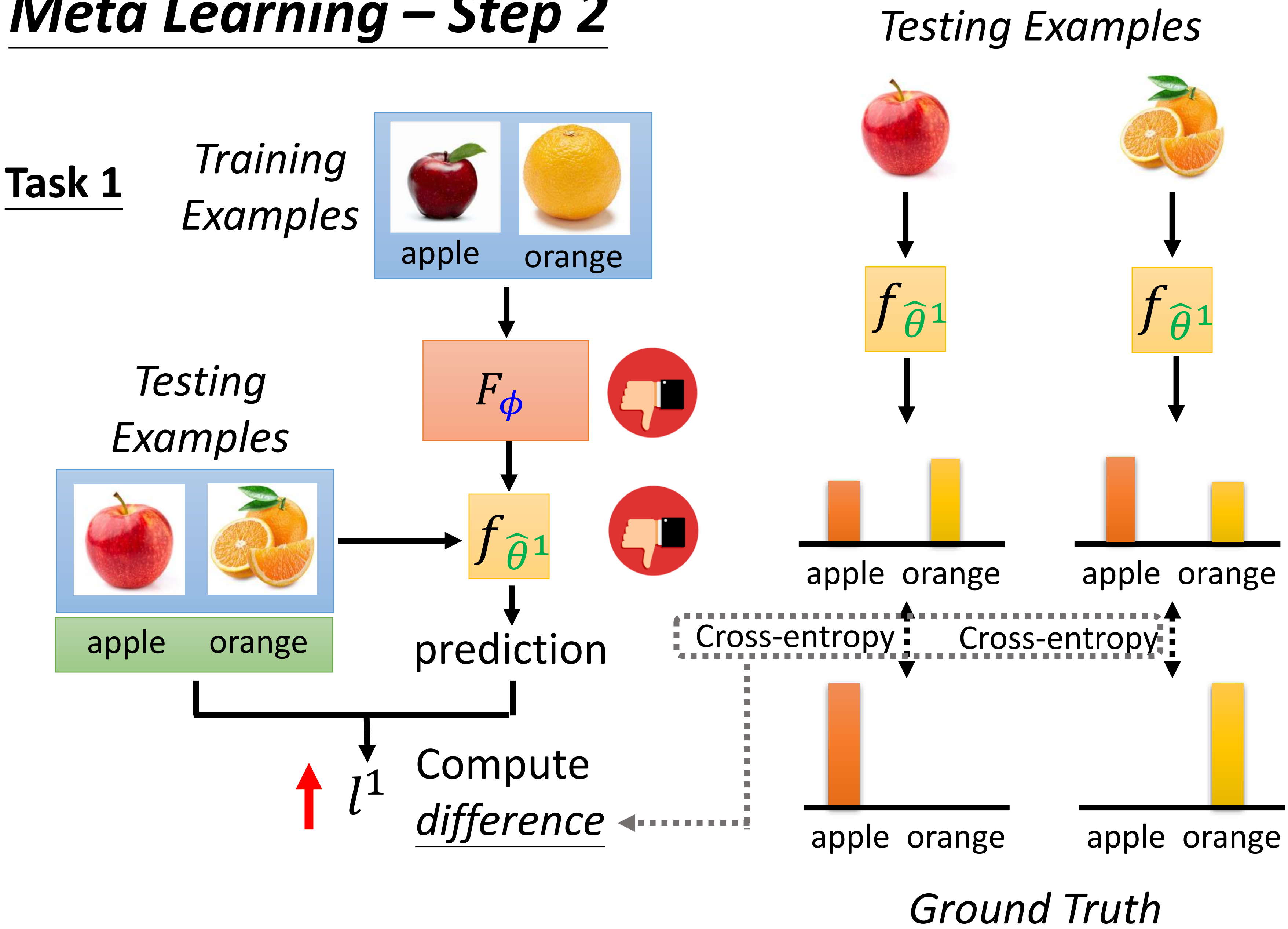
Meta Learning – Step 2



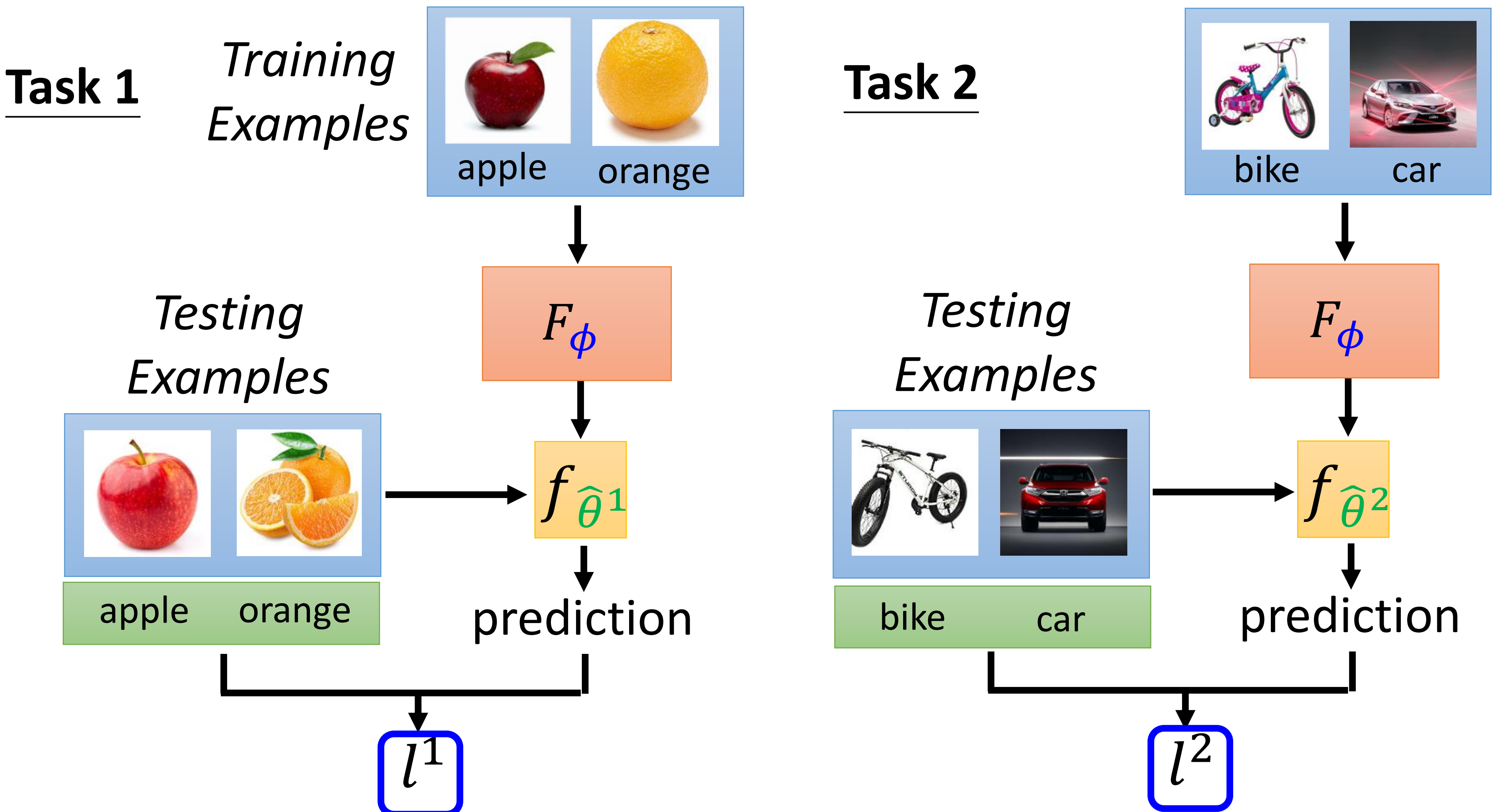
Meta Learning – Step 2



Meta Learning – Step 2

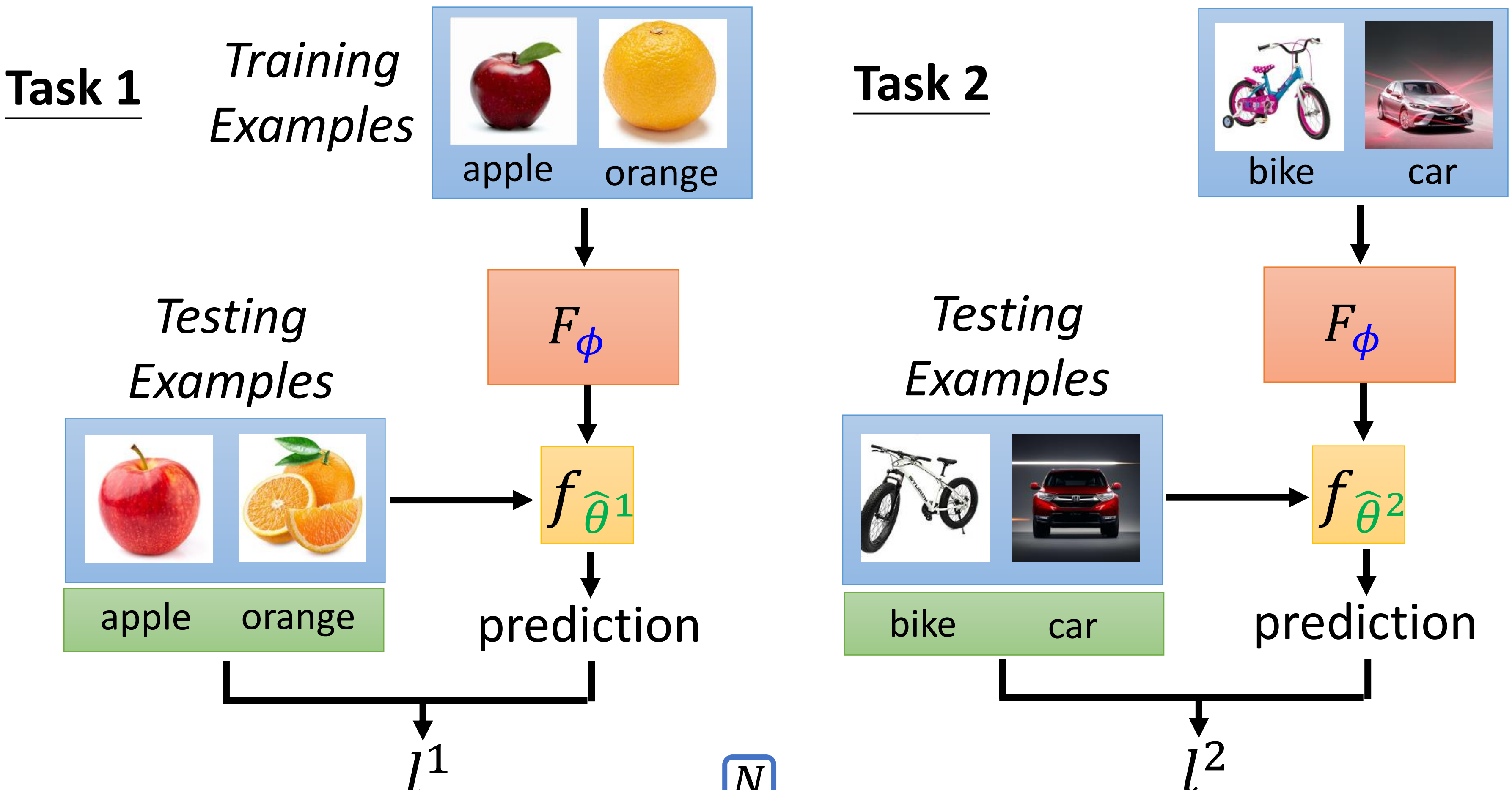


Meta Learning – Step 2



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 $\partial L(\phi)/\partial \phi$

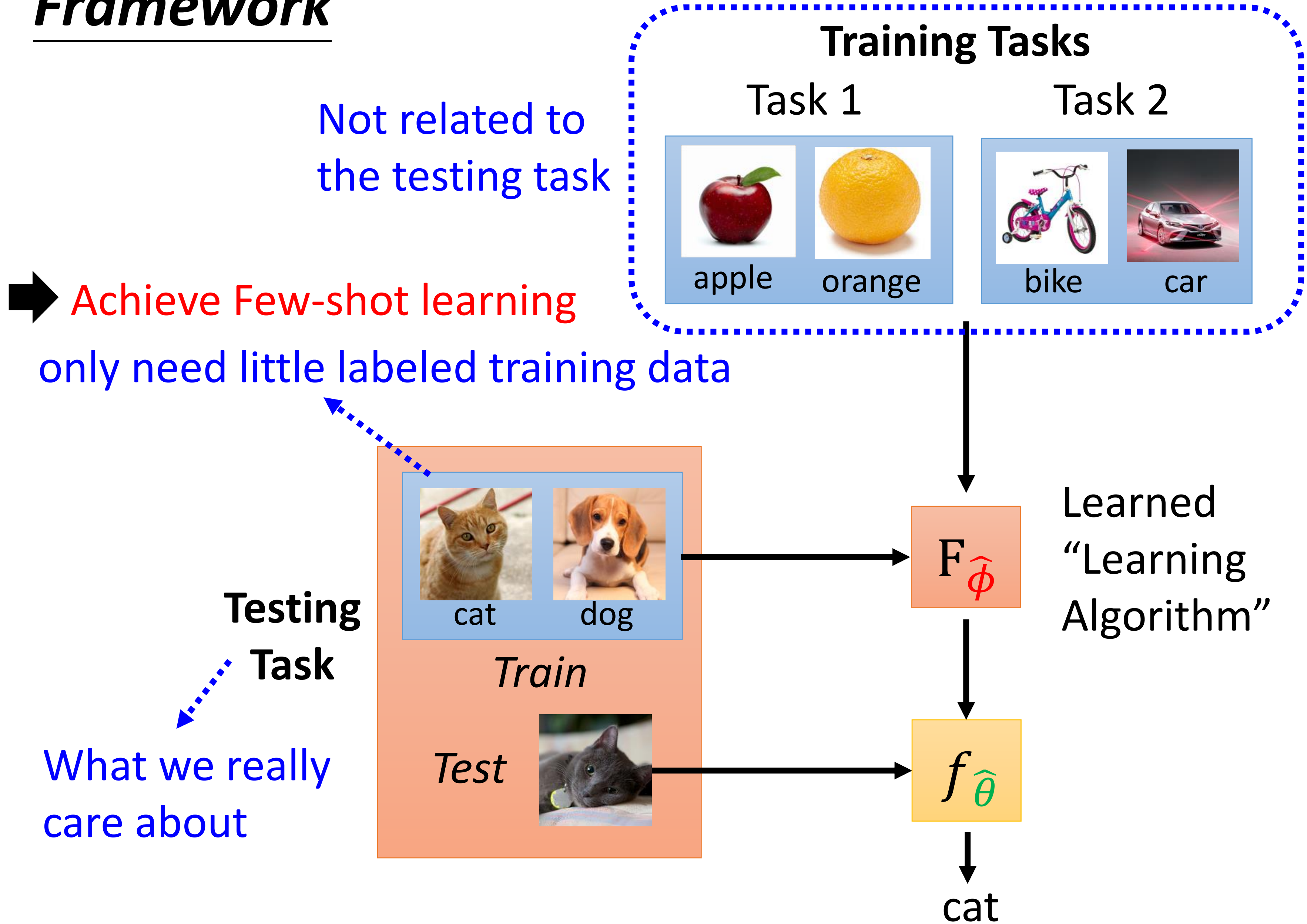
Gradient descent is your friend.

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

Reinforcement Learning / Evolutionary Algorithm

Now we have a learned "learning algorithm" $\mathrm{F}_{\hat{\phi}}$

Framework


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

ML v.s. Meta

Goal

<u>Machine Learning</u> ≈ find a function f

Dog-Cat Classification

$f($  $) =$ "cat"

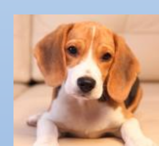
<u>Meta Learning</u>

≈ find a function F that finds a function f

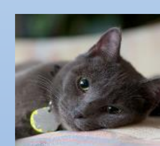
Learning Algorithm

$F($

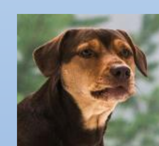
cat



dog



cat



dog

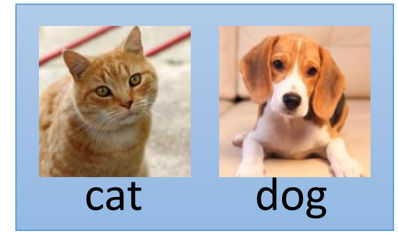
Training Examples

$) = f$

Training Data

Machine Learning

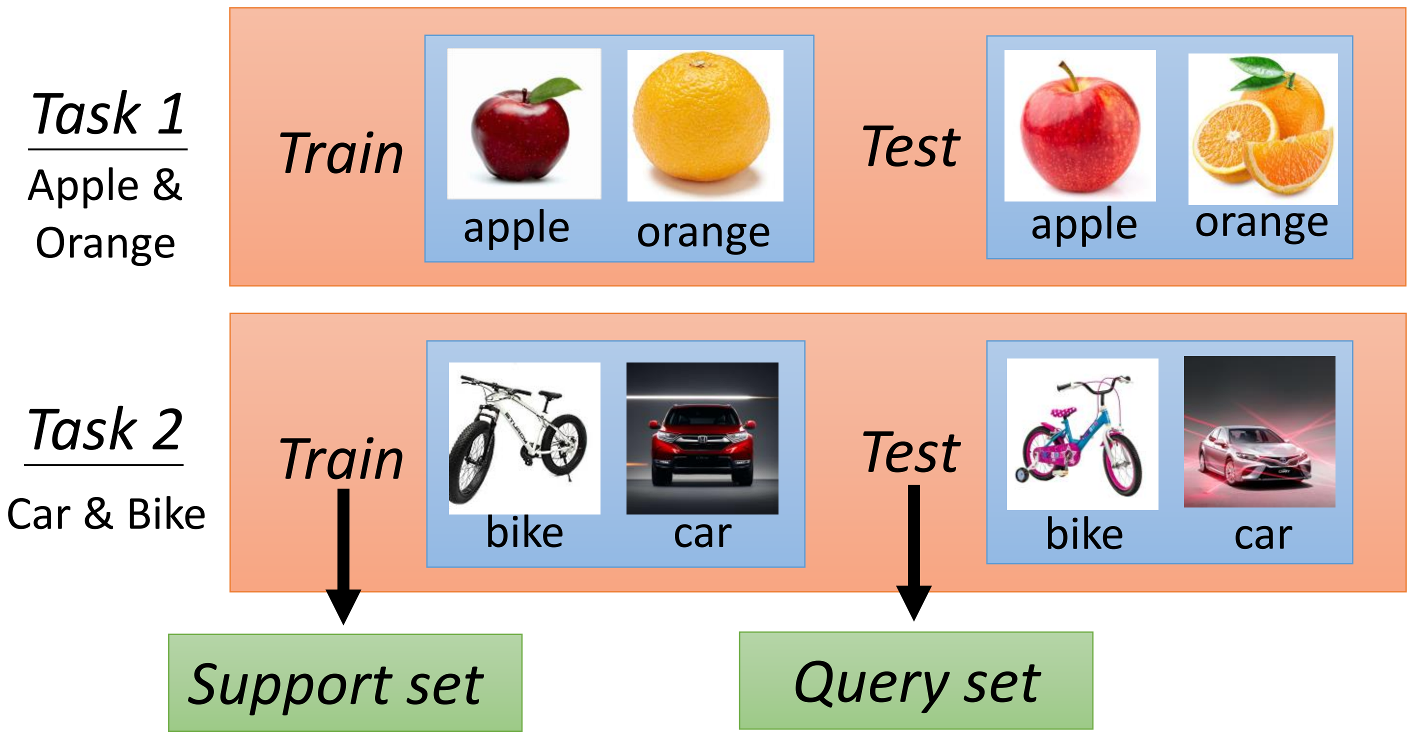
One task



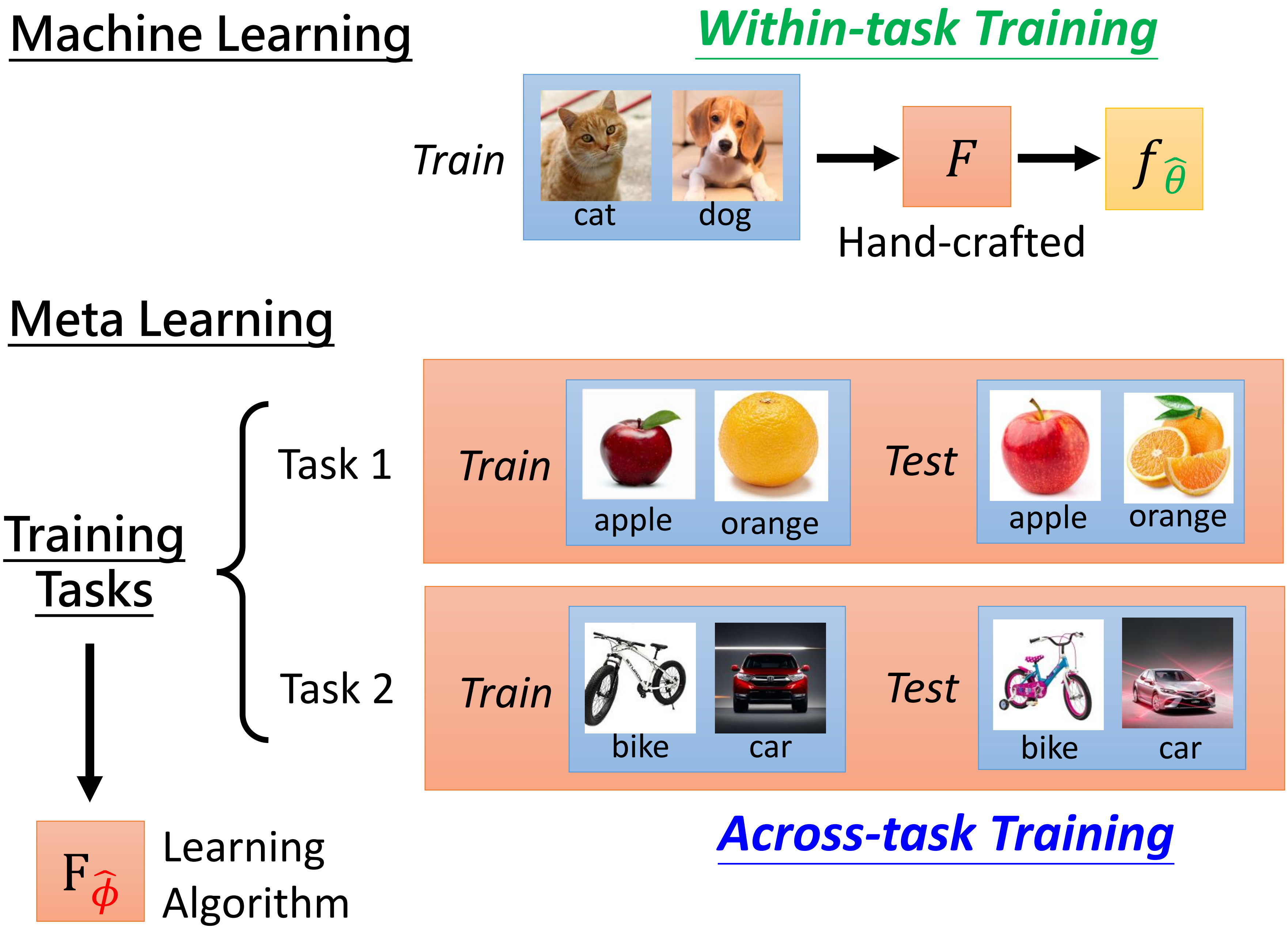
Train

Meta Learning

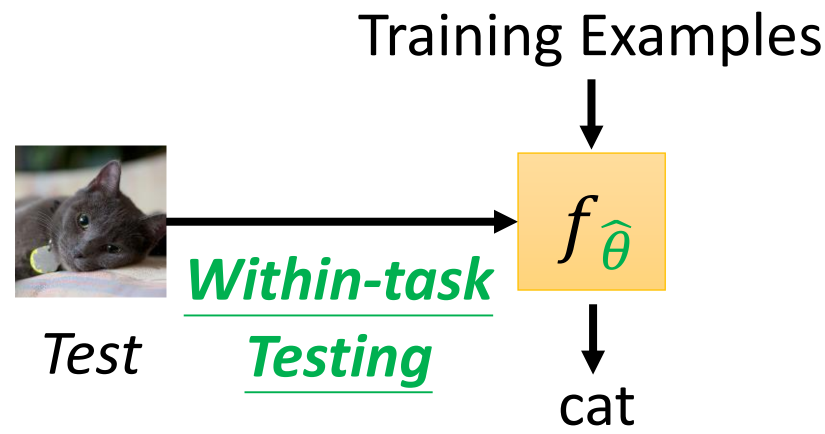
Training tasks



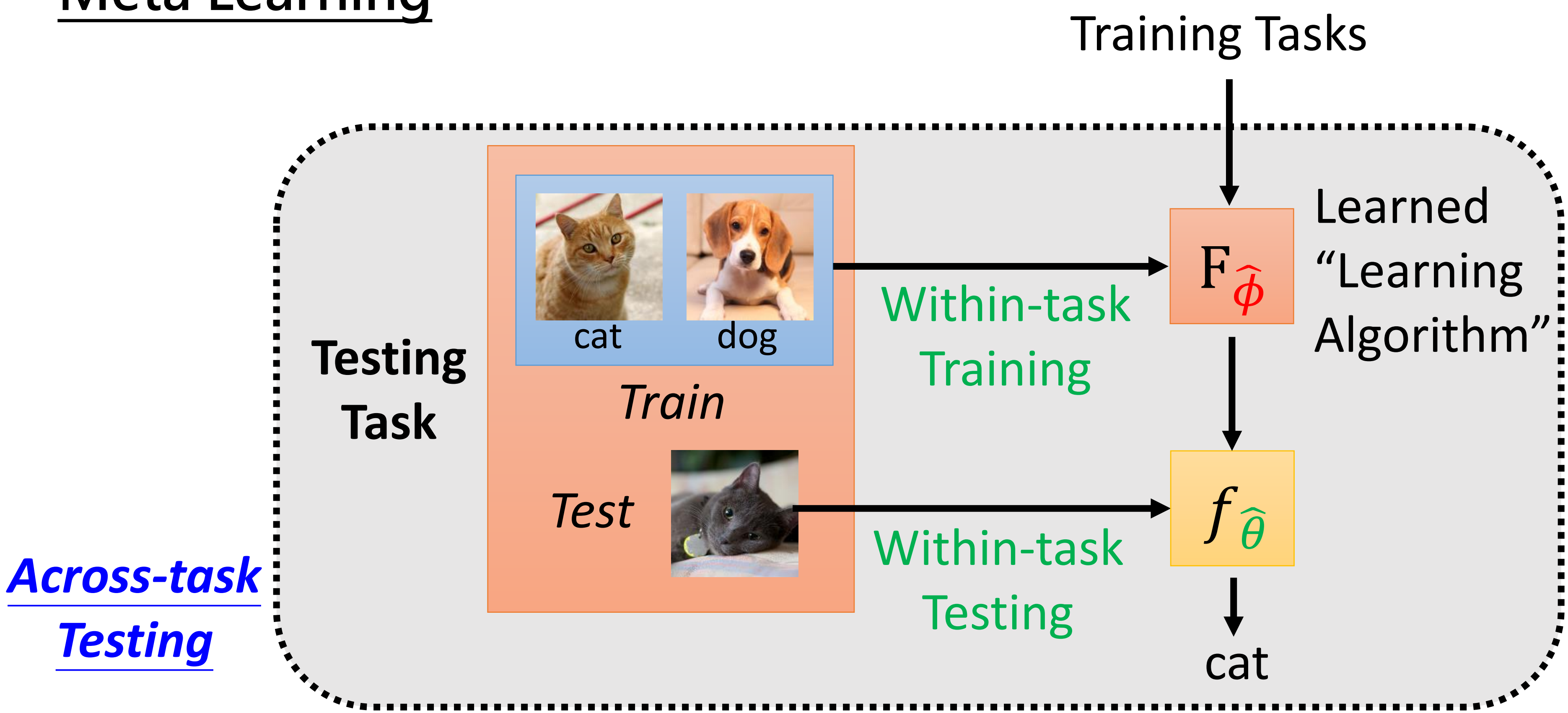
(in the literature of "*learning to compare*")


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

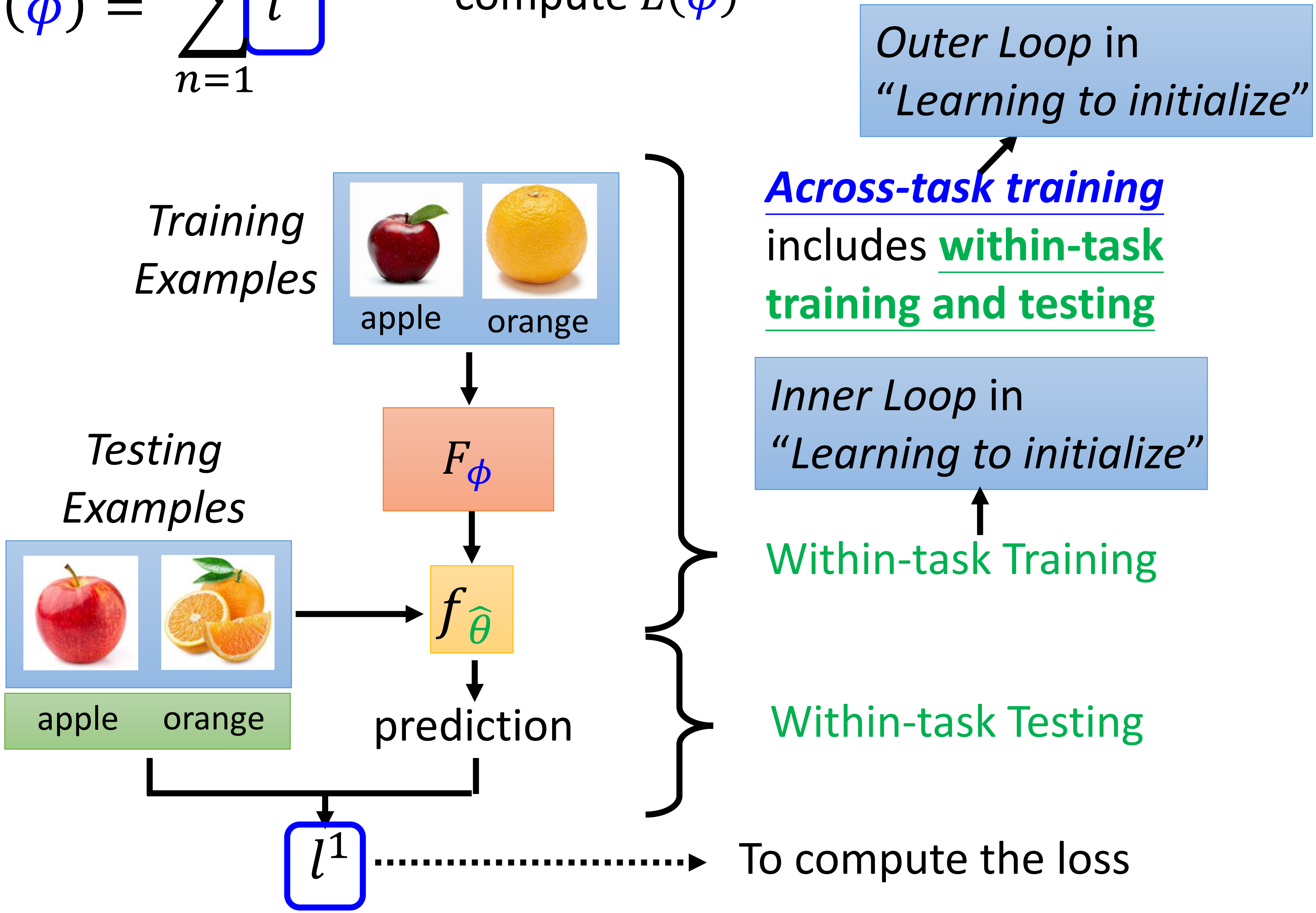
$$L(\phi) = \boxed{\sum_{n=1}^{N} \boxed{l^n}}$$

Sum over testing examples in one task

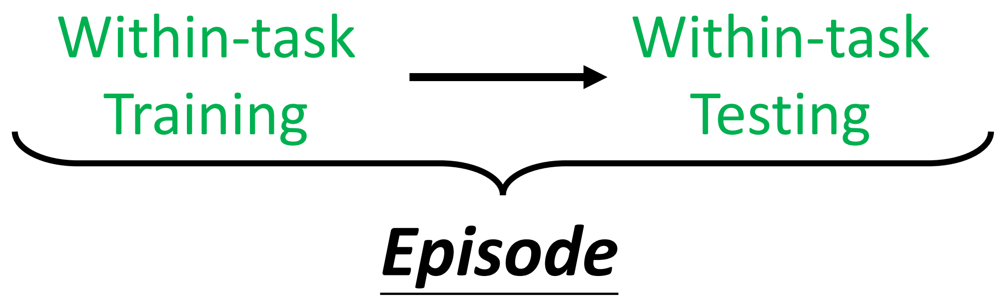
Sum over training tasks

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

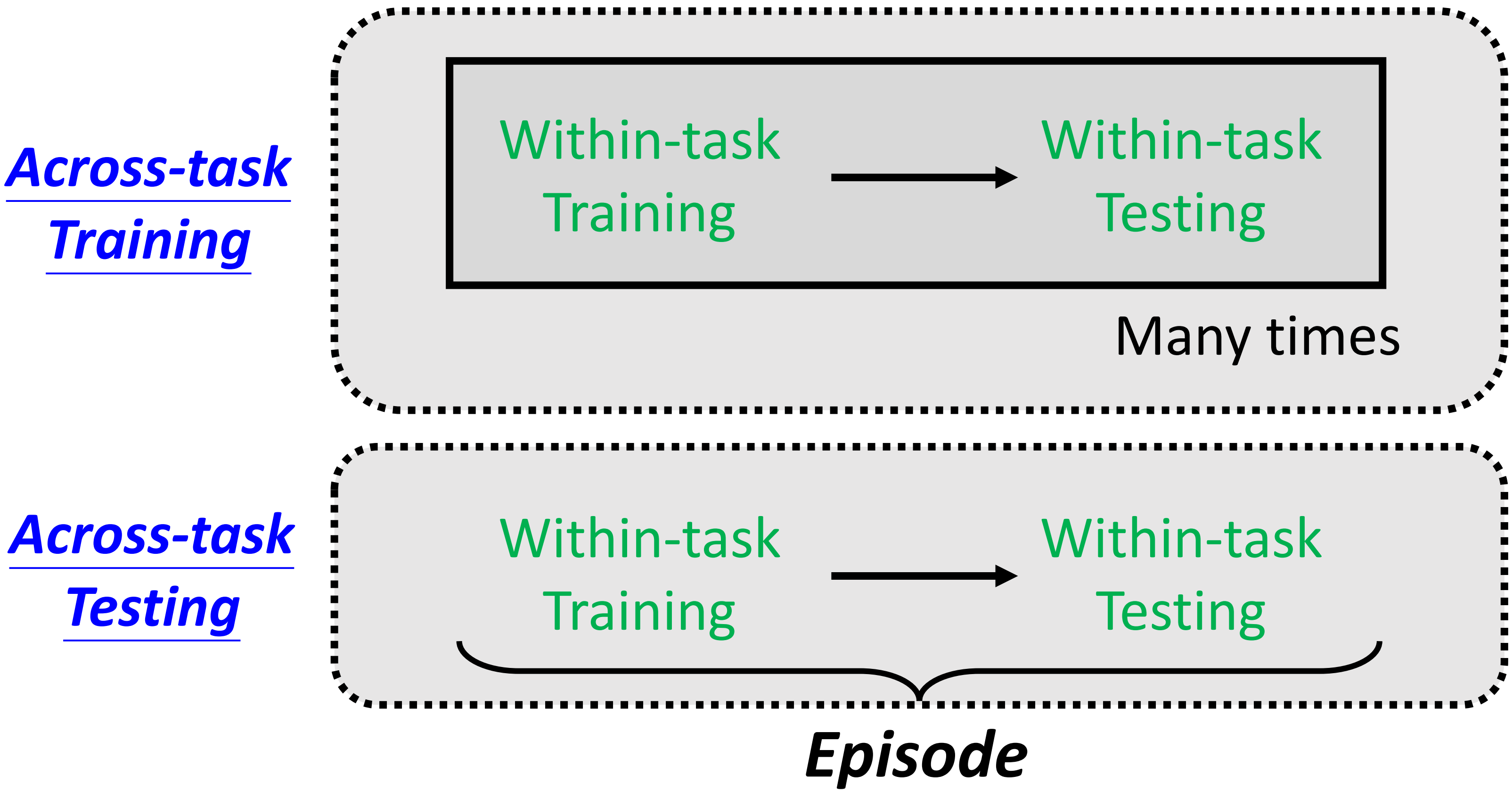
If your optimization method needs to compute $L(\phi)$



Machine Learning



Meta Learning

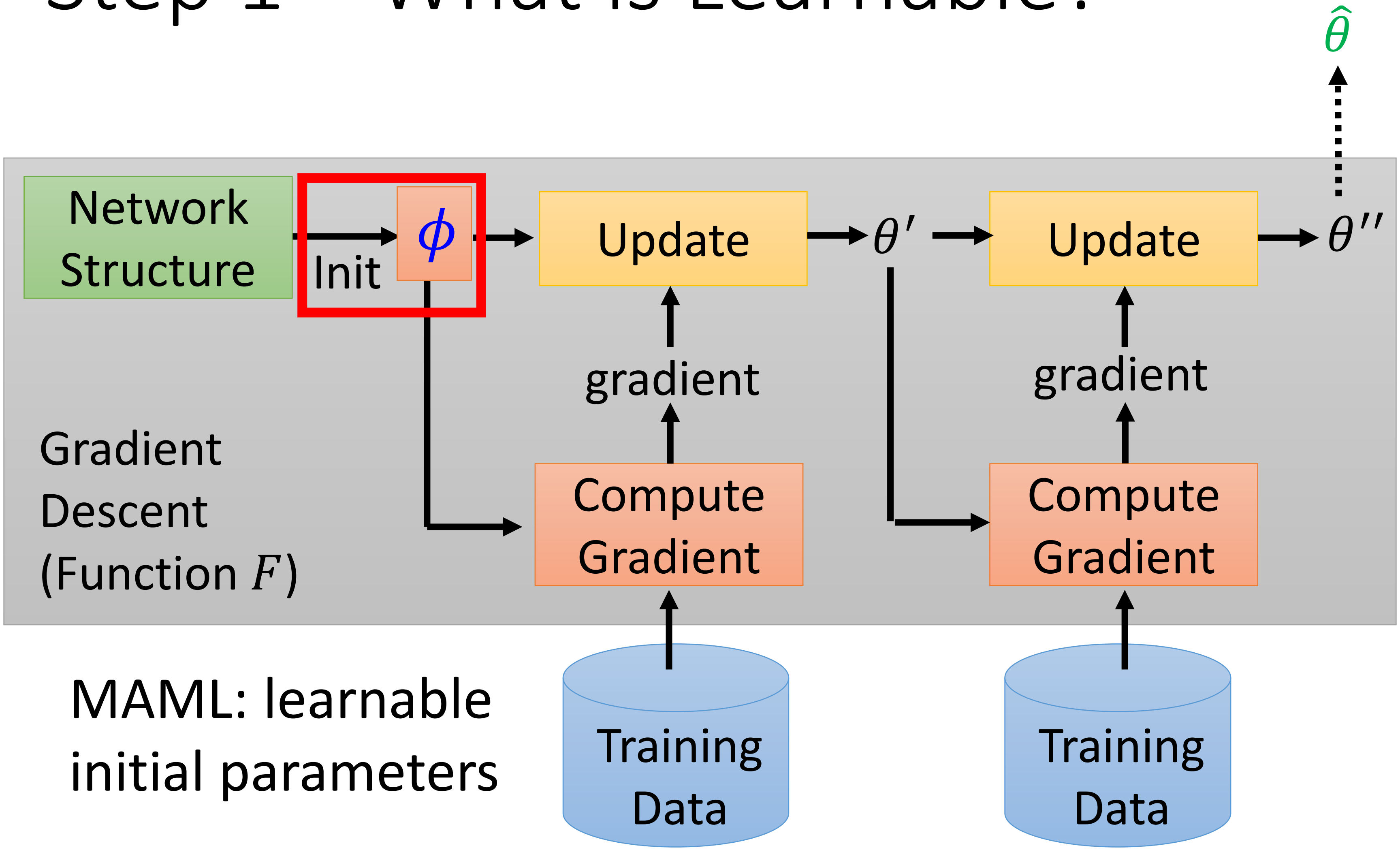


Learning to Initialize

Model-Agnostic Meta-Learning (MAML)

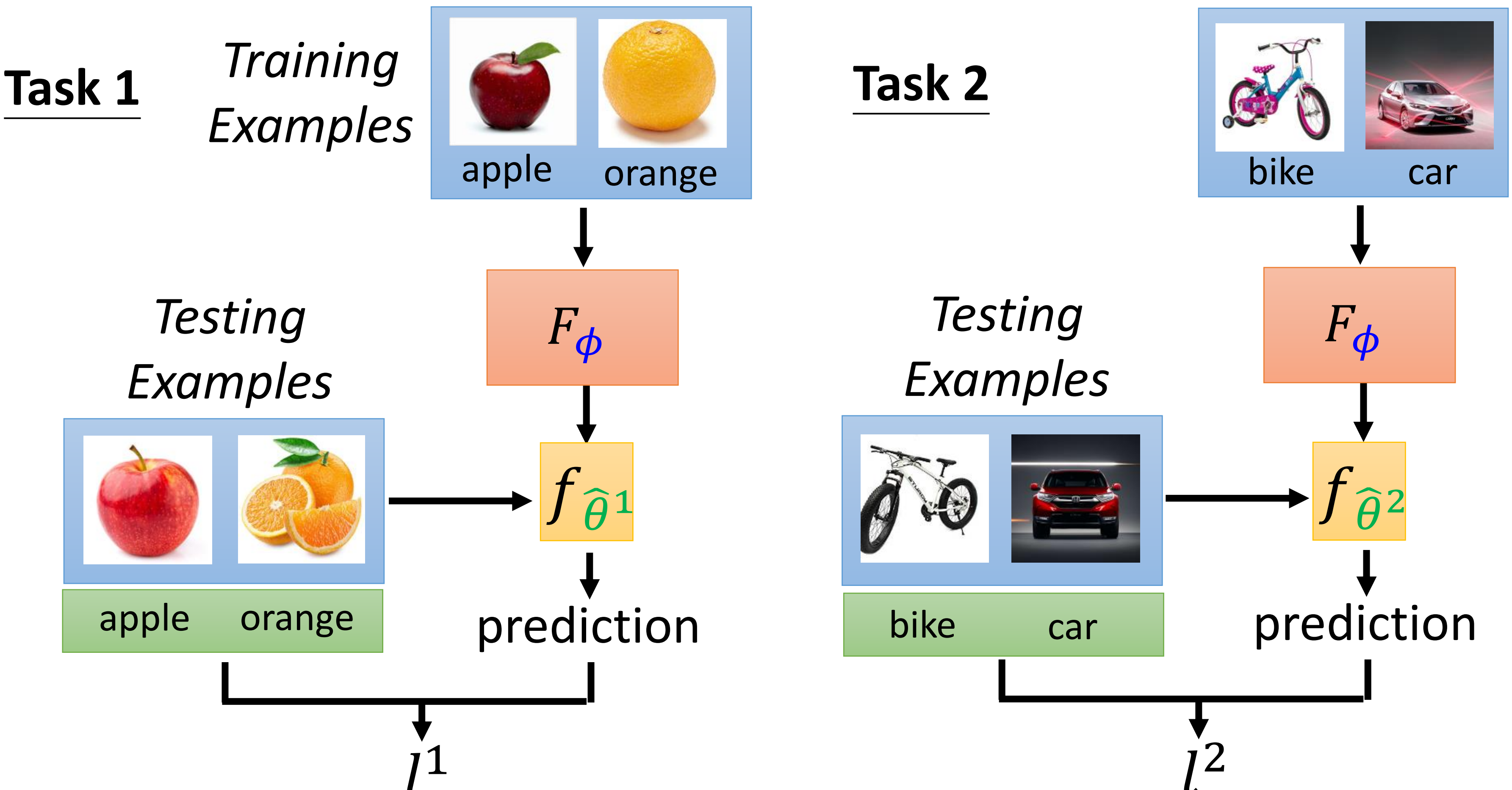
Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”, ICML, 2017

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)

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

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

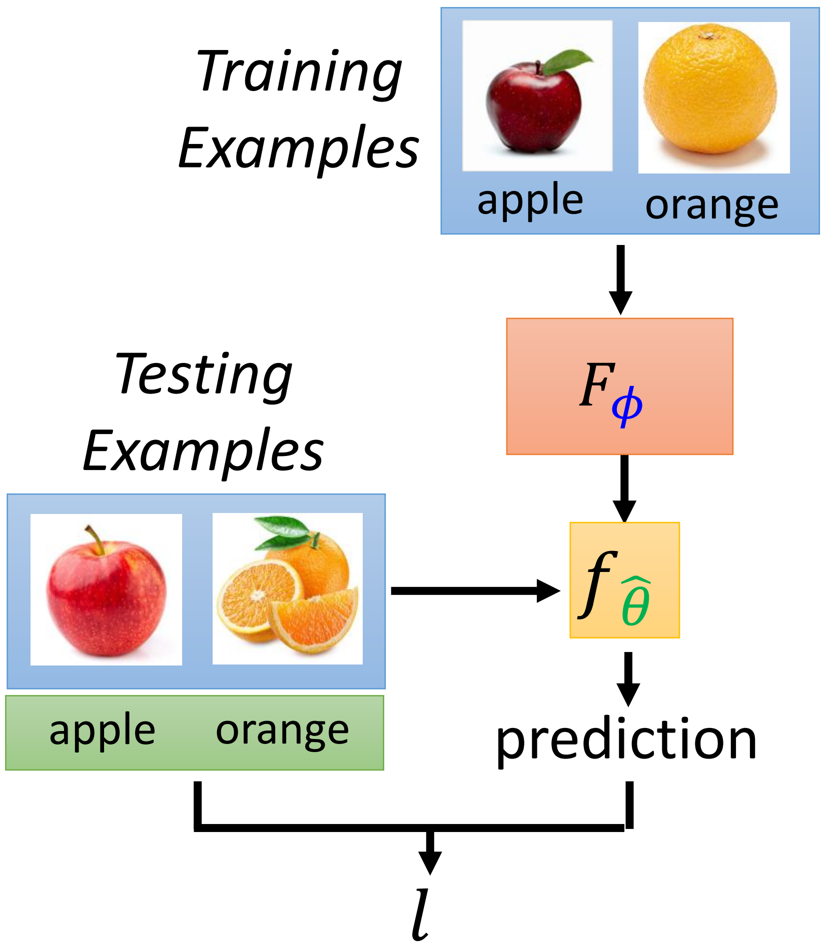
How to compute $\nabla_\phi l$

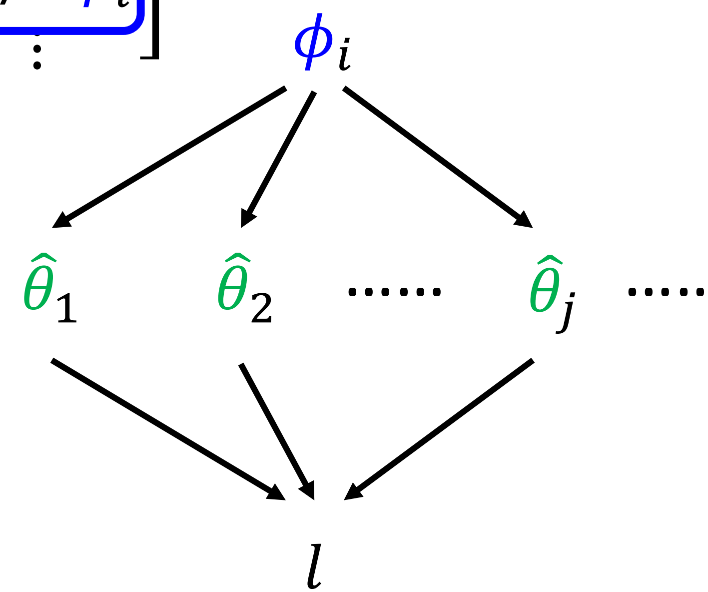
(^{n} is ignored here)

ϕ_i : the i-th parameter of ϕ

Step 3 – Optimization

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





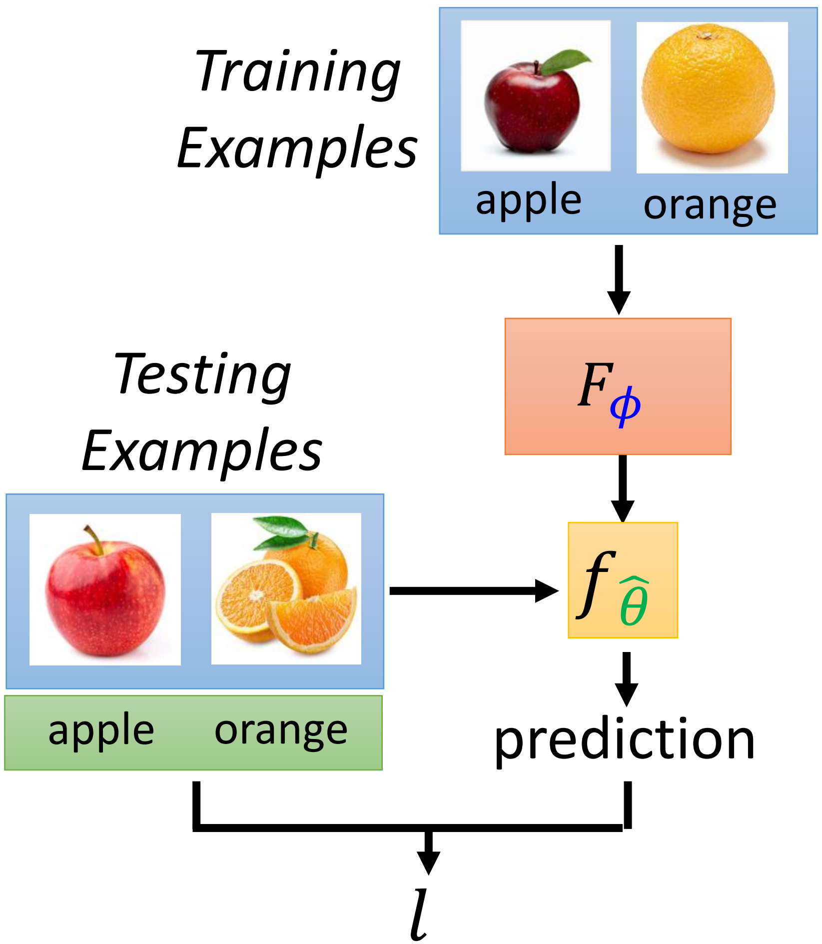
$$\frac{\partial l}{\partial \phi_i} = \sum_j \boxed{\frac{\partial l}{\partial \hat{\theta}_j}} \frac{\partial \hat{\theta}_j}{\partial \phi_i}$$

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

Step 3 – Optimization

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

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



Within-task Training

(inner loop in MAML)

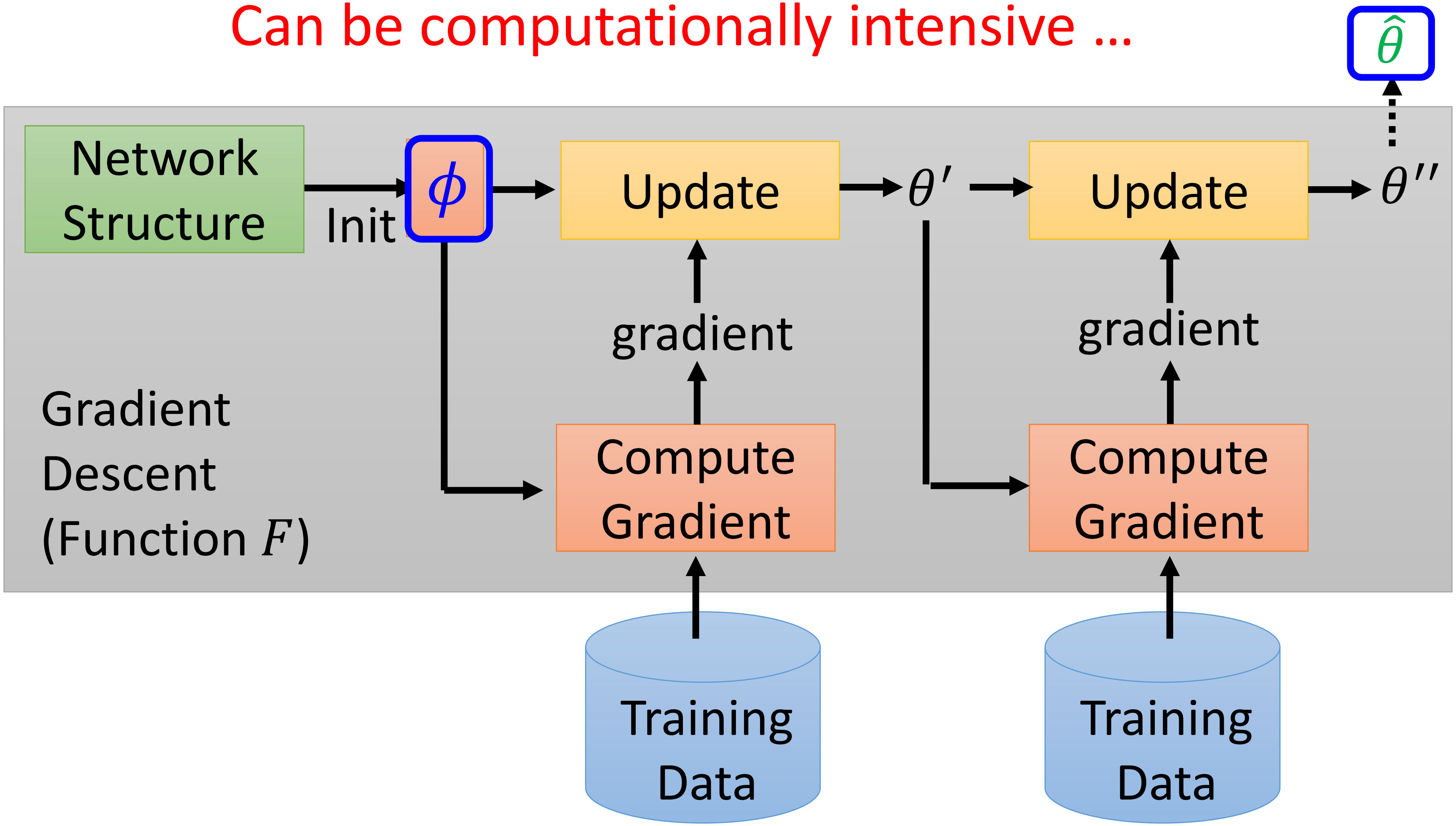
Can be computationally intensive …

Within-task Testing

Step 3 – Optimization

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

Can be computationally intensive …



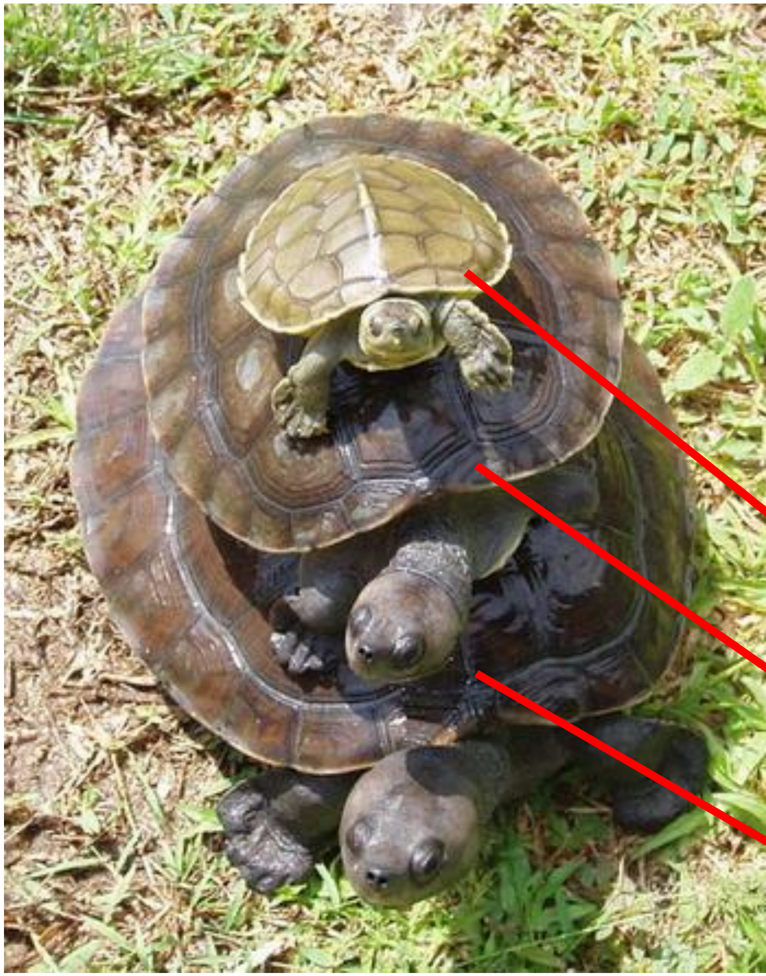
Step 3 – Optimization

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

Can be computationally intensive …

- Reduce the parameter update steps in within-task training (using only *one step* is typical)
- First order approximation: FOMAML, Reptile
  - **Reptile:** Alex Nichol, Joshua Achiam, John Schulman, On First-Order Meta-Learning Algorithms, arXiv, 2018
- Inventing efficient ways to compute gradients: iMAML
  - **iMAML**: Aravind Rajeswaran, Chelsea Finn, Sham Kakade, Sergey Levine, Meta-Learning with Implicit Gradients, NeurIPS, 2019

Turtles all the way down?



- MAML learns the initialization parameter ϕ

by gradient descent

- What is the initialization parameter ϕ^0 for ϕ?

Learn to initialize

Learn to learn to initialize?

Learn to learn to learn to initialize?

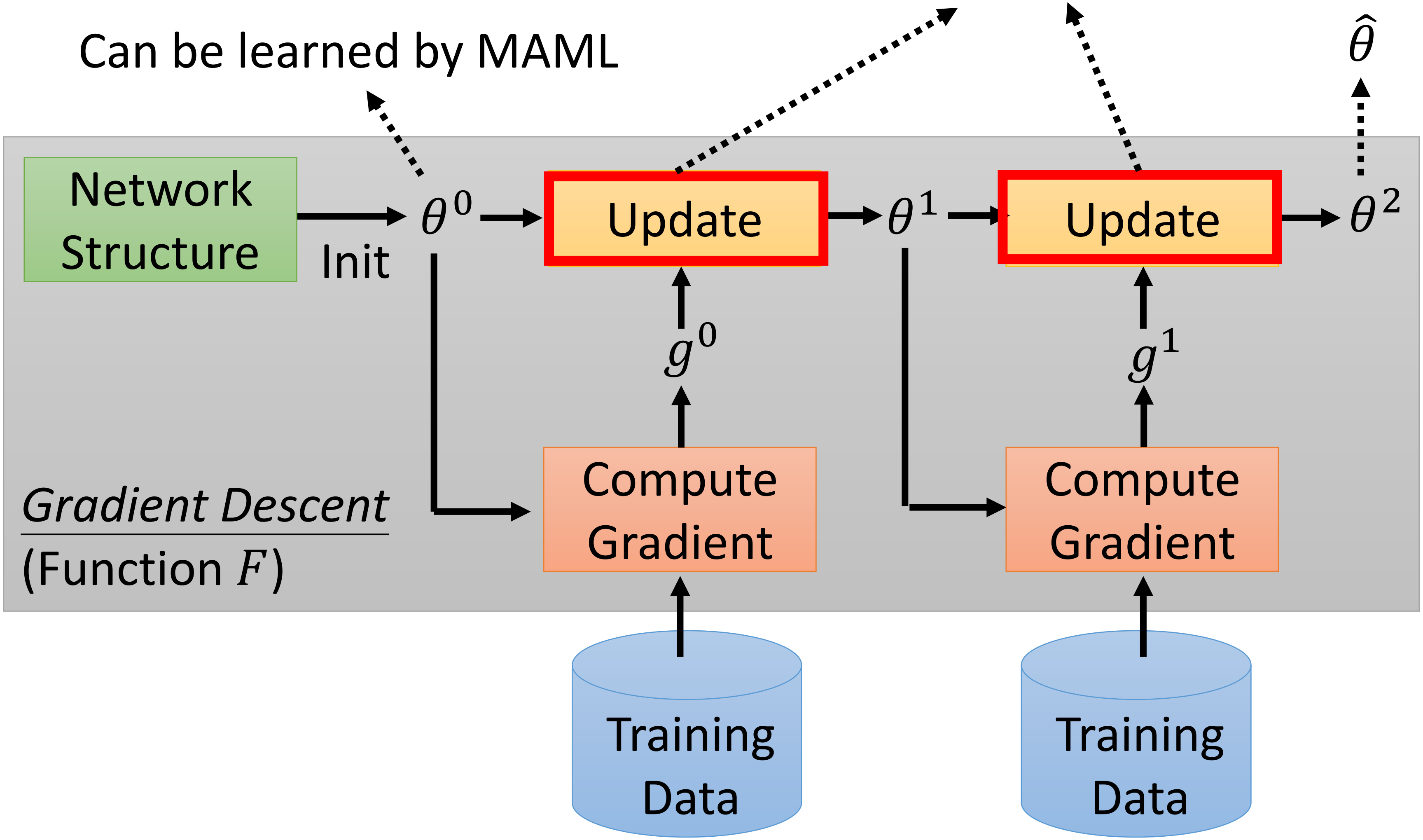
More Approaches

Optimizer

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

Adagrad, RMSprop, NAG, Adam ……

Is the optimizer learnable?

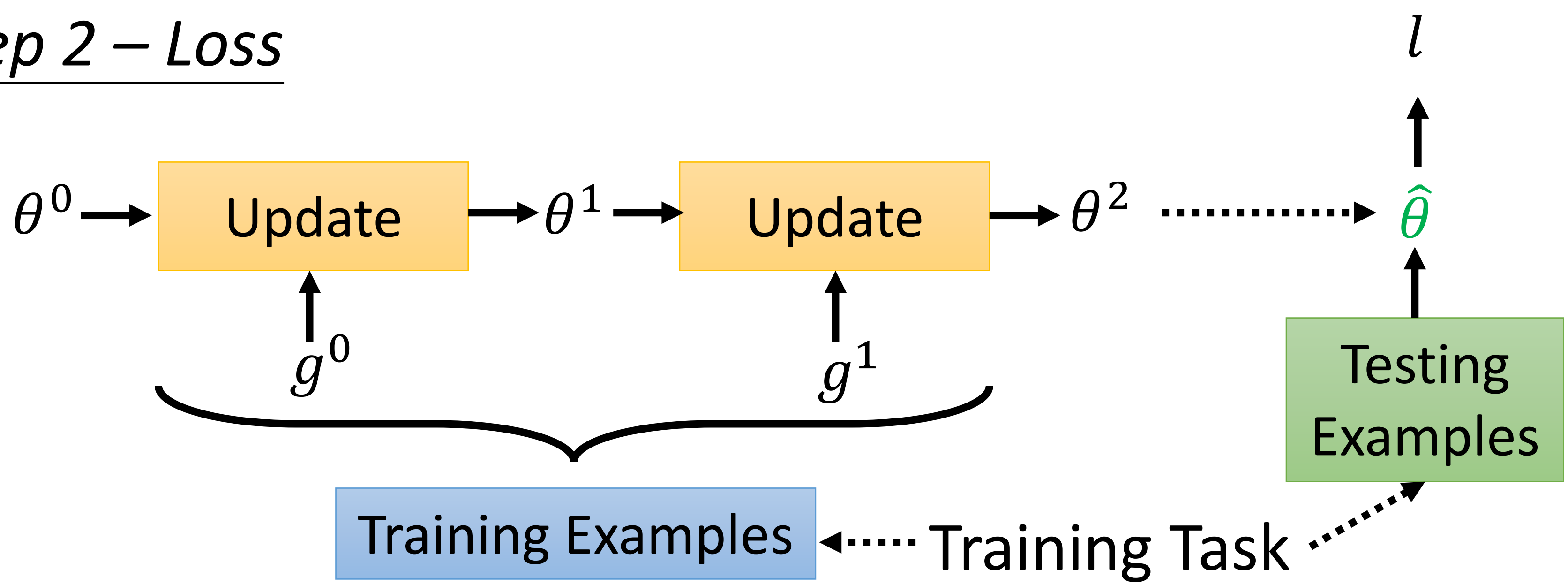


Learning Optimizer

Step 1 – What is learnable?

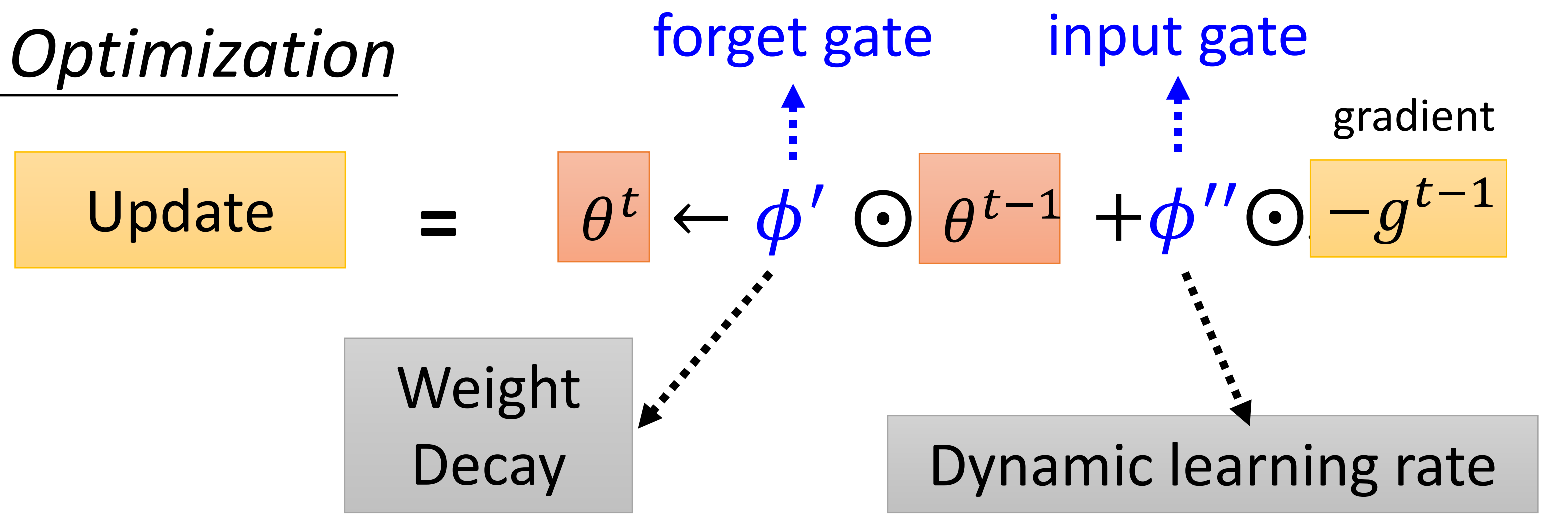
$$\text{Update} = \theta^{t} \leftarrow \underbrace{\phi'}_{\text{Weight Decay}} \odot \theta^{t-1} + \underbrace{\phi''}_{\text{Dynamic learning rate}} \odot \overbrace{-g^{t-1}}^{\text{gradient}}$$

Step 2 – Loss



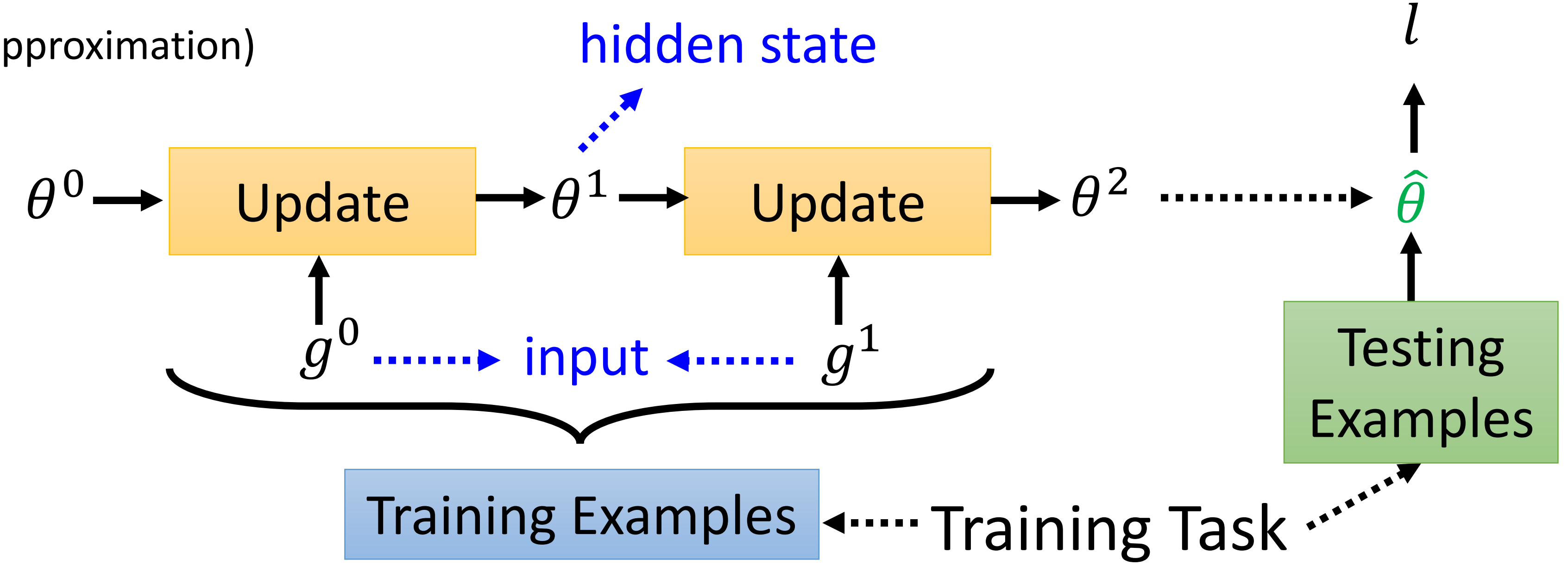
Learning Optimizer

Step 3 – Optimization



This is a "RNN"!

(approximation)



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

Optimizer

forget gate

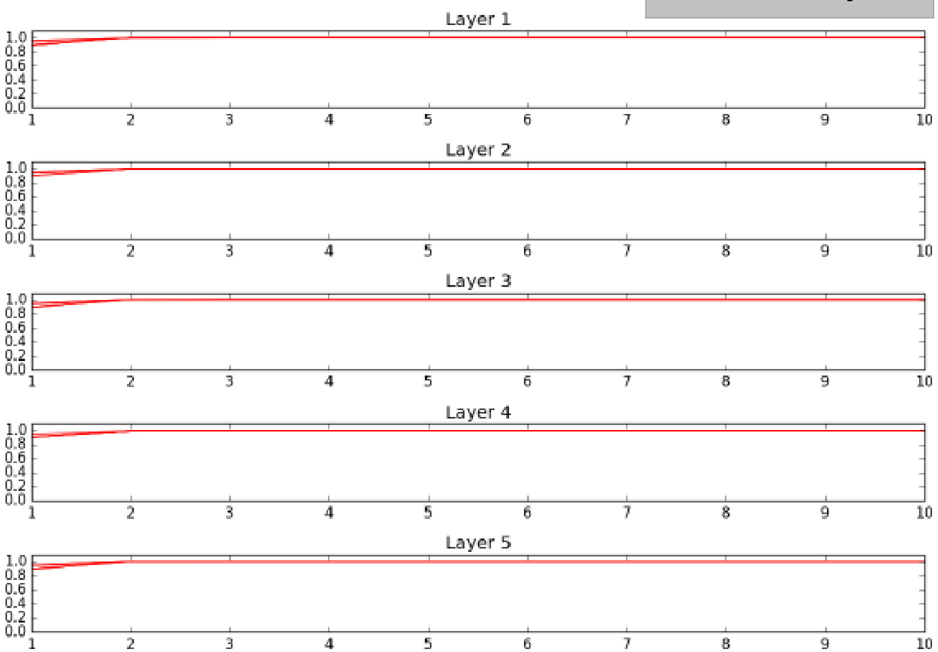
input gate

gradient

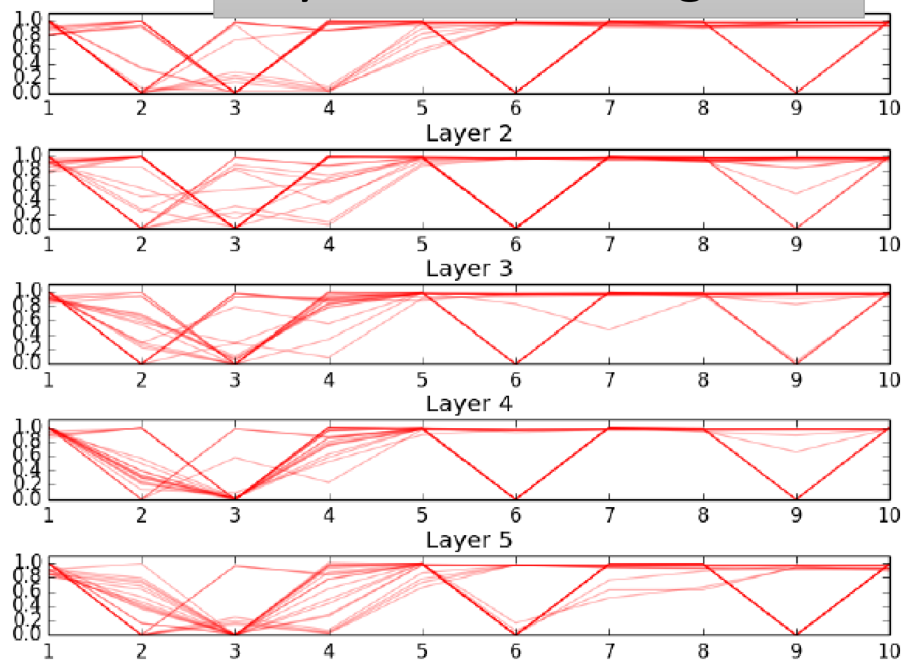
$$\boxed{\text{Update}} = \boxed{\theta^t} \leftarrow \phi' \odot \boxed{\theta^{t-1}} + \phi'' \odot \boxed{-g^{t-1}}$$

Weight Decay

Dynamic learning rate



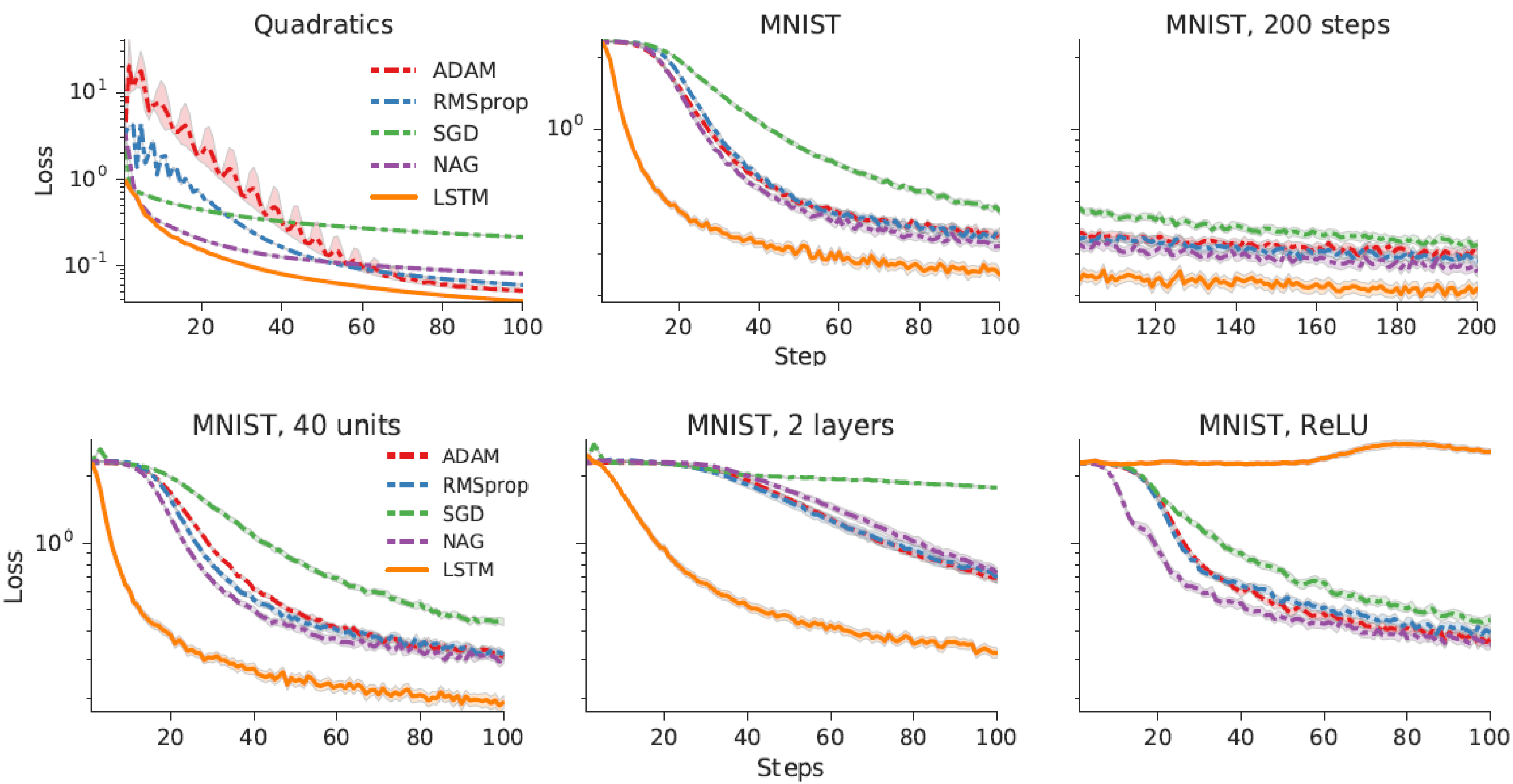
(a) Forget gate values for 1-shot meta-learner



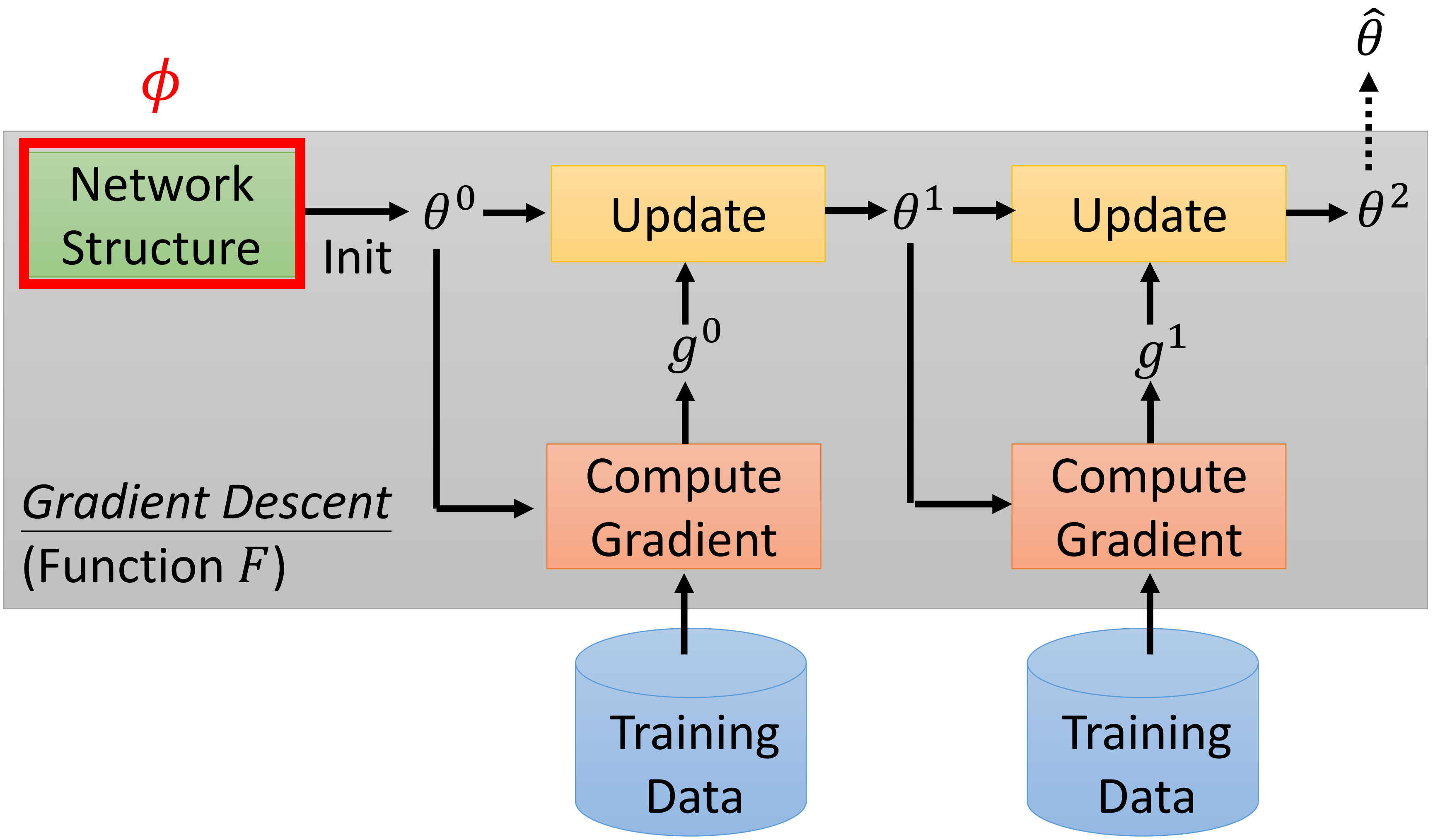
(b) Input gate values for 1-shot meta-learner

Optimizer

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



Network Architecture Search (NAS)



Network Architecture Search (NAS)

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

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

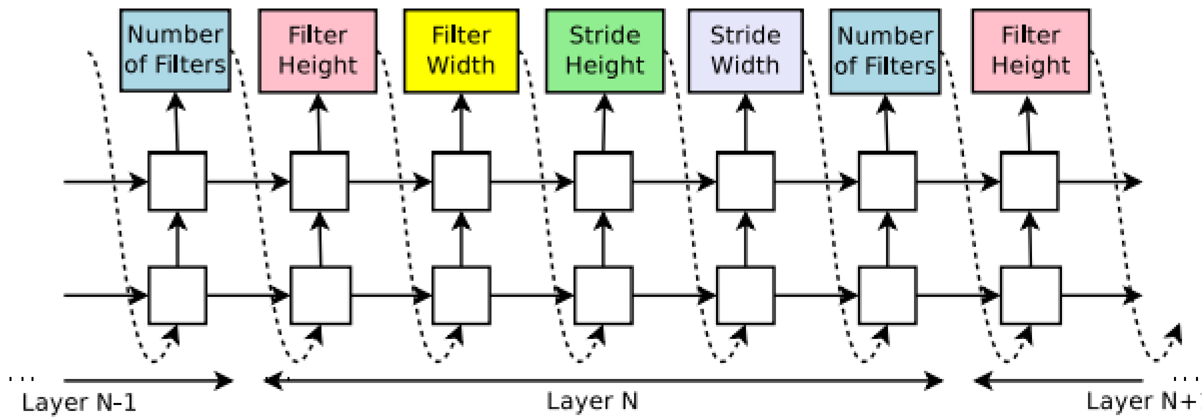
$-L(\phi)$

Reward to be maximized

Network Architecture Search (NAS)

Across-task Training

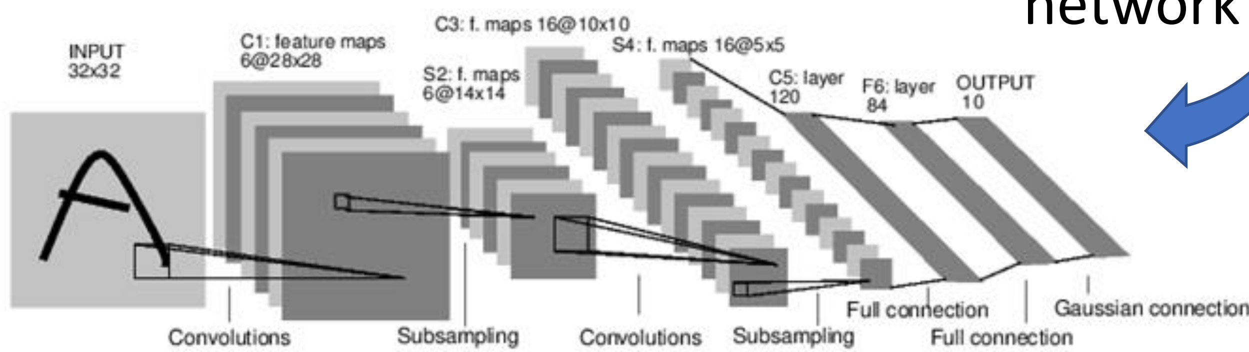
Update ϕ to maximize reward $-L(\phi)$



agent ϕ (RNN)

form a network

$-L(\phi)$



A Full Convolutional Neural Network (LeNet)

Accuracy of the network

Train the network

Within-task Training

Network Architecture Search (NAS)

$$\hat{\phi} = \underset{\phi}{arg\,min}\, L(\phi) \qquad \nabla_{\phi} L(\phi) = ?$$

Network Architecture

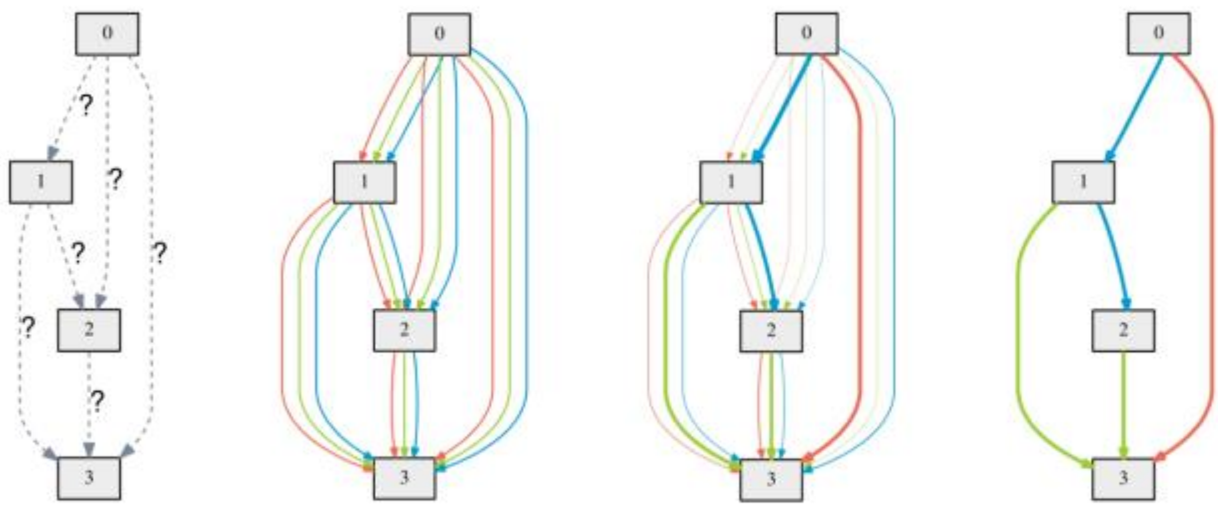
- Reinforcement Learning
  - Barret Zoph, et al., Neural Architecture Search with Reinforcement Learning, ICLR 2017
  - Barret Zoph, et al., Learning Transferable Architectures for Scalable Image Recognition, CVPR, 2018
  - Hieu Pham, et al., Efficient Neural Architecture Search via Parameter Sharing,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

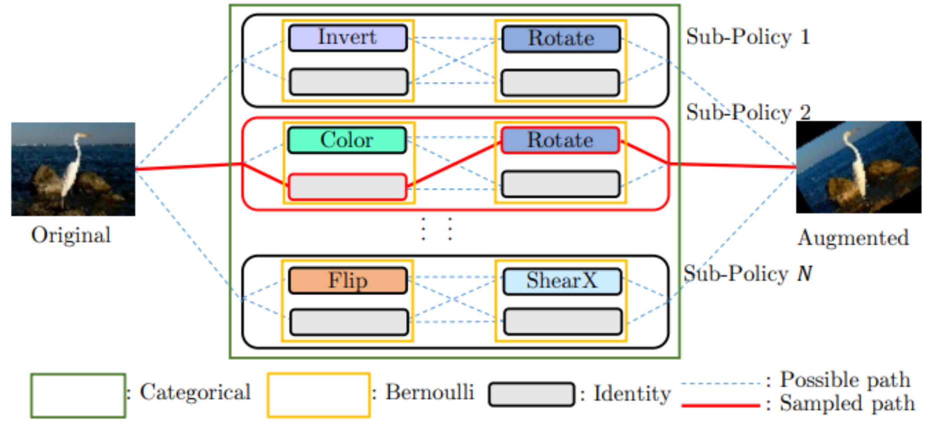
- DARTS Hanxiao Liu, et al., DARTS: Differentiable Architecture Search, ICLR, 2019



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

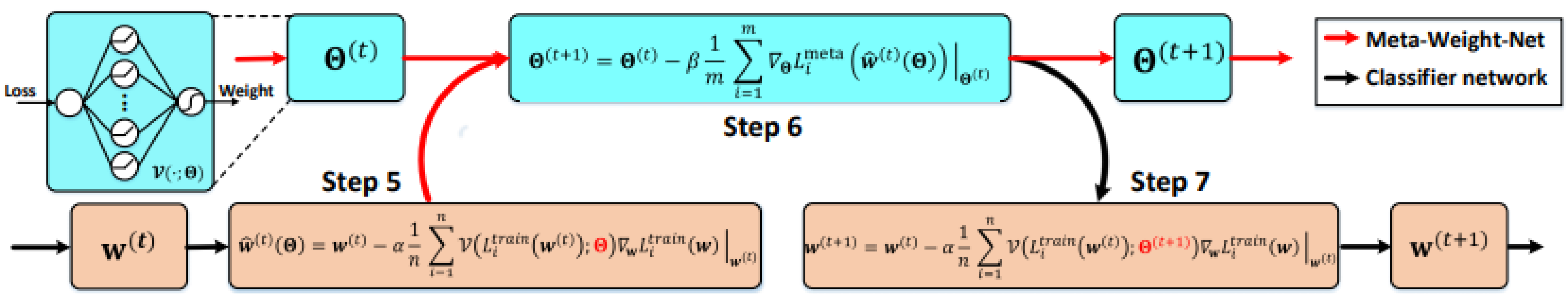
Data Augmentation / Data Reweighting

Data Augmentation



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

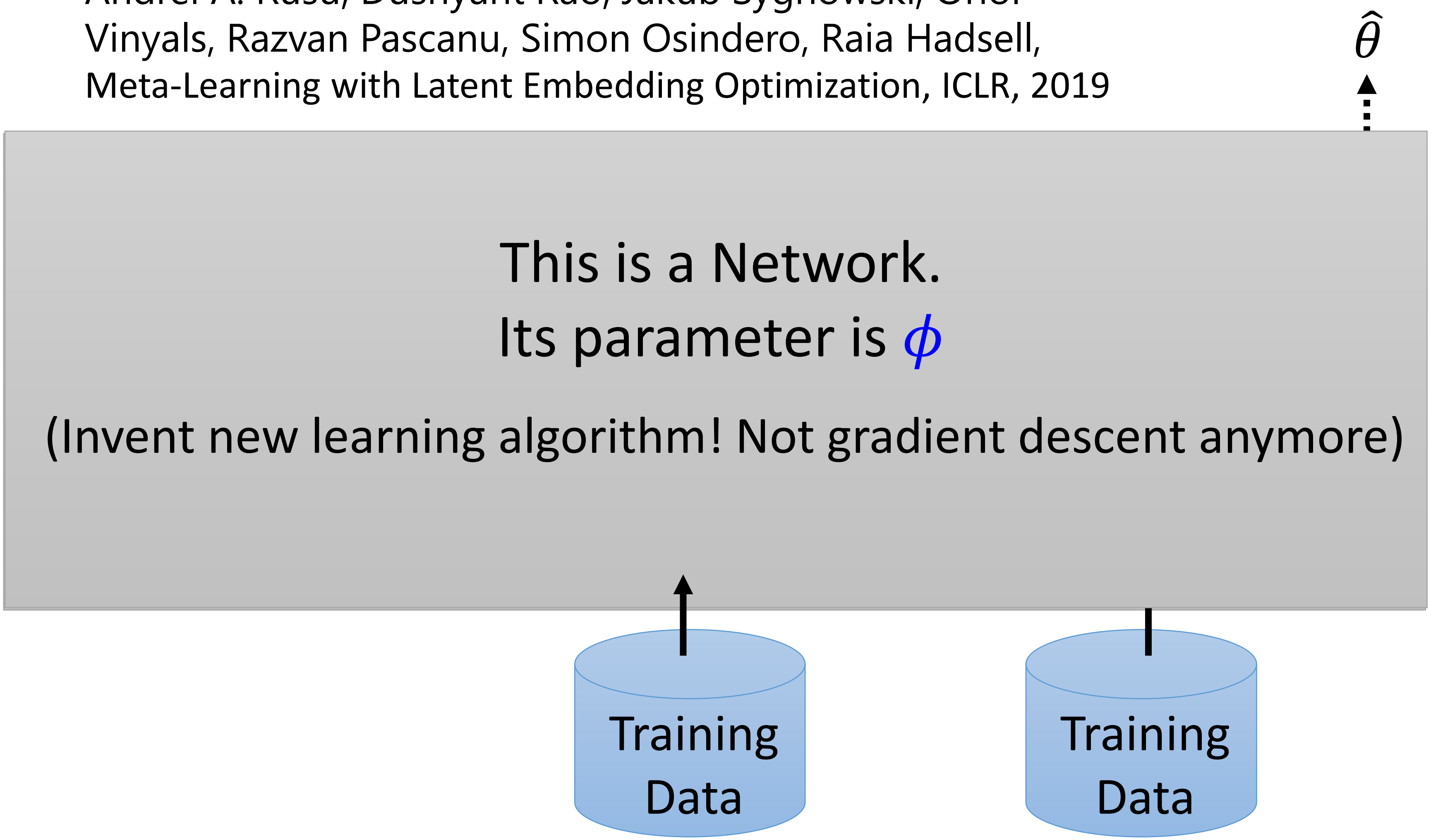
Data Reweighting



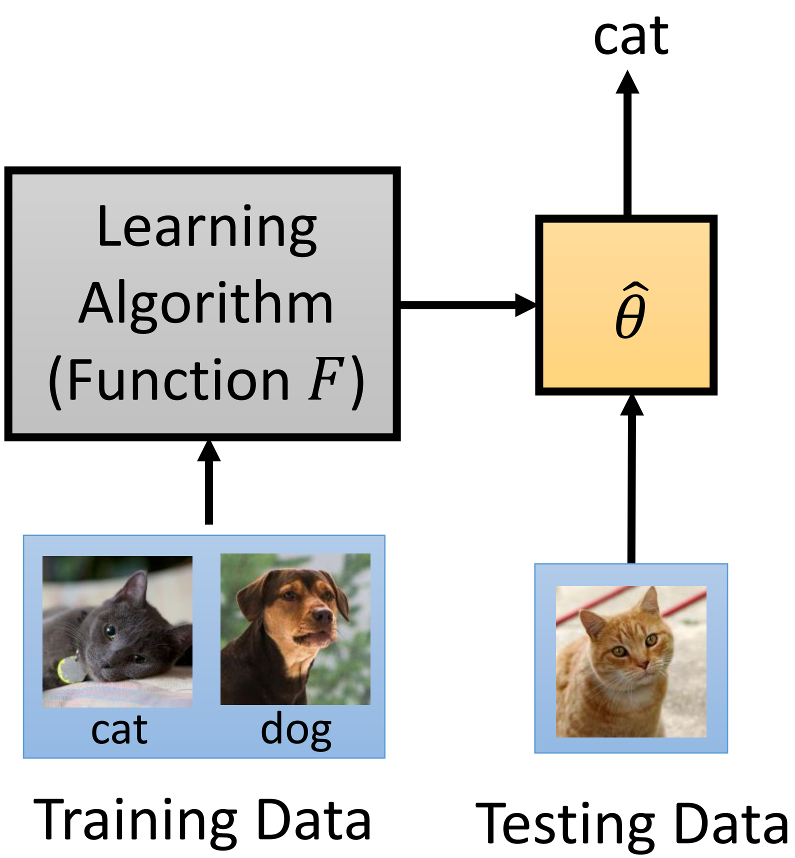
Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, Deyu Meng,
Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting, NeurIPS, 2019

Learning as a Network?

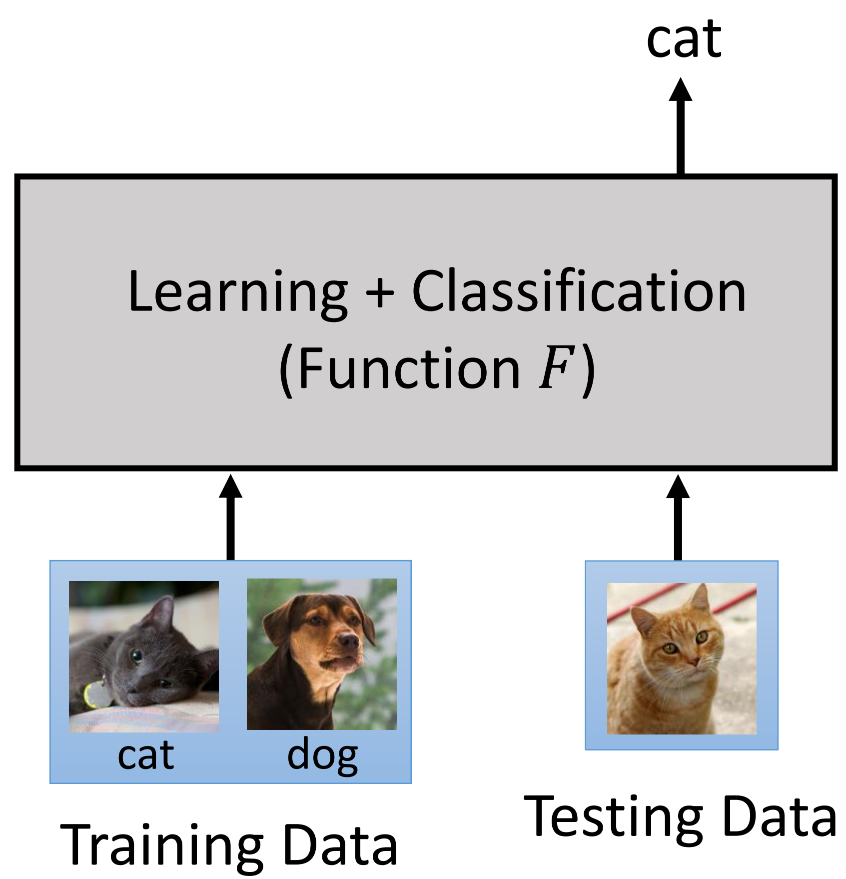
Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, Raia Hadsell, Meta-Learning with Latent Embedding Optimization, ICLR, 2019



Until now ......



Next ......

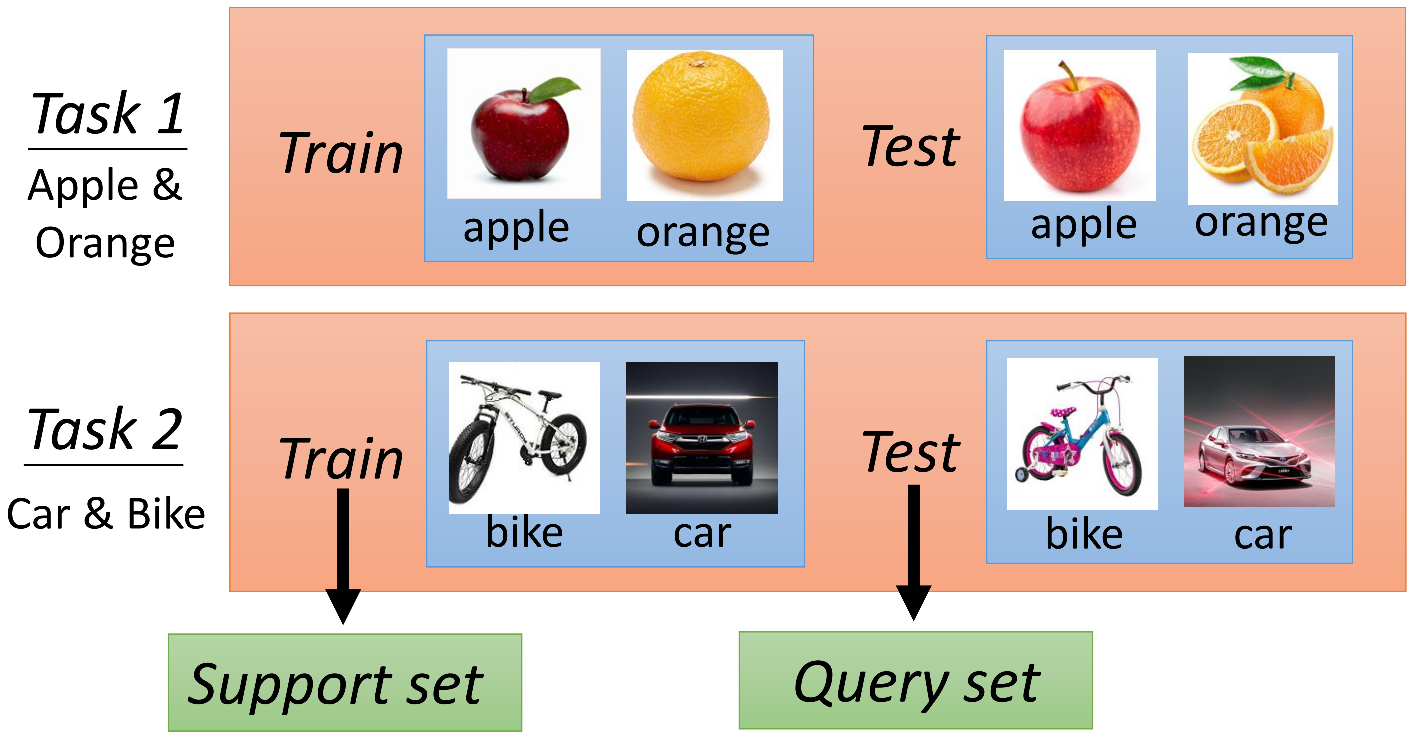


Learning to Compare

Training

Meta Learning

Training tasks

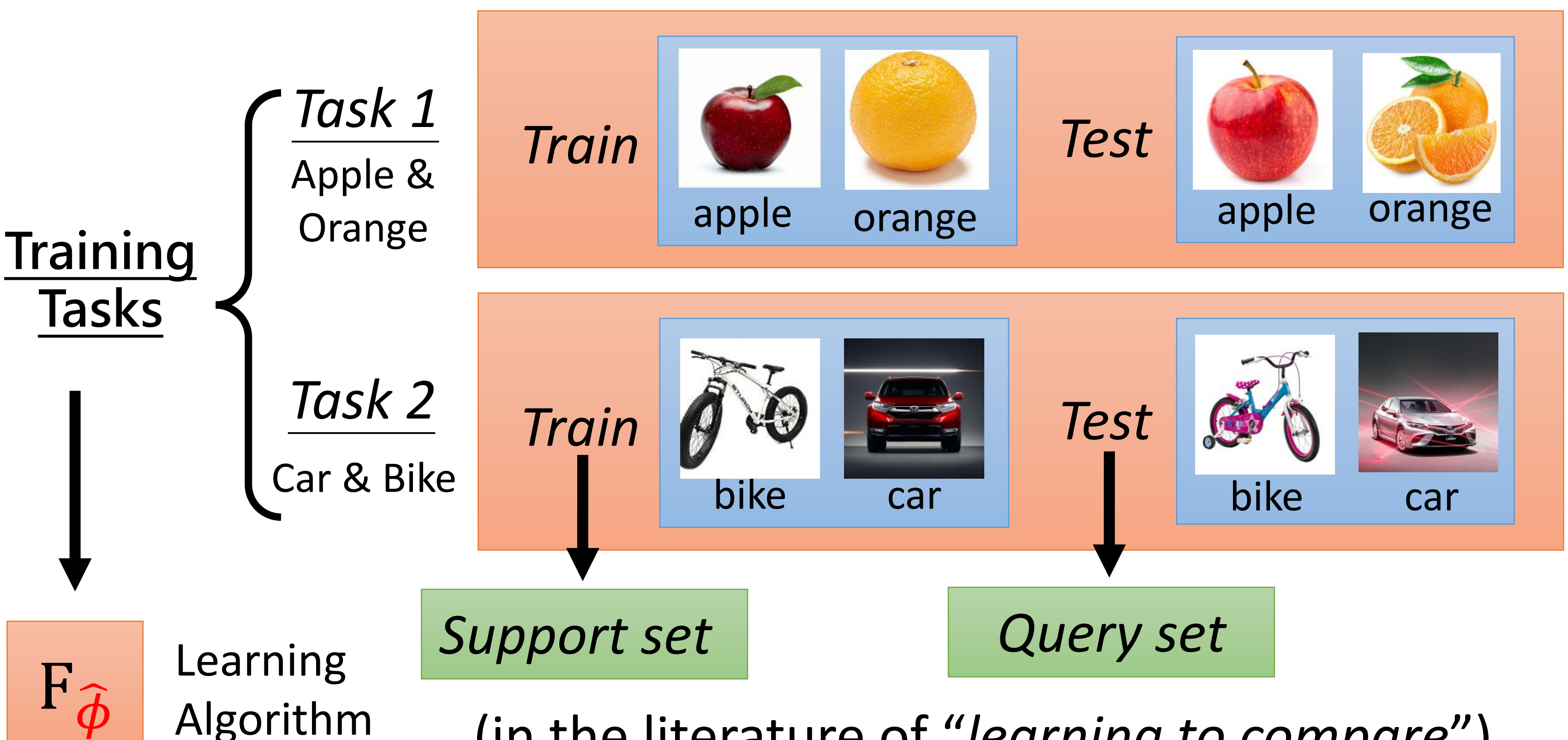


(in the literature of "*learning to compare*")

Training

Meta Learning

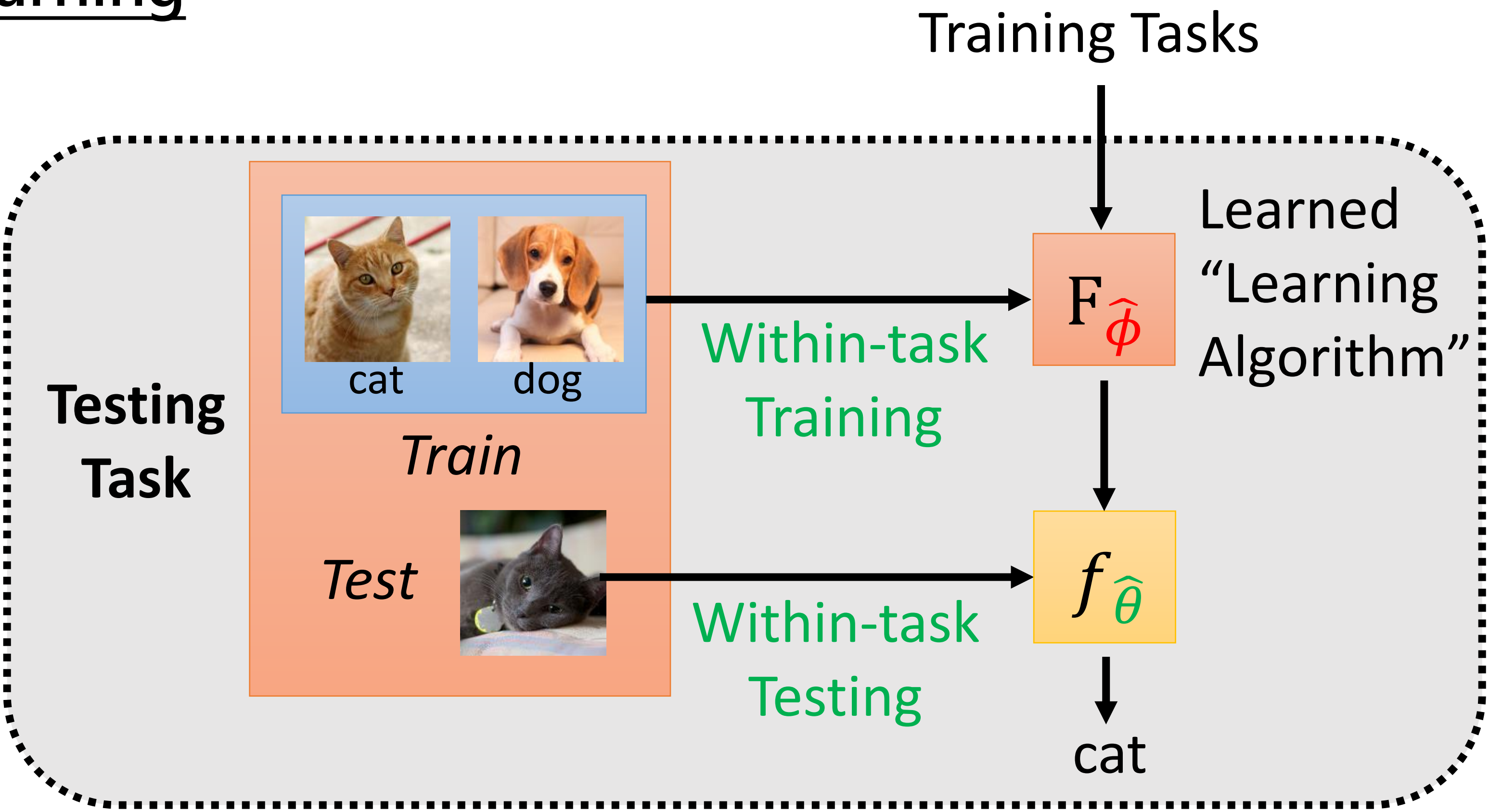
Training tasks



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

Testing

Meta Learning



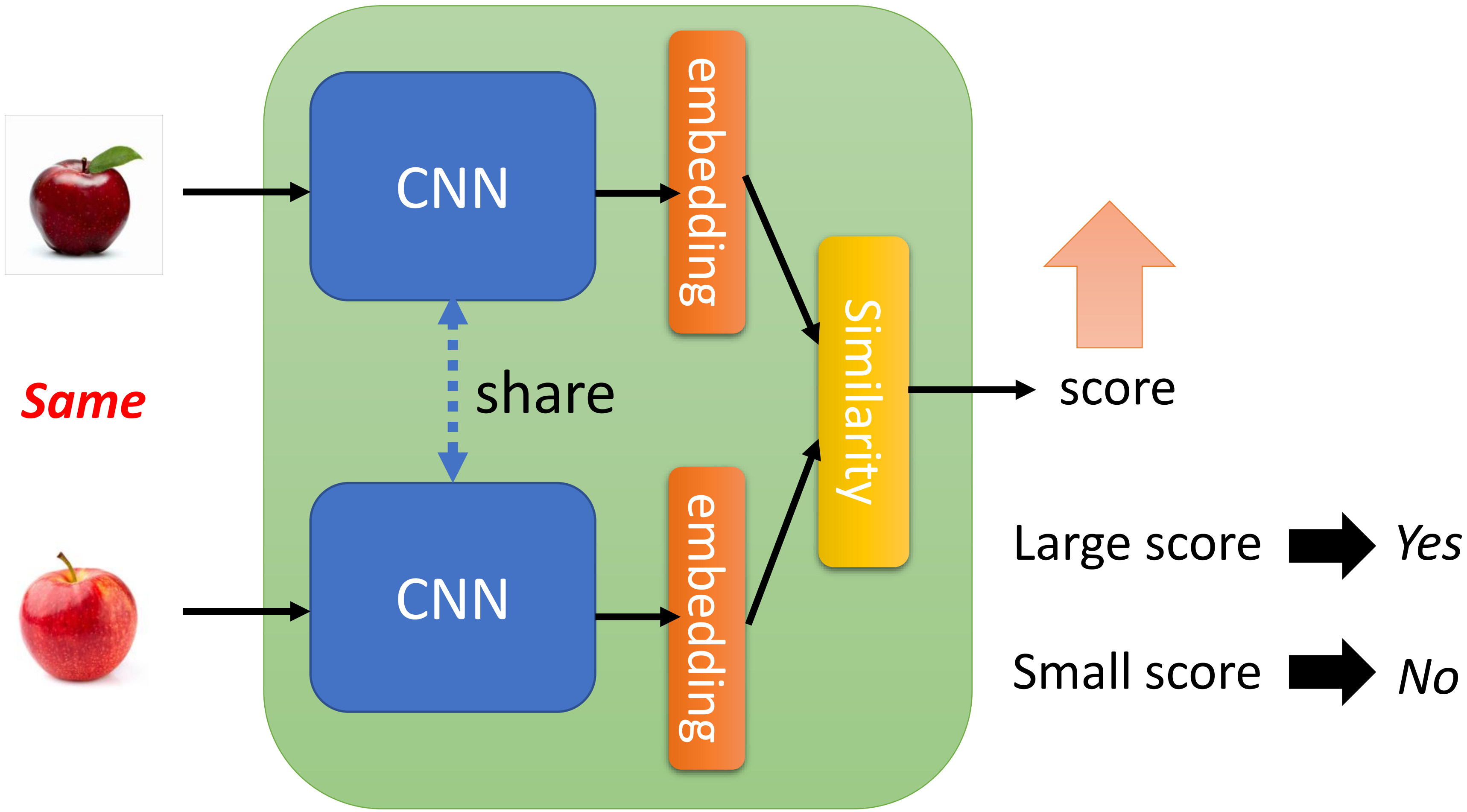
Across-task Testing

Learning to Compare

- What is the learned *learning algorithm* in this case?
- Think about <u>*non parametric models*</u> 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 <u>*trainable*</u> across tasks

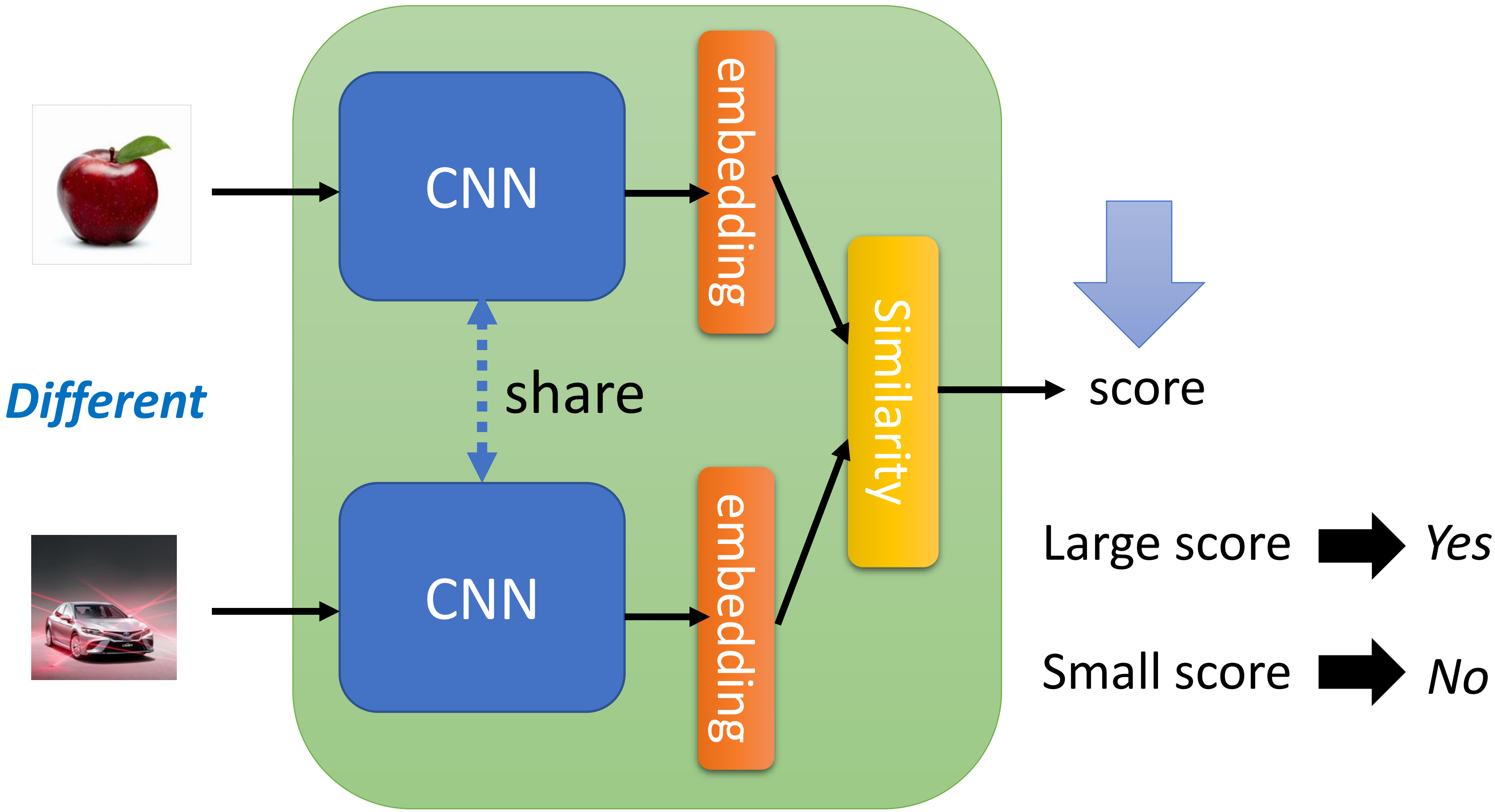
First Example: Siamese Network

Koch, Zemel, Salakhutdinov, 2015



First Example: Siamese Network

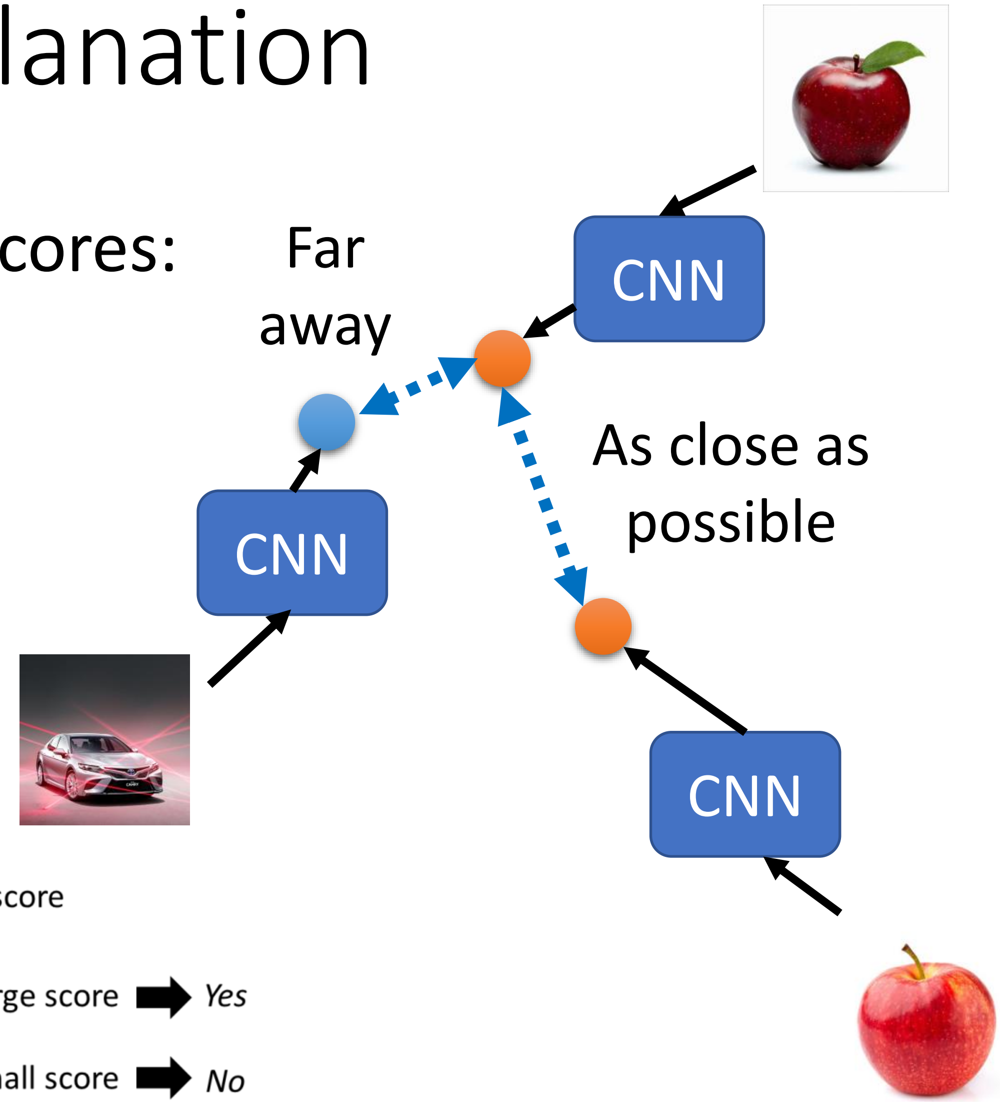
Koch, Zemel, Salakhutdinov, 2015

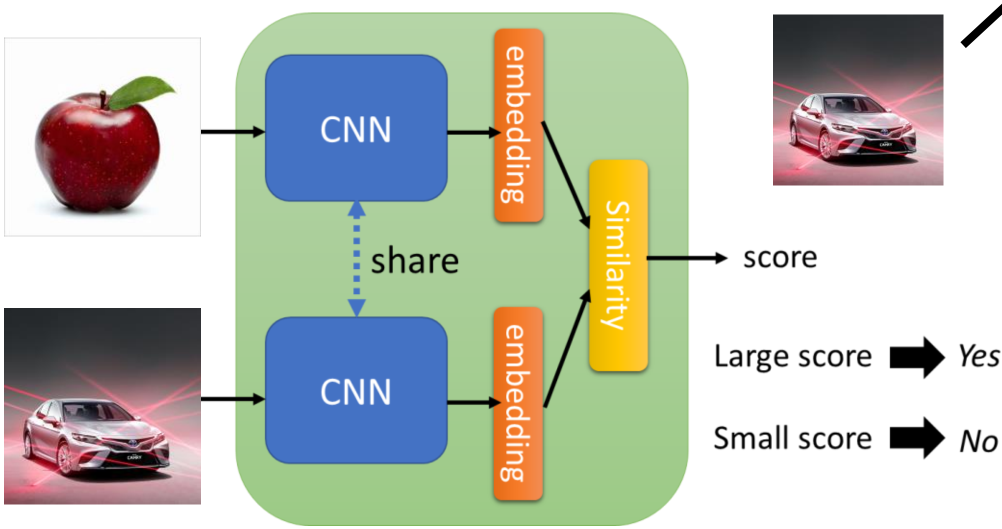


Siamese Network
- Intuitive Explanation

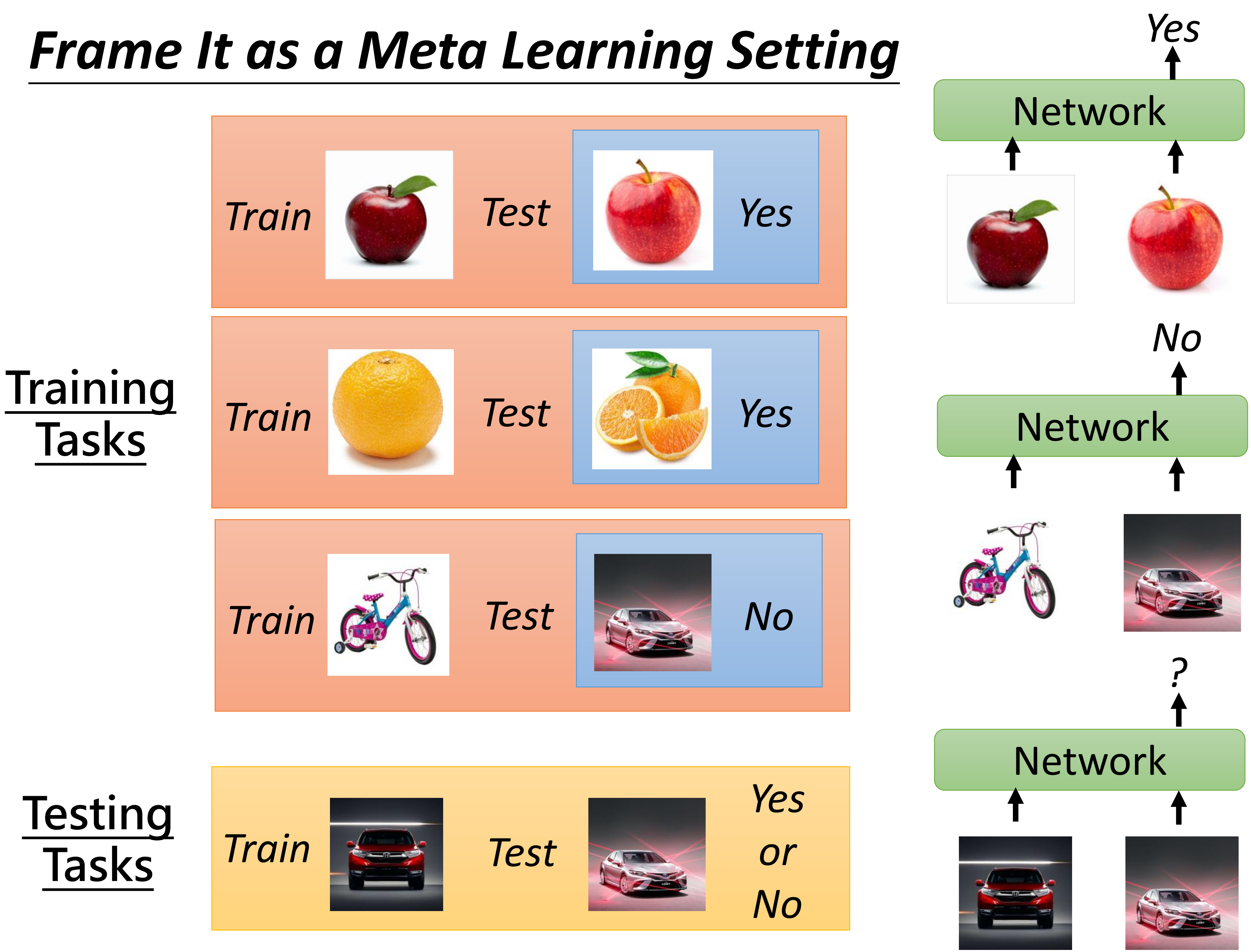
Learning the similarity scores:

- Convolutional NN
- Similarity functions



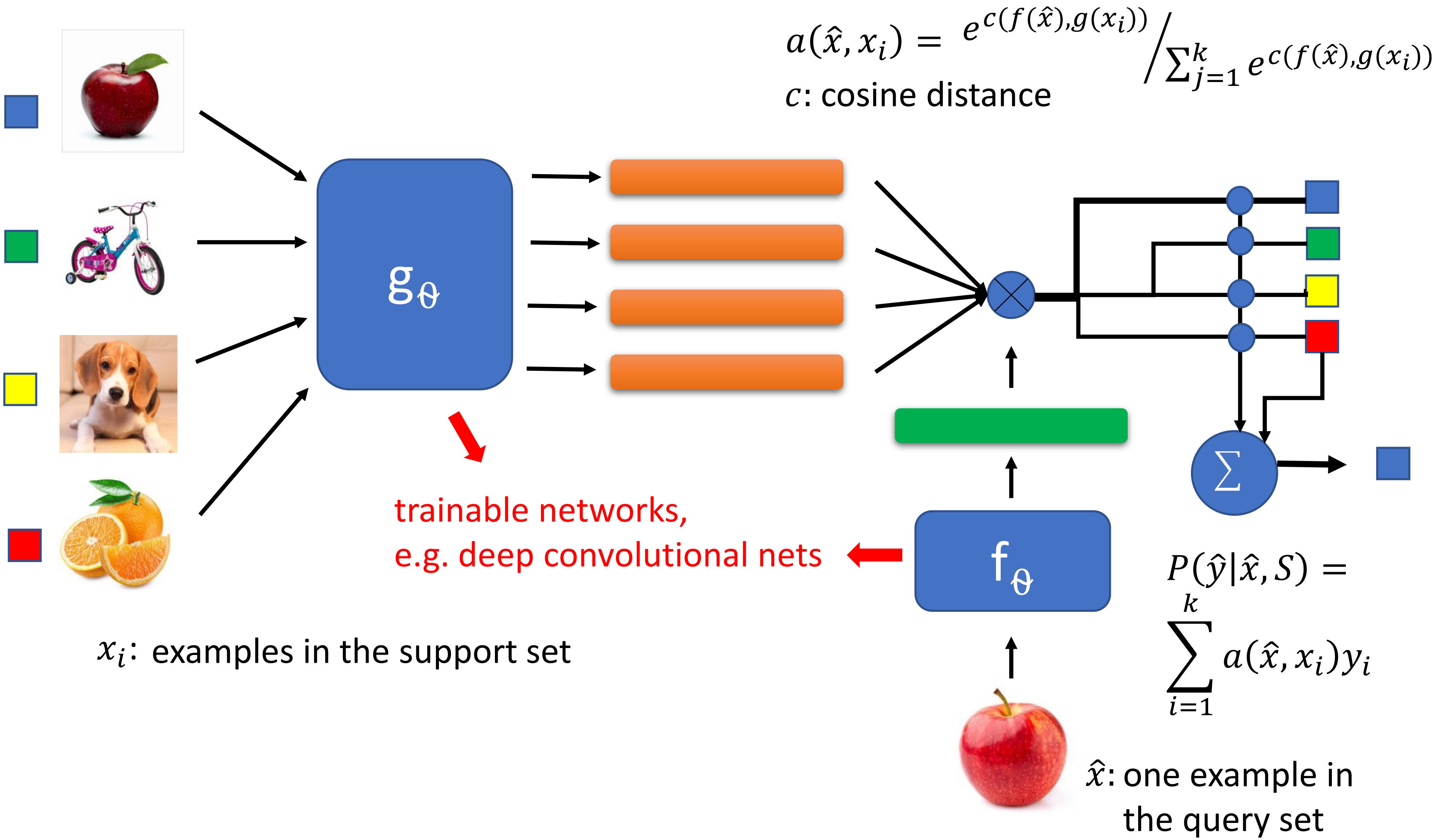


Frame It as a Meta Learning Setting



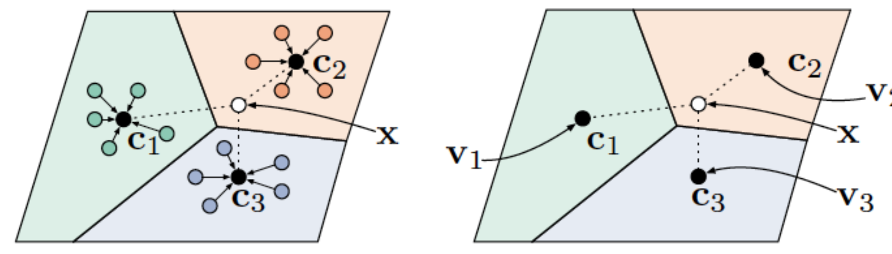
Matching Network

Vinyals, Blundell, Lillicrap, Kavukcupglu, Wierstra, 2017



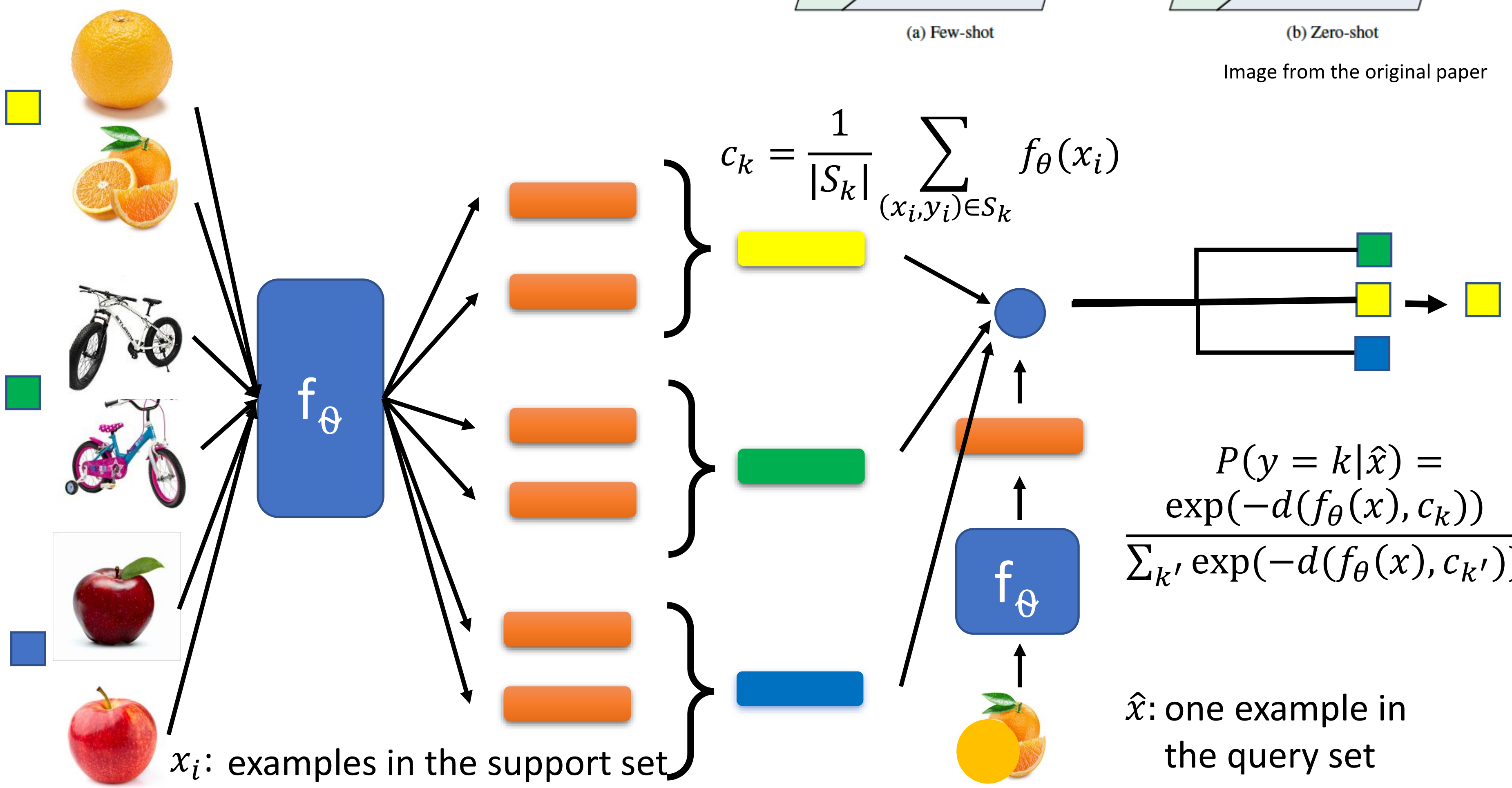
Prototypical Network

Snell, Swersky, Zemel, 2017



(a) Few-shot (b) Zero-shot

Image from the original paper



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

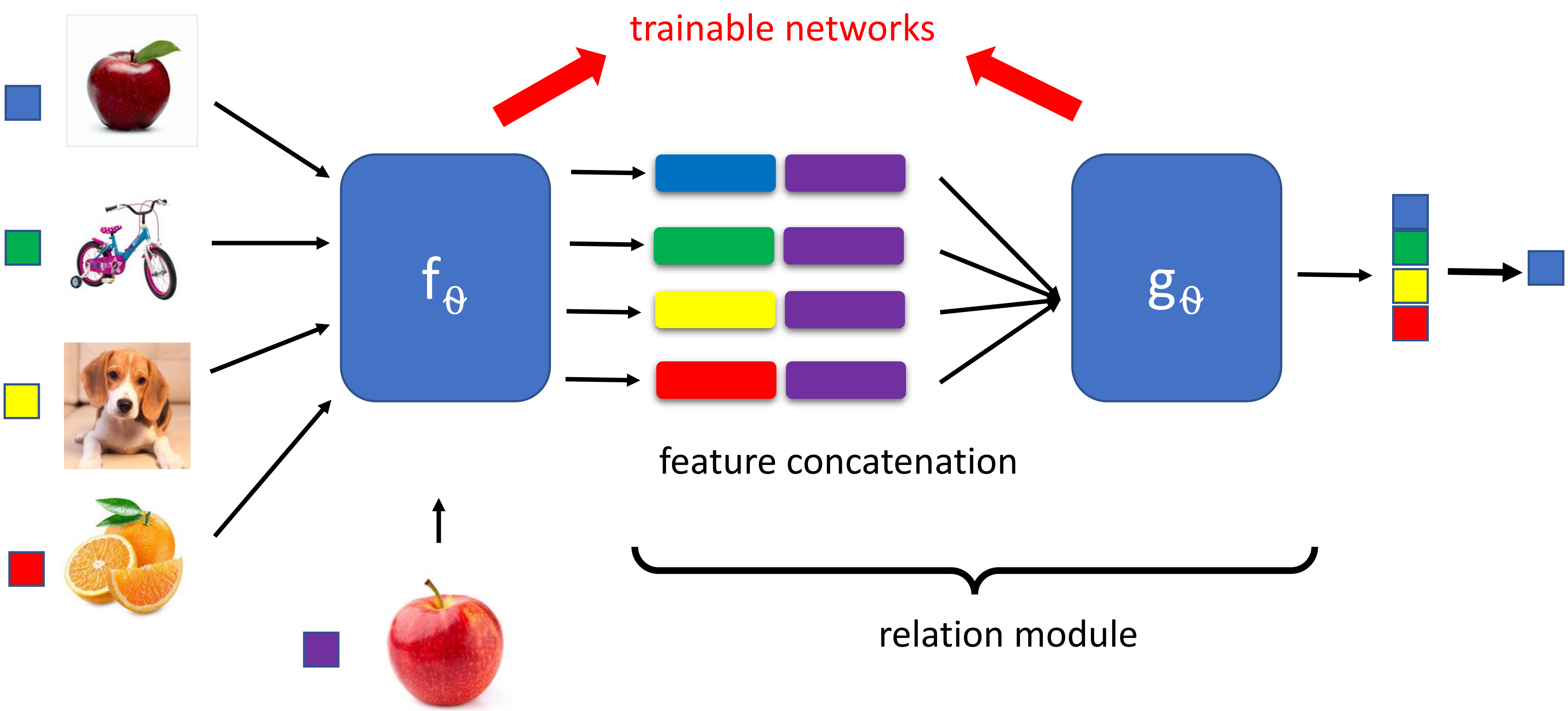
$$P(y = k | \hat{x}) = \frac{\exp(-d(f_\theta(x), c_k))}{\sum_{k'} \exp(-d(f_\theta(x), c_{k'}))}$$

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

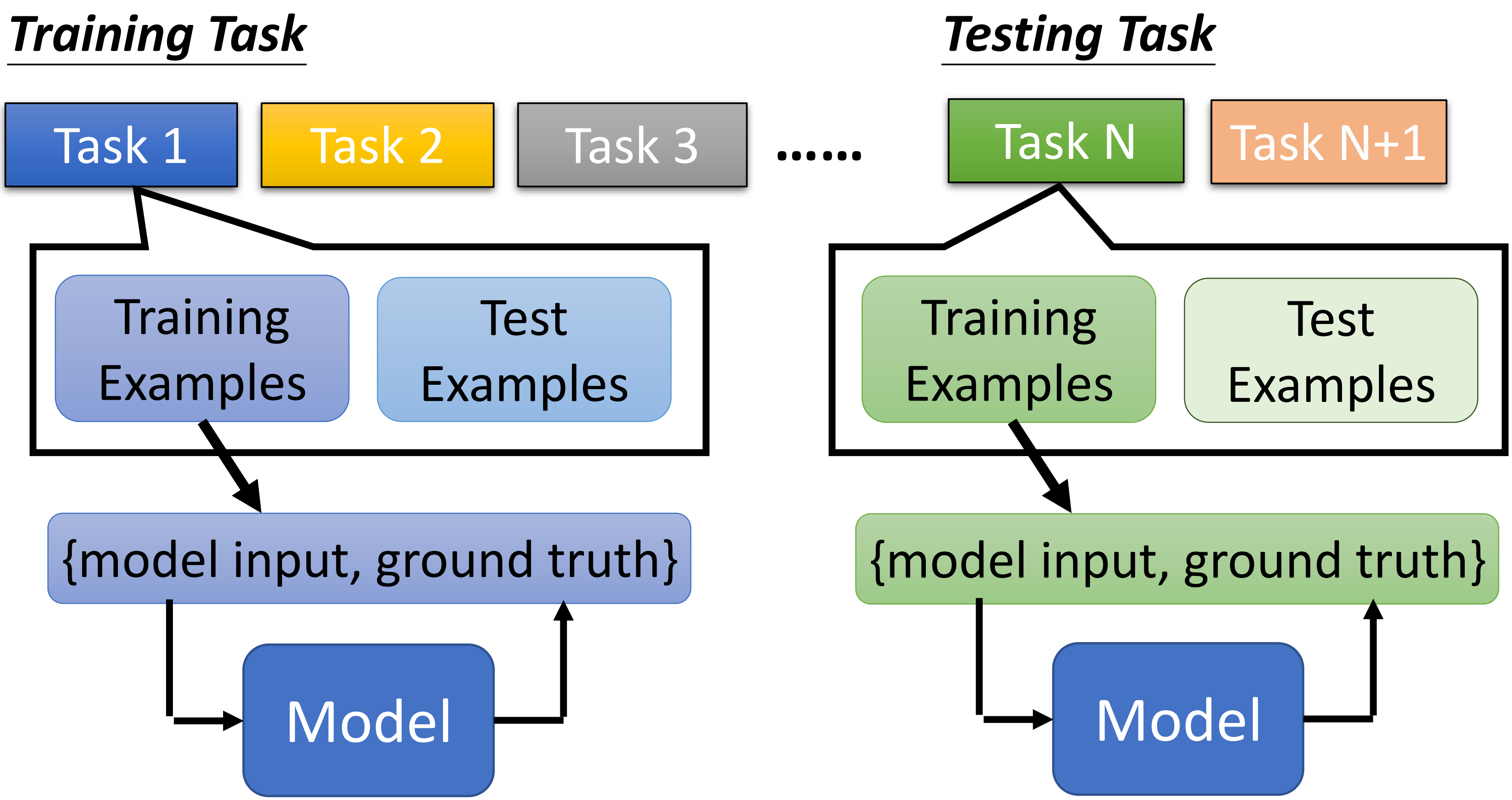
Meta Learning vs. Transfer Learning

- 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 II: Meta Learning to Human Language Processing

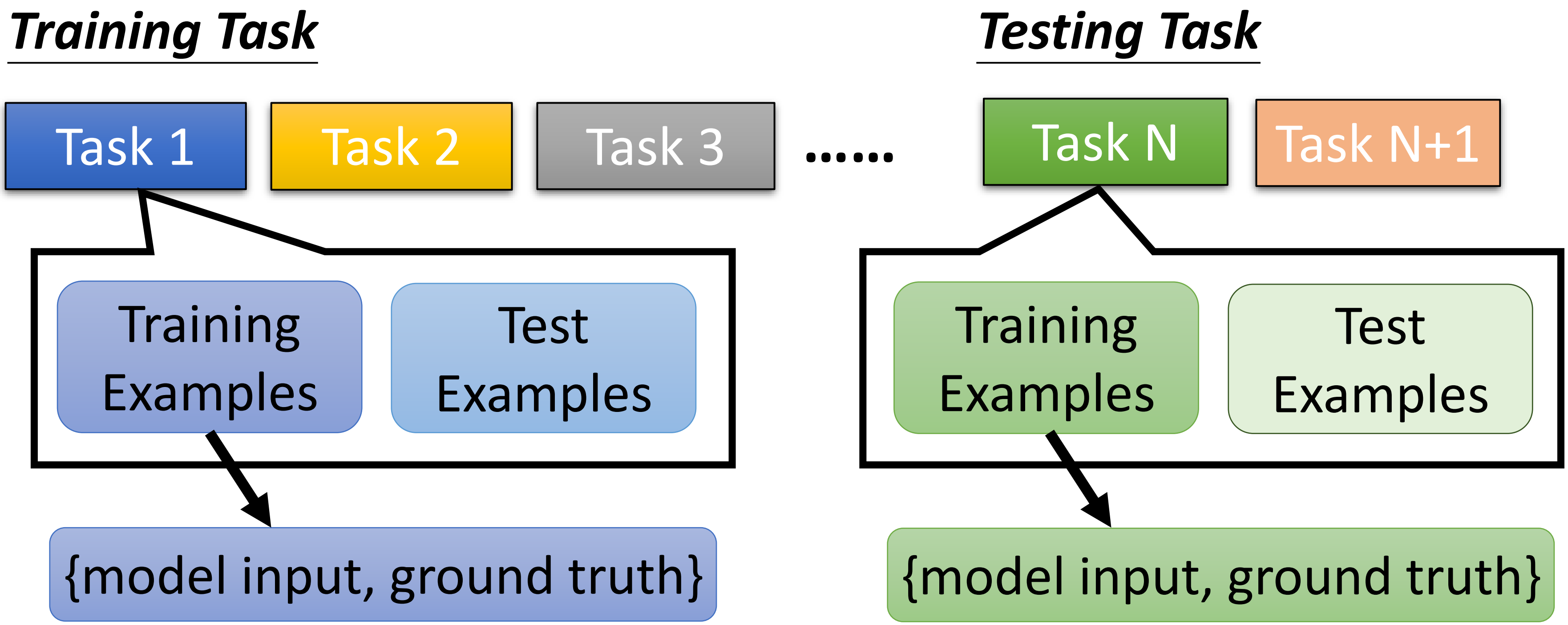
	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Framework of Meta Learning



Constraint of "learning to initialize": All the tasks must use the same model architecture.

General Questions



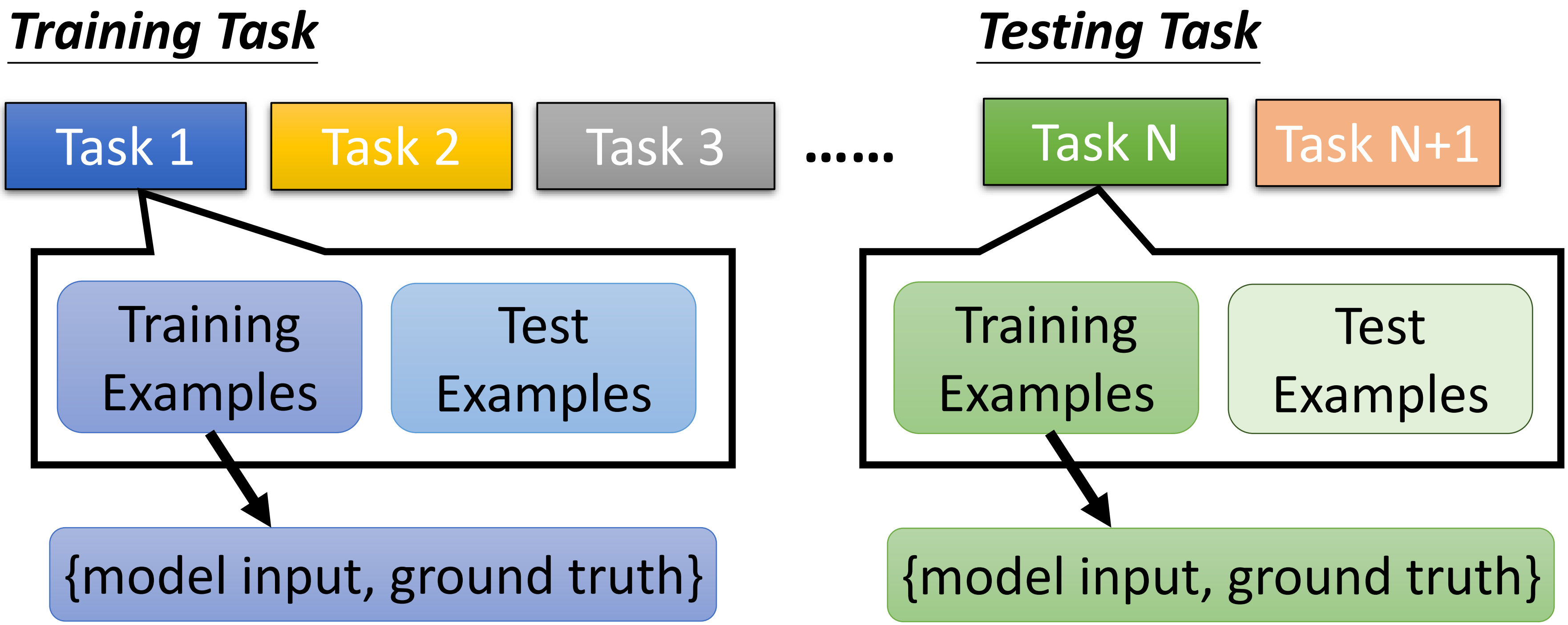
How are you

大家好啊

What if the model input of different tasks are different languages?

Simply use *Multilingual BERT*

General Questions



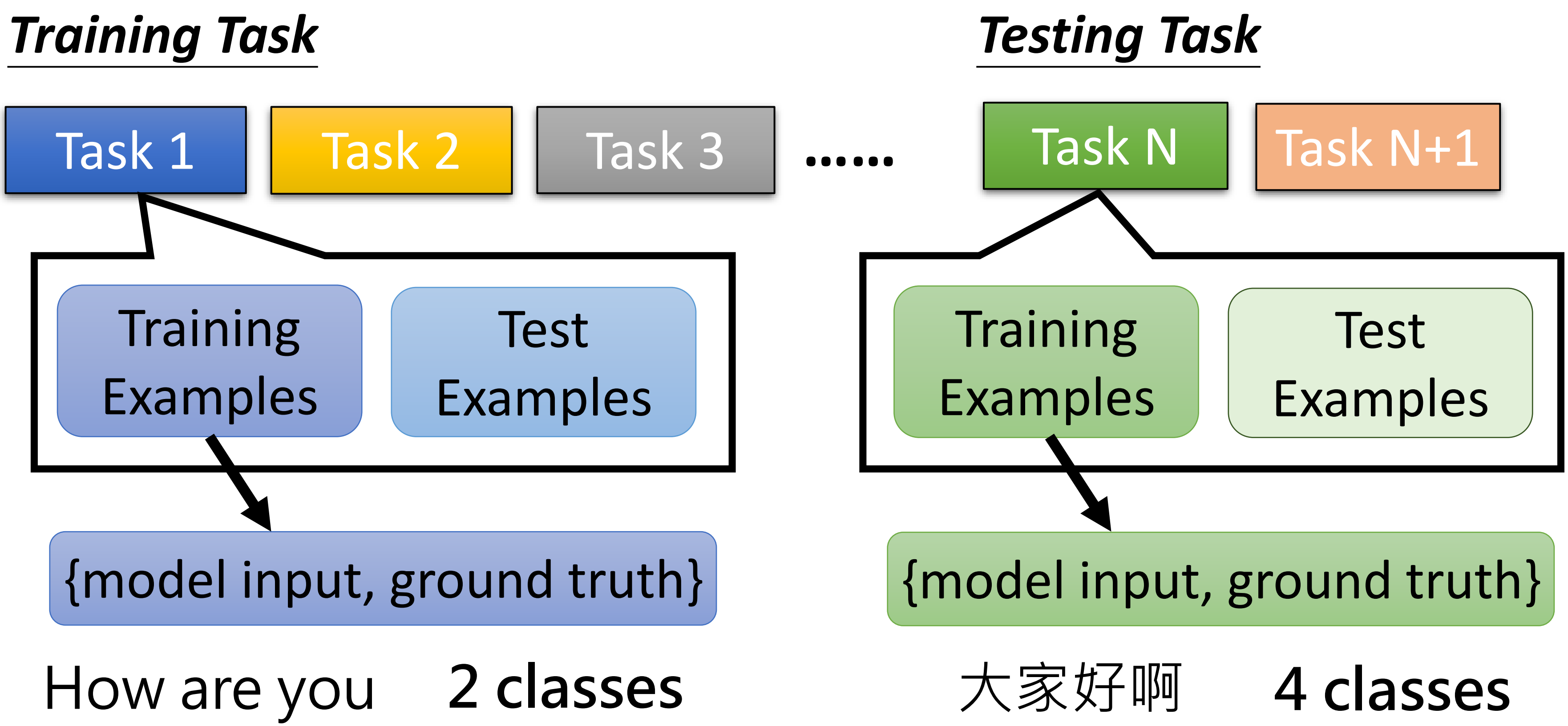
How are you

大家好啊

BERT (and its family) also find good initialization.

Q1: Do we still need "learning to initialize"?

General Questions



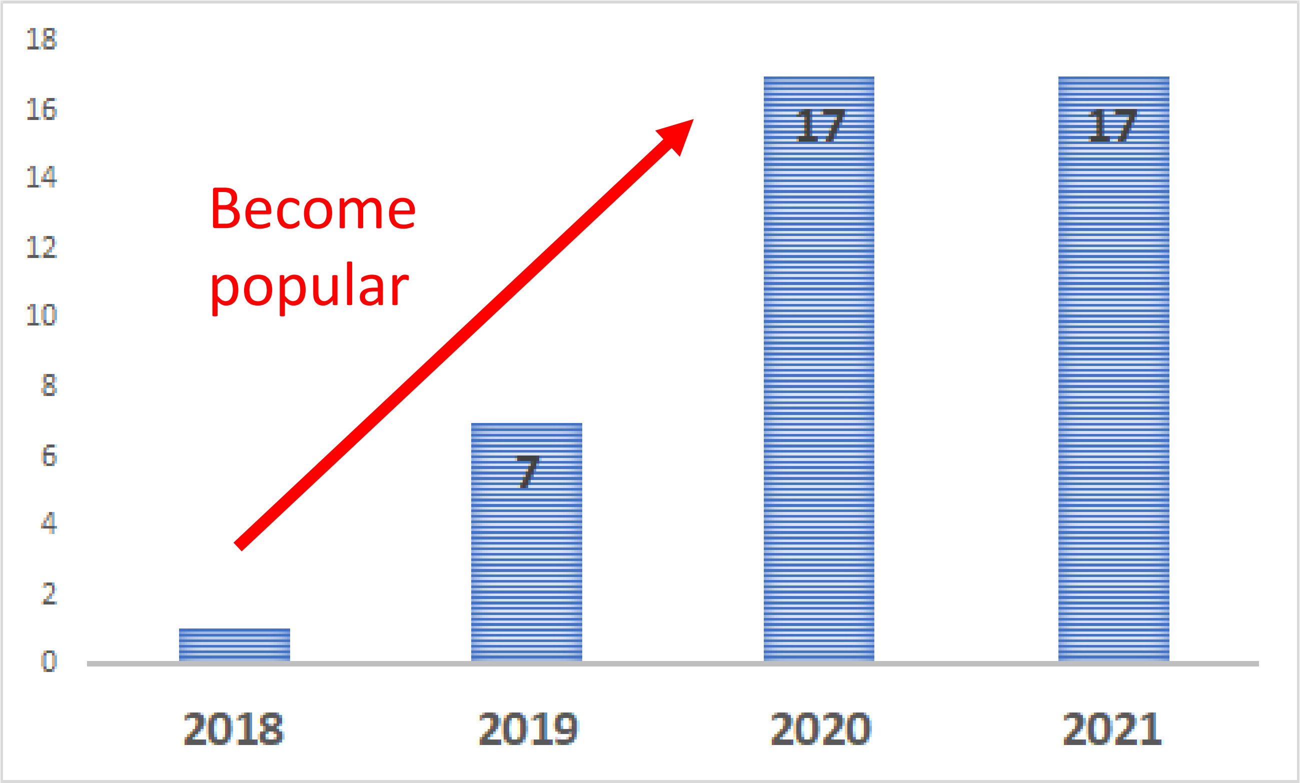
Q2:

What if different tasks have different model output space?

	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

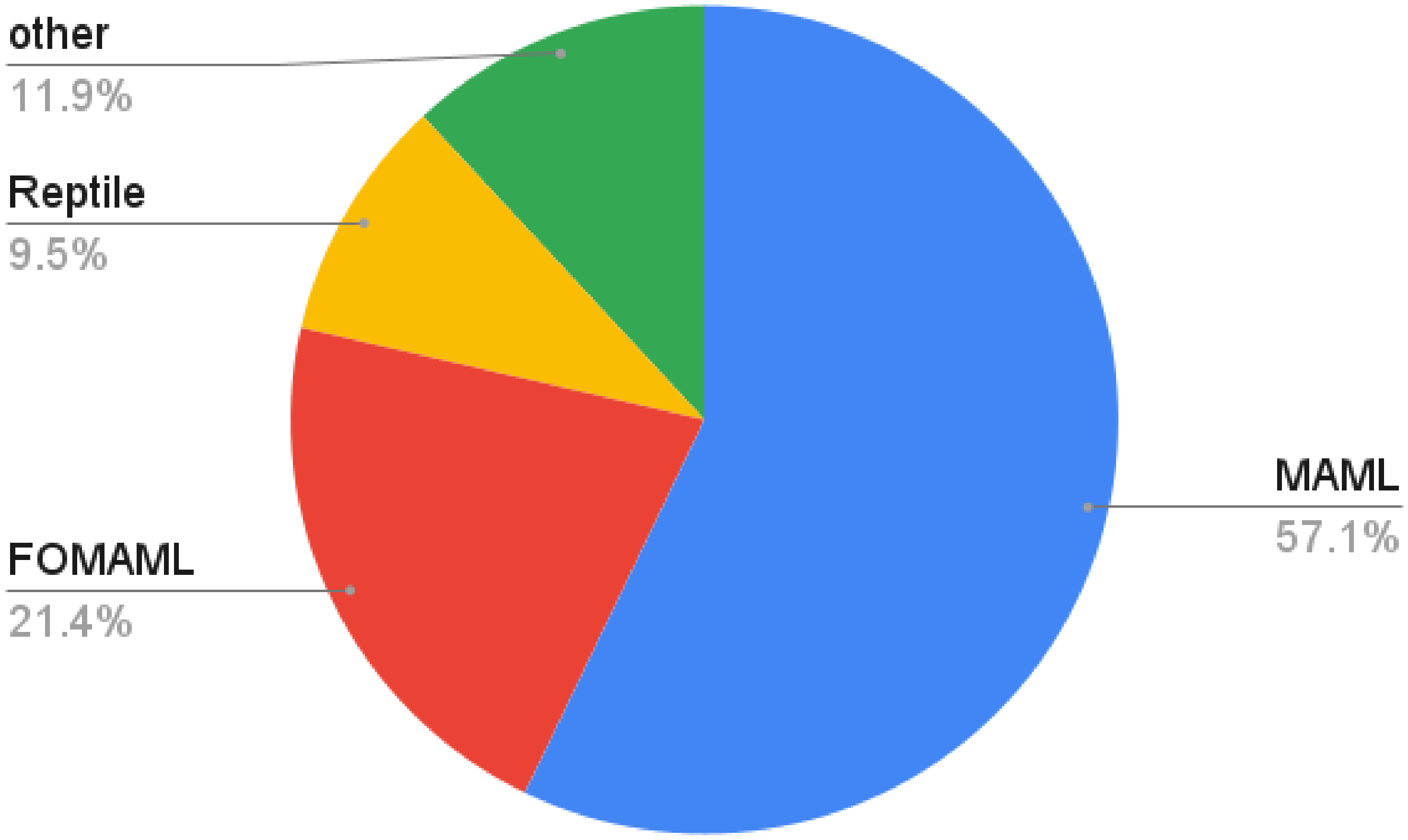
Learning to Initialize

- Go through 42 papers about learning to initialize for speech/NLP applications in the last three years



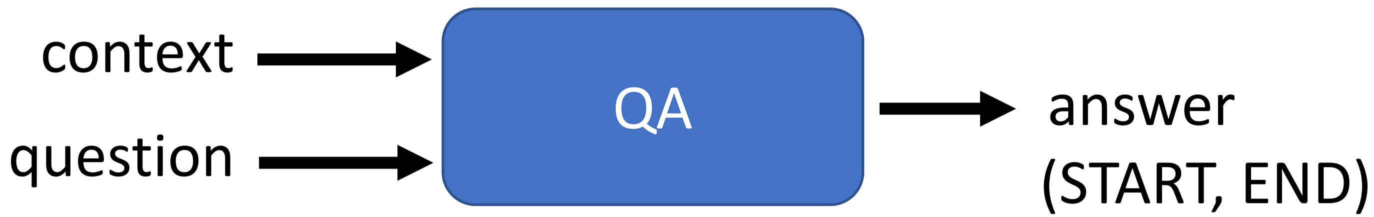
Learning to Initialize

(if a paper uses multiple approaches, we counted the one performs better.)



	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Question Answering



End-to-end models

Training Task ***Testing Task***

Usually used to improve **cross-lingual** transfer learning

Farhad Nooralahzadeh , Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein, Zero-shot cross-lingual transfer with meta learning, EMNLP, 2020

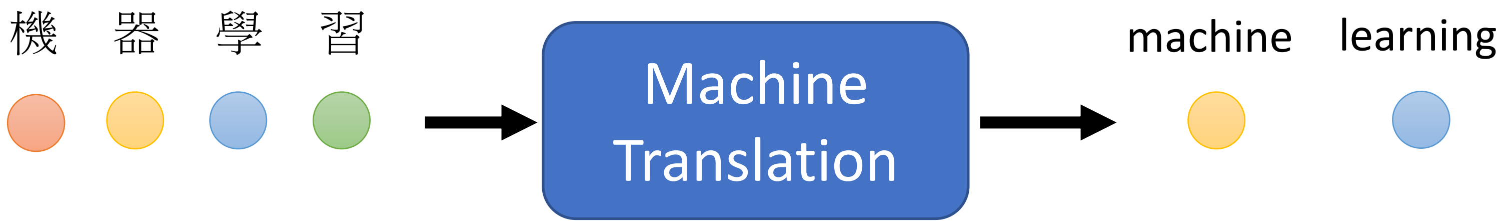
Meryem M'hamdi, Doo Soon Kim, Franck Dernoncourt, Trung Bui, Xiang Ren, and Jonathan May, X-METRAADA: Cross-lingual meta-transfer learning adaptation to natural language understanding and question answering, NAACL, 2021

Training tasks and testing tasks are different QA corpora.

Ming Yan, Hao Zhang, Di Jin, Joey Tianyi Zhou, Multi-source Meta Transfer for Low Resource Multiple-Choice Question Answering, ACL, 2020

	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Machine Translation



End-to-end models

Training Task

Fr-En | Es-En | Pt-En | ……

Testing Task

Ro-En | Fi-En

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

Training Task

Law | Movie | News | ……

Testing Task

Medical

Rumeng Li, Xun Wang, Hong Yu, MetaMT, a Meta Learning Method Leveraging Multiple Domain Data for Low Resource Machine Translation, AAAI, 2020

Machine Translation

Language 1

Language 2

Unsupervised MT
(Training with monolingual data)

Training Task

Law | Movie | News | ......

Testing Task

Medical

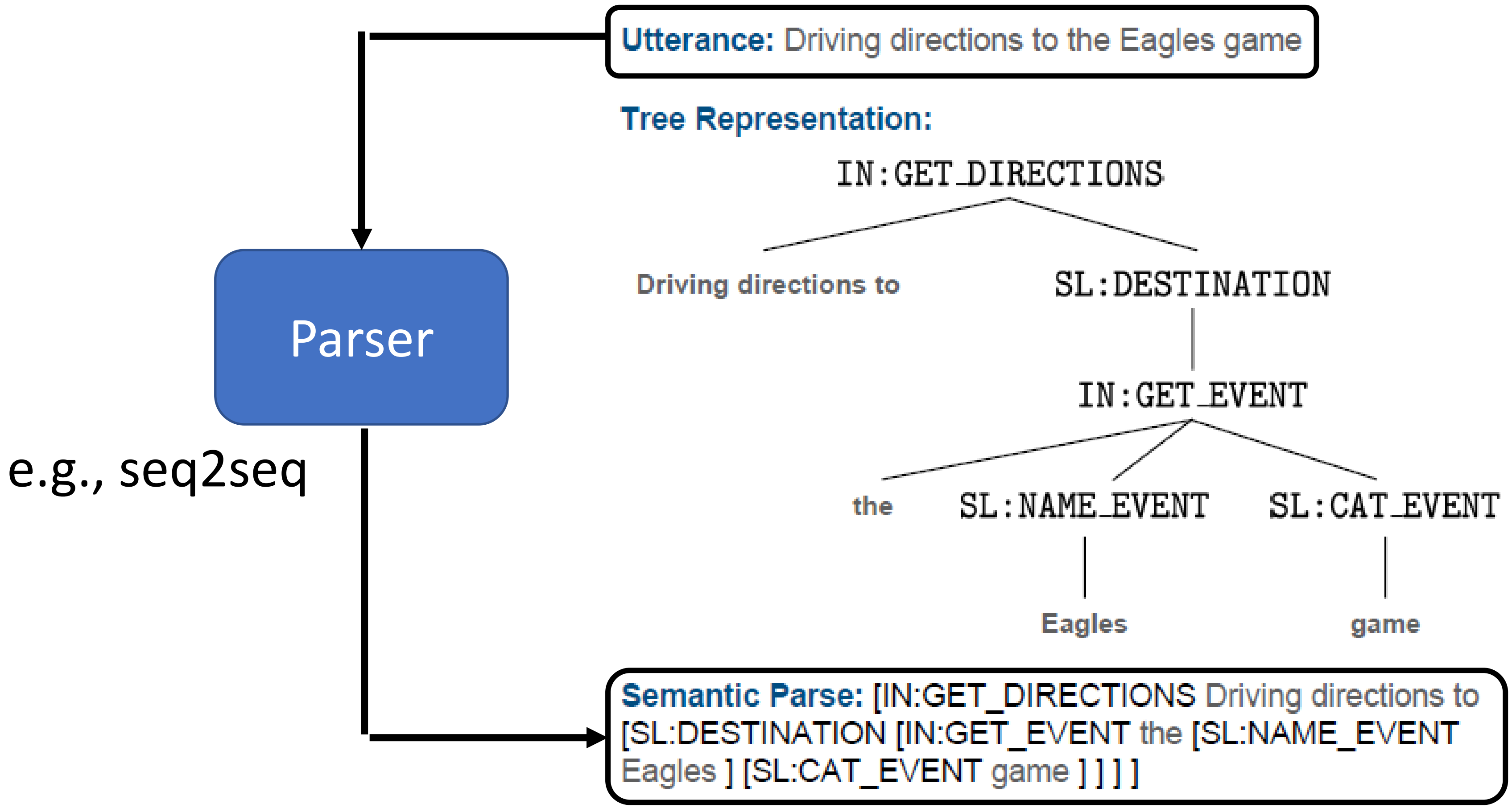
Only unparallel data in each task

Cheonbok Park, Yunwon Tae, Taehee Kim, Soyoung Yang, Mohammad Azam Khan, Eunjeong Park, Jaegul Choo, Unsupervised Neural Machine Translation for Low-Resource Domains via Meta-Learning, ACL 2021

Machine Translation	(Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
Parsing	(Guo et al., 2019) (Huang et al., 2018) (Langedijk et al., 2021) (Chen et al., 2020a) (Wang et al., 2021a)		
Dialogue	(Qian and Yu, 2019) (Madotto et al., 2019) (Mi et al., 2019) (Huang et al., 2020a) (Dingliwal et al., 2021) (Qian et al., 2021) (Dai et al., 2020) (Huang et al., 2020b)		Learning to optimize: (Chien and Lieow, 2019)
Speech Recognition	(Hsu et al., 2020) (Klejch et al., 2019) (Winata et al., 2020a) (Winata et al., 2020b) (Xiao et al., 2021)	(Lux and Vu, 2021)	Learning to optimize: (Klejch et al., 2018) Network architecture search: (Chen et al., 2020c) (Baruwa et al., 2019)
Source Separation	(Wu et al., 2021)		
Keyword Spotting	(Chen et al., 2020b)		Network architecture search: (Mazzawi et al., 2019)
Sound Event Detection		(Shimada et al., 2020) (Chou et al., 2019)	
Voice Cloning			Learning the learning algorithm: (Chen et al., 2019b) (Serrà et al., 2019)
Multi-tasks			Learning to select data: (Tarunesh et al., 2021)
Multi-modal		(Eloff et al., 2019)	Learning the learning algorithm: (Surís et al., 2019) (Xu et al., 2021a)

Parsing

- Example: task-oriented semantic parsing



Parsing

Training Task ***Testing Task***

German | Tamil | French | ...... | Kazakh

Anna Langedijk, Verna Dankers, Phillip Lippe, Sander Bos, Bryan Cardenas Guevara, Helen Yannakoudakis, Ekaterina Shutova, Meta-learning for fast cross-lingual adaptation in dependency parsing, arXiv, 2021

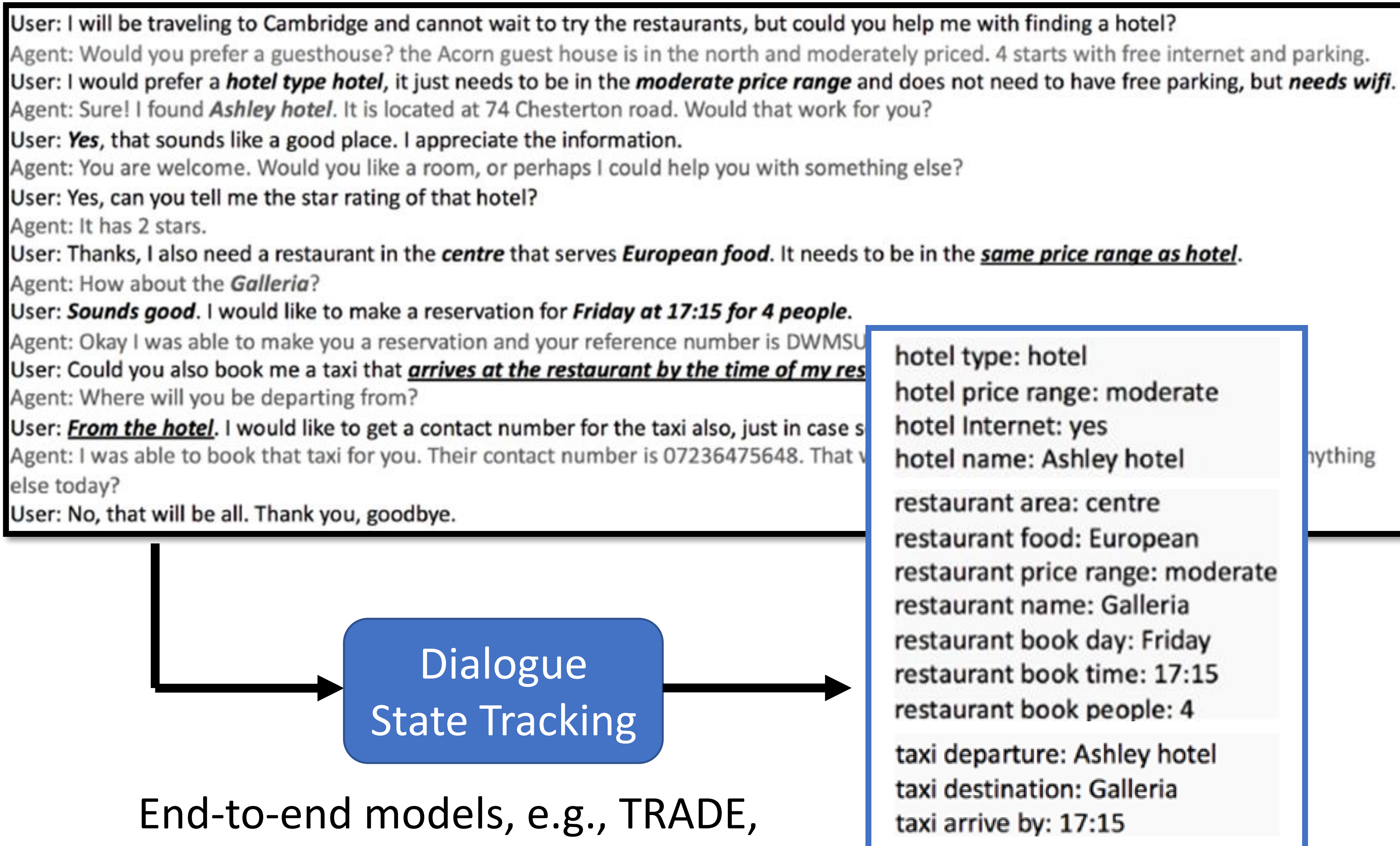
Training Task ***Testing Task***

Alarm | Music | Timer | ...... | Reminder

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, Sonal Gupta, Low-Resource Domain Adaptation for Compositional Task-Oriented Semantic Parsing, EMNLP, 2020

There is a trend to generate task automatically in semantic parsing (Part III).

Dialogue State Tracking



End-to-end models, e.g., TRADE, DST QA, Simple TOD, etc.

State

Dialogue State Tracking

User

A

User

Training Task

Restaurant | Hotel | Train

Testing Task

Taxi | Attraction

Yi Huang, Junlan Feng, Min Hu, Xiaoting Wu, Xiaoyu Du, Shuo Ma, Meta-Reinforced Multi-Domain State Generator for Dialogue Systems, ACL, 2020

Lingxiao Wang, Kevin Huang, Tengyu Ma, Quanquan Gu, Jing Huang, Variance-reduced First-order Meta-learning for Natural Language Processing Tasks, NAACL, 2021

Saket Dingliwal, Bill Gao, Sanchit Agarwal, Chien-Wei Lin, Tagyoung Chung, Dilek Hakkani-Tur, Few Shot Dialogue State Tracking using Meta-learning, EACL, 2021

Dialogue State Tracking

restaurant food: European
restaurant price range: moderate
restaurant name: Galleria
restaurant book day: Friday
restaurant book time: 17:15
restaurant book people: 4

taxi departure: Ashley hotel
taxi destination: Galleria
taxi arrive by: 17:15

End-to-end models, e.g., TRADE, DST QA, Simple TOD, etc.

State

Task-oriented Dialogue / Chatbot

End-to-end Task-oriented Dialogue: Training and testing tasks are different domains.

Kun Qian and Zhou Yu, Domain adaptive dialog generation via meta learning, ACL 2019

Kun Qian, Wei Wei, Zhou Yu, A Student-Teacher Architecture for Dialog Domain Adaptation under the Meta-Learning Setting, AAAI 2021

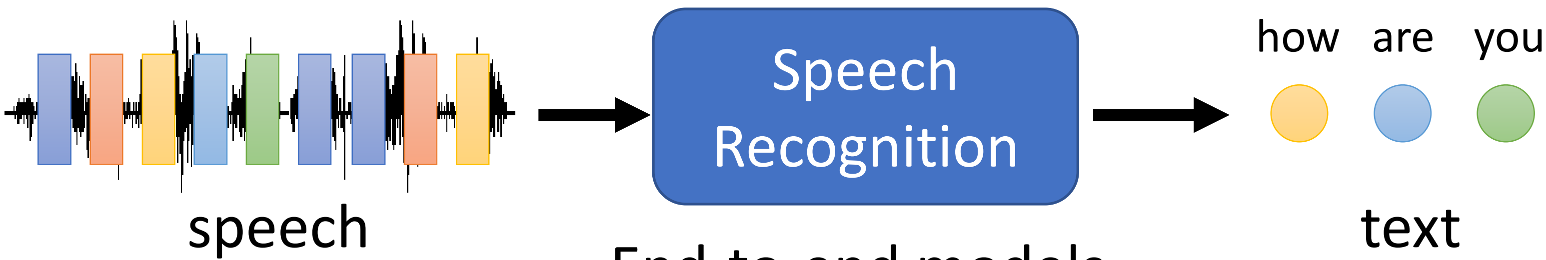
Yinpei Dai, Hangyu Li, Chengguang Tang, Yongbin Li, Jian Sun, Xiaodan Zhu, Learning Low-Resource End-To-End Goal-Oriented Dialog for Fast and Reliable System Deployment, ACL, 2020

End-to-end Chatbot: Training and testing tasks are different personas.

Zhaojiang Lin, Andrea Madotto, Chien-Sheng Wu, Pascale Fung, Personalizing Dialogue Agents via Meta-Learning, ACL, 2019

Machine Translation	(Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
Parsing	(Guo et al., 2019) (Huang et al., 2018) (Langedijk et al., 2021) (Chen et al., 2020a) (Wang et al., 2021a)		
Dialogue	(Qian and Yu, 2019) (Madotto et al., 2019) (Mi et al., 2019) (Huang et al., 2020a) (Dingliwal et al., 2021) (Qian et al., 2021) (Dai et al., 2020) (Huang et al., 2020b)		Learning to optimize: (Chien and Lieow, 2019)
Speech Recognition	(Hsu et al., 2020) (Klejch et al., 2019) (Winata et al., 2020a) (Winata et al., 2020b) (Xiao et al., 2021)	(Lux and Vu, 2021)	Learning to optimize: (Klejch et al., 2018) Network architecture search: (Chen et al., 2020c) (Baruwa et al., 2019)
Source Separation	(Wu et al., 2021)		
Keyword Spotting	(Chen et al., 2020b)		Network architecture search: (Mazzawi et al., 2019)
Sound Event Detection		(Shimada et al., 2020) (Chou et al., 2019)	
Voice Cloning			Learning the learning algorithm: (Chen et al., 2019b) (Serrà et al., 2019)
Multi-tasks			Learning to select data: (Tarunesh et al., 2021)
Multi-modal		(Eloff et al., 2019)	Learning the learning algorithm: (Surís et al., 2019) (Xu et al., 2021a)

Speech Recognition



End-to-end models,
e.g., seq2seq, CTC

Training Task

Bengali | Zulu | Tagalog | ……

a set of languages

Testing Task

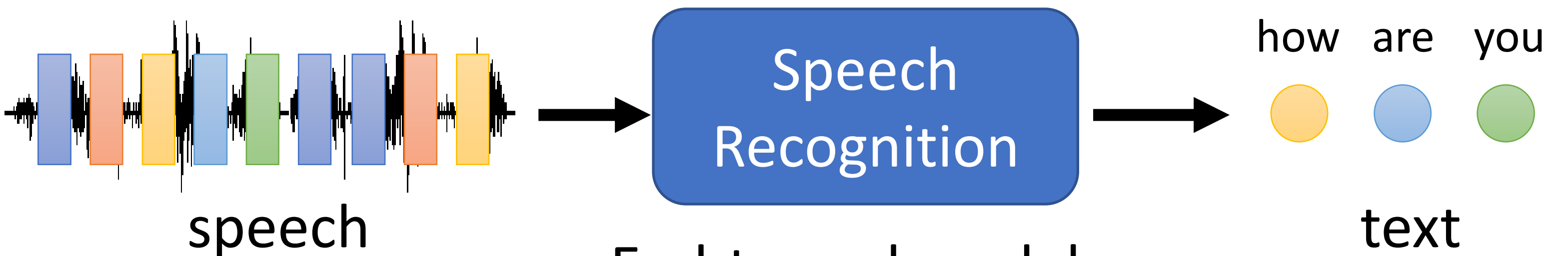
Tamil

new languages

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

Yubei Xiao, Ke Gong, Pan Zhou, Guolin Zheng, Xiaodan Liang, Liang Lin, Adversarial Meta Sampling for Multilingual Low-Resource Speech Recognition, AAAI 2021

Speech Recognition



End-to-end models,
e.g., seq2seq, CTC

Training Task

England | India | Canada ……

a set of English accents

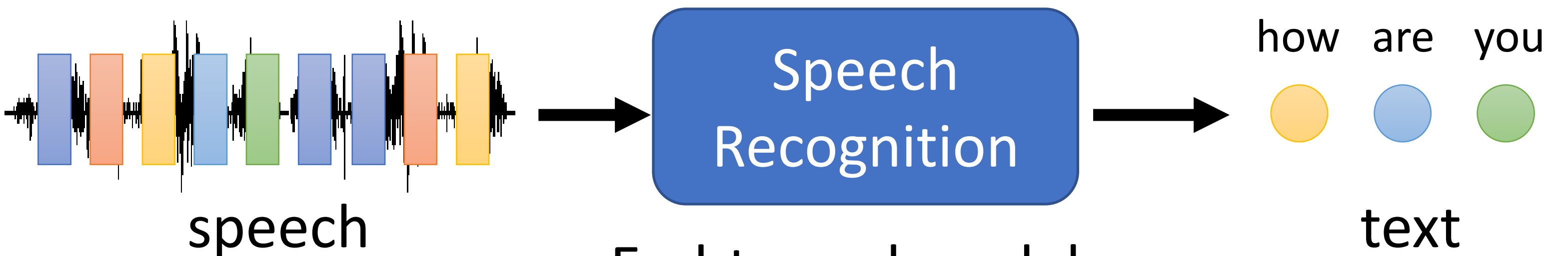
Testing Task

Philippines

new accent

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

Speech Recognition



End-to-end models,
e.g., seq2seq, CTC

Training Task

Speaker 1 | Speaker 2 | Speaker 3 | ……

Testing Task

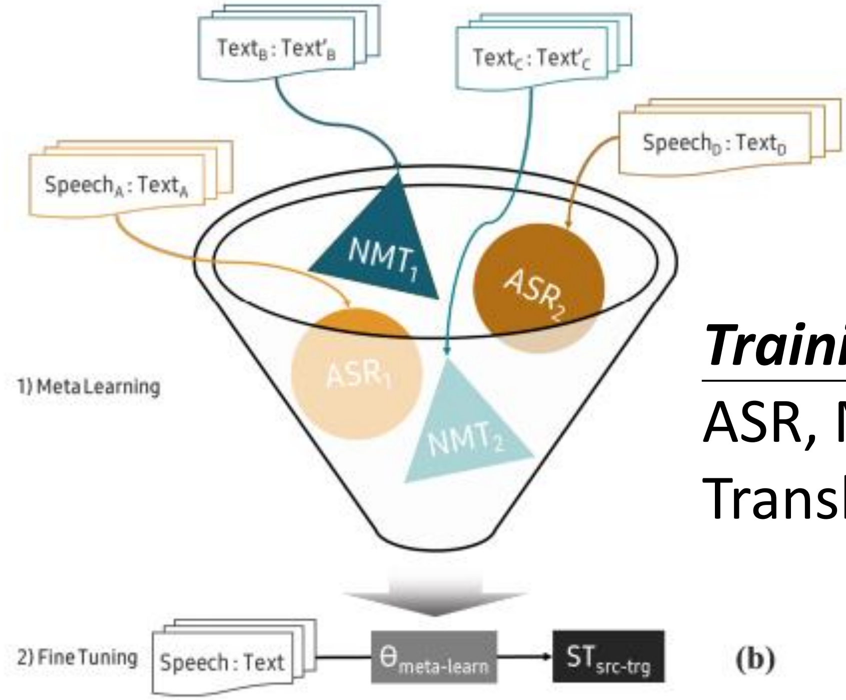
Speaker X

Speaker Adaptive Training?

Yes. New approaches for speaker adaptive training.

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

More ……



Speech Translation

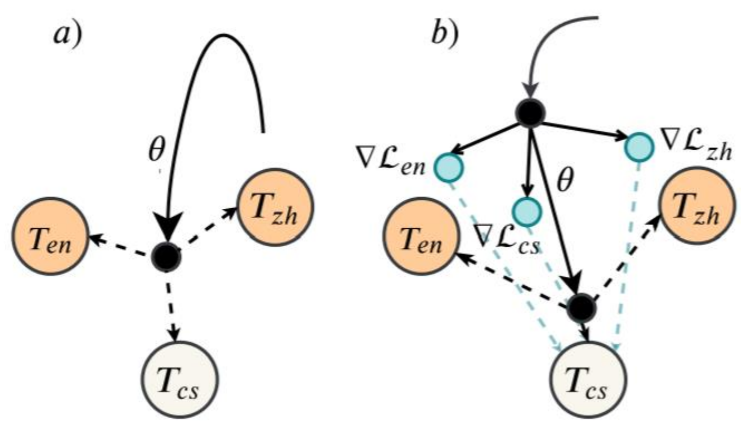
Sathish Indurthi, et al., Data Efficient Direct Speech-to-Text Translation with Modality Agnostic Meta-Learning, ICASSP 2020

Training Task: ASR, Machine Translation

Testing Task: Speech Translation

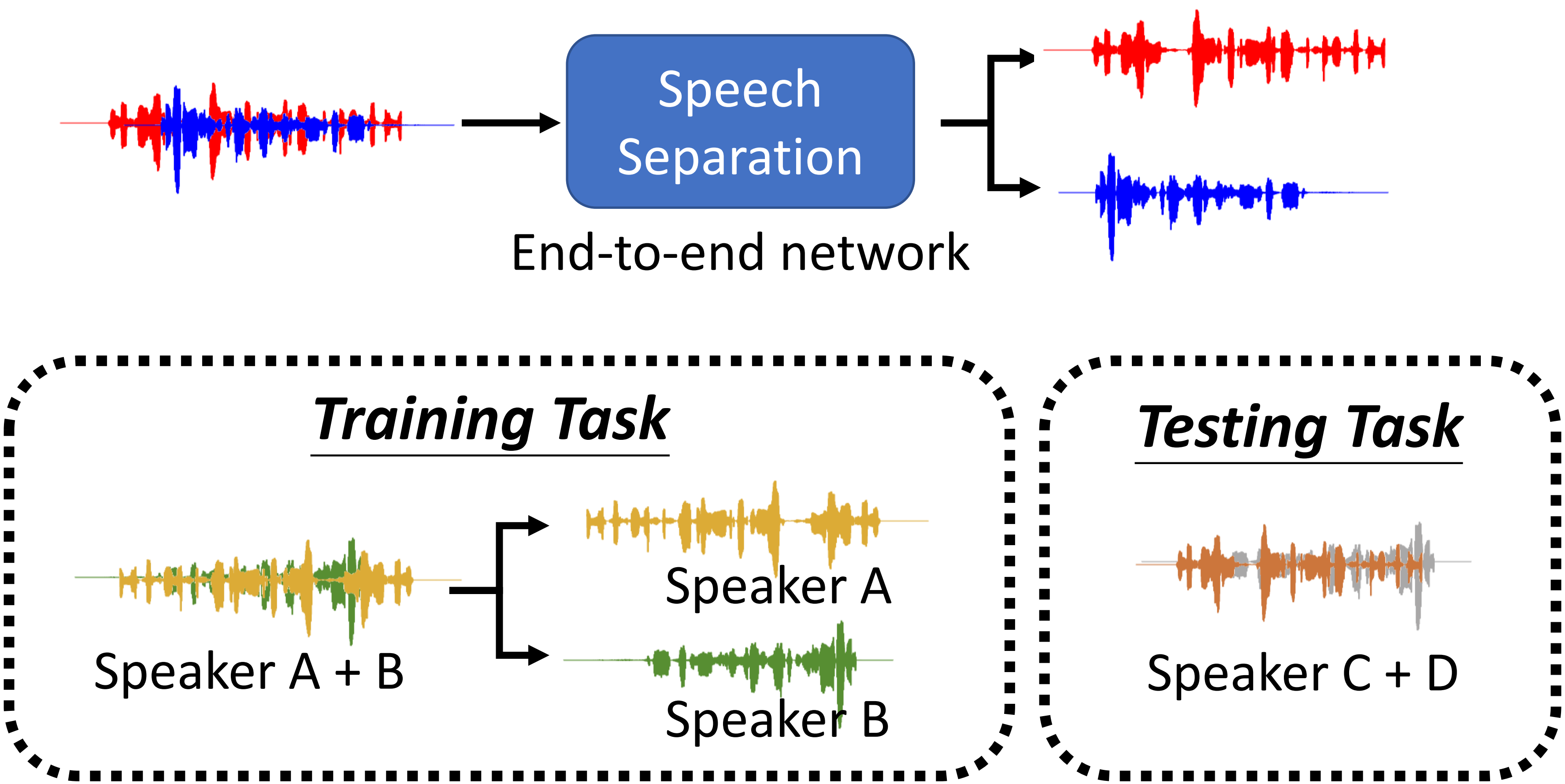
Code Switching

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



Machine Translation	(Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
Parsing	(Guo et al., 2019) (Huang et al., 2018) (Langedijk et al., 2021) (Chen et al., 2020a) (Wang et al., 2021a)		
Dialogue	(Qian and Yu, 2019) (Madotto et al., 2019) (Mi et al., 2019) (Huang et al., 2020a) (Dingliwal et al., 2021) (Qian et al., 2021) (Dai et al., 2020) (Huang et al., 2020b)		Learning to optimize: (Chien and Lieow, 2019)
Speech Recognition	(Hsu et al., 2020) (Klejch et al., 2019) (Winata et al., 2020a) (Winata et al., 2020b) (Xiao et al., 2021)	(Lux and Vu, 2021)	Learning to optimize: (Klejch et al., 2018) Network architecture search: (Chen et al., 2020c) (Baruwa et al., 2019)
Source Separation	(Wu et al., 2021)		
Keyword Spotting	(Chen et al., 2020b)		Network architecture search: (Mazzawi et al., 2019)
Sound Event Detection		(Shimada et al., 2020) (Chou et al., 2019)	
Voice Cloning			Learning the learning algorithm: (Chen et al., 2019b) (Serrà et al., 2019)
Multi-tasks			Learning to select data: (Tarunesh et al., 2021)
Multi-modal		(Eloff et al., 2019)	Learning the learning algorithm: (Surís et al., 2019) (Xu et al., 2021a)

Speech Separation



Yuan-Kuei Wu, Kuan-Po Huang, Yu Tsao, Hung-yi Lee, One Shot Learning for Speech Separation, ICASSP, 2021

	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Question 1: Learn to Init vs. BERT

Learn to Init
(MAML family)

v.s.

Self-supervised
Learning
(Sesame Street)

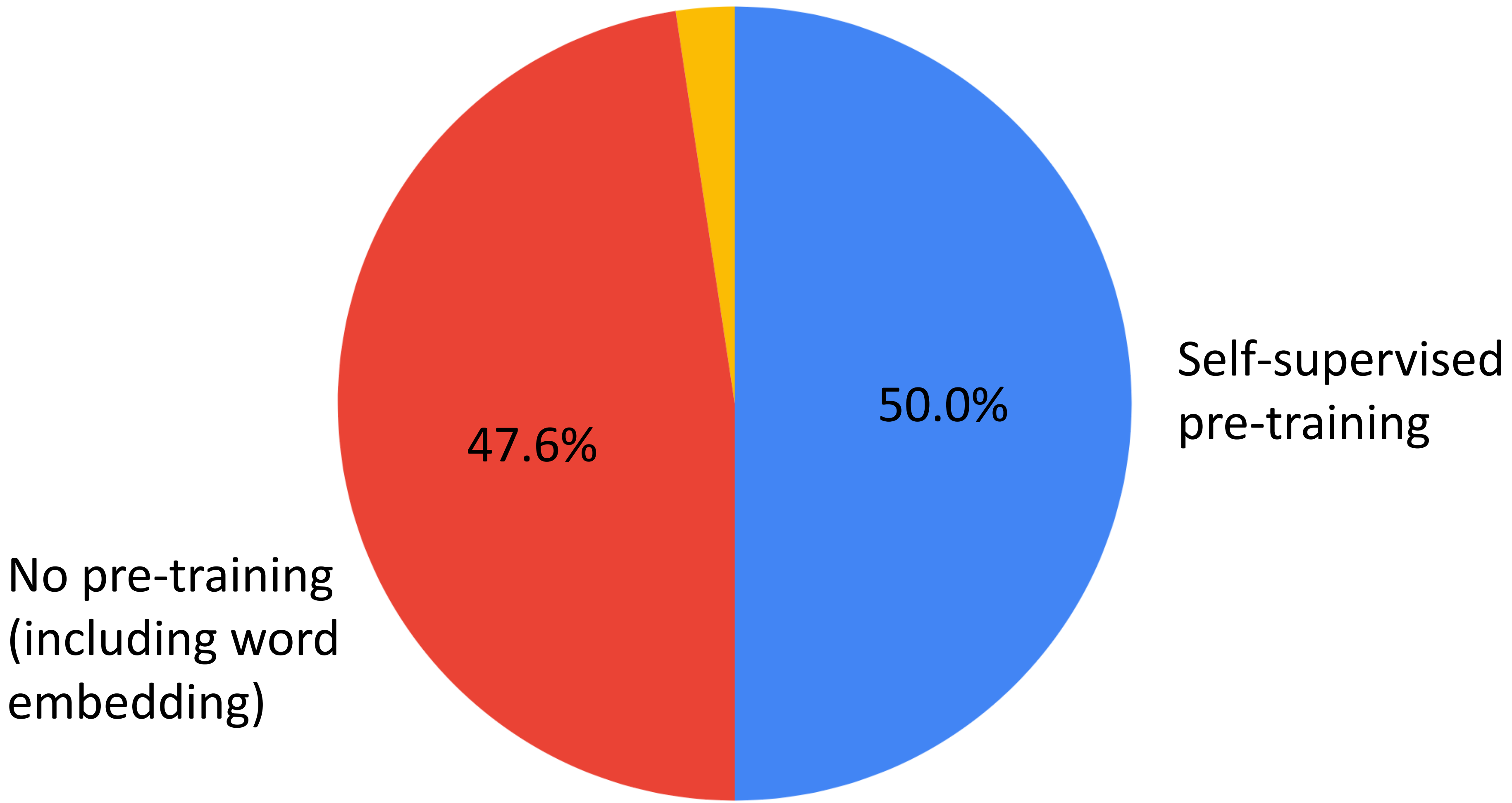
Question 1: Learn to Init vs. BERT

Turtles all the way down?

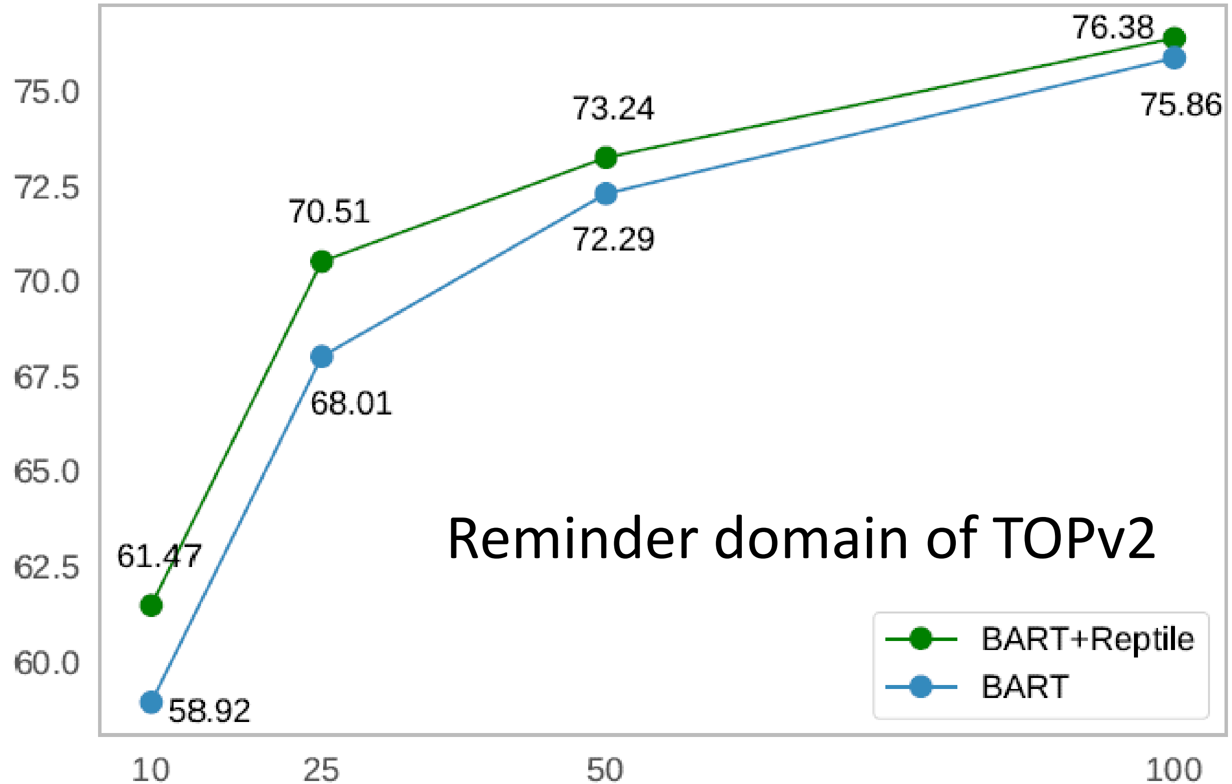
- MAML learns the initialization parameter ϕ by gradient descent
- What is the initialization parameter ϕ^0 for ϕ?

BERT can serve as ϕ^0

Question 1: Learn to Init vs. BERT



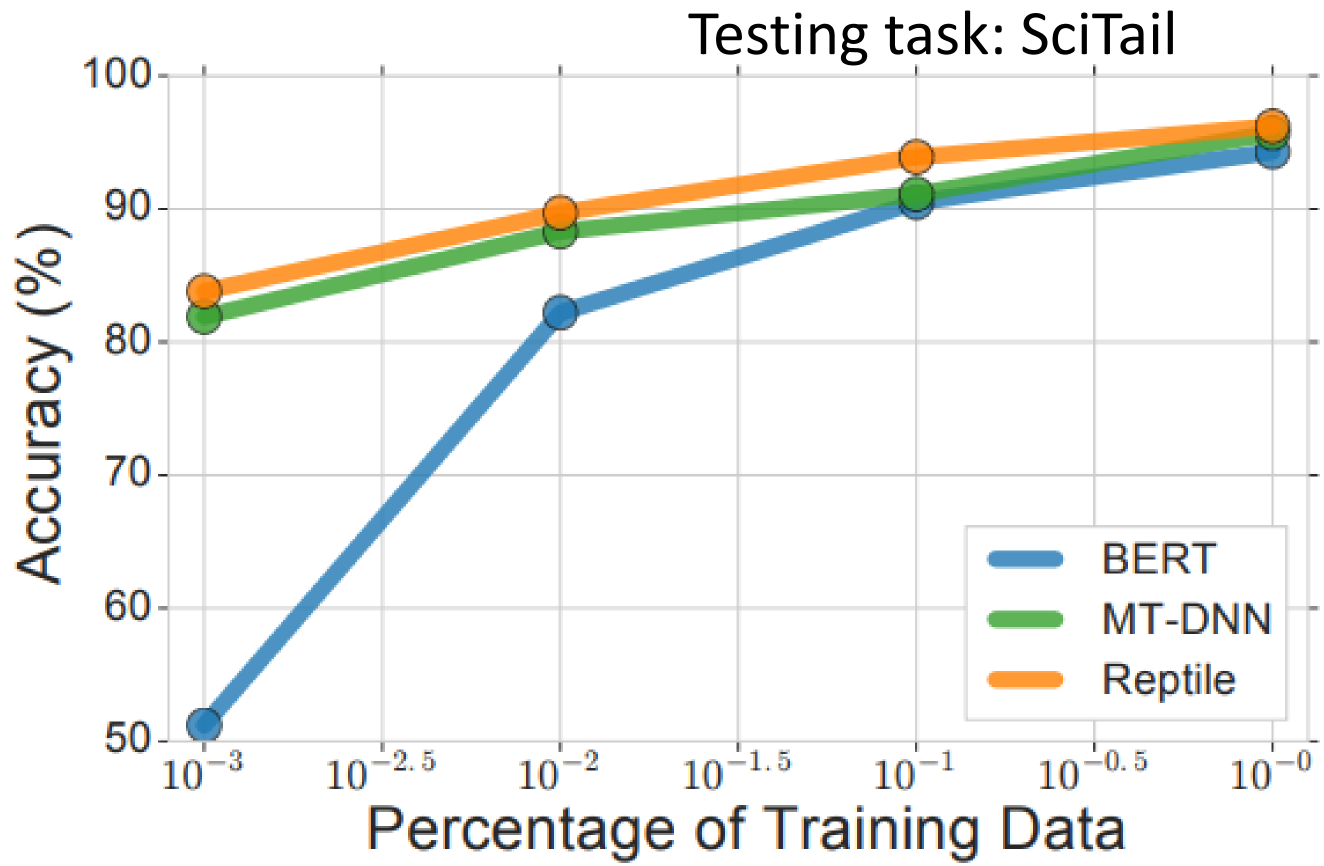
Question 1: Learn to Init vs. BERT



SPIS = samples per intent and slot

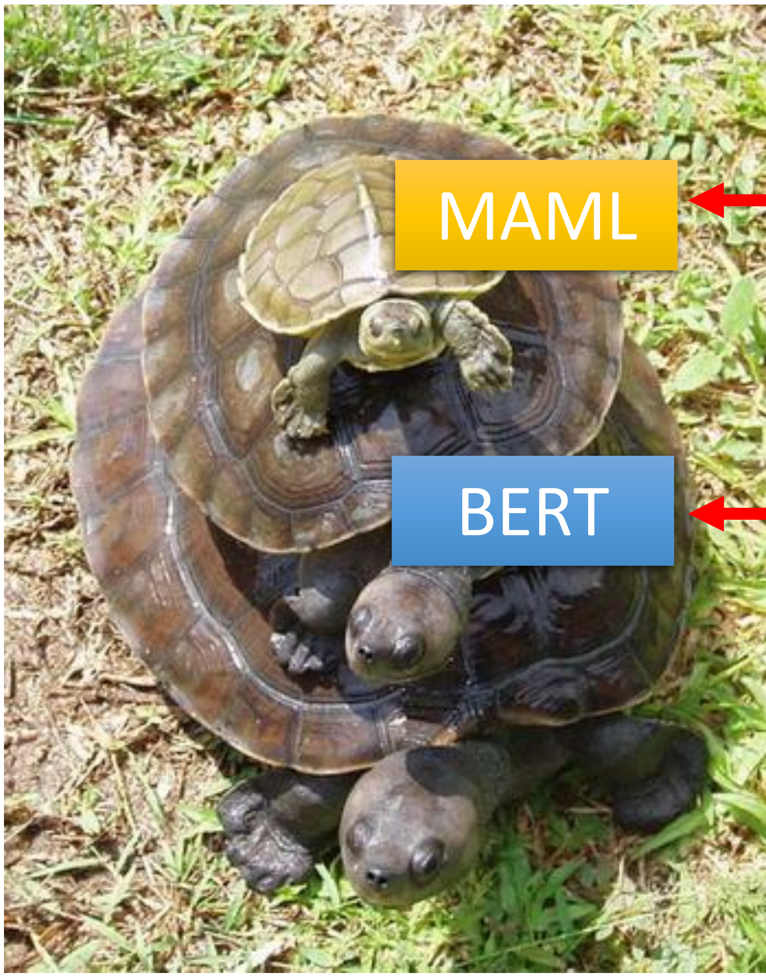
Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, Sonal Gupta, Low-Resource Domain Adaptation for Compositional Task-Oriented Semantic Parsing, EMNLP, 2020

Question 1: Learn to Init vs. BERT



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

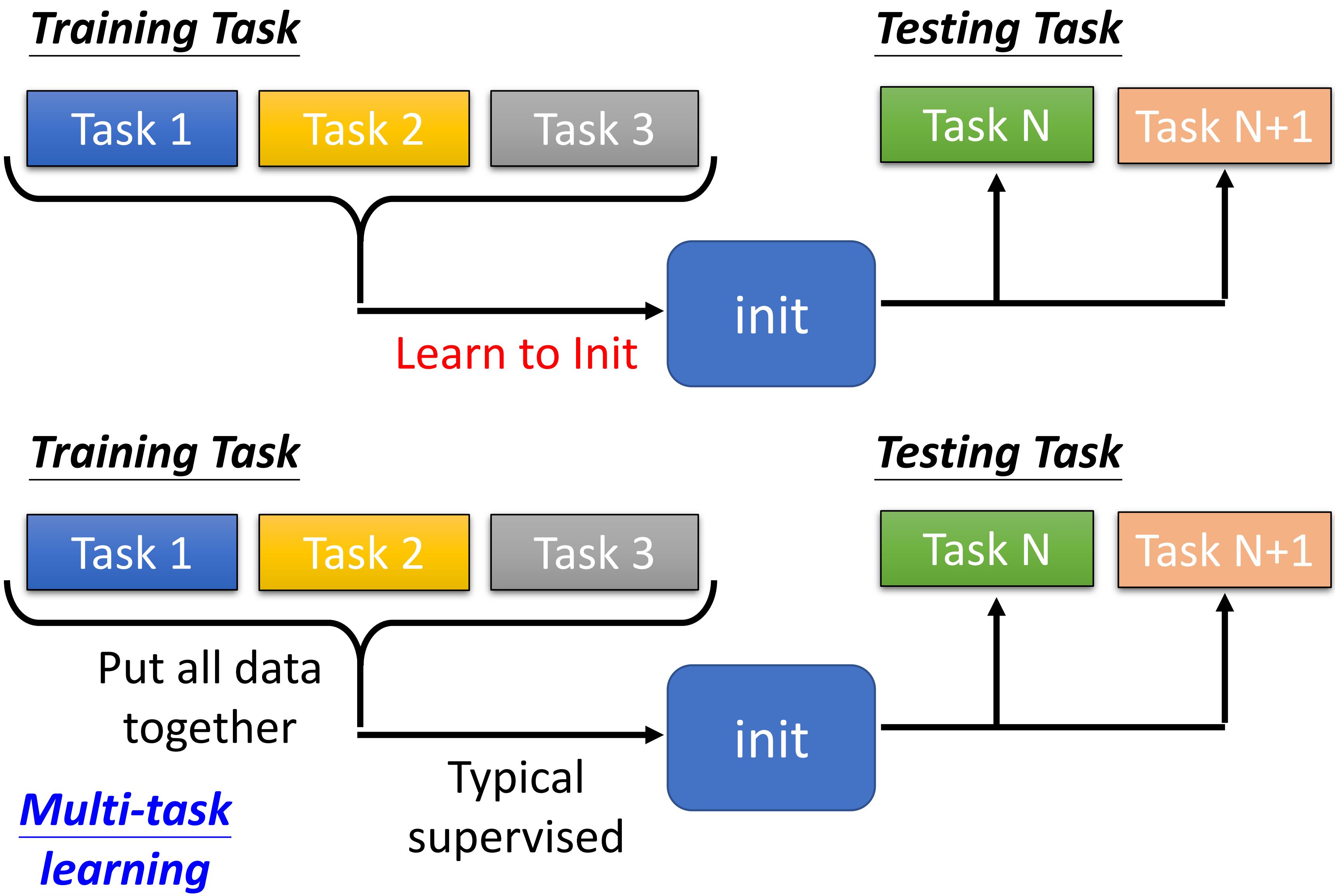
Question 1: Learn to Init vs. BERT



Turtles all the way down?

- Leverage training tasks.
- Learn to achieve good performance on training tasks.
- The self-supervised objectives are different from downstream tasks.
- There is a “learning gap”.

Leveraging Training Task



Leveraging Training Task

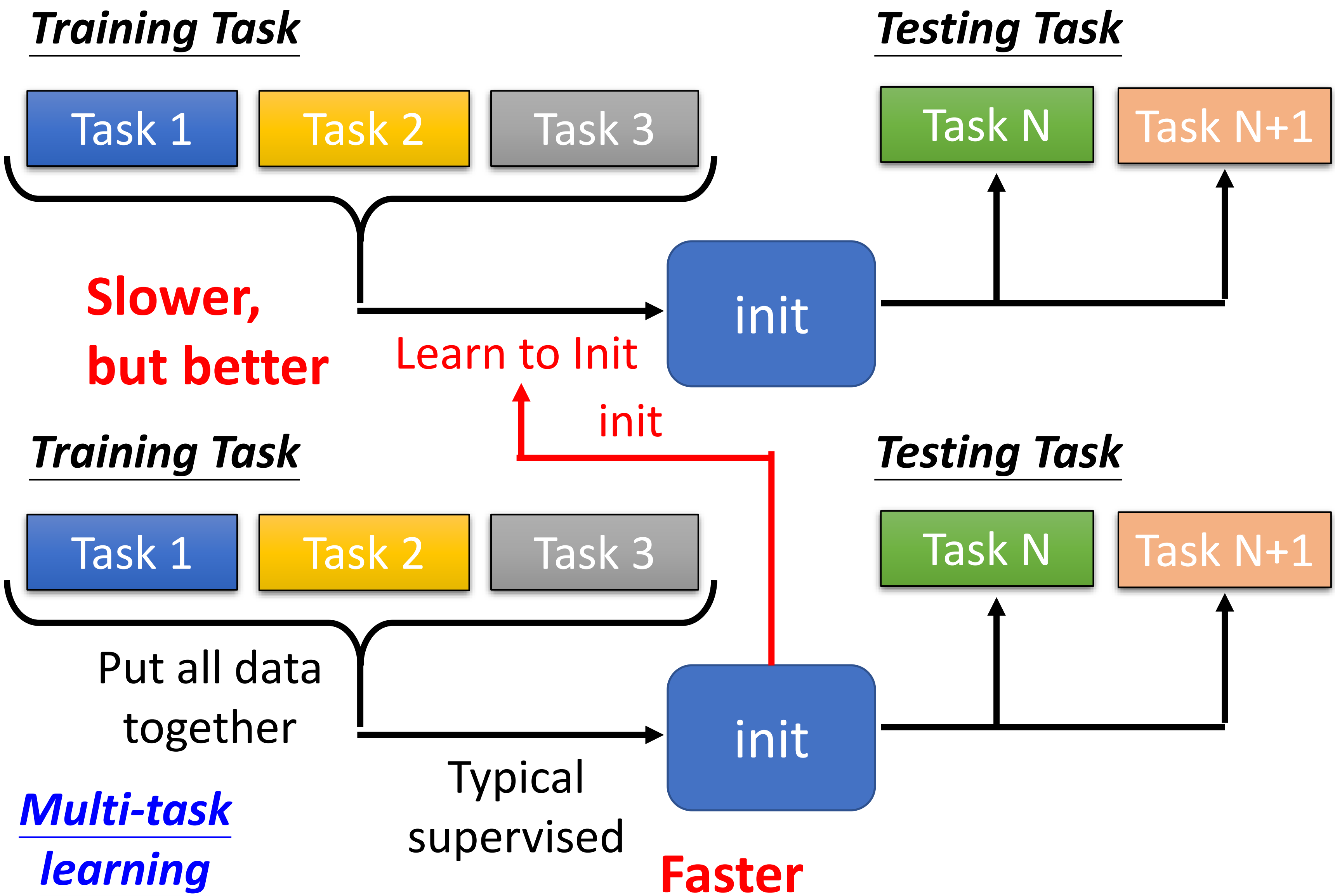
	Learn to Initialization	Multi-task Learning
Performance	Win (?)	
Training Speed		Win

Meta learning: consider the "fine-tuning" stage when learning initialization parameters.

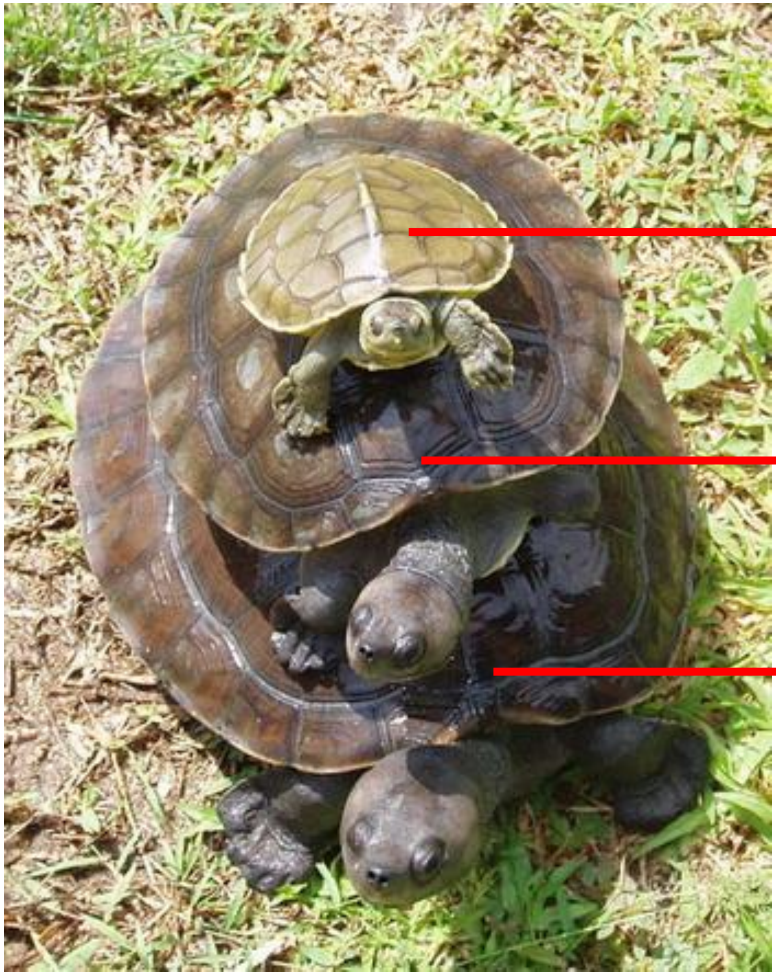
Multi-task learning: do not consider the "fine-tuning" stage at all.

Counterexample: Haoxiang Wang, Han Zhao, Bo Li, Bridging Multi-Task Learning and Meta-Learning: Towards Efficient Training and Effective Adaptation, ICML, 2021

Initialization of "Learn to initialize"



Ultimate Way for Initialization? ☺



Turtles all the way down?

Consider the fine-tuning stage

Learn to Init

Supervised Pre-training

Utilize training tasks

Self-supervised Pre-training

Utilize a large amount of unlabeled data

Language	$\lvert S \rvert = 20$ MAML	$\lvert S \rvert = 20$ MAML-	$\lvert S \rvert = 80$ MAML	$\lvert S \rvert = 80$ MAML-
Low-Resource Languages				
Armenian	**63.84**	59.70	**64.78**	60.03
Breton	**64.18**	59.33	**66.14**	60.84
Buryat†	25.77	**26.02**	**27.33**	27.05
Faroese†	**68.95**	65.30	**71.12**	66.79
Kazakh	**55.07**	53.92	**56.15**	54.99
U.Sorbian†	**56.40**	51.67	**58.78**	52.38
Mean	55.7	52.66	57.38	53.68
High-Resource Languages				
Finnish	**64.89**	61.97	**65.82**	62.47
French	**66.85**	63.42	**67.25**	64.15
German	**76.41**	74.38	**76.72**	74.72
Hungar.	**62.71**	58.47	**62.52**	57.48
Japanese	39.06	**39.72**	**46.81**	43.87
Persian	**52.81**	50.31	**54.74**	51.08
Swedish	**81.36**	77.57	**81.59**	78.10
Tamil	44.34	**46.55**	**50.68**	50.54
Urdu	55.16	**55.4**	**57.60**	56.28
Vietnam.	**43.34**	42.62	**44.33**	43.78
Mean	58.4	55.95	59.52	56.53

Anna Langedijk, Verna Dankers, Phillip Lippe, Sander Bos, Bryan Cardenas Guevara, Helen Yannakoudakis, Ekaterina Shutova, Meta-learning for fast cross-lingual adaptation in dependency parsing, arXiv, 2021

MLQA

Self-supervised

Supervised

Meta

		Model	en	ar	de	es	hi	[illegible]	[illegible]	[illegible]
XLM		Our baseline	**69.80**	48.95	52.64	**58.15**	46.67	48.46	42.64	52.47
	X-MAML	*(One aux. lang.)*	69.39	48.45	53.04	57.68	46.90	[illegible]	[illegible]	[illegible]
		$l \rightarrow X$	*ar*	*hi*	*es*	*en*	*zh*	[illegible]	[illegible]	
	X-MAML	*(Two aux. lang.)*	68.88	**49.76**	**53.18**	58.00	**48.43**	**50.86**	**45.44**	**53.51**
		$(l_1, l_2) \rightarrow X$	*(es,ar)*	*(vi,zh)*	*(vi,zh)*	*(en,zh)*	*(vi,zh)*	*(hi,zh)*	*(es,hi)*	
XLM-R$_{base}$		Liang et al. (2020)	80.1	56.4	62.1	67.9	60.5	67.1	61.4	65.1
		Our baseline	**80.38**	57.23	63.08	67.91	61.46	67.14	62.73	65.70
	X-MAML	*(One aux. lang.)*	80.19	57.97	63.57	67.46	61.70	67.97	64.01	66.12
		$l \rightarrow X$	*vi*	*hi*	*ar*	*vi*	*vi*	*hi*	*hi*	
	X-MAML	*(Two aux. lang.)*	80.31	**58.14**	**64.07**	**68.08**	**62.67**	**68.82**	**64.06**	**66.59**
		$(l_1, l_2) \rightarrow X$	*(ar,vi)*	*(hi,vi)*	*(ar,hi)*	*(ar,hi)*	*(es,ar)*	*(ar,hi)*	*(ar,hi)*	
XLM-R$_{large}$		Hu et al. (2020)	83.5	66.6	70.1	74.1	70.6	74	62.1	71.6
		Our baseline	83.95	66.09	70.62	74.59	70.64	74.13	69.80	72.83
	X-MAML	*(One aux. lang.)*	84.31	66.61	70.84	74.32	**70.94**	**74.84**	**70.74**	73.23
		$l \rightarrow X$	*ar*	*hi*	*ar*	*hi*	*vi*	*ar*	*hi*	
	X-MAML	*(Two aux. lang.)*	**84.60**	**66.95**	**71.00**	**74.62**	70.93	74.73	70.29	**74.30**
		$(l_1, l_2) \rightarrow X$	*(hi,vi)*	*(hi,vi)*	*(ar,vi)*	*(en,vi)*	*(ar,vi)*	*(es,hi)*	*(en,vi)*	

Farhad Nooralahzadeh , Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein, Zero-shot cross-lingual transfer with meta learning, EMNLP, 2020

Mixed Results

	method	p.t.	f.t.	libri	vctk	libri_n	vctk_n
(1)	MAML	best	m	**9.84**	7.76	7.56	5.99
(2)		-	m	9.38	**8.62**	7.54	**7.18**
(3)	ANIL_s	best	a_s	9.67	7.92	**7.64**	6.17
(4)		-	a_s	9.48	7.57	7.53	6.16
(5)	ANIL_c	best	a_c	8.89	6.52	7.03	5.33

Yuan-Kuei Wu, Kuan-Po Huang, Yu Tsao, Hung-yi Lee, One Shot Learning for Speech Separation, ICASSP, 2021

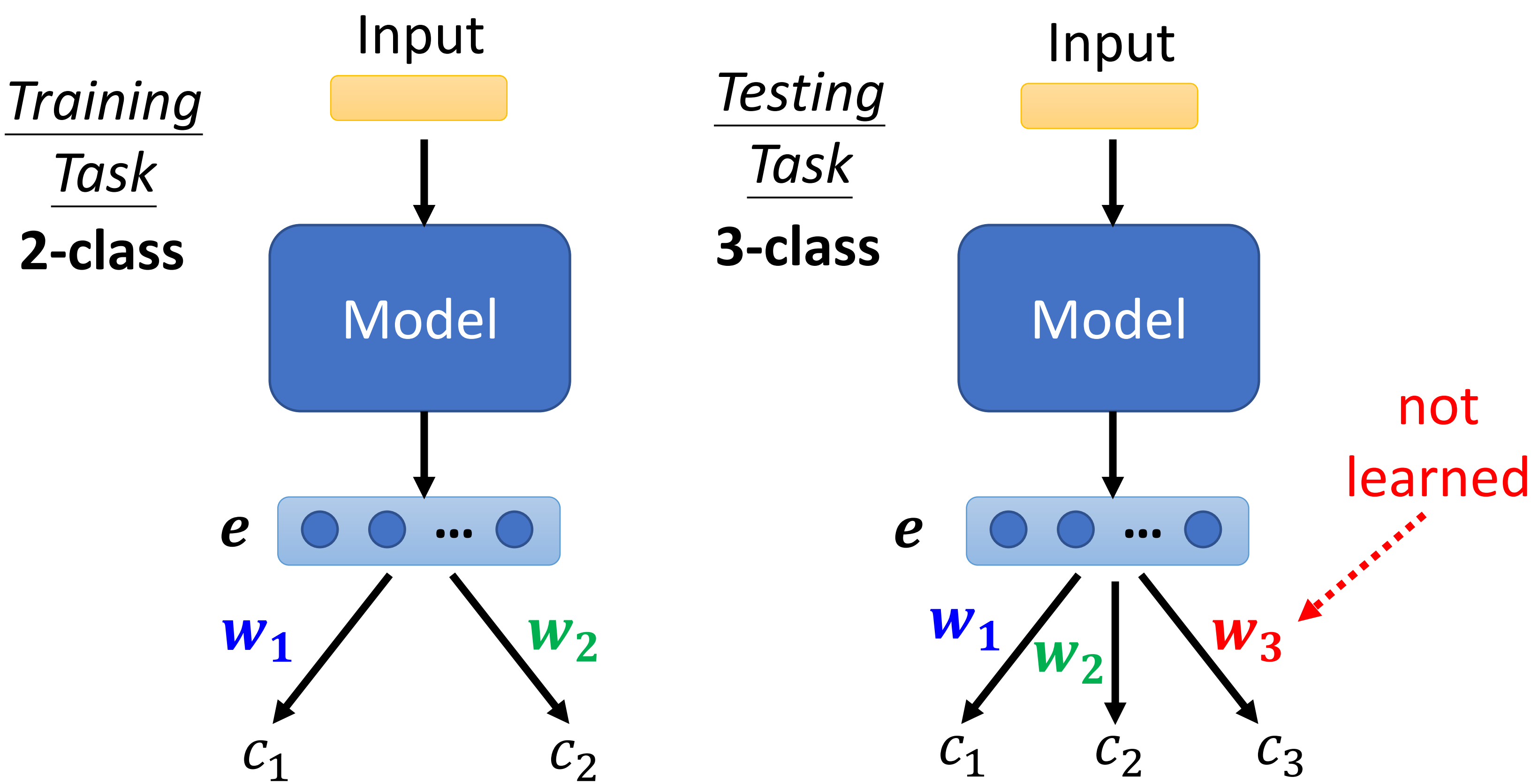
Supervised pre-training is added.

Mixed Results

Method	**Limited-resource setting**					**High-resource setting**				
	de	**fr**	**ja**	**zh**	Diff	**de**	**fr**	**ja**	**zh**	Diff
ProtoNet	91.1	90.9	87.1	85.5	+0.75	91.3	91.1	87.4	88.7	+1.44
foMAML	90.8	87.4	87.3	85.2	-0.75	91.7	91.2	87.2	88.1	-1.13
foProtoMAMLn	87.7	87.8	83.9	84.4	-3.1	90.8	89.8	86.2	82.3	-3.96
Reptile	89.3	90.2	86.7	85.5	+0.35	90.0	89.3	87.1	85.7	-1.04

Niels van der Heijden, Helen Yannakoudakis, Pushkar Mishra, Ekaterina Shutova, Multilingual and cross-lingual document classification: A meta-learning approach, EACL, 2021

Question 2: Different Output



$c_i = \boldsymbol{e} \cdot \boldsymbol{w_i}$ $\qquad$ $c_i = \boldsymbol{e} \cdot \boldsymbol{w_i}$

Question 2: Different Output

LEOPARD — Trapit Bansal, Rishikesh Jha, Andrew McCallum, Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks, COLING, 2020

ProtoMAML — Niels van der Heijden, Helen Yannakoudakis, Pushkar Mishra, Ekaterina Shutova, Multilingual and cross-lingual document classification: A meta-learning approach, EACL, 2021

Training Task

GLUE

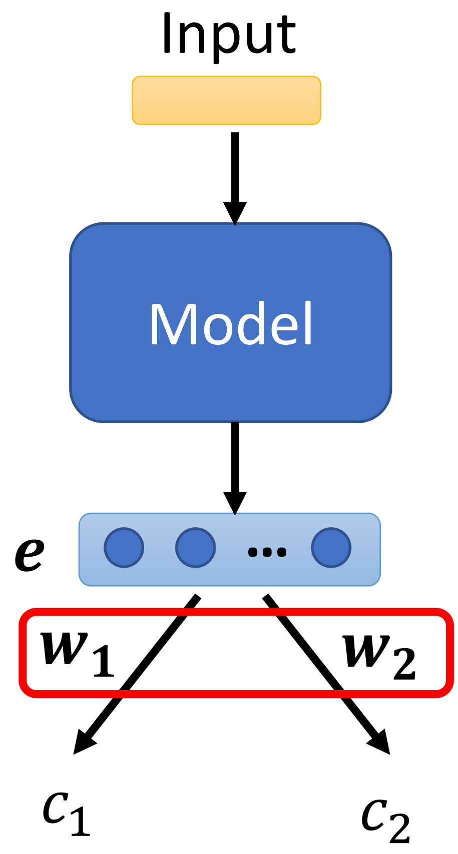
Testing Task

Other classification tasks

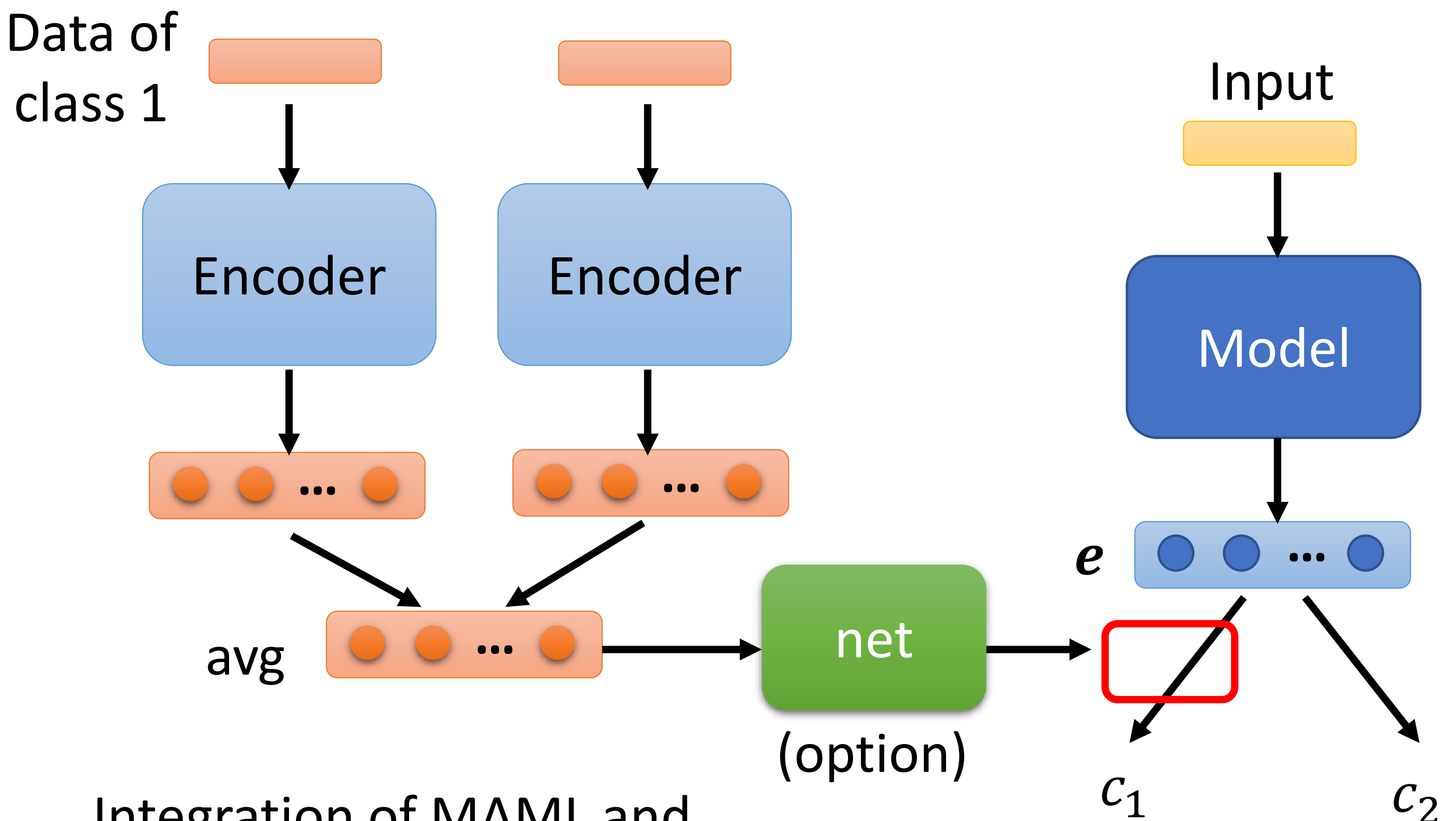
Question 2: Different Output

We do not learn class-specific parameters.

The class-specific parameters are generated from data.



Question 2: Different Output



Integration of MAML and metric-based approach

Learning to Compare in NLP

Thang Vu

Overview

	(A) Learning to initialize	(B) Learning to compare	(C) Other
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Overview

	(A) Learning to initialize	(B) Learning to compare	(C) Other
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	1 (Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	2 (Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	3 (Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		4 (Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Overview

	(A) Learning to initialize	(B) Learning to compare	(C) Other
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	1 (Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	2 (Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	3 (Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		4 (Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

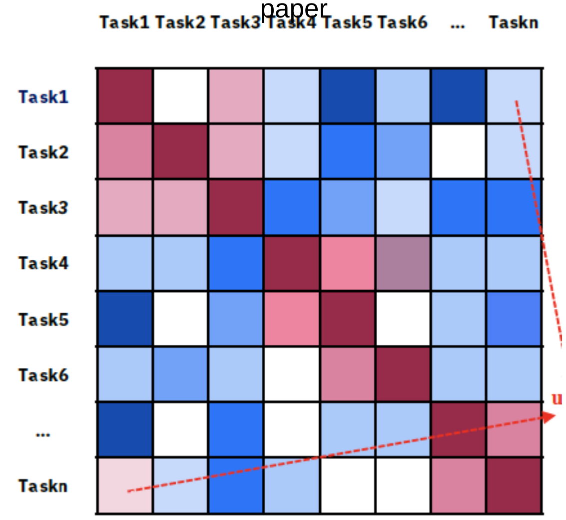
Diverse Few-Shot Text Classification with Multiple Metrics

- 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



Cross-task transfer performance matrix

- How to cluster tasks:
  - Create a transfer performance matri
  - Apply scores filtering and matrix completion
  - Apply spectral clustering
- How to combine decisions:
  - Linearly combine decisions from different task clusters
  - Linear coefficients are adaptable parameters

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

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

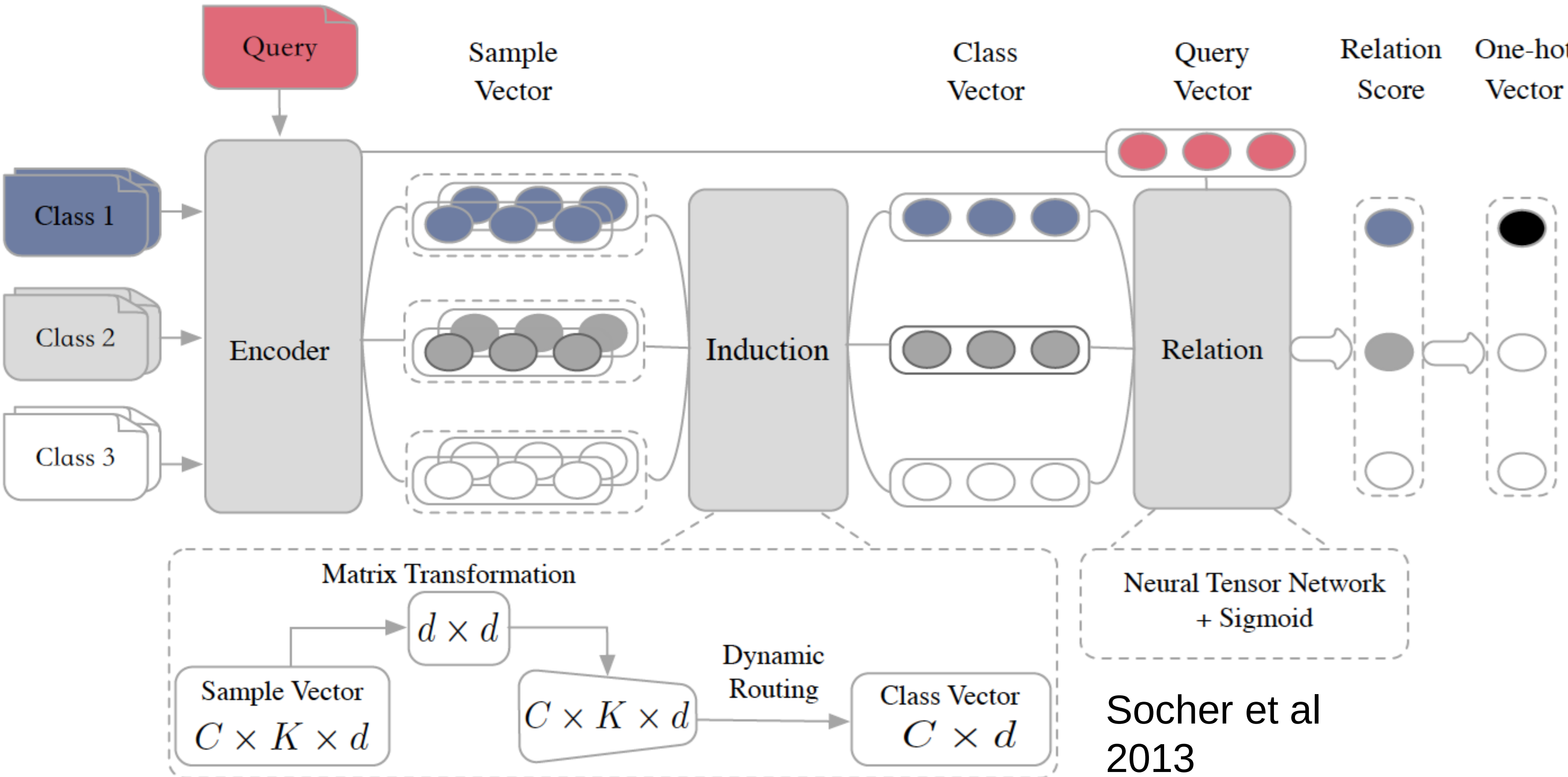
Induction Networks for Few-Shot Text Classification

- 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



Socher et al 2013

Sabour et al 2017

Overview

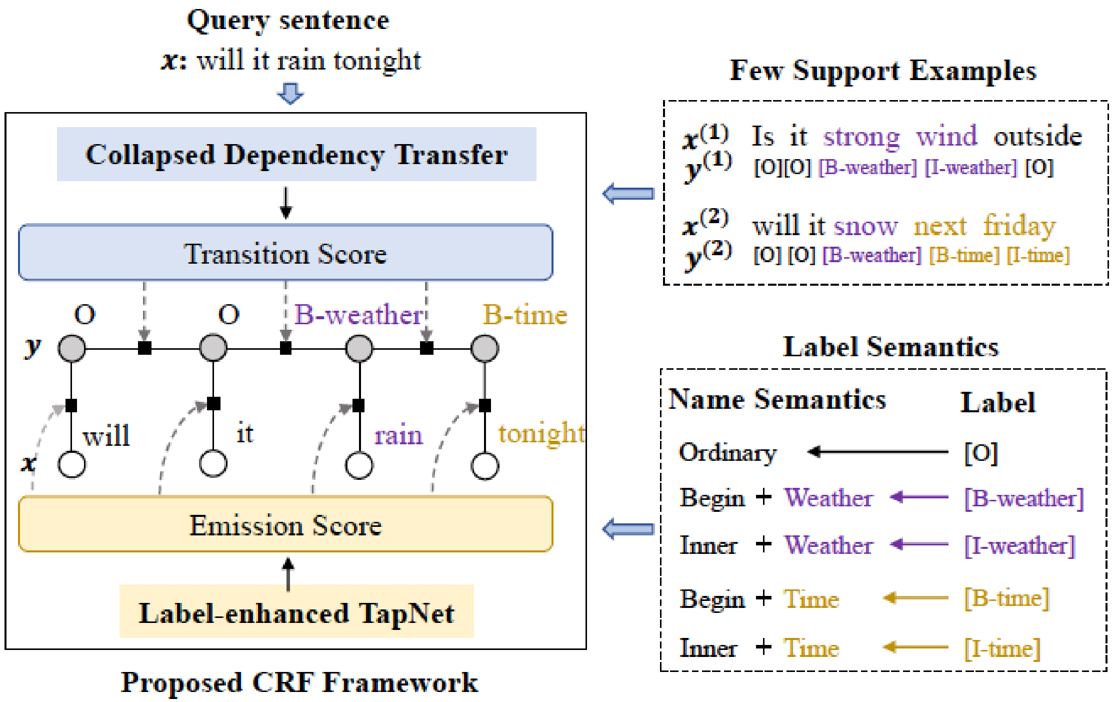
	(A) Learning to initialize	(B) Learning to compare	(C) Other
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	1 (Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	2 (Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	3 (Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		4 (Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network

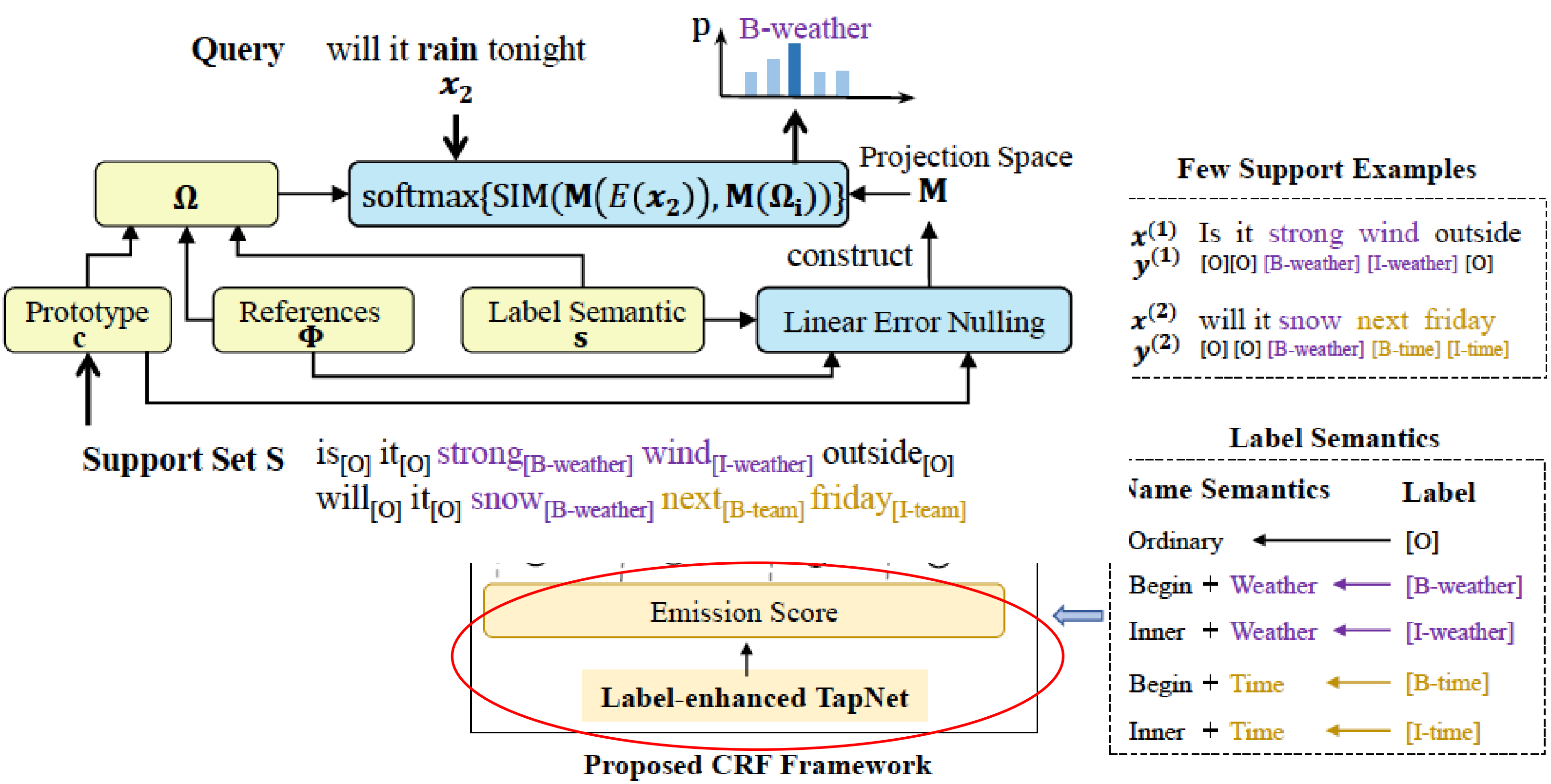
- Key ideas and take-home messages
  - Leverage the CRF framework for sequence labeling task
  - Novelties lie on methods to compute transition scores and emission scores
  - The proposed emission scoring method is based on learning to compare methods

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, Ting Liu. Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network, ACL 2020

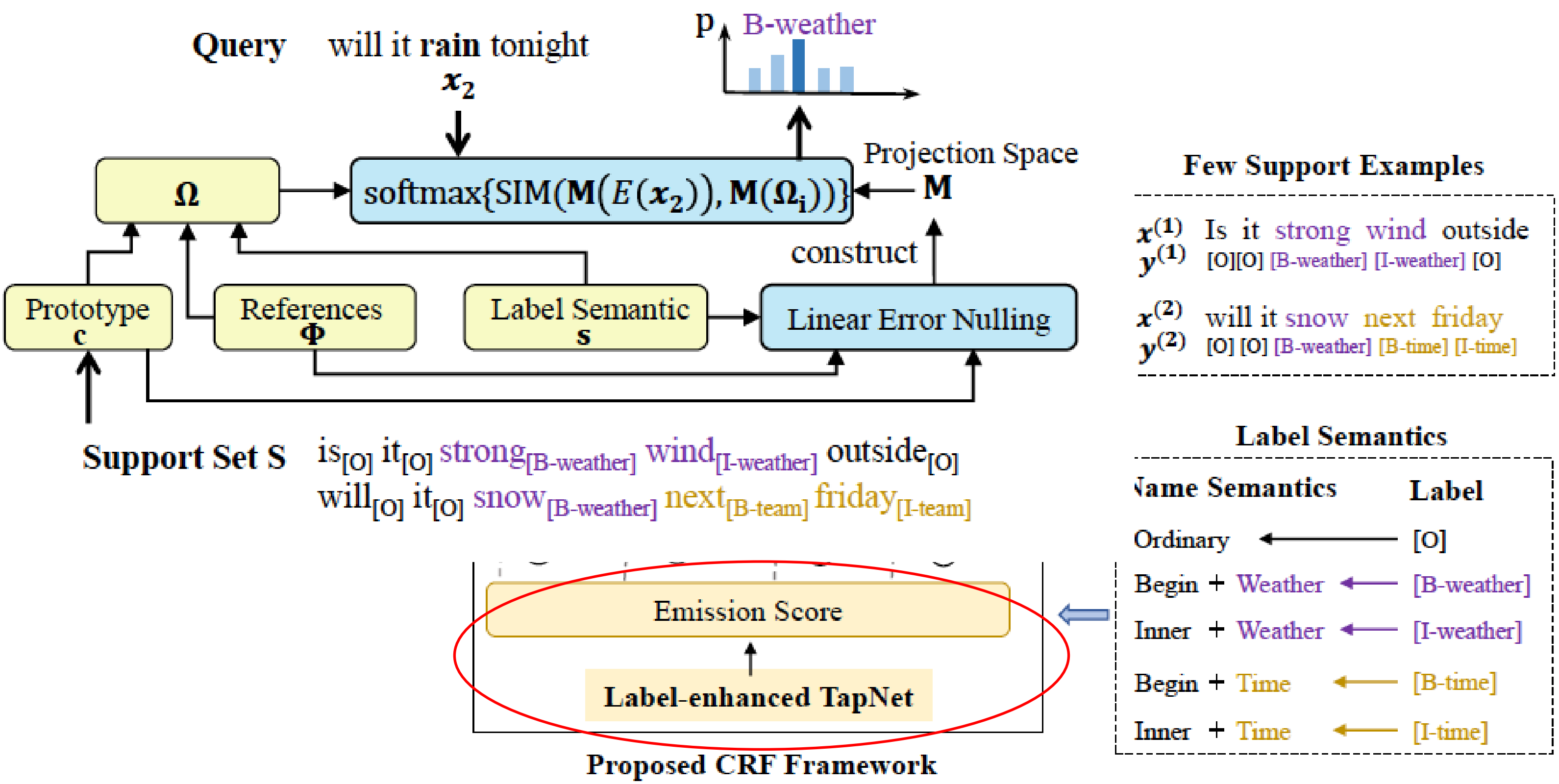
Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network



Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network



Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network



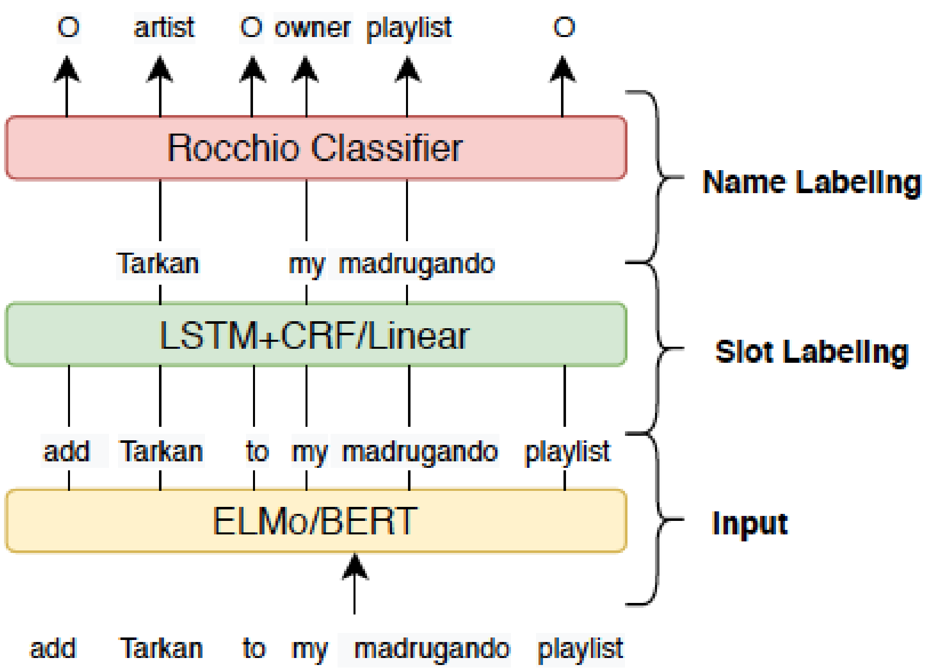
TapNet (Yoon et al 2019)

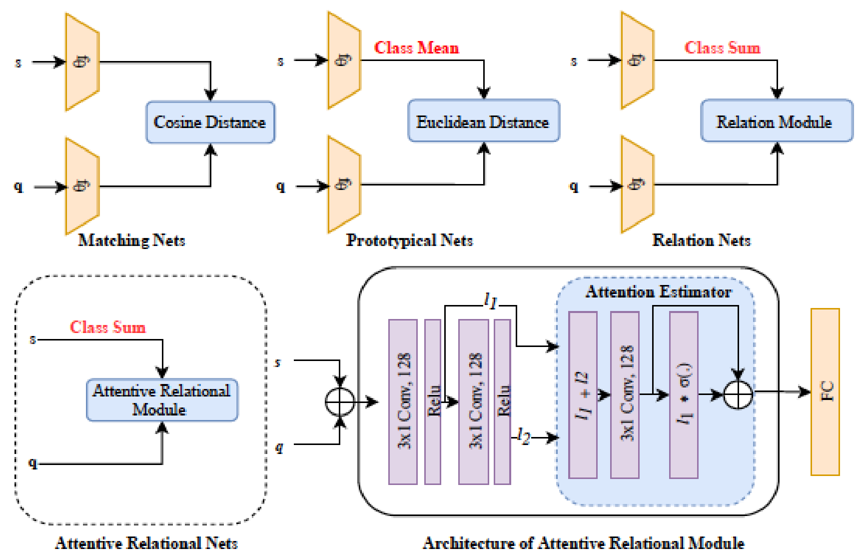
Few-shot Learning for Slot Tagging with Attentive Relational Network

- Key ideas and take-home messages
  - Propose a two step approach to exclude O class
  - Based on Relation Nets
  - Propose an attentive relational module to compute the relation score

Cennet Oguz, Ngoc Thang Vu. Few-shot Learning for Slot Tagging with Attentive Relational Network. EACL 2021.

Few-shot Learning for Slot Tagging with Attentive Relational Network





2-step
approach

Advance relation nets
for
name labeling step

Overview

	(A) Learning to initialize	(B) Learning to compare	(C) Other
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	1 (Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	2 (Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	3 (Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		4 (Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

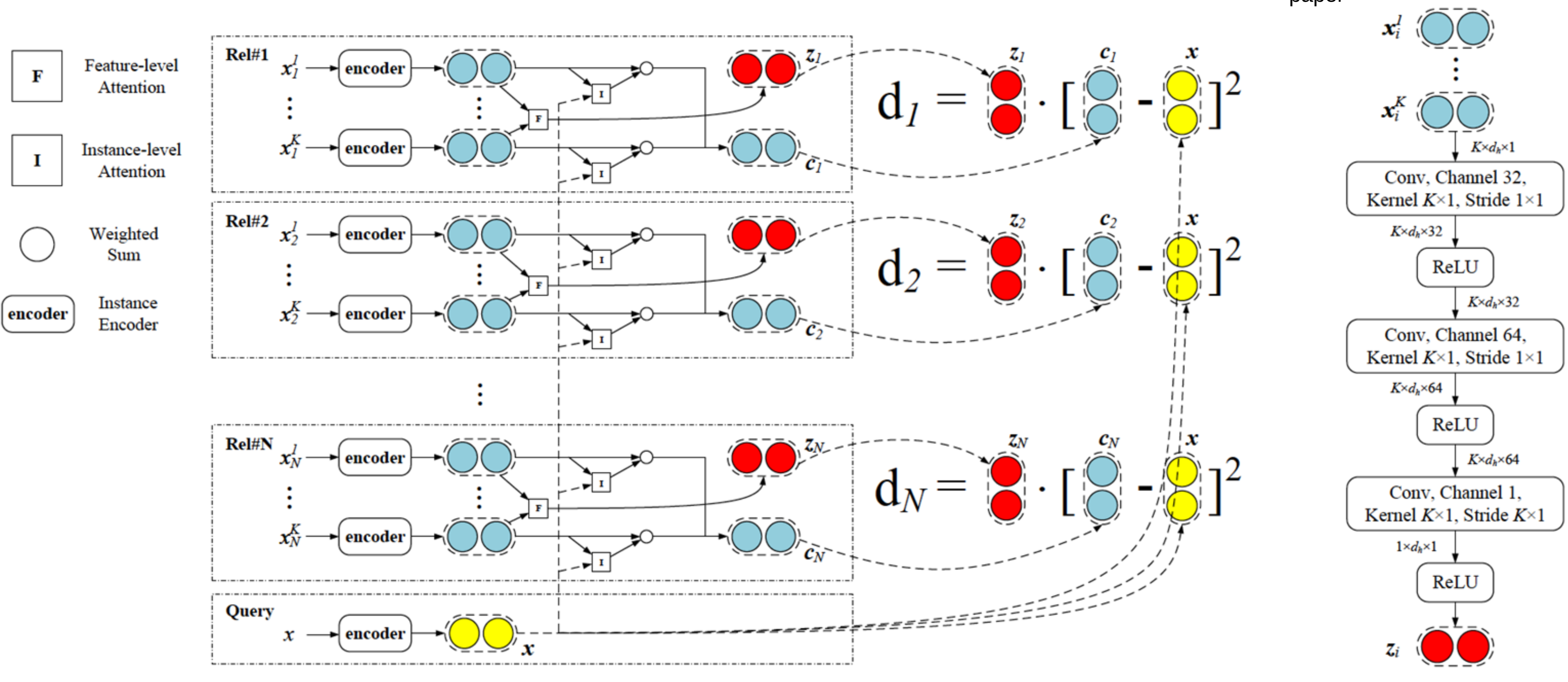
Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification

- 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



(a) (b)

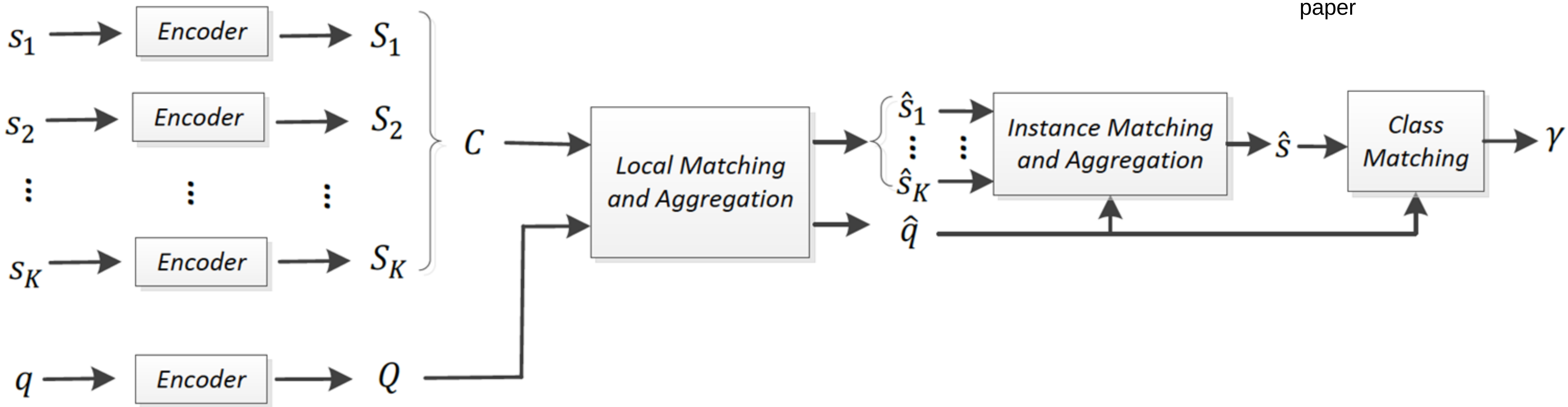
Multi-Level Matching and Aggregation Network for Few-Shot Relation Classification

- 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

Overview

	(A) Learning to initialize	(B) Learning to compare	(C) Other
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	1 (Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	2 (Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	3 (Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		4 (Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

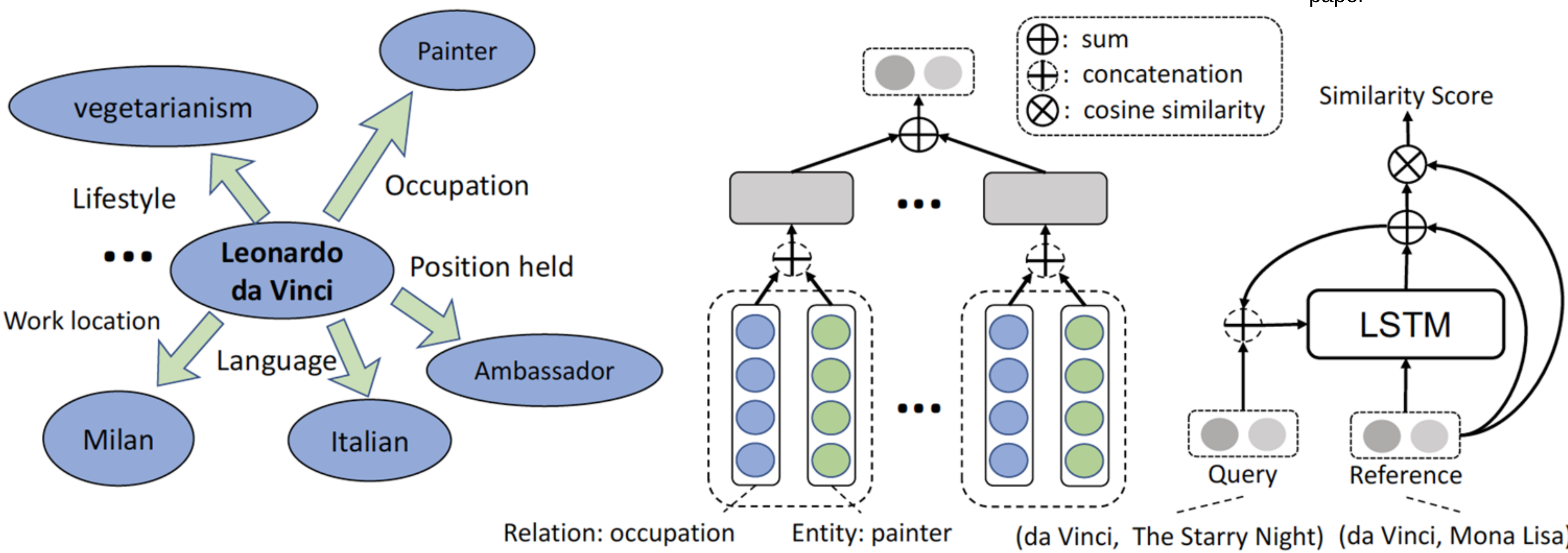
One-Shot Relational Learning for Knowledge Graphs

- (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



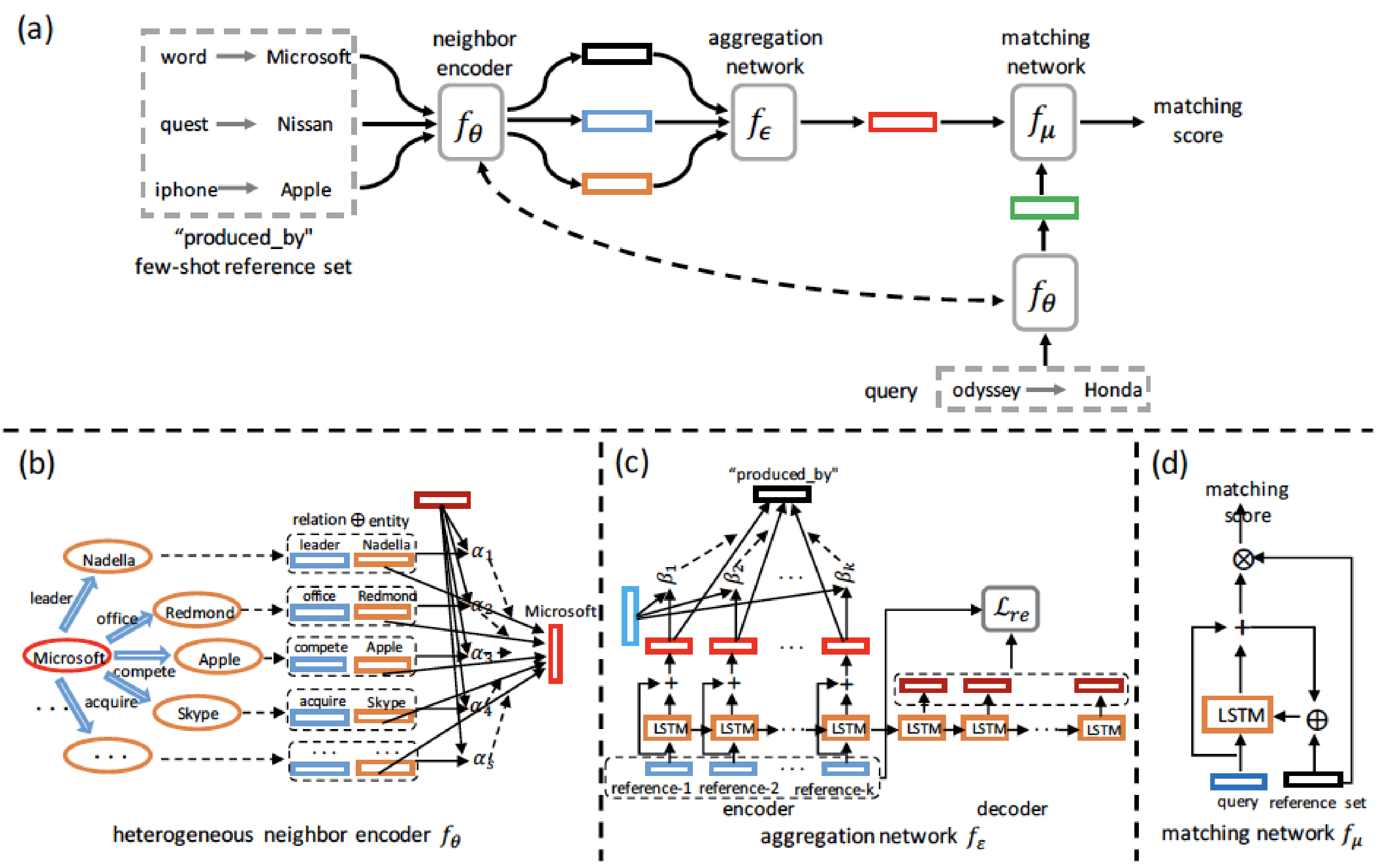
a) Local graph of entity ***Leonardo da Vinci*** **b) Neighbor Encoder** **c) Matching Processor**

Few-Shot Knowledge Graph Completion

- Key ideas and take-home messages:
  - The proposed architecture is based on matching network
  - Apply attention mechanism for neighbor encoder
  - Leverage auto encoder framework for aggregation that allows few-shot classification and interaction among examples in the support set

Chuxu Huang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, Nitesh V. Chawla. Few-Shot Knowledge Graph Completion. AAAI, 2020.

Few-Shot Knowledge Graph Completion

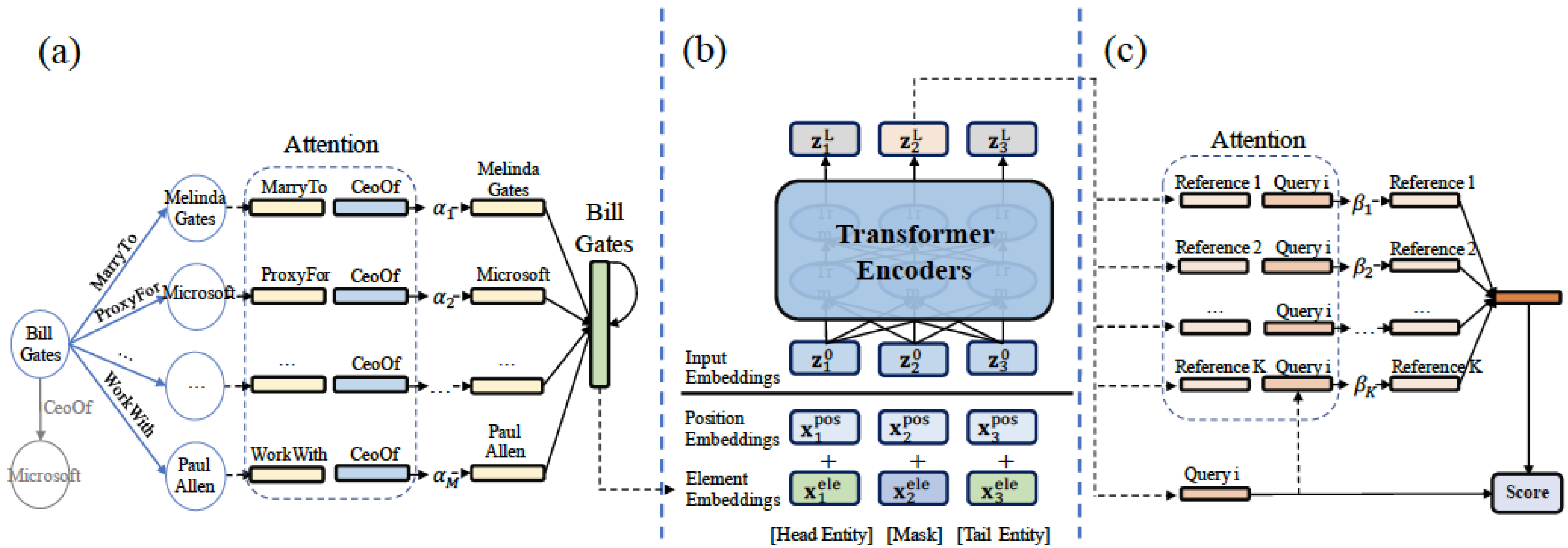


Adaptive Attentional Network for Few-Shot Knowledge Graph Completion

- Key ideas and take-home messages:
  - The proposed method is based on relation network
  - As previous paper, apply attention mechanism for neighbor encoder
  - Leverage transformer to model the relation between head and tail entities
  - Apply attention mechanism in the scoring function

Jiawei Sheng, Shu Gou, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, Hungbo Xu. Adaptive Attentional Network for Few-Shot Knowledge Graph Completion, EMNLP, 2020.

Adaptive Attentional Network for Few-Shot Knowledge Graph Completion



Summary: General Patterns

- Mostly based on:
  - Matching Network
  - Prototypical Network
  - Relation Network
- The main novelties focus on:
  - Representation learning
    - For a single instance
    - For prototypes/classes
  - Scoring functions
    - Distance/similarity
    - Relation scores

Network architecture search, learning to optimize, learning the learning algorithm, and more

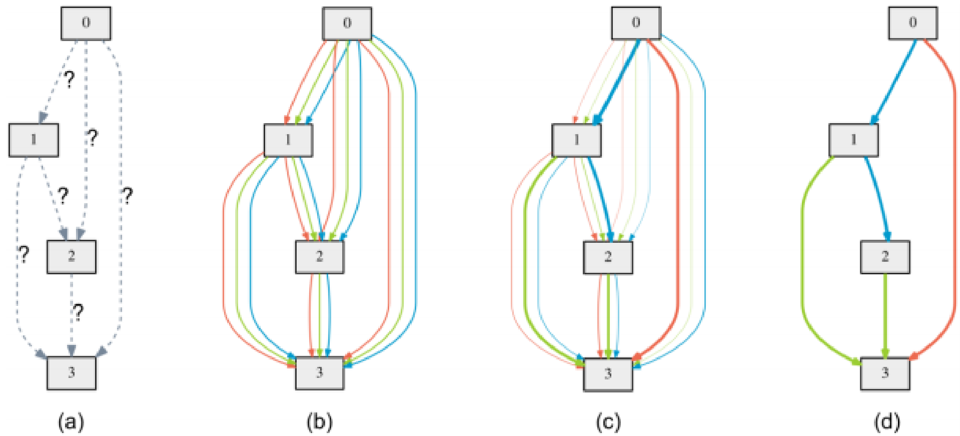
	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

NAS for LM/NER

Yufan Jiang, et al., Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition, EMNLP, 2019

- Differentiable Architecture Search (DARTs)[1]



- Improve DARTs with simpler and more suitable search space for NLP (RNN)

Training Task

Tr-ep 1 | Tr-ep 2 | Tr-ep 3 | ……

Testing Task

Val-ep 1 | Val-ep 2

[1] Hanxiao Liu, et al., DARTS: Differentiable Architecture Search. ICLR, 2019

NAS for LM/NER

Yufan Jiang, et al., Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition, EMNLP, 2019

- Results
  - Competitive LM / NER against baselines with popular architectures
  - Better than ENAS / DARTs baselines

LSTM	60.7	58.8
LSTM + SC	60.9	58.3
LSTM + SE	58.1	56.0
ENAS	60.8	58.6
DARTS	58.3	56.1
Random RNNs	63.7	61.2
I-DARTS ($n = 1$)	58.0	56.0
I-DARTS ($n = 2$)	-	-

LM on PTB (ppl val/test)

BiLSTM-CRF (Lample et al., 2016)	90.94
BiLSTM-CRF+ELMo (Peters et al., 2018)	92.22
BERT Base (Devlin et al., 2018)	92.40
BERT Large (Devlin et al., 2018)	92.80
Random RNNs	92.89
DARTS	93.13
I-DARTS ($n = 2$)	93.14
I-DARTS ($n = 1$)	93.47

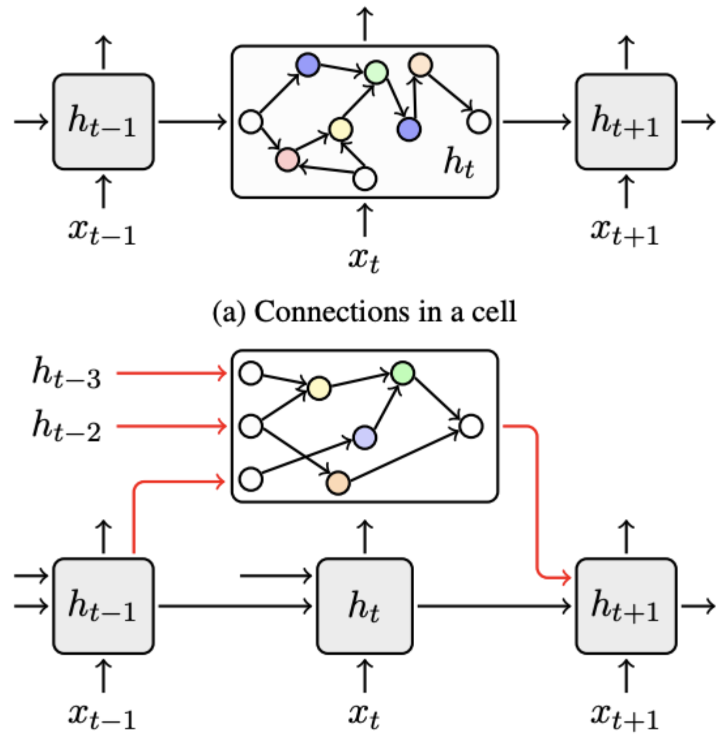
NER on CoNLL-2003 English (F1)

NAS for LM/NER

Yinqiao Li, et al., Learning Architectures from an Extended Search Space for Language Modeling, ACL, 2020

- Extend previous paper to include more architectures for RNN
  - Search cell architecture (a) and how cells are concatenated (b)
  - Each cell for one time stamp (token)
- LM (ppl)
  - > Transformer/SOTA/I-DARTs (PTB)
  - Transformer > ESS > others (WT-103)

Dataset	Method	Search Space		Params	Perplexity	
		intra-cell	inter-cell		valid	test
PTB	AWD-LSTM (Merity et al., 2018c)	-	-	24M	61.2	58.8
	Transformer-XL (Dai et al., 2019)	-	-	24M	56.7	54.5
	Mogrifier LSTM (Melis et al., 2019)	-	-	23M	**51.4**	**50.1**
	ENAS (Pham et al., 2018)	✓	-	24M	60.8	58.6
	RS (Li and Talwalkar, 2019)	✓	-	23M	57.8	55.5
	DARTS†	✓	-	23M	55.2	53.0
	ESS	-	✓	23M	54.1	52.3
	ESS	✓	✓	23M	**47.9**	**45.6**
WT-103	QRNN (Merity et al., 2018a)	-	-	151M	32.0	33.0
	Hebbian + Cache (Rae et al., 2018)	-	-	-	29.9	29.7
	Transformer-XL (Dai et al., 2019)	-	-	151M	**23.1**	**24.0**
	DARTS†	✓	-	151M	31.4	31.6
	ESS	✓	✓	156M	**28.8**	**29.2**



(a) Connections in a cell

(b) Connections among cells

Figure 1: Examples of intra and inter-cell architectures.

NAS for LM/NER

Yinqiao Li, et al., Learning Architectures from an Extended Search Space for Language Modeling, ACL, 2020

- Transferability of searched architecture
  - Search on WikiText-103 and evaluate on NER corpora (CoNLL-2003 and more in paper)
  - SOTA / Near SOTA

Models	F1
LSTM-CRF (Lample et al., 2016)	90.94
LSTM-CRF + ELMo (Peters et al., 2018)	92.22
LSTM-CRF + Flair (Akbik et al., 2019)	93.18
GCDT + $\text{BERT}_{\text{LARGE}}$ (Liu et al., 2019b)	93.47
CNN Large + ELMo (Baevski et al., 2019)	**93.50**
DARTS + Flair (Jiang et al., 2019)	93.13
I-DARTS + Flair (Jiang et al., 2019)	93.47
ESS	91.78
ESS + Flair	**93.62**

NAS for text classification

Ramakanth Pasunuru, et al., FENAS: Flexible and Expressive Neural Architecture Search, EMNLP, 2020

- Extend ENAS[1] search space
  - (accuracy) more activation functions and operations to contain GRU/LSTM etc.
  - (efficiency) allowing to initialize search with well-known human-designed structure

Training Task ***Testing Task***

- Performance on GLUE
  - FENAS > ENAS > LSTM (all ~24M parameters)
- FENAS about 5x slower than ENAS

Architecture	CoLA	SST-2	MRPC	QQP	STS-B	MNLI	QNLI	RTE	WNLI	AVG
LSTM	17.1	86.9	71.0/78.9	83.2/62.7	67.8/65.6	64.9/65.8	77.4	52.1	65.1	64.3
ENAS-RL	14.7	84.1	74.5/82.6	83.8/63.0	72.6/70.7	66.0/66.6	78.5	51.0	65.1	64.8
ENAS-RS	16.7	85.6	73.7/81.6	81.9/61.5	72.5/70.4	66.9/67.5	78.8	53.1	65.1	65.3
FENAS	16.4	86.6	71.0/78.9	84.9/63.7	73.2/71.0	66.6/66.0	79.1	52.7	65.1	65.6

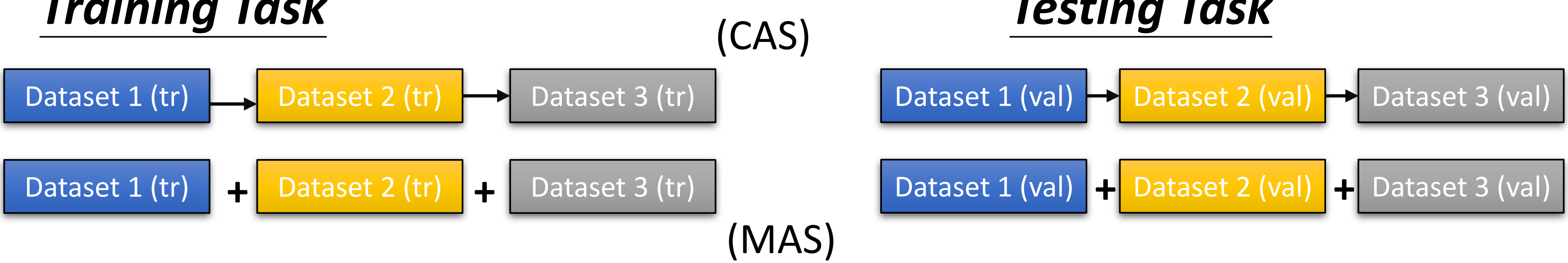
[1] Hieu Pham, et al., Efficient neural architecture search via parameters sharing.. ICML, 2018

NAS for text classification

Ramakanth Pasunuru, et al., Continual and Multi-Task Architecture Search, ACL, 2019

- ENAS
- Continual architecture search (CAS)
  - Sequentially training networks on several tasks without forgetting previously learned objective
  - Designed loss to encourage parameter updates from dataset to dataset orthogonal
- Multi-Task Architecture Search (MAS)
  - Multi-task version of architecture search to optimize a unified structure for many tasks
- Results
  - QNLI, RTE, WNLI from GLUE
  - CAS > ENAS / BiLSTM+ELMo
  - Similar trend in MAS

Training Task ***Testing Task***



	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Machine Translation	(Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
Parsing	(Guo et al., 2019) (Huang et al., 2018) (Langedijk et al., 2021) (Chen et al., 2020a) (Wang et al., 2021a)		
Dialogue	(Qian and Yu, 2019) (Madotto et al., 2019) (Mi et al., 2019) (Huang et al., 2020a) (Dingliwal et al., 2021) (Qian et al., 2021) (Dai et al., 2020) (Huang et al., 2020b)		Learning to optimize: (Chien and Lieow, 2019)
Speech Recognition	(Hsu et al., 2020) (Klejch et al., 2019) (Winata et al., 2020a) (Winata et al., 2020b) (Xiao et al., 2021)	(Lux and Vu, 2021)	Learning to optimize: (Klejch et al., 2018) Network architecture search: (Chen et al., 2020c) (Baruwa et al., 2019)
Source Separation	(Wu et al., 2021)		
Keyword Spotting	(Chen et al., 2020b)		Network architecture search: (Mazzawi et al., 2019)
Sound Event Detection		(Shimada et al., 2020) (Chou et al., 2019)	
Voice Cloning			Learning the learning algorithm: (Chen et al., 2019b) (Serrà et al., 2019)
Multi-tasks			Learning to select data: (Tarunesh et al., 2021)
Multi-modal		(Eloff et al., 2019)	Learning the learning algorithm: (Surís et al., 2019) (Xu et al., 2021a)

Learning the learning algorithm for NLP

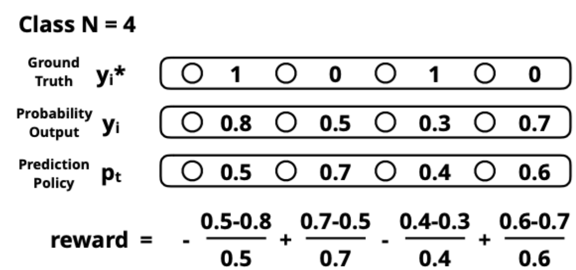
Jiawei Wu, et al., Learning to learn and predict: A meta-learning approach for multi-label classification, EMNLP, 2019

- Multi-label classification
  - Learning to learn:
    learn the weight (w_i) of loss over each label *i* and example *j*

$$L(\theta_t^C) = -\sum_i^{B_t}\sum_j^{N} w_t^{(j)} N\{y_i^{*(j)} \log y_i^{(j)} + (1 - y_i^{*(j)}) \log(1 - y_i^{(j)})\},$$

  - Learning to predict: learn threshold p_i for predicting *i* as True
  - Meta-learn a GRU iteratively predicting w, p based on w', p' in previous time stamps
  - Reinforcement learning (policy gradient) to update the meta learner

$$r_t = \sum_i^{B_t}\sum_{j=1}^{N}(-1)^{y_i^{*(j)}} \frac{p_t^{(j)} - y_i^{(j)}}{p_t^{(j)}}$$

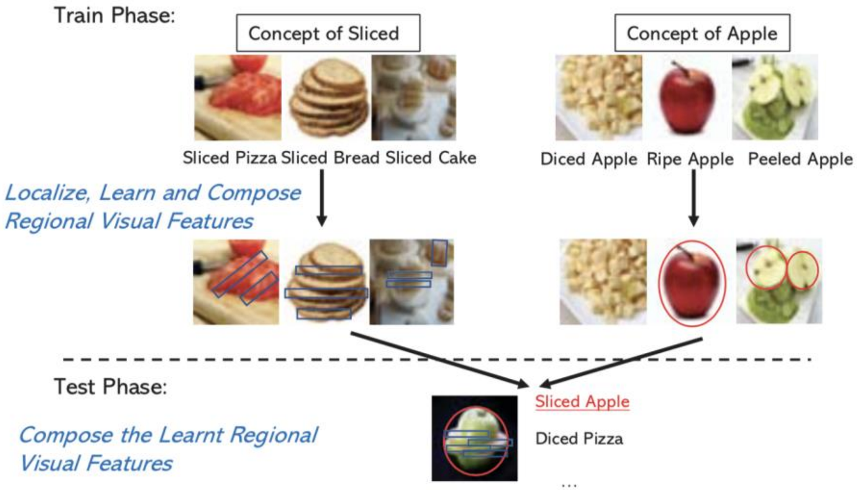


- Results
  - Entity type classification: FIGER, OntoNotes, and BBN
  - Text classification: Reuters-21578 and RCV1-V2
  - SOTA results

Learning the learning algorithm for NLP

Guangyue Xu, et al., Zero-Shot Compositional Concept Learning, MetaNLP workshop at ACL, 2021

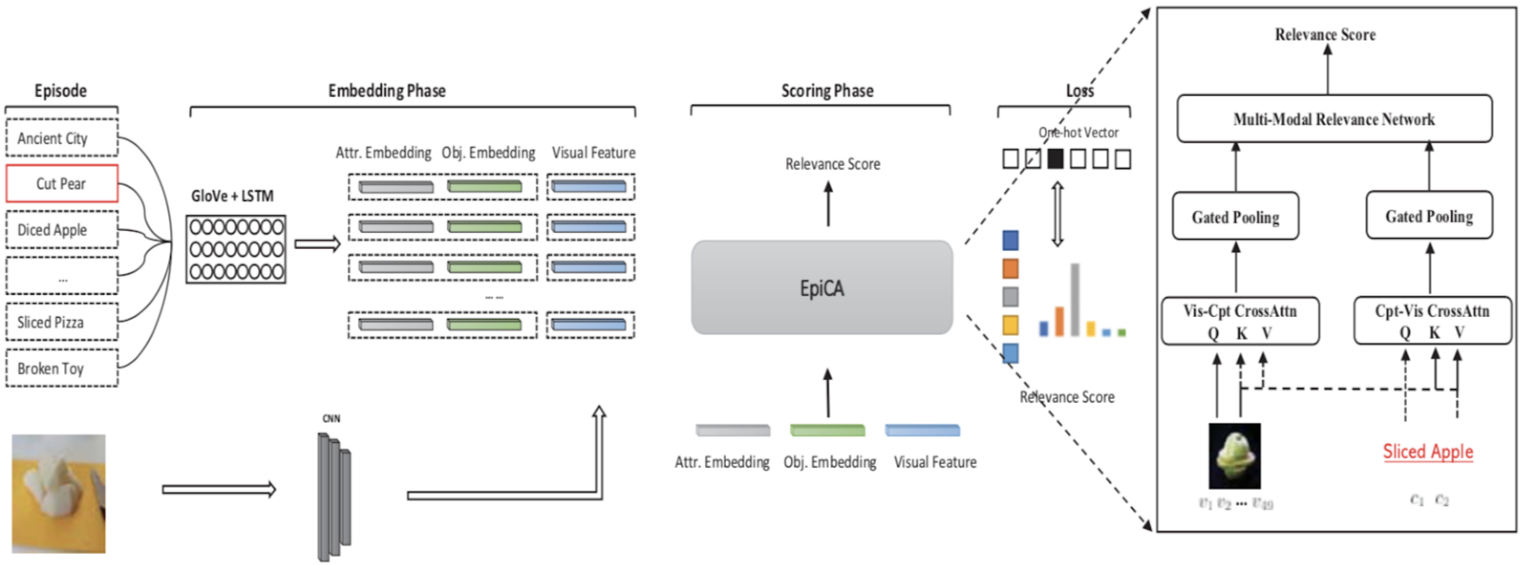
- Zero-shot, multimodal concept learning
  - Input: 1 image, 1 correct concept (text), n incorrect concept -> predict the correct one



Learning the learning algorithm for NLP

Guangyue Xu, et al., Zero-Shot Compositional Concept Learning, MetaNLP workshop at ACL, 2021

- Learning to learn the fusion mechanisms over multimodalities
  - Image, text encoded by CNN / GloVe
  - Cross attention + gated pooling + Multi-Modal Relevance Network



- Datasets: MIT-States, UT-Zappos
- SOTA or competitive performance

	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Machine Translation	(Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
Parsing	(Guo et al., 2019) (Huang et al., 2018) (Langedijk et al., 2021) (Chen et al., 2020a) (Wang et al., 2021a)		
Dialogue	(Qian and Yu, 2019) (Madotto et al., 2019) (Mi et al., 2019) (Huang et al., 2020a) (Dingliwal et al., 2021) (Qian et al., 2021) (Dai et al., 2020) (Huang et al., 2020b)		Learning to optimize: (Chien and Lieow, 2019)
Speech Recognition	(Hsu et al., 2020) (Klejch et al., 2019) (Winata et al., 2020a) (Winata et al., 2020b) (Xiao et al., 2021)	(Lux and Vu, 2021)	Learning to optimize: (Klejch et al., 2018) Network architecture search: (Chen et al., 2020c) (Baruwa et al., 2019)
Source Separation	(Wu et al., 2021)		
Keyword Spotting	(Chen et al., 2020b)		Network architecture search: (Mazzawi et al., 2019)
Sound Event Detection		(Shimada et al., 2020) (Chou et al., 2019)	
Voice Cloning			Learning the learning algorithm: (Chen et al., 2019b) (Serrà et al., 2019)
Multi-tasks			Learning to select data: (Tarunesh et al., 2021)
Multi-modal		(Eloff et al., 2019)	Learning the learning algorithm: (Surís et al., 2019) (Xu et al., 2021a)

Learning to optimize for NLP

Weijia Xu, et al., Soft Layer Selection with Meta-Learning for Zero-Shot Cross-Lingual Transfer, MetaNLP workshop at ACL, 2021

- Zero-shot cross-lingual transfer
- Meta-optimizer
  - Soft-select portion of pretrained parameters to be frozen during fine-tuning
  - Parameterized by λ $\quad \boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} - \boldsymbol{\lambda} \odot \Delta \boldsymbol{\theta}^t$
  - Learn λ episodically similar to MAML (simulating zero-shot transfer scenario)

Training Task

En | Fr | De | ……

Testing Task

Zh | Hi

- Results
  - NLI on XNLI dataset
  - Meta-optimizer > (vanilla) fine-tuning, X-MAML

	fr	es	de	ar	ur	bg	sw	th	tr	vi	zh	ru	el	hi	avg
Devlin et al. (2019)	–	74.30	70.50	62.10	58.35	–	–	–	–	–	63.80	–	–	–	–
Wu and Dredze (2019)	74.60	74.90	72.00	66.10	58.60	69.80	49.40	55.70	62.00	71.90	70.40	69.80	67.90	61.20	66.02
Nooralahzadeh et al. (2020)	74.42	75.07	71.83	66.05	61.51	69.45	49.76	55.39	61.20	71.82	71.11	70.19	67.95	62.20	66.28
Aux. language	el	el	el	el	el	el	el	el	el	el	ur	ur	ur	ur	
Fine-tuning baseline	75.42	75.77	72.57	67.22	61.08	70.23	**51.70**	**51.03**	**64.26**	71.61	**72.52**	69.97	69.16	55.40	66.28
Meta-Optimizer	**75.78**	**75.87**	**73.15**	**67.34**	**62.00**	**70.47**	51.22	50.54	63.96	**72.06**	72.32	**70.20**	**69.34**	**55.88**	**66.44**
Aux. language: el + ur															
Fine-tuning baseline	74.87	75.78	72.27	66.96	62.73	70.16	50.21	48.20	63.86	71.61	71.97	70.24	69.64	56.04	66.04
Meta-Optimizer	**75.53**	**75.93**	**72.68**	**67.04**	**63.33**	**70.88**	**51.51**	**49.89**	**64.33**	**72.06**	**72.36**	**70.32**	**70.38**	**56.29**	**66.61**

Learning to optimize for NLP

Jen-Tzung Chien, et al., Meta learning for hyperparameter optimization in dialogue system, INTERSPEECH, 2019

- Dialog management
  - Usually many hyperparameters to tune
  - Gaussian process (GP) for HPO but slow

- Meta learn RNN for multifidelity Bayesian optimization
  - RNN: guide hyperparameter search
  - *h* = current state of search
  - *input* = hyperparameters and performance of algorithm trained with the hyperparameters
  - 2 level of RNNs: cheap, low fidelity; small-region, high fidelity search
- Experiments
  - Proposed approach (LSTM-MF) > GP in success rate, reward and faster to run

Method	Success rate	Reward
DQN	51.7%	3.5
DQN-VIME	53.5%	3.7
DQN-GP	55.1%	3.9
DQN-LSTM	55.8%	4.1
DQN-LSTM-MF	**56.2%**	**4.5**

	(A) **Learning to initialize**	(B) **Learning to compare**	(C) **Other**
Text Classification	(Dou et al., 2019) (Bansal et al., 2019) (Holla et al., 2020) (Zhou et al., 2021b) (van der Heijden et al., 2021) (Bansal et al., 2020) (Murty et al., 2021)	(Yu et al., 2018) (Tan et al., 2019) (Geng et al., 2019) (Sun et al., 2019) (Geng et al., 2020)	Learning the learning algorithm: (Wu et al., 2019) Network architecture search: (Pasunuru and Bansal, 2020) (Pasunuru and Bansal, 2019) Learning to optimize (Xu et al., 2021b) Learning to select data: (Zheng et al., 2021)
Sequence Labelng	(Wu et al., 2020) (Xia et al., 2021)	(Hou et al., 2020) (Yang and Katiyar, 2020) (Oguz and Vu, 2021)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Relation Classification	(Obamuyide and Vlachos, 2019) (Bose et al., 2019) (Lv et al., 2019)	(Ye and Ling, 2019) (Chen et al., 2019a) (Xiong et al., 2018a) (Gao et al., 2019) (Ren et al., 2020)	
Knowledge Graph Completion		(Xiong et al., 2018b) (Wang et al., 2019) (Zhang et al., 2020) (Sheng et al., 2020)	
Word Embedding	(Hu et al., 2019)	(Sun et al., 2018)	Network architecture search: (Li et al., 2020b) (Jiang et al., 2019)
Question Answering	(M'hamdi et al., 2021) (Nooralahzadeh et al., 2020) (Yan et al., 2020) (Hua et al., 2020)		
Machine Translation	(Gu et al., 2018) (Indurthi et al., 2020) (Li et al., 2020a) (Park et al., 2021)		Network architecture search: (Wang et al., 2020b) Learning to select data: (Wang et al., 2020d) (Pham et al., 2021)
	(Guo et al., 2019)		

Part III: Advanced topics in Meta learning for NLP

Advanced topics in Meta learning for NLP

- Data Selection
- Domain Generalization
- Task Augmentation
- Inference efficiency
- Meta knowledge distillation
- Mitigating catastrophic forgetting

Meta-learning for data selection

- Selecting from multi-lingual (& multi-task) corpora
  - Xinyi Wang, et al., Balancing Training for Multilingual Neural Machine Translation, ACL, 2020
  - Ishan Tarunesh, et al., Meta-Learning for Effective Multi-task and Multilingual Modelling, EACL, 2021
  - Hieu Pham, et al., Meta Back-Translation, ICLR, 2021
- Selecting from noisy labels
  - Guoqing Zheng, et al., Meta Label Correction for Noisy Label Learning, AAAI, 2021
  - Jun Shu, et al., Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting, NeurIPS, 2019

Selecting from multi-lingual corpora

Xinyi Wang, et al., Balancing Training for Multilingual Neural Machine Translation, ACL, 2020

Training Task

En-Fr | En-Es | En-Pt | ......

Testing Task

En-Aze | En-Bel

- Differential Data Selection (DDS)
  - Parameterize sampling strategies, the prob. of sampling task *i* = $P_{\mathcal{D}}(i) = e^{\psi_i} / \sum_j e^{\psi_j}$
  - Iteratively optimizing ψ with J and θ with L

$$\psi^* = \underset{\psi}{\operatorname{argmin}} \; J(\theta^*(\psi), \mathcal{D}_{dev})$$

$$\theta^*(\psi) = \underset{\theta}{\operatorname{argmin}} \; E_{x,y \sim P(T;\psi)}[l(x, y; \theta)]$$

  - Update ψ with REINFORCE (J is non-differentiable)

$$\psi_{t+1} \leftarrow \psi_t + R(x,y;\theta_t) \cdot \nabla_{\psi} log(P(x,y;\psi))$$

Selecting from multi-lingual corpora

Xinyi Wang, et al., Balancing Training for Multilingual Neural Machine Translation, ACL, 2020

Training Task

En-Fr | En-Es | En-Pt | ……

Testing Task

En-Aze | En-Bel

- Experiments
  - Model backbone = 6-layer transformers
  - 58-languages-to-English translation TED talk datasets[1] (across task train on all pairs and eval on 8 pairs separately)
  - DDS outperforms naïve sampling baselines

	Method	M2O Related	M2O Diverse
Baseline	Uni. (τ=∞)	22.63	24.81
	Temp. (τ=5)	24.00	26.01
	Prop. (τ=1)	24.88	26.68
Ours	MultiDDS	25.26	26.65
	MultiDDS-S	**25.52**	**27.00**

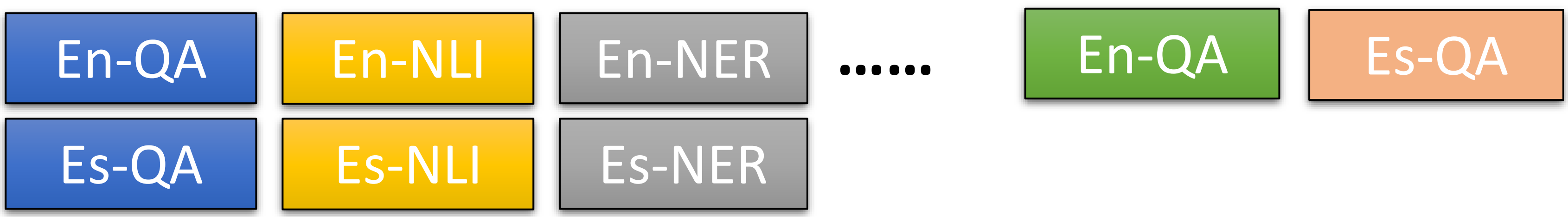
	Method	Avg.	aze	bel	glg	slk	tur	rus	por	ces
M2O	Prop.	24.88	11.20	17.17	27.51	28.85	**23.09***	**22.89**	**41.60**	26.80
	MultiDDS-S	**25.52**	**12.20***	**19.11***	**29.37***	**29.35***	22.81	22.78	41.55	**27.03**

[1] Ye Qi, et al., When and why are pre-trained word embeddings useful for neural machine translation?, NAACL, 2018

Selecting from multi-lingual & multi-task corpora

Ishan Tarunesh, et al., Meta-Learning for Effective Multi-task and Multilingual Modelling, EACL, 2021

Training Task ***Testing Task***



- Combine DDS with Reptile
- Extend the across task training to multi- tasks and languages
  - Tasks: QA, NLI, paraphrase identification, POS, and NER
  - Languages - en hi es de fr zh

Selecting from multi-lingual & multi-task corpora

Ishan Tarunesh, et al., Meta-Learning for Effective Multi-task and Multilingual Modelling, EACL, 2021

Training Task

En-QA | En-NLI | En-NER | ......

Es-QA | Es-NLI | Es-NER

Testing Task

En-QA | Es-QA

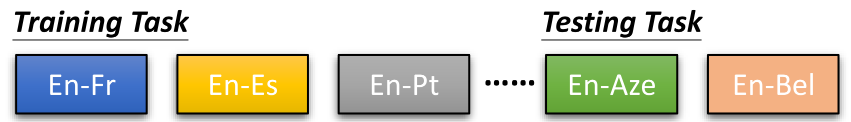
- Results
  - Meta-learned models outperform multi-tasks learning baselines (seen or unseen, i.e., zero-shot, target tasks/languages)

Model	SS	QA (F1) en	QA (F1) hi	QA (F1) es	QA (F1) de	NLI (Acc.) en	NLI (Acc.) es	NLI (Acc.) de	NLI (Acc.) fr	PA (Acc.) en	PA (Acc.) es	PA (Acc.) de	PA (Acc.) fr	PA (Acc.) zh
Baselines		79.94	59.94	65.83	63.17	81.39	78.37	76.82	77.30	92.35	89.75	87.45	89.61	83.32
Lang-Limited MTL		69.80	53.24	62.29	58.91	80.49	76.10	75.18	74.94	93.75	87.75	85.35	88.55	80.49
Task-Limited MTL		74.04	57.77	64.28	61.47	80.95	78.15	75.90	77.14	93.65	86.65	86.25	86.82	81.24
All TLPs MTL		63.22	42.94	54.05	51.61	80.05	76.48	74.86	76.18	93.50	90.30	88.45	89.71	82.66
Lang-Limited	Temp	-0.04	-0.24	-0.27	+0.07	+0.06	+0.39	+0.03	-0.70	+0.45	+0.05	+0.35	+0.40	-0.06
	mDDS	+0.07	-0.12	+0.06	+0.14	+0.02	-0.61	-0.80	-0.60	-0.25	-0.05	0.00	-0.30	-1.41
Task-Limited	Temp	+0.55	+0.43	+0.50	+0.40	+1.65	+1.12	+1.25	+0.79	+0.20	-0.15	-0.55	+0.85	-0.15
	mDDS	+0.21	+0.62	-0.67	+1.06	+1.32	+1.10	+1.39	+0.48	+0.50	-0.65	-0.35	+1.45	+1.06
All TLPs	Temp	+0.53	+0.47	+0.32	+0.47	+1.90	+1.22	+1.45	+0.95	+0.35	+0.45	+1.20	+1.05	+0.85
	mDDS-Lang	+0.08	+0.50	-1.57	+0.08	+0.76	+0.26	-0.10	+0.32	+0.25	+0.85	+0.75	+0.75	+1.11
	mDDS-Task	+0.18	+0.60	+0.11	+0.54	+1.50	+0.90	+0.72	+0.72	+0.10	+0.80	+1.27	+1.10	+1.16

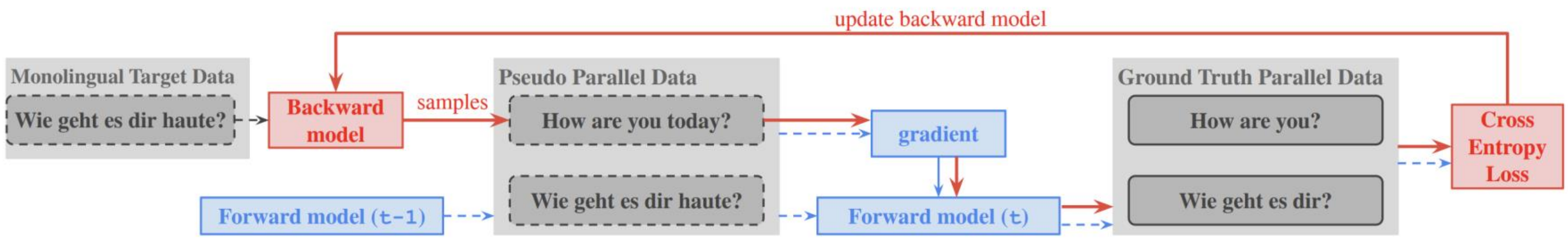
Model	SS	NER (Acc.) en	NER (Acc.) hi	NER (Acc.) es	NER (Acc.) de	NER (Acc.) fr	NER (Acc.) zh	POS (Acc.) en	POS (Acc.) hi	POS (Acc.) es	POS (Acc.) de	POS (Acc.) zh
Baselines		93.23	95.72	95.84	97.32	95.48	94.34	96.15	93.57	96.02	97.37	92.60
Lang-Limited MTL		92.54	92.67	95.14	96.40	94.38	92.97	95.08	92.43	95.19	97.19	89.71
Task-Limited MTL		93.51	93.94	95.77	97.09	95.27	93.72	95.70	93.34	95.73	97.35	92.52
All TLPs MTL		92.28	91.95	94.90	96.18	94.38	92.53	94.70	91.89	95.10	97.03	89.92
Lang-Limited	Temp	+0.60	+0.06	+0.09	+0.24	-0.09	-0.47	-0.06	-0.01	+0.10	+0.04	-0.17
	mDDS	-0.21	-0.85	-0.20	-0.10	-0.57	-0.55	-0.27	-0.02	-0.19	-0.06	-0.37
Task-Limited	Temp	+0.79	-0.46	0.00	-0.07	-0.18	-0.51	-0.22	-0.05	-0.21	+0.02	-0.09
	mDDS	-0.10	-1.61	0.00	-0.16	-0.33	-0.69	-0.38	-0.02	-0.22	+0.05	-0.12
All TLPs	Temp	-0.15	-0.70	+0.13	0.00	-0.16	-0.39	-0.22	-0.09	-0.21	+0.03	-0.16
	mDDS-Lang	-0.16	-0.09	+0.11	-0.08	-0.14	-0.65	-0.21	-0.10	-0.11	+0.03	-0.17
	mDDS-Task	-0.27	-0.42	+0.08	-0.14	-0.07	-0.58	-0.22	-0.14	-0.19	+0.02	-0.09

Selecting from multi-lingual corpora

Hieu Pham, et al., Meta Back-Translation, ICLR, 2021



- Formulate back translation as data sampling
  - y / x utterances in target (T) / source (S) languages
  - Generate x with y and $\widehat{P}(\mathbf{x}|\mathbf{y}) \triangleq P(\mathbf{x}|\mathbf{y};\psi)$
  - Train $P(\mathbf{y}|\mathbf{x};\theta)$ with (generated) x and y



  - Inner loop $\theta^*(\psi) = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{y\sim\mathrm{Uniform}(D_T)}\mathbb{E}_{x\sim\widehat{P}(\mathbf{x}|y)}[\ell(x,y;\theta)]$
  - Outer loop $\psi^* = \underset{\psi}{\operatorname{argmax}}\, \mathrm{Performance}(\theta^*(\psi), D_{\mathrm{MetaDev}})$

- Multilingual settings
  - Back translate T -> S and T -> S'
- Back translate vs. DDS
  - Granularity: sampling weights on tokens vs. examples/corpora

Selecting from multi-lingual corpora

Hieu Pham, et al., Meta Back-Translation, ICLR, 2021

- Experiments
  - Model backbone = transformer-base
  - 58-languages-to-English translation TED talk datasets[1] (across task train on all pairs and eval on 4 pairs separately)

BT Model Objective	Multilingual az-en	be-en	gl-en	sk-en
No BT	11.50	17.00	28.44	28.19
MLE (Edunov et al., 2018)	11.30	17.40	29.10	28.70
DualNMT (Xia et al., 2016)	11.69	14.81	25.30	27.07
Meta Back-Translation	**11.92***	**18.10***	**30.30***	**29.00**

[2]

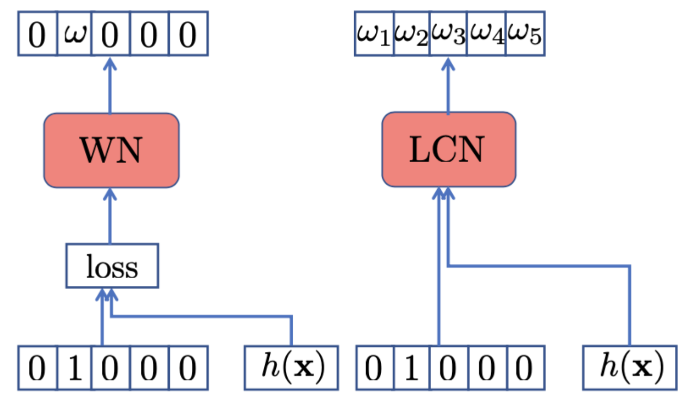
	Method	Avg.	aze	bel	glg	slk	tur	rus	por	ces
M2O	Prop.	24.88	11.20	17.17	27.51	28.85	**23.09***	**22.89**	**41.60**	26.80
	MultiDDS-S	**25.52**	**12.20***	**19.11***	**29.37***	**29.35***	22.81	22.78	41.55	**27.03**

[1] Ye Qi, et al., When and why are pre-trained word embeddings useful for neural machine translation?, NAACL, 2018
[2] Xinyi Wang, et al., Balancing Training for Multilingual Neural Machine Translation, ACL, 2020 (DDS)

Selecting from noisy labels

[1] *Jun Shu, et al., Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting, NeurIPS, 2019*
[2] *Guoqing Zheng, et al., Meta Label Correction for Noisy Label Learning, AAAI, 2021*

- Noisy labels
  - Meta-learner predicts weights[1] / rewrites labels[2] based on noisy labels and representation of input x
  - α, w: meta-parameters & parameters
  - y', y^c: noisy/corrected labels
  - 1, 2, 3, 4: inner loop
  - y_j, x_j: (clean) examples from meta-training set
  - 5, 6: outer loop



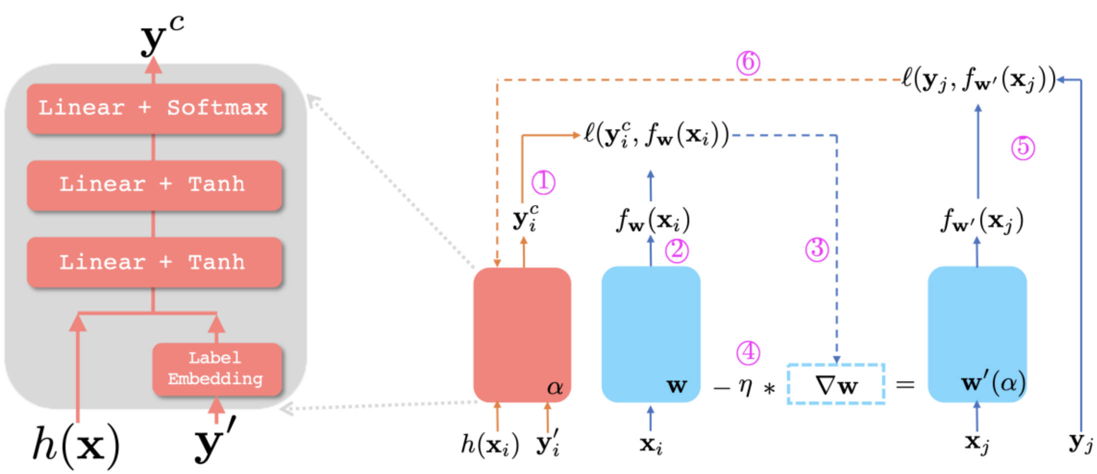
Training Task

Tr-ep 1 | Tr-ep 2 | Tr-ep 3

……

Testing Task

Val-ep 1 | Val-ep 2



Selecting from noisy labels

[1] *Jun Shu, et al., Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting, NeurIPS, 2019*
[2] *Guoqing Zheng, et al., Meta Label Correction for Noisy Label Learning, AAAI, 2021*

- Experiments
  - Real noise on image classification (Clothing1M dataset)
  - Meta-selection > vanilla training

Method	Forward (Patrini et al. 2017)	Joint Learning (Tanaka et al. 2018)	MLNT (Li et al. 2019)	MW-Net [1]	GLC (Hendrycks et al. 2018)	MLC [2]
Accuracy	69.84	72.23	73.47	73.72	73.69	**75.78**

  - Text classification, synthesized noise (2 types and 10 levels / probabilities)
  - AG news, Amazon reviews, Yelp reviews and Yahoo answers
  - No comparison to vanilla training

Datasets (# clean labels)	**AG** (4×100)	**Yelp-5** (5×100)	**Amazon-5** (5×100)	**Yahoo** (10×100)
MW-Net [1]	75.91	51.27	49.49	60.18
GLC (Hendrycks et al. 2018)	83.88	60.12	60.31	68.03
MLC [2]	**85.27**	**62.61**	**61.21**	**73.72**

Meta Learning for Domain Generalization

Domain Shift

- Training examples and testing examples have different distributions. → Domain shift

cat dog

Training Examples

Testing Examples

Can meta learning help?

Domain Shift

Domain Adaptation

Testing Examples

Target domain

Training Examples

cat

dog

cat

dog

dog

Source domain

Target domain

- Use little data from target domain to adapt.
- This is a few-shot learning problem.

→ It is intuitive to apply meta learning here.

Domain Shift

Testing Examples

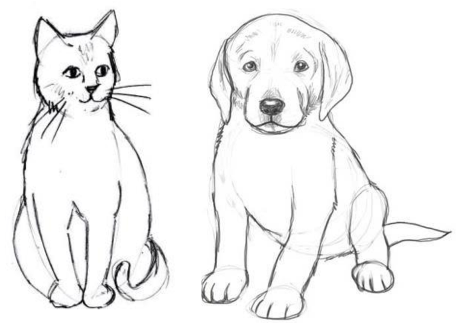
Target domain

Domain Generalization

Training Examples

cat dog

Domain 1



cat dog

Domain 2

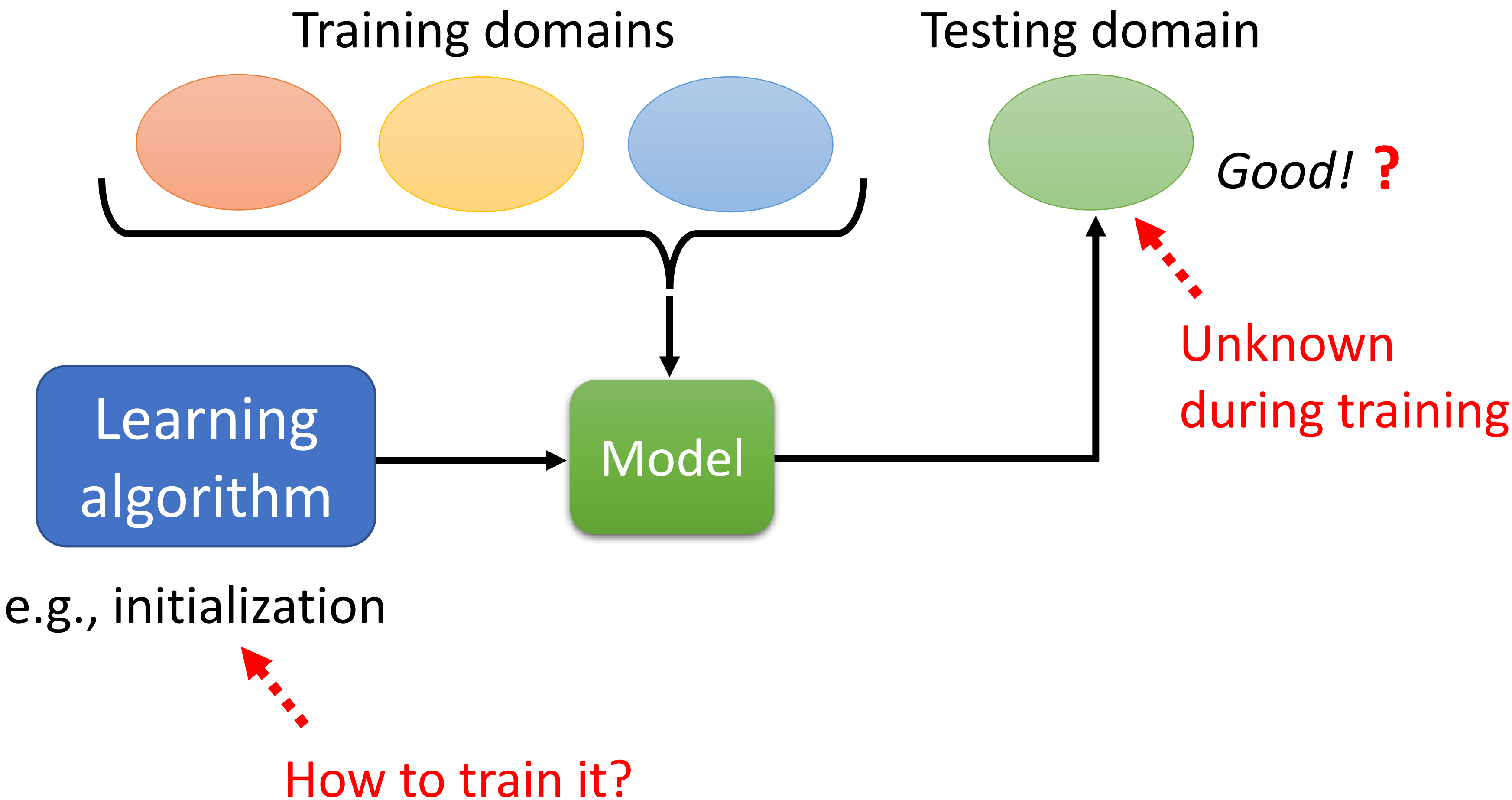
cat dog

Domain 3

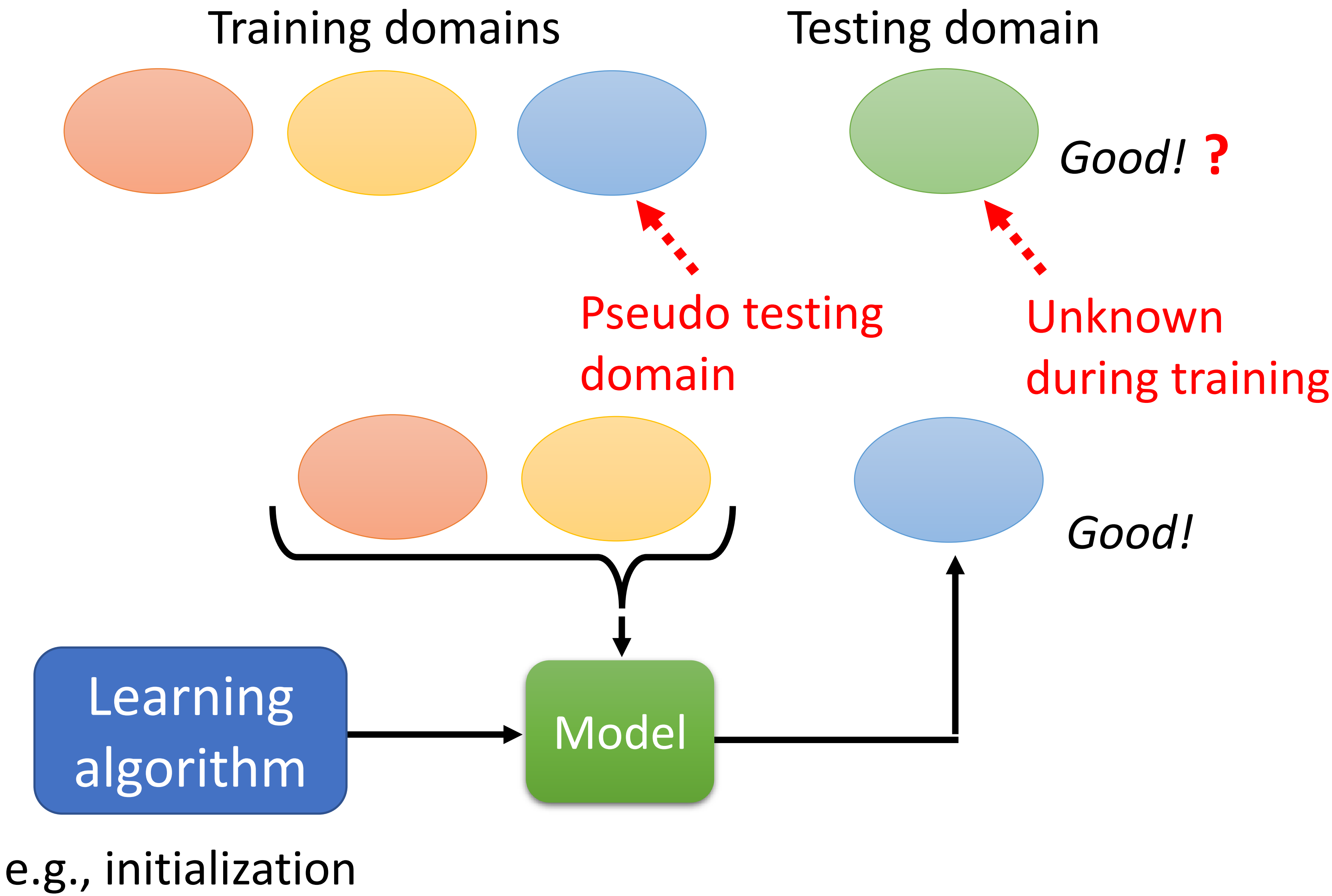
- The training data may include multiple domains.
- But we know nothing about the target domain.

How to use meta learning to improve domain generalization?

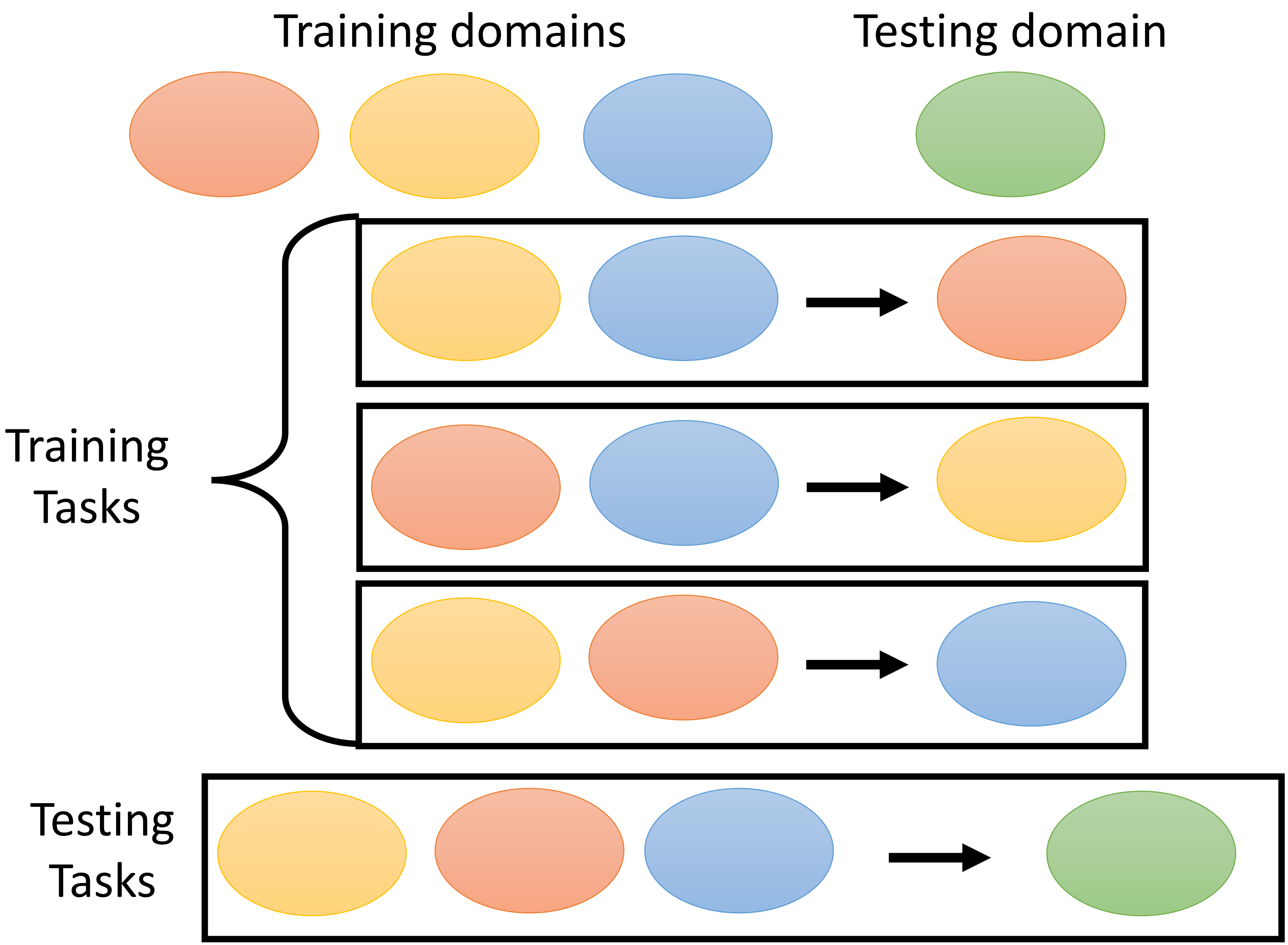
Meta Learning for Domain Generalization



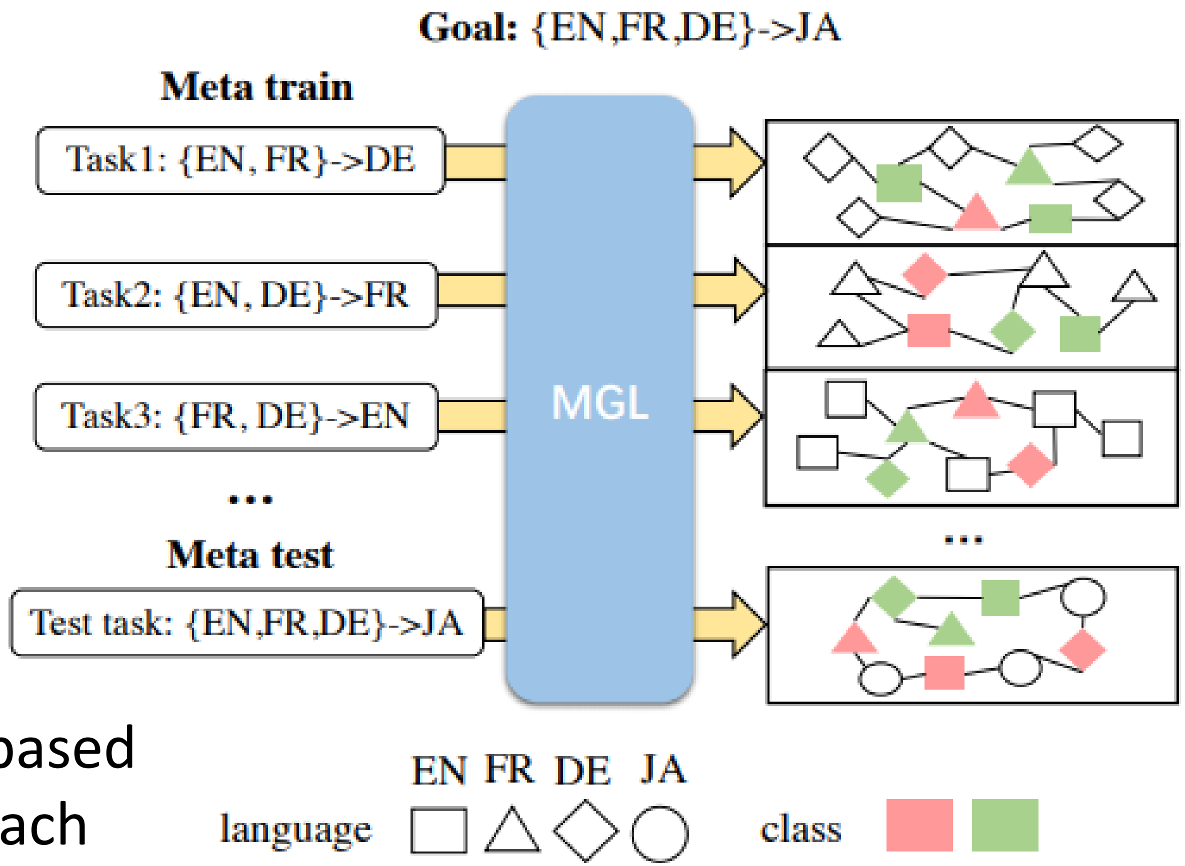
Meta Learning for Domain Generalization



Meta Learning for Domain Generalization



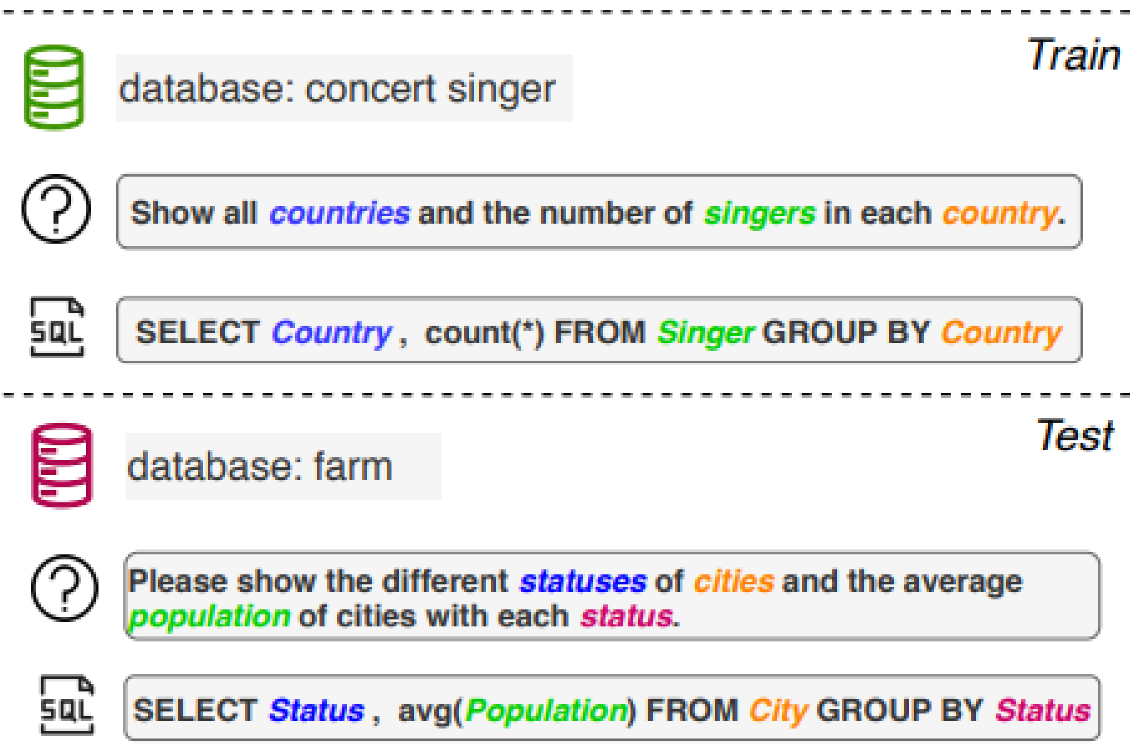
Example – Text Classification



Metric-based Approach

Zheng Li, Mukul Kumar, William Headden, Bing Yin, Ying Wei, Yu Zhang, Qiang Yang, Learn to Cross-lingual Transfer with Meta Graph Learning Across Heterogeneous Languages, EMNLP, 2020

Example – Semantic Parsing



Bailin Wang, Mirella Lapata, Ivan Titov, Meta-Learning for Domain Generalization in Semantic Parsing, NAACL, 2021
Henry Conklin, Bailin Wang, Kenny Smith, Ivan Titov, Meta-Learning to Compositionally Generalize, ACL 2021

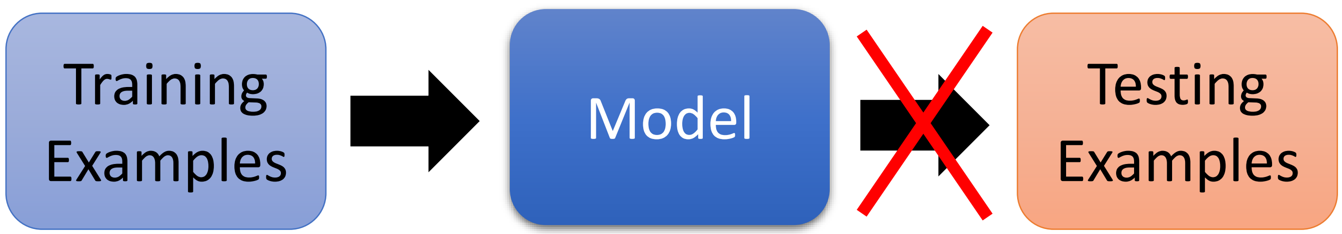
To learn more …

- Da Li, Yongxin Yang, Yi-Zhe Song, Timothy M. Hospedales, Learning to Generalize: Meta-Learning for Domain Generalization, AAAI 2018
- Yogesh Balaji, Swami Sankaranarayanan, Rama Chellappa, MetaReg: Towards Domain Generalization using Meta-Regularization, NeurIPS, 2018
- Fengchun Qiao, Long Zhao, Xi Peng, Learning to Learn Single Domain Generalization, CVPR, 2020
- Vinay Kumar Verma, Dhanajit Brahma, Piyush Rai, Meta-Learning for Generalized Zero-Shot Learning, AAAI, 2020
- Yun Li, Zhe Liu, Lina Yao, Xianzhi Wang, Can Wang, Attribute-Modulated Generative Meta Learning for Zero-Shot Classification, arXiv, 2021

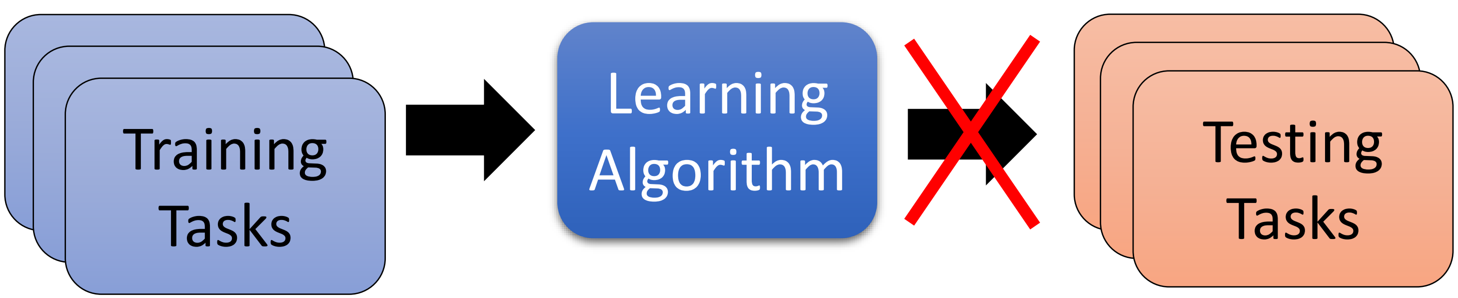
(general idea of applying meta learning to domain generalization, not related to HLP)

Problem of another level ……

- The training examples and testing examples may have different distributions.



- The training tasks and testing tasks can also have different distributions.



Huaxiu Yao, Longkai Huang, Linjun Zhang, Ying Wei, Li Tian, James Zou, Junzhou Huang, Zhenhui Li, Improving generalization in meta-learning via task augmentation, ICML, 2021

Advanced Topics in Meta Learning for NLP: Task Augmentation

Thang Vu

The Main Motivation

- Generate tasks to be able to leverage the advantages of meta learning methods
- Generate tasks to improve the performance of meta learning and to overcome overfitting problem

The Main Motivation

- Generate tasks to be able to leverage the advantages of meta learning methods
- Generate tasks to improve the performance of meta learning and to overcome overfitting problem

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

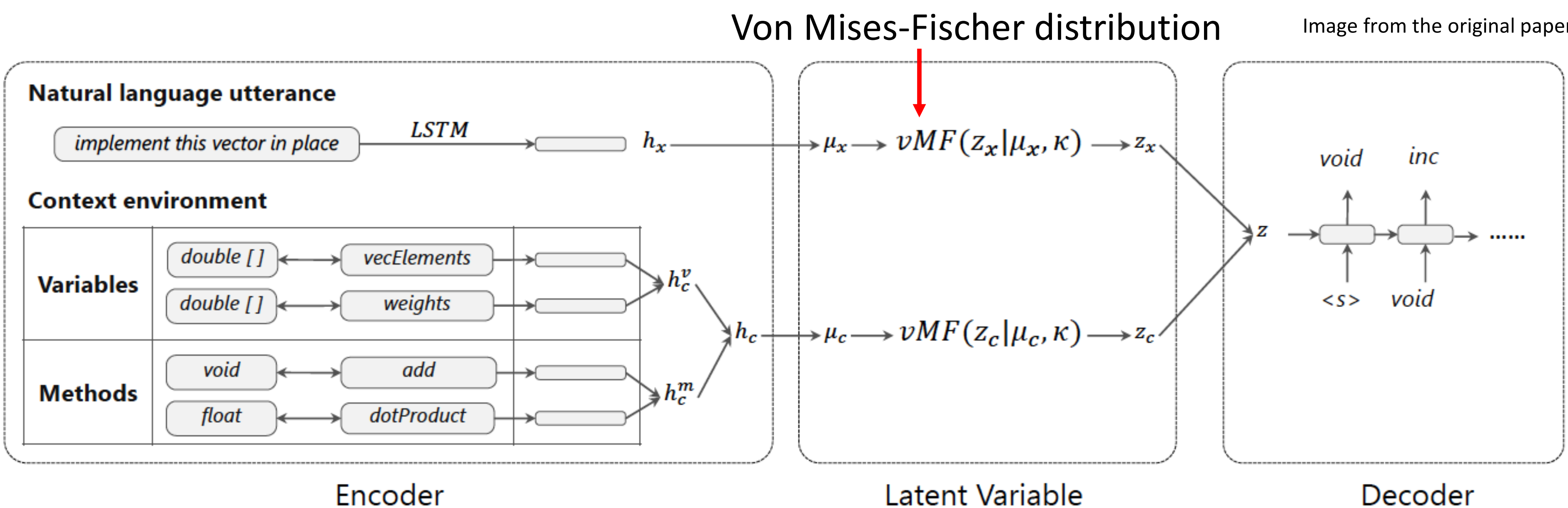
Coupling Retrieval and Meta-Learning for Context-Dependent Semantic Parsing

- 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

Image from the original paper



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

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

The Main Motivation

- Generate tasks to be able to leverage the advantages of meta learning methods
- Generate tasks to improve the performance of meta learning and to overcome overfitting problem

Self-Supervised Meta-Learning for Few-Shot Natural Language Classification Tasks

- Key ideas and take-home messages
  - Generate tasks called Subset Masked Language Modeling Tasks from unlabelled text

Subset: {Democratic, Capital}

Support set

Sentence	Class
A member of the [m] Party, he was the first African American to be elected to the presidency.	1
The [m] Party is one of the two major contemporary political parties in the United States, along with its rival, the Republican Party.	1
Honolulu is the [m] and largest city of the U.S. state of Hawaii.	2
Washington, D.C., formally the District of Columbia and commonly referred to as Washington or D.C., is the [m] of the United States.	2

Query: New Delhi is an urban district of Delhi which serves as the [m] of India

Correct Prediction: 2

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, Andrew McCallum. Self-supervised Meta-Learning for Few-Shot Natural Language Classsification Tasks. EMNLP 2020.

Self-Supervised Meta-Learning for Few-Shot Natural Language Classification Tasks

Subset: {**Democratic**, **Capital**}

Support set

Sentence	Class
A member of the [m] Party, he was the first African American to be elected to the presidency.	1
The [m] Party is one of the two major contemporary political parties in the United States, along with its rival, the Republican Party.	1
Honolulu is the [m] and largest city of the U.S. state of Hawaii.	2
Washington, D.C., formally the District of Columbia and commonly referred to as Washington or D.C., is the [m] of the United States.	2

Query: New Delhi is an urban district of Delhi which serves as the [m] of India

Correct Prediction: 2

Define N classes
by choosing N unique words

Consider all sentences which contain these words and choose randomly a subset for training

Mask the chosen words with [m]

Self-Supervised Meta-Learning for Few-Shot Natural Language Classification Tasks

Task	N	k	BERT	**SMLMT**	MT-BERT$_{\text{softmax}}$	MT-BERT	LEOPARD	**Hybrid-SMLMT**
CoNLL	4	4	50.44 $\pm$ 08.57	46.81 $\pm$ 4.77	52.28 $\pm$ 4.06	55.63 $\pm$ 4.99	54.16 $\pm$ 6.32	**57.60** $\pm$ 7.11
		8	50.06 $\pm$ 11.30	61.72 $\pm$ 3.11	65.34 $\pm$ 7.12	58.32 $\pm$ 3.77	67.38 $\pm$ 4.33	**70.20** $\pm$ 3.00
		16	74.47 $\pm$ 03.10	75.82 $\pm$ 4.04	71.67 $\pm$ 3.03	71.29 $\pm$ 3.30	76.37 $\pm$ 3.08	**80.61** $\pm$ 2.77
		32	83.27 $\pm$ 02.14	84.01 $\pm$ 1.73	73.09 $\pm$ 2.42	79.94 $\pm$ 2.45	83.61 $\pm$ 2.40	**85.51** $\pm$ 1.73
MITR	8	4	49.37 $\pm$ 4.28	46.23 $\pm$ 3.90	45.52 $\pm$ 5.90	50.49 $\pm$ 4.40	49.84 $\pm$ 3.31	**52.29** $\pm$ 4.32
		8	49.38 $\pm$ 7.76	61.15 $\pm$ 1.91	58.19 $\pm$ 2.65	58.01 $\pm$ 3.54	62.99 $\pm$ 3.28	**65.21** $\pm$ 2.32
		16	69.24 $\pm$ 3.68	69.22 $\pm$ 2.78	66.09 $\pm$ 2.24	66.16 $\pm$ 3.46	70.44 $\pm$ 2.89	**73.37** $\pm$ 1.88
		32	78.81 $\pm$ 1.95	78.82 $\pm$ 1.30	69.35 $\pm$ 0.98	76.39 $\pm$ 1.17	78.37 $\pm$ 1.97	**79.96** $\pm$ 1.48

……

Rating Kitchen	3	4	34.76 $\pm$ 11.20	40.75 $\pm$ 7.33	40.41 $\pm$ 5.33	36.77 $\pm$ 10.62	50.21 $\pm$ 09.63	**52.13** $\pm$ 10.18
		8	34.49 $\pm$ 08.72	43.04 $\pm$ 5.22	48.35 $\pm$ 7.87	47.98 $\pm$ 09.73	53.72 $\pm$ 10.31	**58.13** $\pm$ 07.28
		16	47.94 $\pm$ 08.28	46.82 $\pm$ 3.94	52.94 $\pm$ 7.14	53.79 $\pm$ 09.47	57.00 $\pm$ 08.69	**61.02** $\pm$ 05.55
		32	50.80 $\pm$ 04.52	51.71 $\pm$ 4.64	54.26 $\pm$ 6.37	53.23 $\pm$ 5.14	61.12 $\pm$ 04.83	**64.69** $\pm$ 02.40
Overall Average		4	38.13	40.95	40.13	40.10	45.99	**48.71**
		8	36.99	46.37	45.89	44.25	50.86	**53.70**
		16	48.55	51.61	49.93	49.07	55.50	**58.41**
		32	55.30	56.23	52.65	55.42	57.02	**60.81**

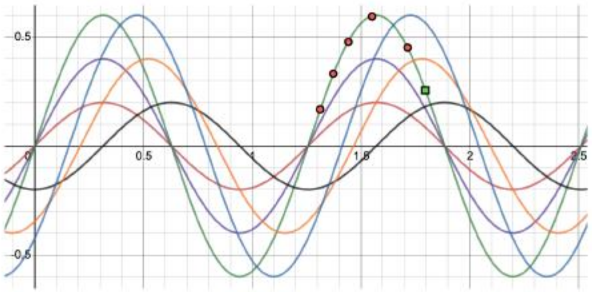
DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

- Key ideas and take-home messages:
  - Explore the overfitting problem of meta learning
  - Propose a task augmentation strategy
    - Apply clustering on BERT vectors to create tasks

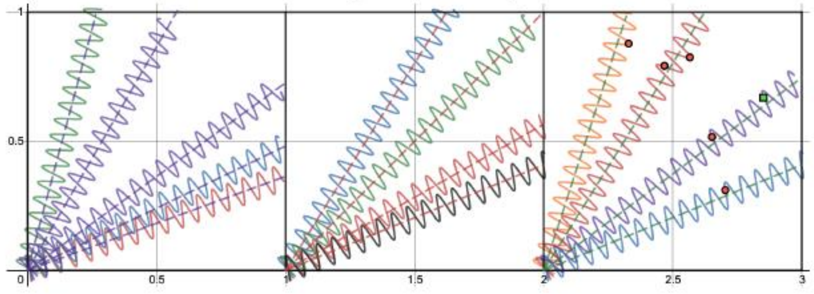
Shikhar Murty, Tatsunori B. Hashimoto, Christopher Manning. DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference. NAACL 2021.

DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

- Explore the overfitting problem of meta learning



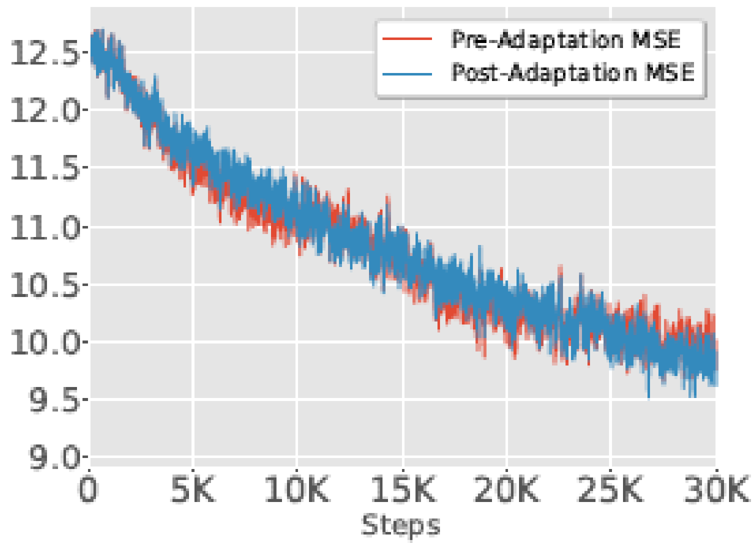
(a) 1D sine wave regression (Finn et al., 2017). Each task is a sine-wave with a fixed amplitude and phase offset.



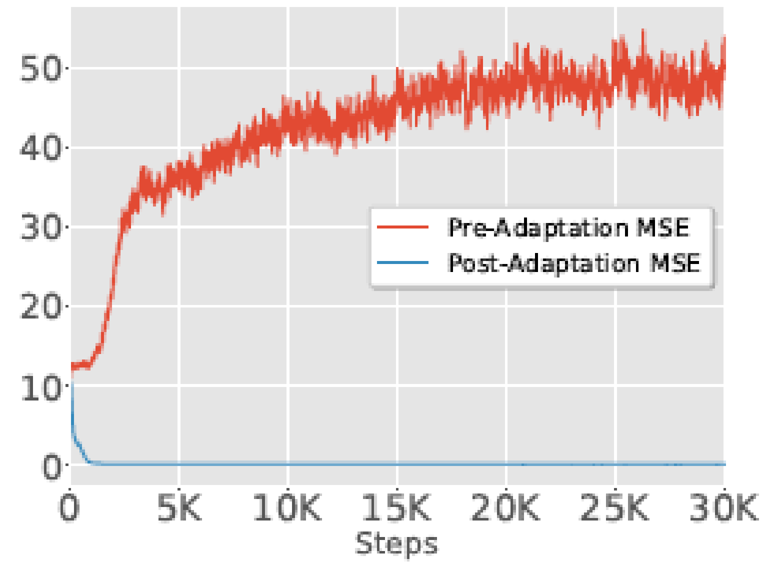
(b) Three datasets from our 2D sine wave regression. Each dataset is a unit square with multiple reasoning categories; A reasoning category is a distinct sinusoid along a ray that maps $x = (x_1, x_2)$ to the value of the sine-wave y at that point.

DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

- Explore the overfitting problem of meta learning



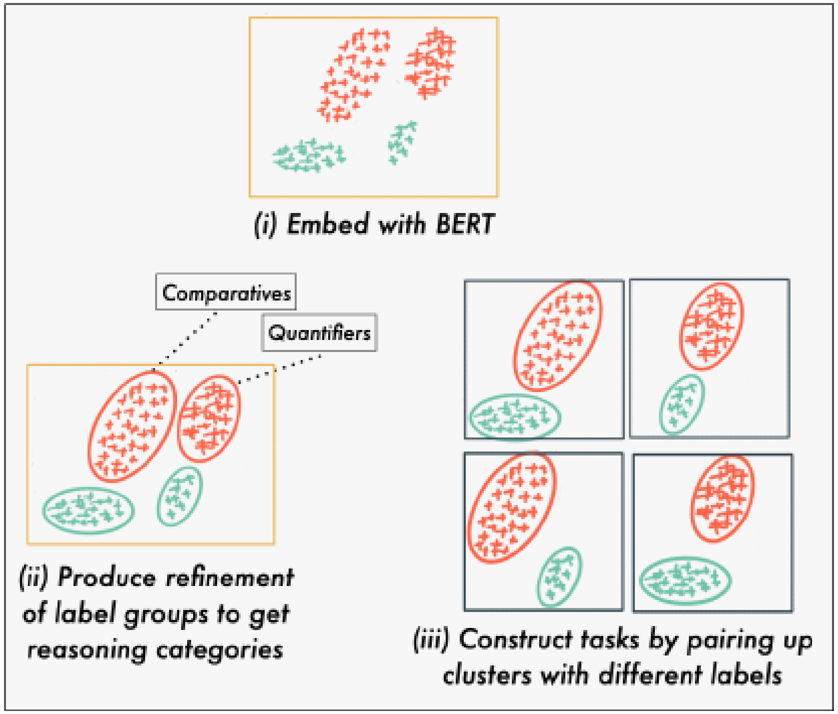
(a)



(b)

DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

- Apply clustering on BERT vectors to create tasks



DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

- Apply clustering on BERT vectors to create tasks

Model	CombinedNLI-QANLI	CombinedNLI-RTE	GLUE-SciTail
Multitask (Finetune)	69.66 ± 0.39	65.47 ± 3.19	75.80 ± 2.58
Multitask (k-NN)	68.97 ± 1.26	63.69 ± 6.65	69.76 ± 3.74
Multitask (Finetune + k-NN)	67.38 ± 2.61	66.52 ± 5.48	76.44 ± 1.77
MAML-Base	69.43 ± 0.81	72.61 ± 0.85	76.38 ± 1.25
SMLMT (Bansal et al., 2020b)	–	–	76.75 ± 2.08
MAML-DReCa	$\mathbf{71.98 \pm 0.79}$	$\mathbf{75.36 \pm 0.69}$	$\mathbf{77.91 \pm 1.60}$

DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

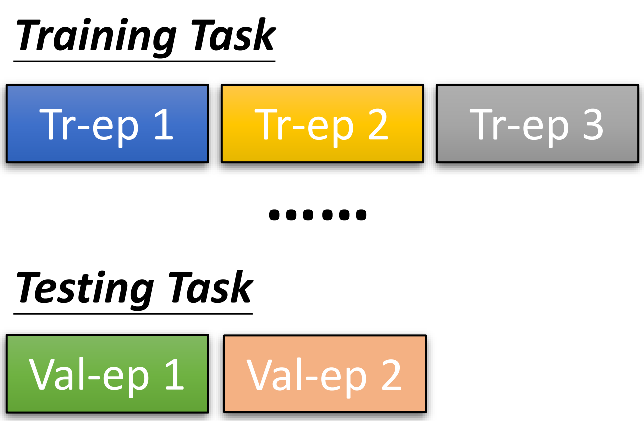
- Apply clustering on BERT vectors to create tasks

Model	CombinedNLI-QANLI	CombinedNLI-RTE	GLUE-SciTail
Multitask (Finetune)	69.66 ± 0.39	65.47 ± 3.19	75.80 ± 2.58
Multitask (k-NN)	68.97 ± 1.26	63.69 ± 6.65	69.76 ± 3.74
Multitask (Finetune + k-NN)	67.38 ± 2.61	66.52 ± 5.48	76.44 ± 1.77
MAML-Base	69.43 ± 0.81	72.61 ± 0.85	76.38 ± 1.25
SMLMT (Bansal et al., 2020b)	–	–	76.75 ± 2.08
MAML-DReCa	$\mathbf{71.98 \pm 0.79}$	$\mathbf{75.36 \pm 0.69}$	$\mathbf{77.91 \pm 1.60}$

Meta-learning for hardware-aware inference efficiency

Hanrui Wang, et al., HAT: Hardware-Aware Transformers for Efficient Natural Language Processing, ACL, 2020

- Search Transformers architecture based on hardware
- Efficient search
  - SuperTransformer for weight sharing (sampling searched architectures and inheriting weights from it)
  - Latency predictor inferring latency from architectures
- Evolution search
  - Deciding next generations based on inferred latency and performance on val.
  - Retrain searched architecture and evaluate on test



Meta-learning for hardware-aware inference efficiency

Hanrui Wang, et al., HAT: Hardware-Aware Transformers for Efficient Natural Language Processing, ACL, 2020

- Machine translation task
  - 3x speedup and 3x size reduction compared to Transformer
  - similar BLEU
  - 12000x faster than Evolved transformer (NAS baseline for searching transformer architecture)

		Hardware-Aware	Hetero. Layers	Latency	#Params	FLOPs (G)	BLEU	GPU Hours	CO_2e (lbs)	Cloud Comp. Cost
IWSLT'14 De-En	Transformer	✗	✗	3.3s	32M	1.5	34.5	2	5	$12 - $40
	HAT (Ours)	✓	✓	**2.1s**	**23M**	**1.1**	**34.5**	4	9	$24 - $80
WMT'14 En-Fr	Transformer	✗	✗	23.2s	176M	10.6	41.2	240	68	$178 - $595
	Evolved Trans.	✗	✗	20.9s	175M	10.8	41.3	2,192,000	626,000	$1.6M - $5.5M
	HAT (Ours)	✓	✓	**7.8s**	**48M**	**3.4**	**41.4**	216	61	$159 - $534
	HAT (Ours)	✓	✓	9.1s	57M	3.9	**41.8**	224	64	$166 - $555
WMT'14 En-De	Transformer	✗	✗	20.5s	176M	10.6	28.4	184	52	$136 - $456
	Evolved Trans.	✗	✗	7.6s	47M	2.9	28.2	2,192,000	626,000	$1.6M - $5.5M
	HAT (Ours)	✓	✓	**6.0s**	**44M**	**2.7**	28.2	184	52	$136 - $456
	HAT (Ours)	✓	✓	6.9s	48M	3.0	**28.4**	200	57	$147 - $495

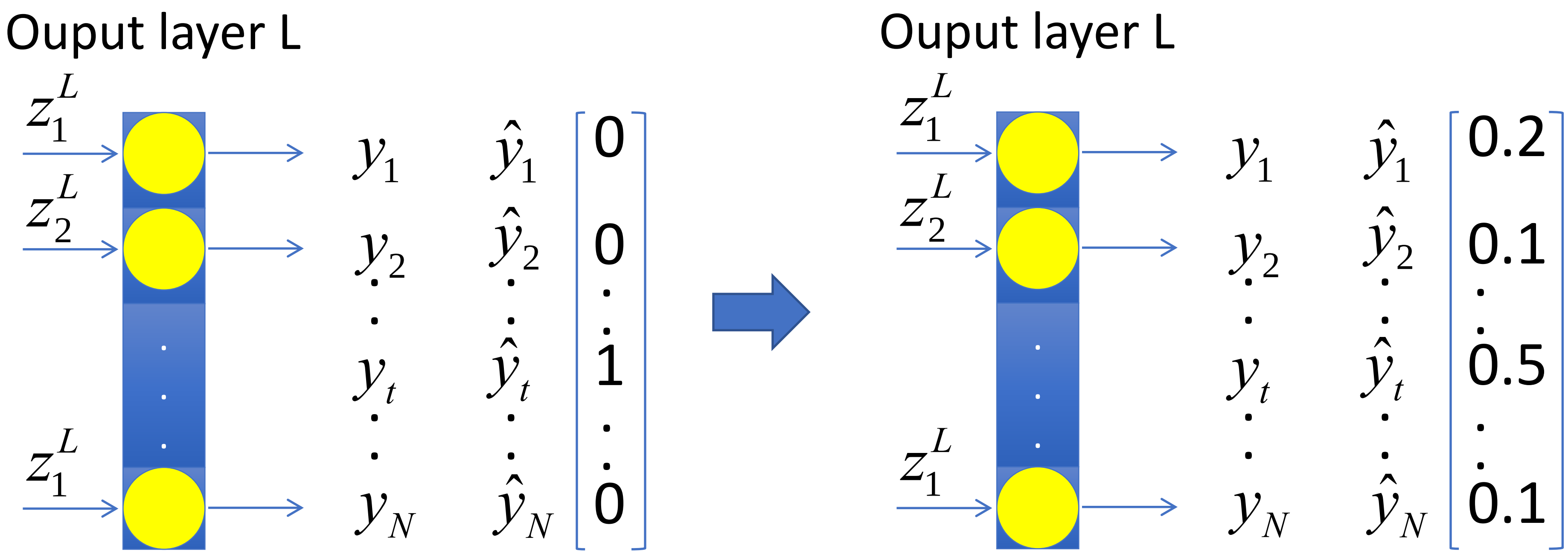
Table 2: Comparisons of latency, model size, FLOPs, BLEU and training cost in terms of CO_2 emissions (lbs) and cloud computing cost (USD) for Transformer, the Evolved Transformer and HAT. The training cost estimation is adapted from Strubell et al. (2019). The training time is for one Nvidia V100 GPU, and the latency is measured on the Raspberry Pi ARM CPU. The cloud computing cost is based on AWS.

Advanced Topics in Meta Learning for NLP: Meta Knowledge Distillation

Thang Vu

Knowledge Distillation [Hinton et al 2014]

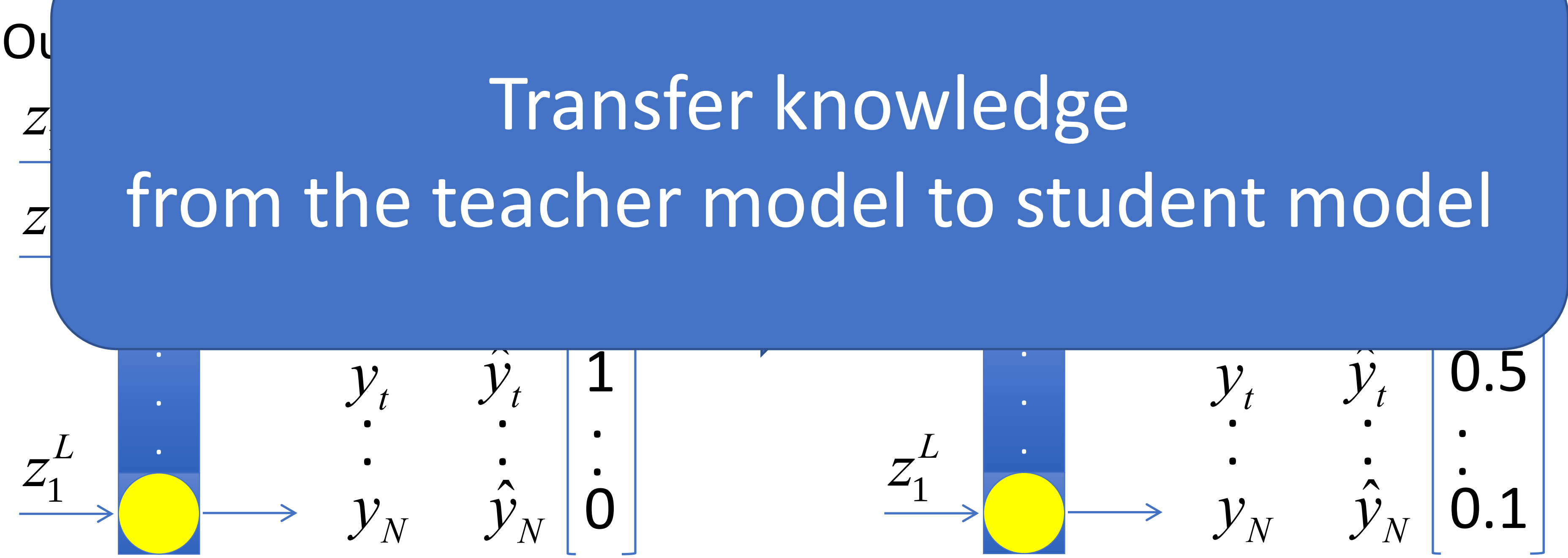
- Use the class probabilities produced by a teacher model as the soft target to train a student model



Knowledge Distillation [Hinton et al 2014]

- Use the class probabilities produced by a teacher model as the soft target to train a student model

Transfer knowledge
from the teacher model to student model

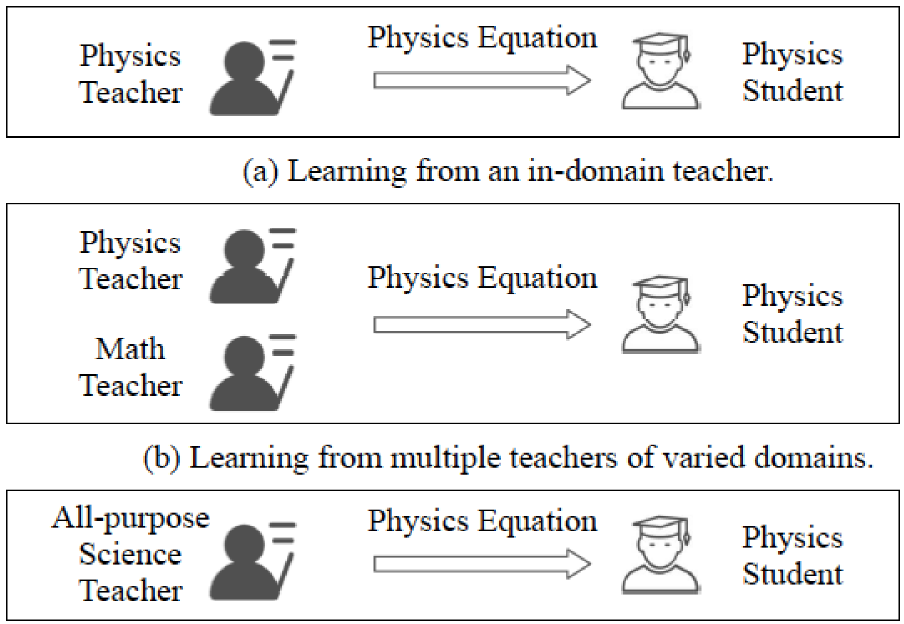


Meta Knowledge Distillation

Learn to Transfer knowledge
from the teacher model to student model

Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains

- High level ideas:

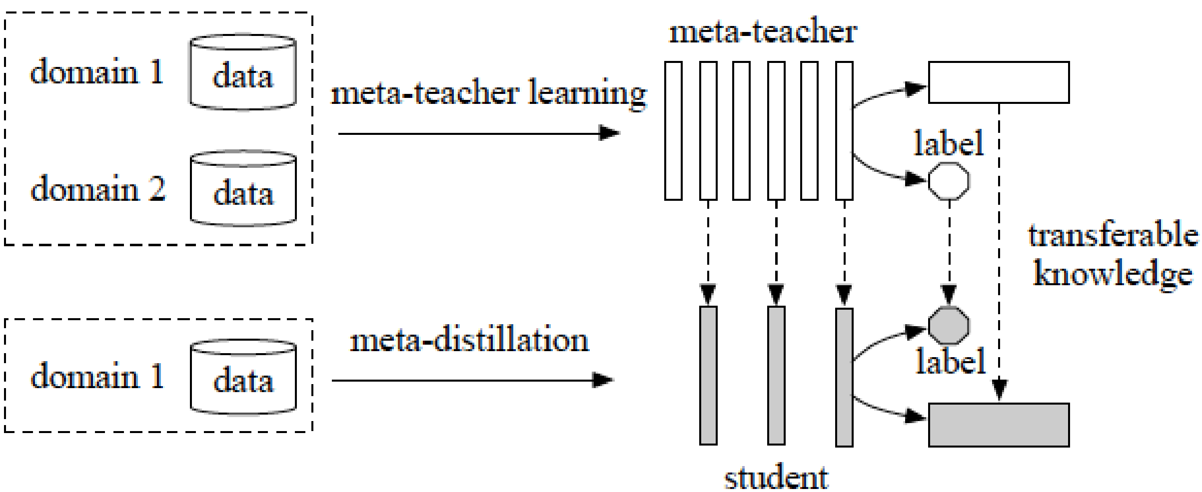


(c) Learning from the meta-teacher with multi-domain knowledge.

Haojie Pan, Chengyu Wang, Minghui Qiu, Yichang Zhang, Yaliang Ji, Hun Huang. Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains. Arxiv Dec 2020.

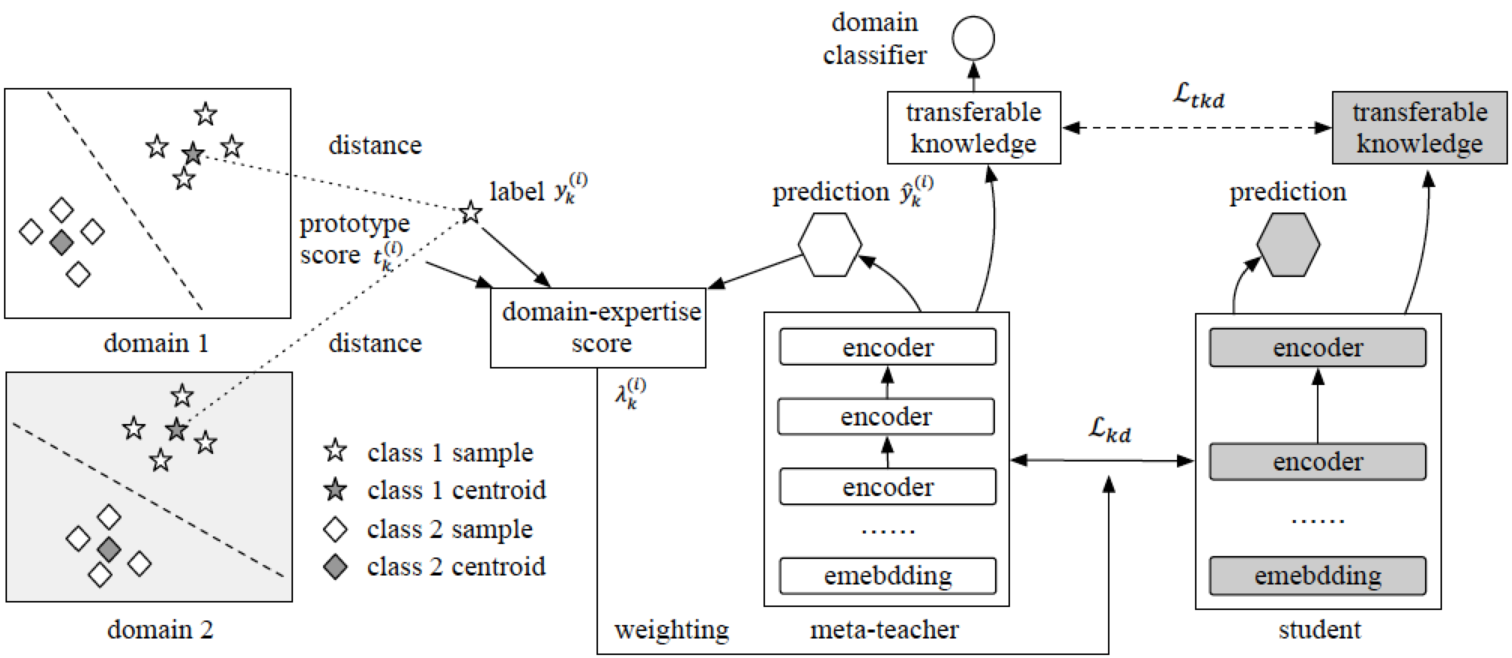
Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains

- High level ideas:

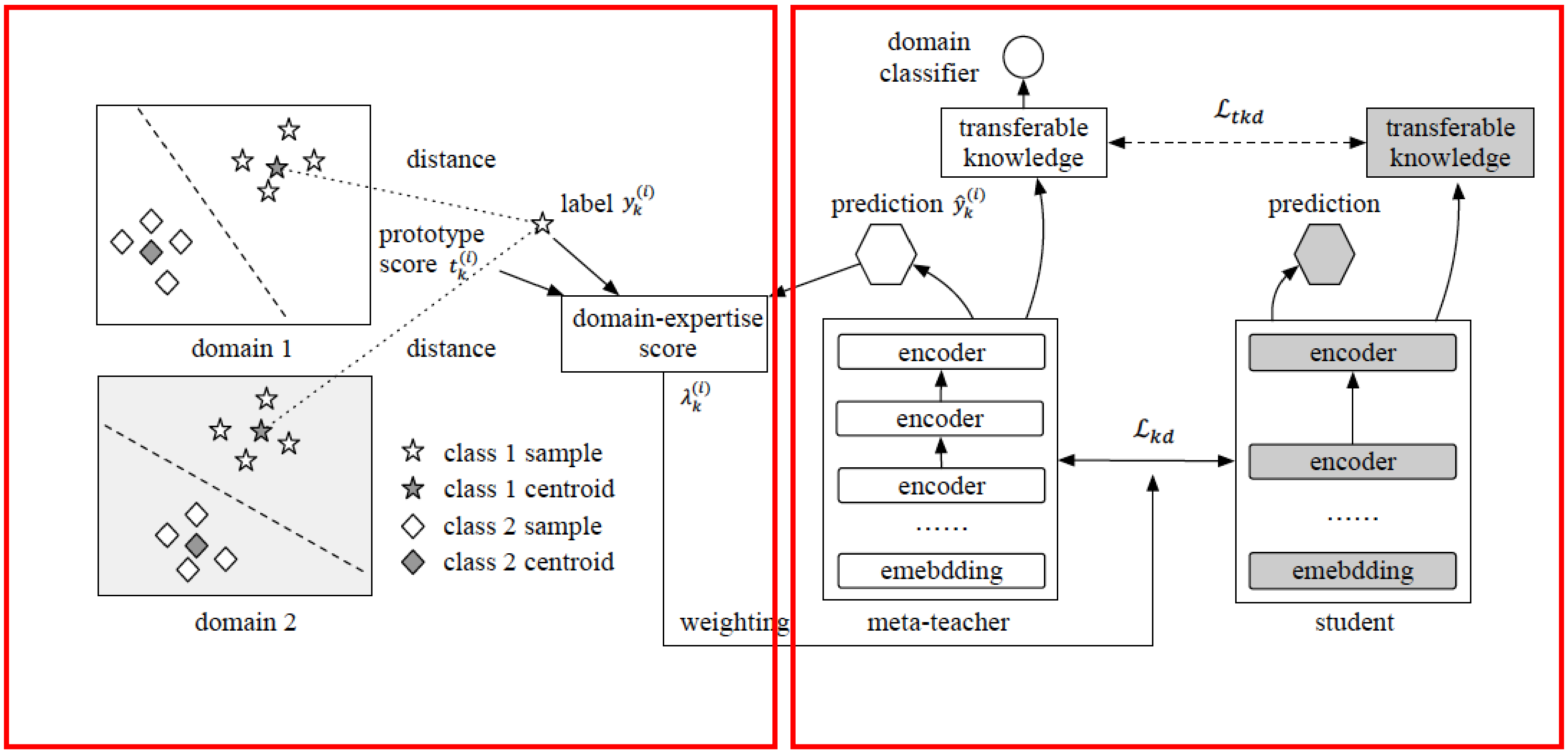


Haojie Pan, Chengyu Wang, Minghui Qiu, Yichang Zhang, Yaliang Ji, Hun Huang. Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains. Arxiv Dec 2020.

Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains



Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains



Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains

- Results on MNLI with five domains

Methods	Fiction	Government	Slate	Telephone	Travel	Average
BERT_B-single	82.2	84.2	76.7	82.4	84.2	81.9
BERT_B-mix	84.8	87.2	80.5	83.8	85.5	84.4
BERT_B-mtl	83.7	87.1	80.6	83.9	85.8	84.2
Meta-teacher	85.1	86.5	81.0	83.9	85.5	84.4
$\text{BERT}_\text{B}\text{-single} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	78.8	83.2	73.6	78.8	81.9	79.3
$\text{BERT}_\text{B}\text{-mix} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	79.6	83.3	74.8	79.0	81.5	79.6
$\text{BERT}_\text{B}\text{-mtl} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	79.7	83.1	74.2	79.3	82.0	79.7
$\text{Multi-teachers} \xrightarrow{\text{MTN-KD}} \text{BERT}_\text{S}$	77.4	81.1	72.2	77.2	78.0	77.2
$\text{Meta-teacher} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	80.3	83.0	**75.1**	80.2	81.6	80.0
$\text{Meta-teacher} \xrightarrow{\text{Meta-distillation}} \text{BERT}_\text{S}$	**80.5**	**83.7**	75.0	**80.5**	**82.1**	**80.4**

Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains

- Results on Amazon Review with four domains

Methods	Books	DVD	Electronics	Kitchen	Average
BERT_B-single	87.9	83.8	89.2	90.6	87.9
BERT_B-mix	89.9	85.9	90.1	92.1	89.5
BERT_B-mtl	90.5	86.5	91.1	91.1	89.8
Meta-teacher	92.5	87.0	91.1	89.2	89.9
$\text{BERT}_\text{B}\text{-single} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	83.4	83.2	89.2	91.1	86.7
$\text{BERT}_\text{B}\text{-mix} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	88.4	81.6	89.7	89.7	87.3
$\text{BERT}_\text{B}\text{-mtl} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	90.5	81.6	88.7	90.1	87.7
$\text{Multi-teachers} \xrightarrow{\text{MTN-KD}} \text{BERT}_\text{S}$	83.9	78.4	88.7	87.7	84.7
$\text{Meta-teacher} \xrightarrow{\text{TinyBERT-KD}} \text{BERT}_\text{S}$	89.9	84.3	87.3	**91.6**	88.3
$\text{Meta-teacher} \xrightarrow{\text{Meta Distillation}} \text{BERT}_\text{S}$	**91.5**	**86.5**	**90.1**	89.7	**89.4**

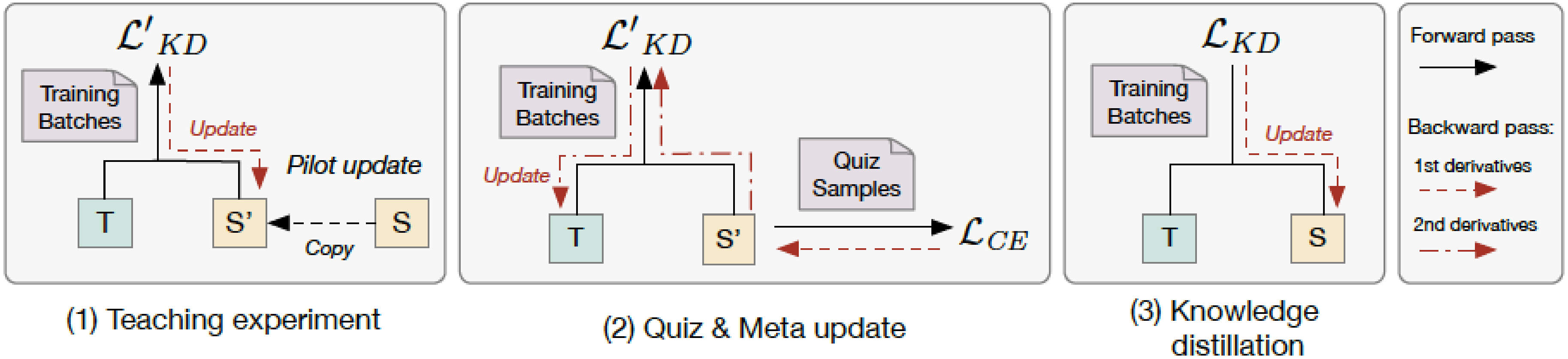
Meta Learning for Knowledge Distillation

- Starting point:
  - The teacher is unaware of the student
  - The teacher is not optimized for distillation
- High-level ideas:
  - Student-centered learning
  - Teacher models can be updated using feedback from student models
- Novelty:
  - propose pilot update that aligns the learning of the student and the teacher model

Wangchunshu Zhou, Canwen Xu, Julian McAuley. Meta Learning for Knowledge Distillation. Arxiv June 2021.

Meta Learning for Knowledge Distillation

- Key ideas and take-home messages



Wangchunshu Zhou, Canwen Xu, Julian McAuley. Meta Learning for Knowledge Distillation. Arxiv June 2021.

Meta Learning for Knowledge Distillation

- Results on dev sets

Method	**CoLA** (8.5K)	**MNLI** (393K)	**MRPC** (3.7K)	**QNLI** (105K)	**QQP** (364K)	**RTE** (2.5K)	**SST-2** (67K)	**STS-B** (5.7K)
				Dev. Set				
BERT-Base (teacher) (2019)	58.9	84.6/84.9	91.6/87.6	91.2	88.5/91.4	71.4	93.0	90.2/89.8
BERT-6L (student) (2019)	53.5	81.1/81.7	89.2/84.4	88.6	86.9/90.4	67.9	91.1	88.1/87.9
				Pretraining Distillation				
TinyBERT‡ (2019)	54.0	84.5/84.5	90.6/86.3	91.1	88.0/91.1	73.4	93.0	90.1/89.6
MiniLM (2020b)	49.2	84.0/ -	88.4/ -	91.0	- /91.0	71.5	92.0	-
MiniLM v2 (2020a)	52.5	84.2/ -	88.9/ -	90.8	- /91.1	72.1	92.4	-
				Task-specific Distillation				
KD† (2015)	53.9	82.7/83.2	89.8/85.2	89.4	87.4/90.7	67.6	91.4	88.5/88.1
PKD† (2019)	54.3	82.9/83.4	89.5/84.8	89.8	87.6/90.8	67.5	91.2	88.8/88.2
TinyBERT w/o DA†	52.5	83.5/83.8	90.6/86.4	89.7	87.8/90.9	67.9	91.8	89.1/88.7
RCO† (2019)	53.4	82.3/82.9	89.7/85.2	89.6	87.5/90.6	67.4	91.3	88.6/88.3
TAKD† (2020)	53.7	82.7/83.1	89.5/84.9	89.5	87.3/90.6	68.2	91.1	88.5/88.3
DML† (2018)	53.6	82.5/83.0	89.8/85.2	89.7	87.6/90.5	68.5	91.6	88.5/88.0
ProKT† (2021)	54.4	82.9/83.3	90.6/86.4	89.9	87.7/90.8	68.4	91.5	88.9/88.4
MetaDistil *(ours)*	**58.5**	**83.6/83.9**	**91.2/87.0**	**90.4**	**88.2/91.2**	**69.5**	**92.4**	**89.6/89.2**
w/o pilot update	56.4	83.2/83.6	90.8/86.7	90.0	88.1/88.7	67.8	92.1	89.3/89.1

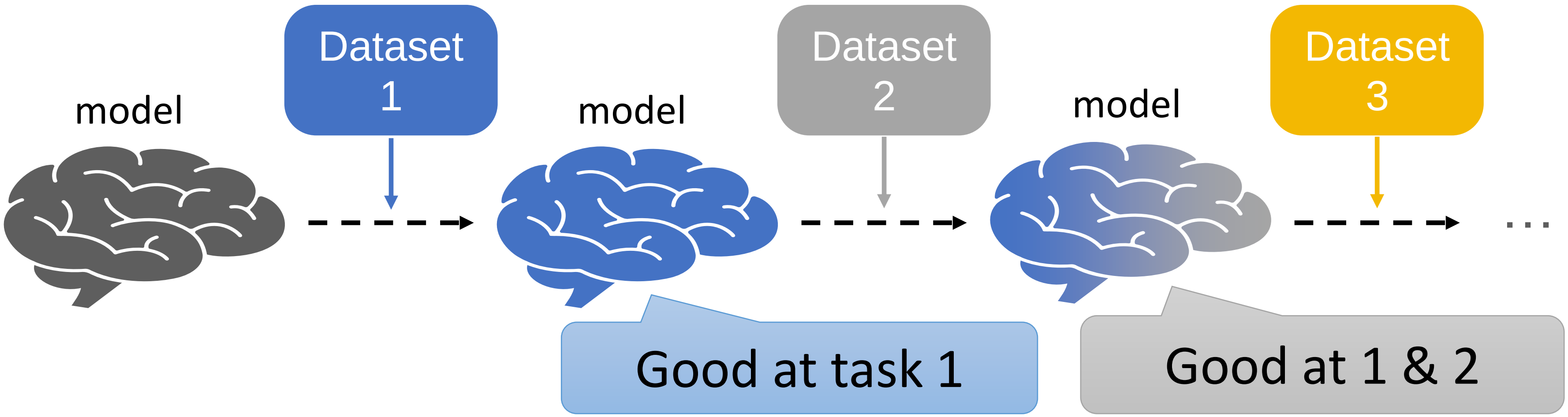
Meta Learning for Knowledge Distillation

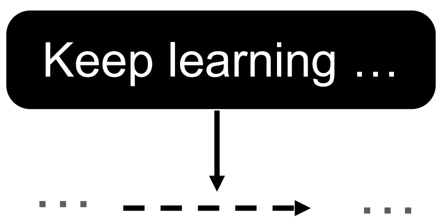
- Results on test sets

				Test Set				
BERT-Base (teacher) (2019)	52.1	84.6/83.4	88.9/84.8	90.5	71.2/89.2	66.4	93.5	87.1/85.8
				Pretraining Distillation				
DistilBERT (2019)	45.8	81.6/81.3	87.6/83.1	88.8	69.6/88.2	54.1	92.3	71.0/71.0
TinyBERT‡ (2019)	51.1	84.3/83.4	88.8/84.5	91.6	70.5/88.3	70.4	92.6	86.2/84.8
				Task-specific Distillation				
KD (2019)	-	82.8/82.2	86.8/81.7	88.9	70.4/88.9	65.3	91.8	-
PKD (2019)	43.5	81.5/81.0	85.0/79.9	89.0	70.7/88.9	65.5	92.0	83.4/81.6
Theseus (2020)	47.8	82.4/82.1	87.6/83.2	89.6	**71.6/89.3**	66.2	92.2	85.6/84.1
ProKT (2021)	-	82.9/82.2	87.0/82.3	89.7	70.9/88.9	-	93.3	-
DML† (2018)	48.5	82.6/81.6	86.5/81.2	89.5	70.7/88.7	66.3	92.7	85.5/84.0
RCO† (2019)	48.2	82.3/81.2	86.8/81.4	89.3	70.4/88.7	66.5	92.6	85.3/84.1
TAKD† (2020)	48.4	82.4/81.7	86.5/81.3	89.4	70.6/88.8	66.8	92.9	85.4/84.1
MetaDistil *(ours)*	**50.7**	**83.8/83.2**	**88.7/84.7**	**90.2**	71.1/88.9	**67.2**	**93.5**	**86.1/85.0**
w/o pilot update	49.1	83.3/82.8	88.2/84.1	89.9	71.0/88.7	66.6	**93.5**	85.9/84.6

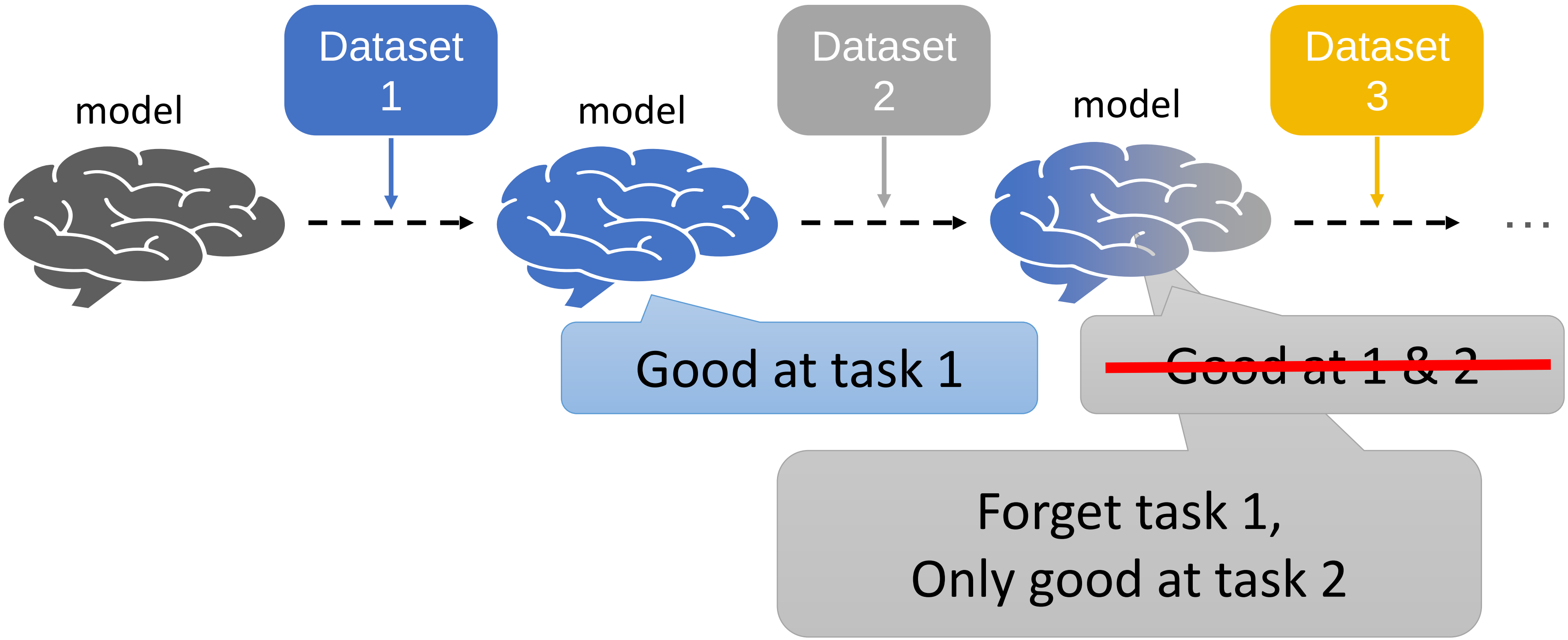
Mitigating Catastrophic Forgetting by Meta Learning

Lifelong Learning Scenario



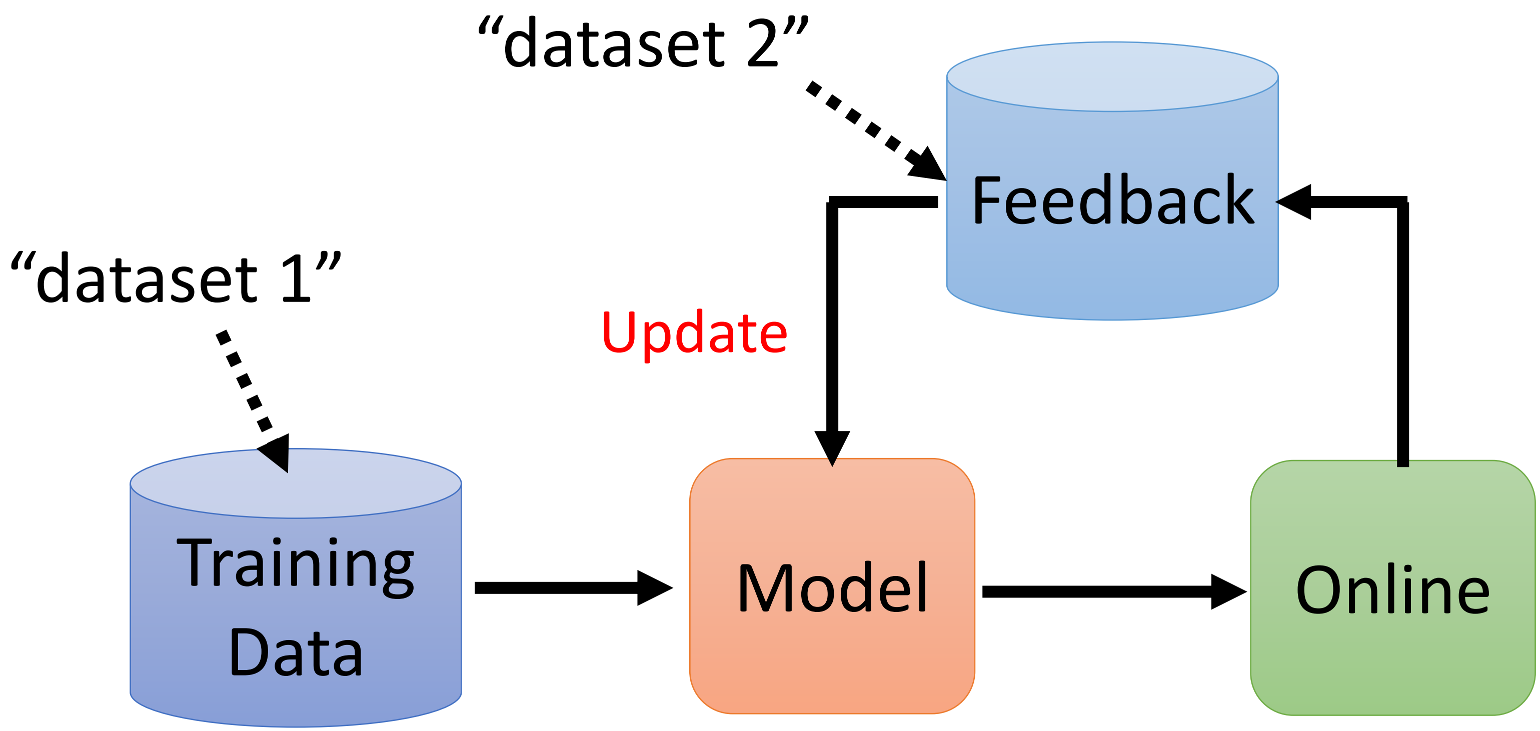


Lifelong Learning Scenario

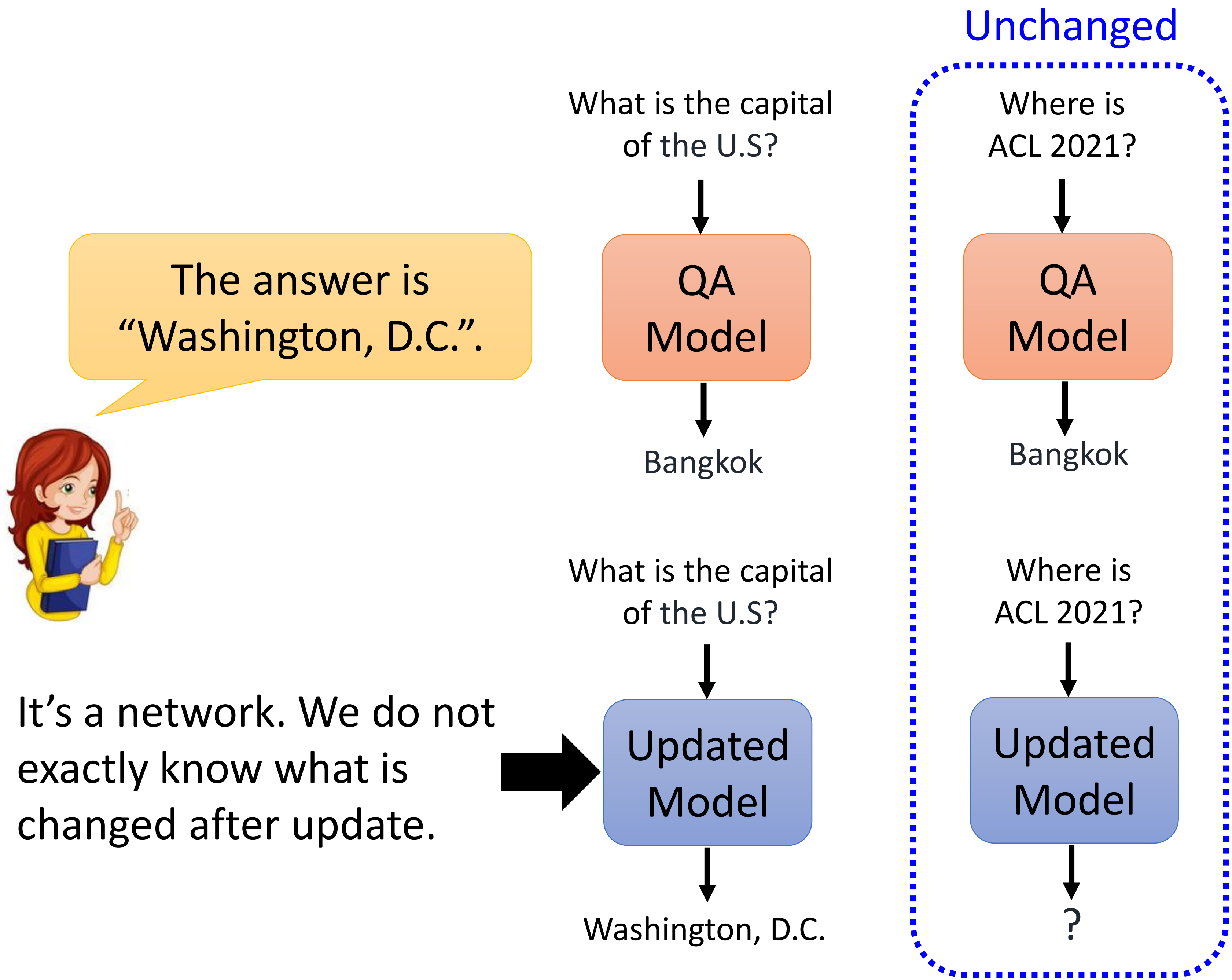


Catastrophic forgetting!

Lifelong Learning in real-world applications



Catastrophic forgetting may happen!


Unchanged
The answer is “Washington, D.C.”.
What is the capital of the U.S?
QA Model
Bangkok
Where is ACL 2021?
QA Model
Bangkok
What is the capital of the U.S?
Updated Model
Washington, D.C.
Where is ACL 2021?
Updated Model
?
It’s a network. We do not exactly know what is changed after update.

Mitigating Catastrophic Forgetting

Selective Synaptic Plasticity

Regularization-based

Additional Neural Resource Allocation

Memory Replay

- There are already lots of research along each direction.
- Can meta learning enhance these approaches?

Regularization-based

Dataset 1

cat dog

Dataset 2

cat dog

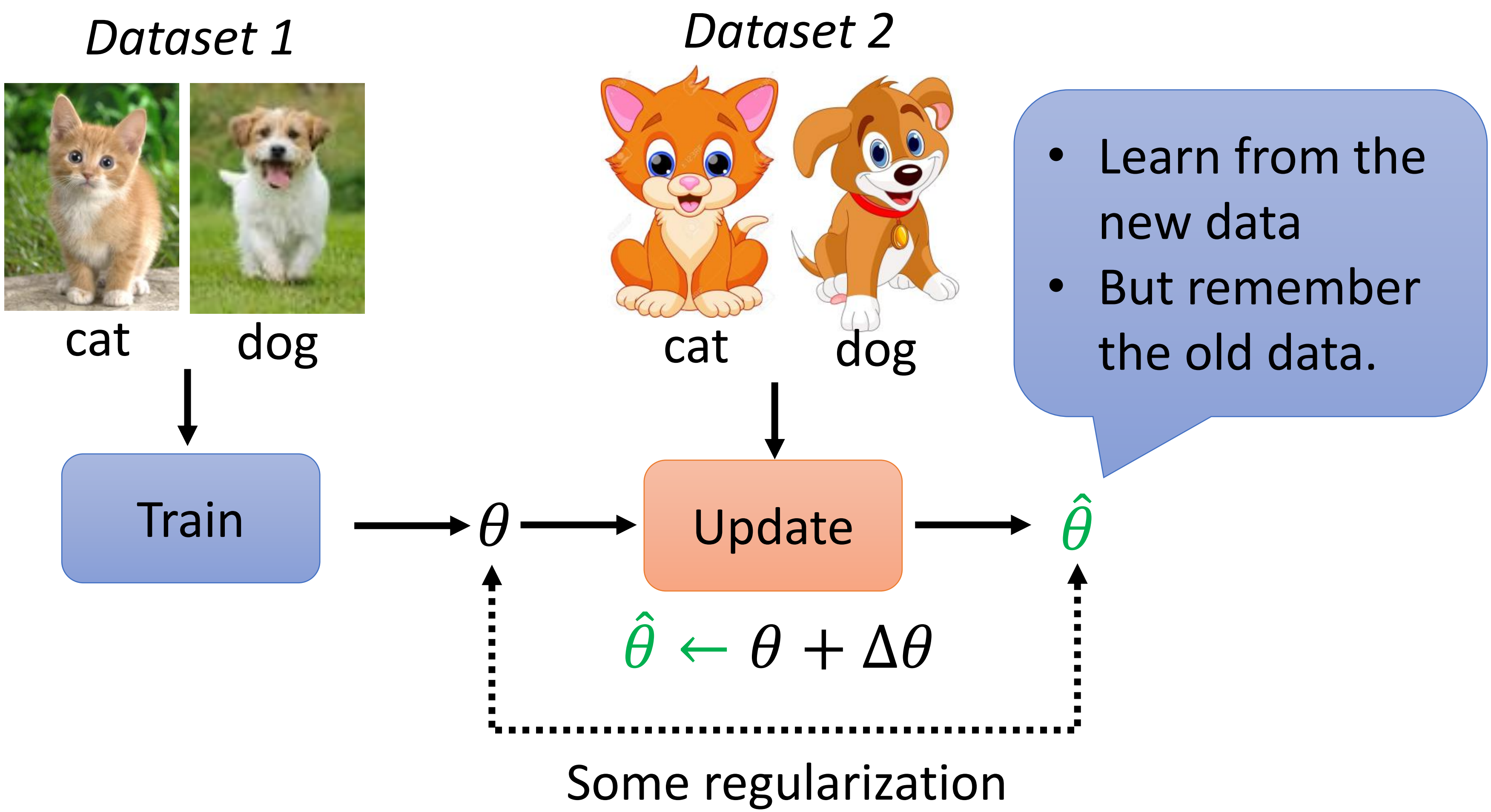
- Learn from the new data
- But remember the old data.

Train → θ → Update → $\hat{\theta}$

$\hat{\theta} \leftarrow \theta + \Delta\theta$

Gradient computed based on new data

Regularization-based



L2 does not work. For prevent forgetting: EWC, SI, MAS ……

Regularization-based

Dataset 1

cat dog

Train → θ → Update → $\hat{\theta}$

Dataset 2

cat dog

- Learn from the new data
- But remember the old data.

$$\hat{\theta} \leftarrow \theta + \boxed{\Delta\theta}$$

satisfy

- Not simply use gradient
- Learn how to compute "proper" update from new data

Nicola De Cao, Wilker Aziz, Ivan Titov, Editing Factual Knowledge in Language Models, arXiv, 2021

Application: Fact checking, QA

Regularization-based

Dataset 1

cat dog

Train $\rightarrow$ $\boxed{\theta}$ $\rightarrow$ Update $\rightarrow$ $\hat{\theta}$

Dataset 2

cat dog

satisfy

- Learn from the new data
- But remember the old data.

$$\hat{\theta} \leftarrow \theta + \boxed{\Delta\theta}$$

Gradient computed based on new data

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, Artem Babenko, Editable Neural Networks, ICLR, 2020

Application: Machine translation

Mitigating Catastrophic Forgetting

Selective Synaptic Plasticity

Regularization-based

Additional Neural Resource Allocation

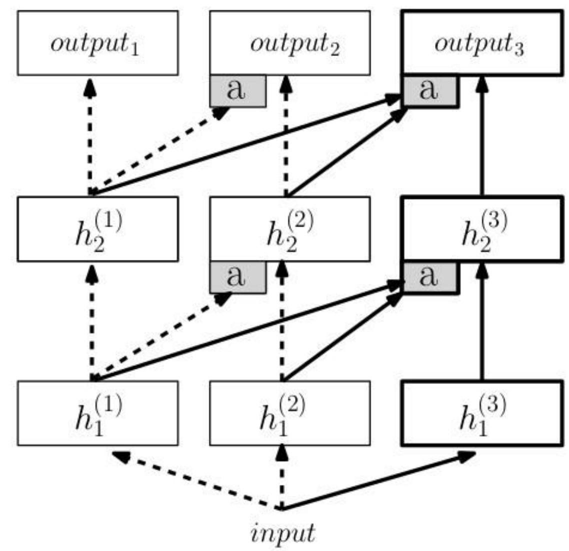
Memory Replay

- There are already lots of research along each direction.
- Can meta learning enhance these approaches?

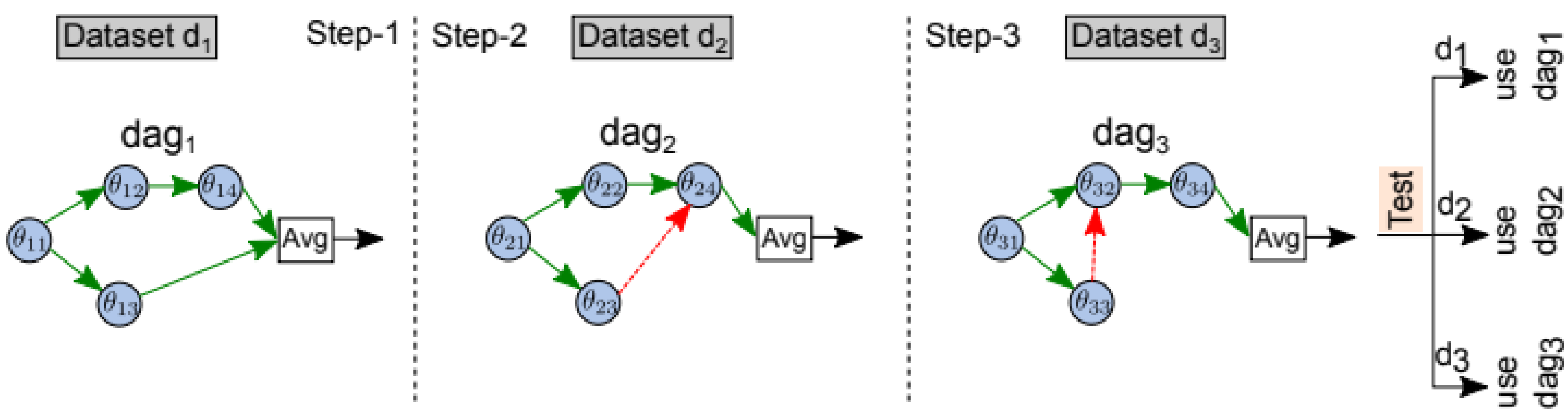
Additional Neural Resource Allocation

Expand the network when there are new dataset.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, Raia Hadsell, Progressive Neural Networks, 2016



Network architecture search can be used when you want to change the network architecture given new dataset.



Ramakanth Pasunuru, Mohit Bansal, Continual and Multi-Task Architecture Search, ACL, 2019

Mitigating Catastrophic Forgetting

Selective Synaptic Plasticity

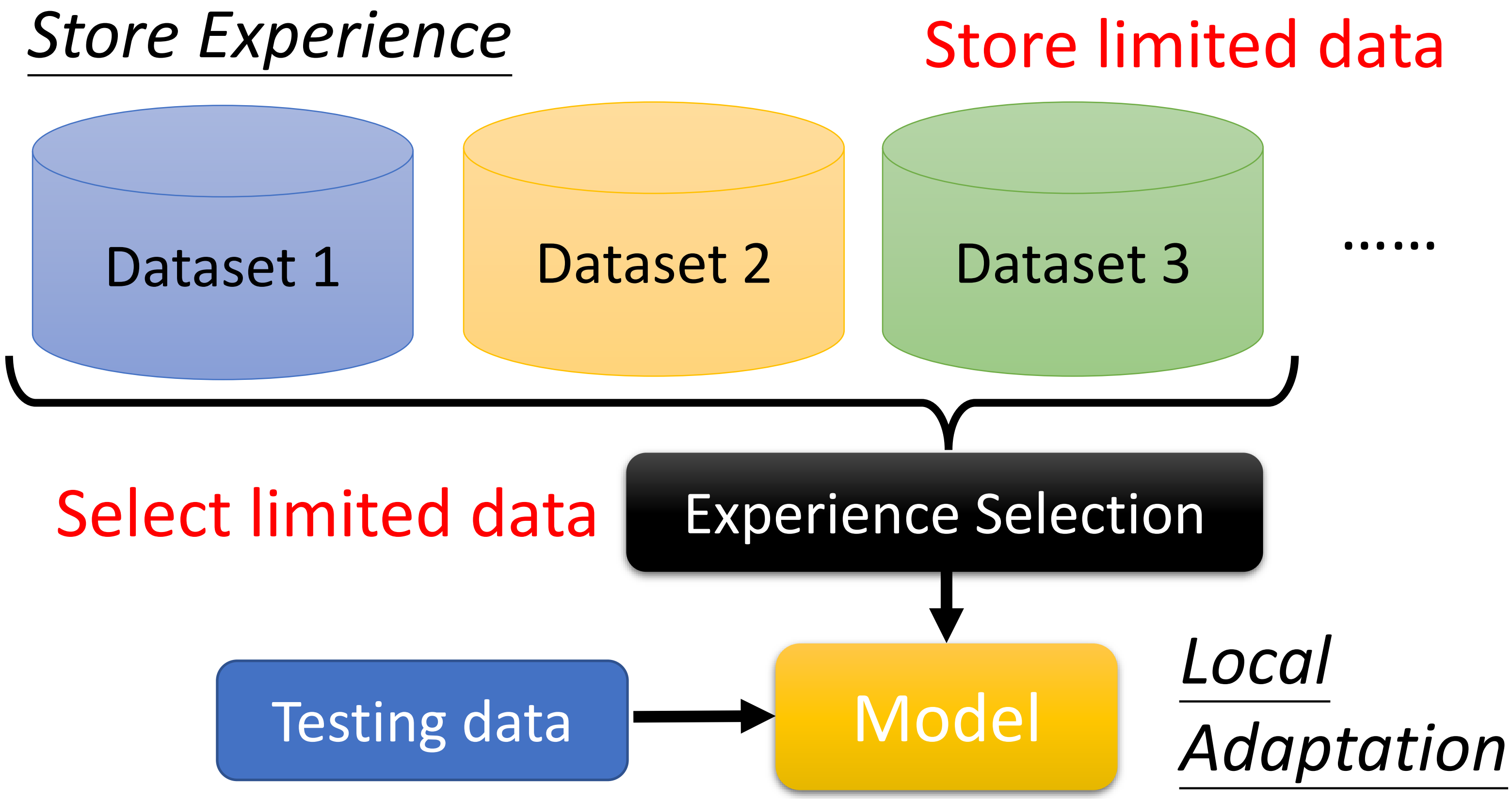
Regularization-based

Additional Neural Resource Allocation

Memory Replay

- There are already lots of research along each direction.
- Can meta learning enhance these approaches?

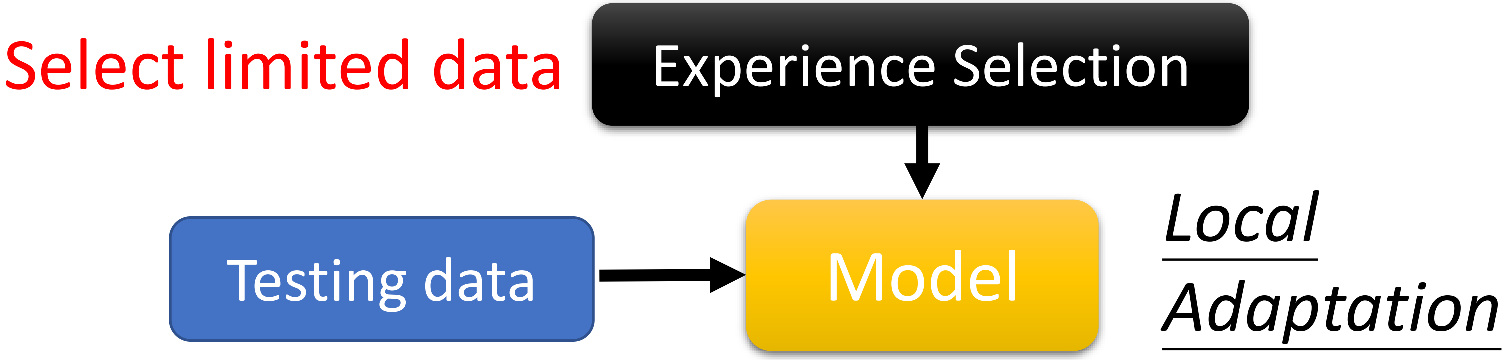
Memory-based Parameter Adaptation (MbPA)



Pablo Sprechmann, Siddhant M. Jayakumar, Jack W. Rae, Alexander Pritzel, Adrià Puigdomènech Badia, Benigno Uria, Oriol Vinyals, Demis Hassabis, Razvan Pascanu, Charles Blundell, Memory-based Parameter Adaptation, ICLR, 2018

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, Dani Yogatama, Episodic Memory in Lifelong Language Learning, NeurIPS, 2019

Memory-based Parameter Adaptation (MbPA)



This is few-shot learning problem. → Meta Learning!

Text Classification, QA

Zirui Wang, Sanket Vaibhav Mehta, Barnabás Póczos, Jaime Carbonell, Efficient Meta Lifelong-Learning with Limited Memory, EMNLP, 2020

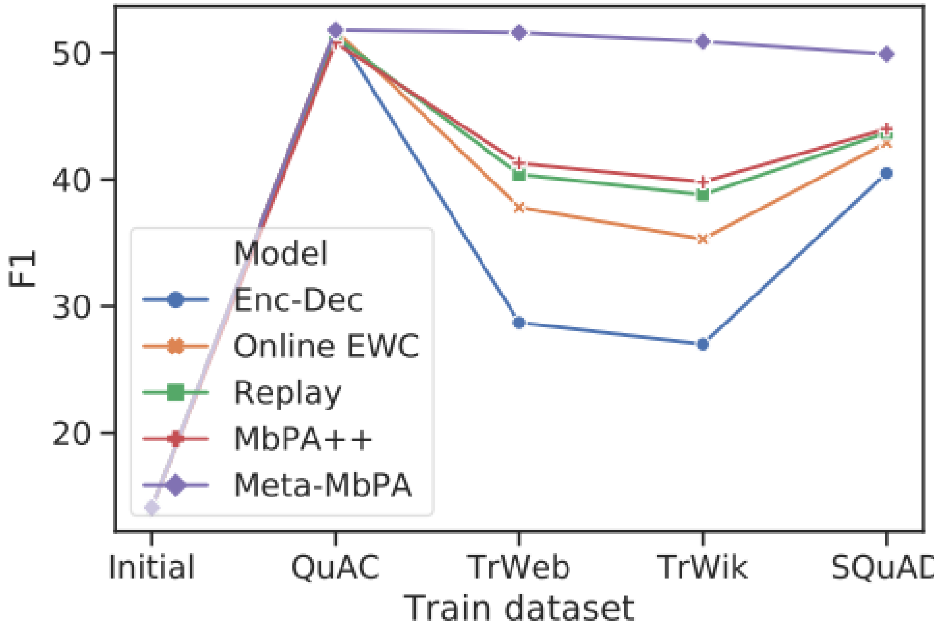
Relation Extraction

Abiola Obamuyide, Andreas Vlachos, Meta-learning improves lifelong relation extraction, RepL4NLP, 2019

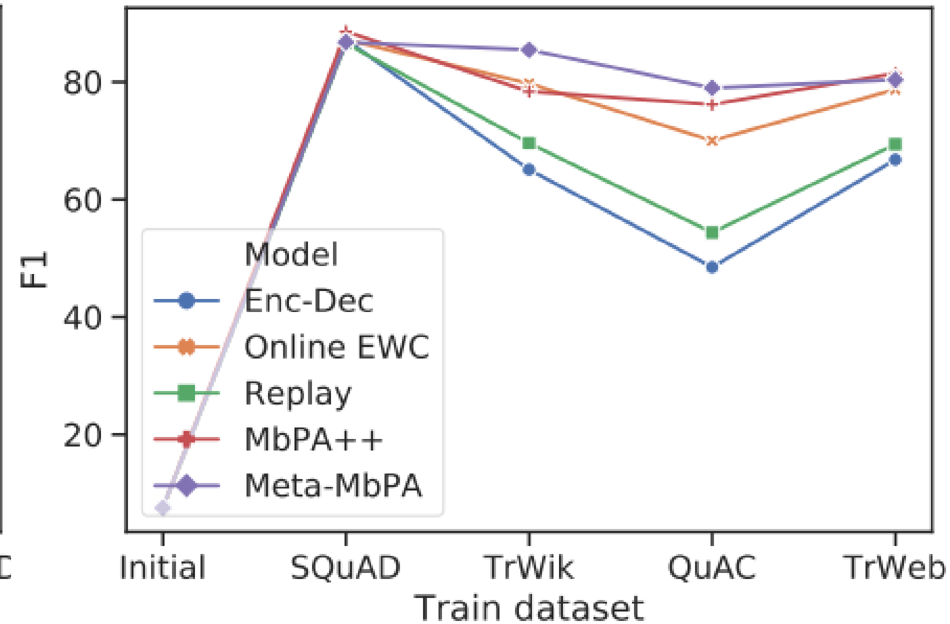
Tongtong Wu, Xuekai Li, Yuan-Fang Li, Reza Haffari, Guilin Qi, Yujin Zhu, Guoqiang Xu, Curriculum-Meta Learning for Order-Robust Continual Relation Extraction, AAAI, 2021

Memory-based Parameter Adaptation (MbPA)

+ Meta Learning



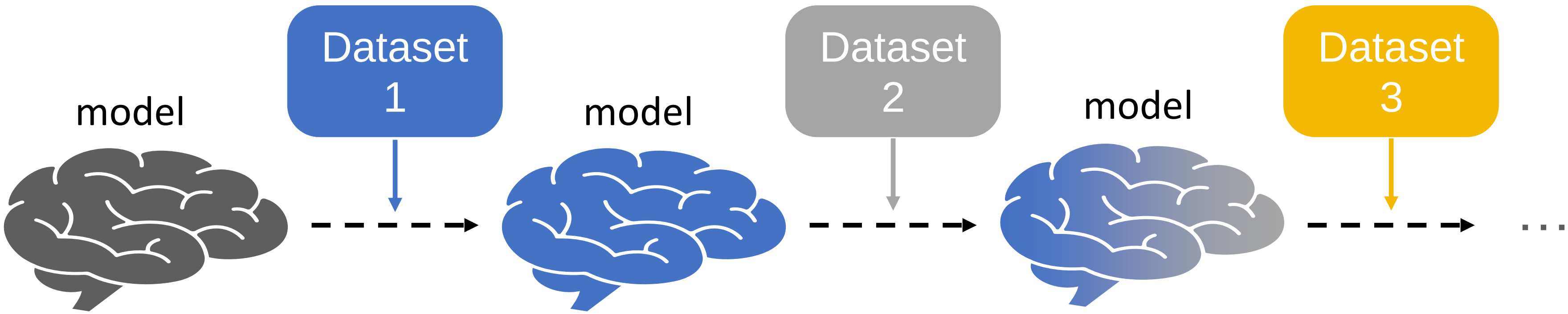
QuAC



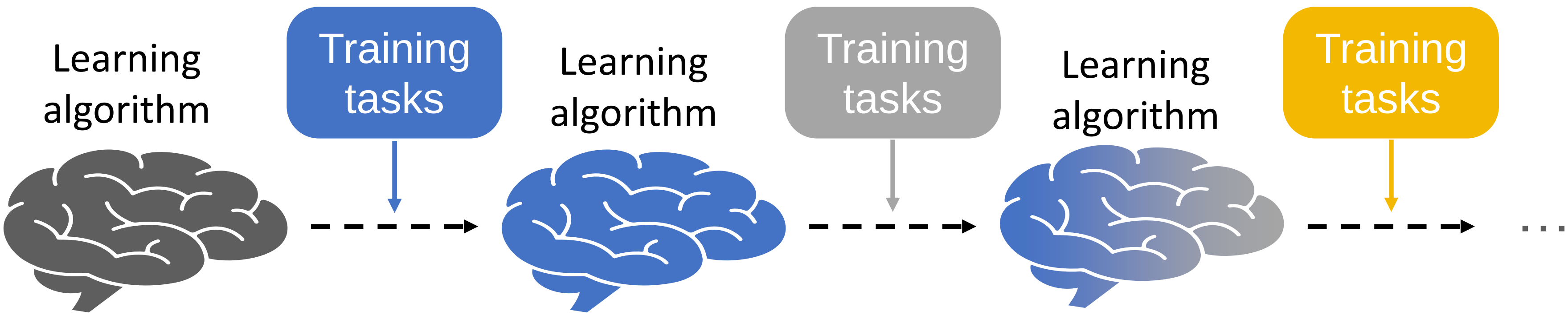
SQuAD

Zirui Wang, Sanket Vaibhav Mehta, Barnabás Póczos, Jaime Carbonell, Efficient Meta Lifelong-Learning with Limited Memory, EMNLP, 2020

Problem of Another Level ……



Meta learning can help.



Meta learning itself also face the issue of catastrophic forgetting!

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, Sergey Levine, Online Meta-Learning, ICML, 2019

Pauching Yap, Hippolyt Ritter, David Barber, Addressing Catastrophic Forgetting in Few-Shot Problems, ICML, 2021

Concluding Remarks

Part I: Basic Idea of Meta Learning

Part II: Applications to Human Language Processing

- Check this! https://jeffeuxmartin.github.io/meta-learning-hlp/

Part III: Advanced Topics

- Data Selection
- Domain Generalization → Generalization of learned model
- Task Augmentation → Generalization of meta learning itself
- Inference efficiency
- Meta knowledge distillation
- Mitigating catastrophic forgetting

Beyond accuracy

Meta Learning and Its Applications to Natural Language Processing @ ACL 2021

Andreas Vlachos

University of Cambridge

Chelsea Finn

Stanford University

Eric Xing

Carnegie Mellon University

Invited Speakers

Heng Ji

University of Illinois Urbana-Champaign

Zhou Yu

Columbia University

Thank you for your attention.