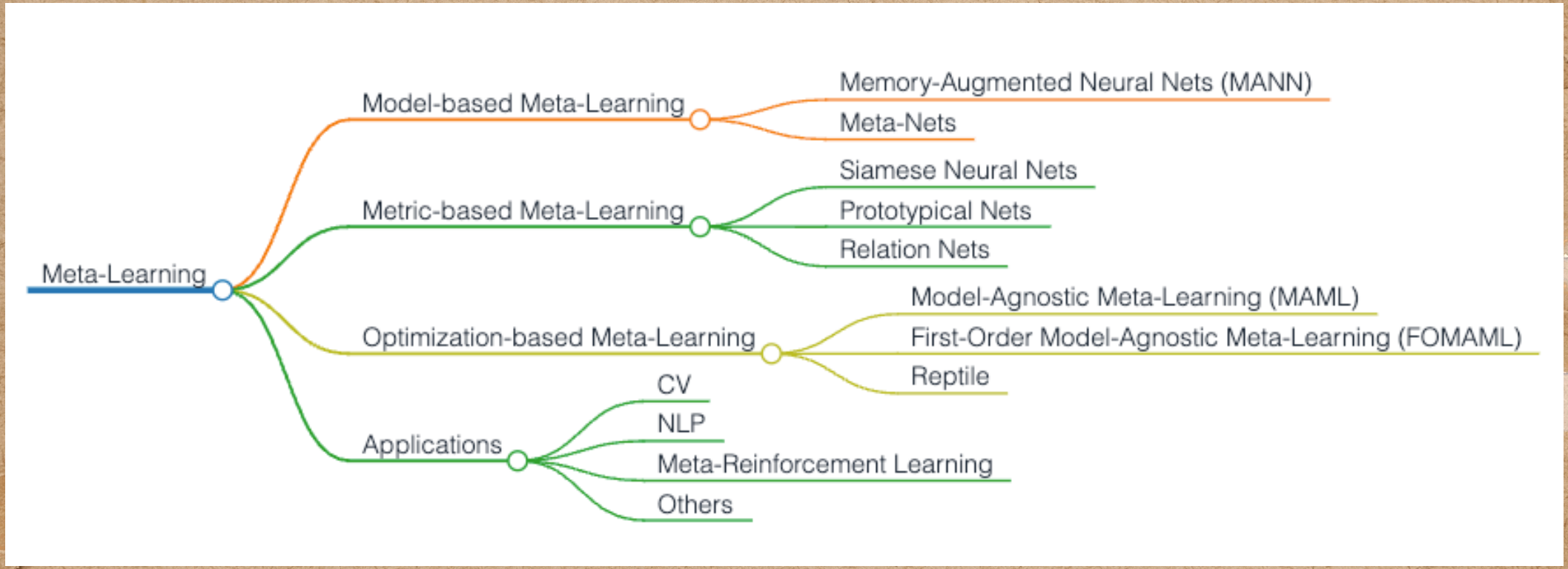
Map of Content to Meta-Learning

aka a map to learn how to learn
"learning to learn"

Map of Content



Map of Resources

- Stanford CS 330 Deep Multi-Task and Meta Learning *(For Depth)*
- Machine Learning Course (National Taiwan University), in Mandarin *(For Breadth)*
- Meta-Learning: Theory, Algorithms and Applications
- Meta-Learning in PyTorch Youtube Playlist
- Meta Learning for Natural Language Processing: A Survey

Machine Learning & Meta Learning

- **Machine Learning** ≈ find a function f

Dog-Cat Classification

f_θ () = “cat”

- **Meta Learning** ≈ find a **function F** that finds a function f

Learning Algorithm

F_ϕ (    ) = f

Training Examples

Dog Cat Cat Dog Cat

Adapted from https://youtu.be/xoastiYx9JU?t=1745

Formulating a Meta Learning Problem

Given data from $\mathcal{T}_1$, …, $\mathcal{T}_n$, solve new task $\mathcal{T}_{test}$ more quickly / proficiently / stably

Key assumption: meta-training tasks and meta-test task drawn i.i.d. from same task distribution

$$\mathcal{T}_1 , \ldots, \mathcal{T}_n \sim p(\mathcal{T}), \mathcal{T}_{test} \sim p(\mathcal{T})$$

Tasks must share structure.

Sample Tasks (in NLP context):

- The tasks belong to the same NLP problem.
  - All $\mathcal{T}_n$ are QA tasks but different corpora
- Tasks that belong to various NLP problems

Adapted from https://youtu.be/cjTs7lH5eUs?t=53

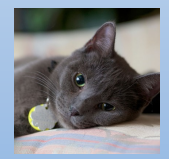
Terminology

Machine Learning


cat

dog

Train


cat

dog

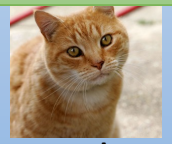
Test

Training Tasks

Task 1

Support set

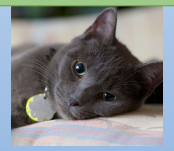
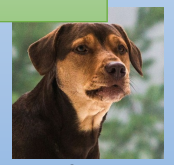
Train


cat

dog

Query set

Test


cat

dog

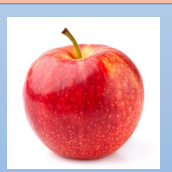
Task 2

Train


apple

orange

Test

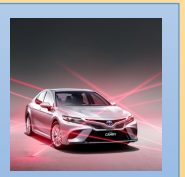

apple

orange

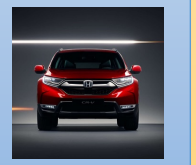
Sometimes you need validation tasks

Testing Tasks

Train

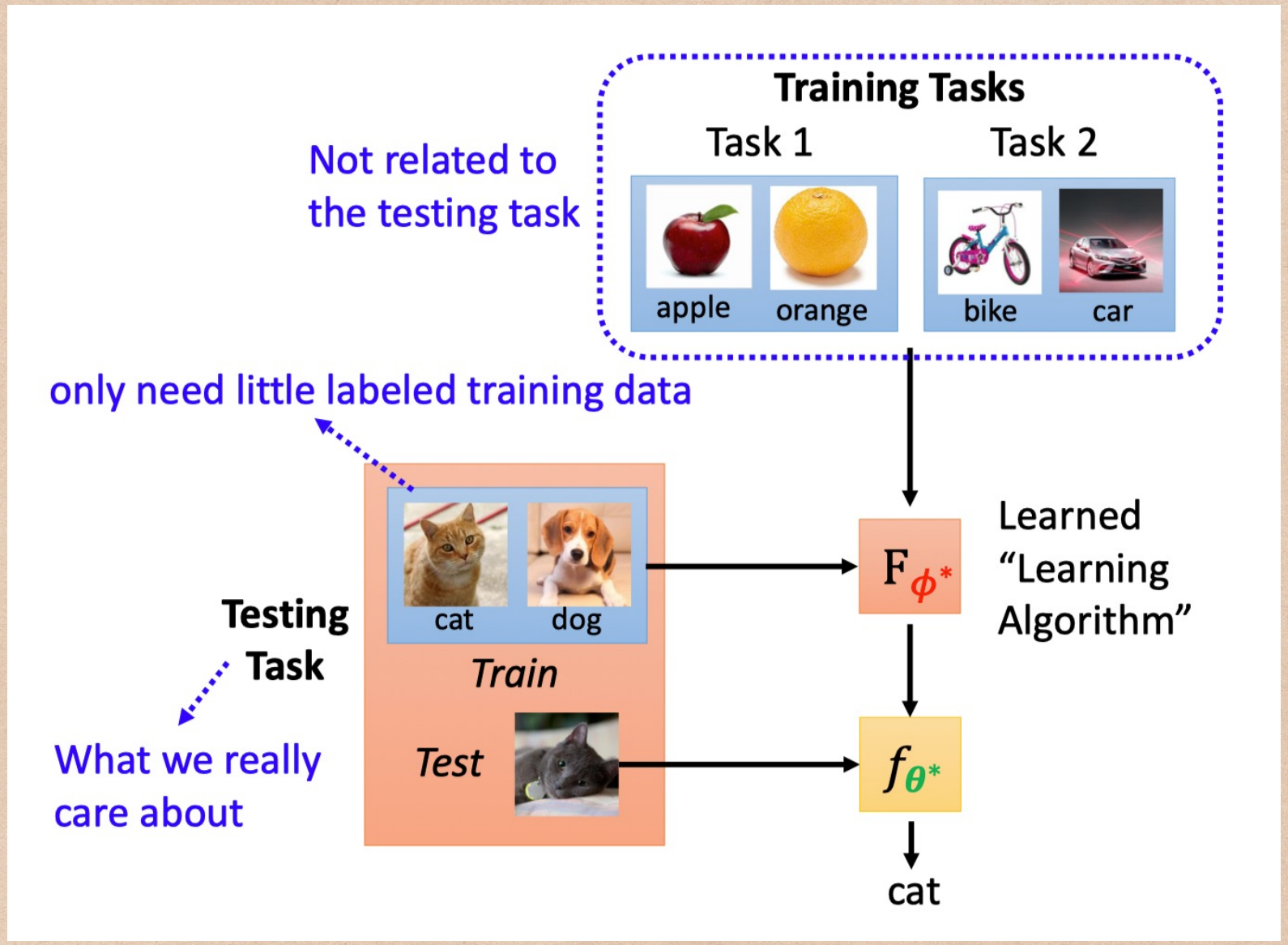

bike

car

Test


bike

car

Adapted from https://www.youtube.com/watch?v=EkAqYbpCYAc

Framework



Adapted from https://www.youtube.com/watch?v=xoastiYx9JU

Few shot Classification

- N-ways K-shot classification: In each training and test tasks, there are N classes, each has K examples.

20 ways

1 shot

Each character represents a class

Testing set (Query set)

Training set (Support set)

- Split your characters into training and testing characters
  - Sample N training characters, sample K examples from each sampled characters → one training task
  - Sample N testing characters, sample K examples from each sampled characters → one testing task

Omniglot dataset:
1623 characters
Each has 20 examples

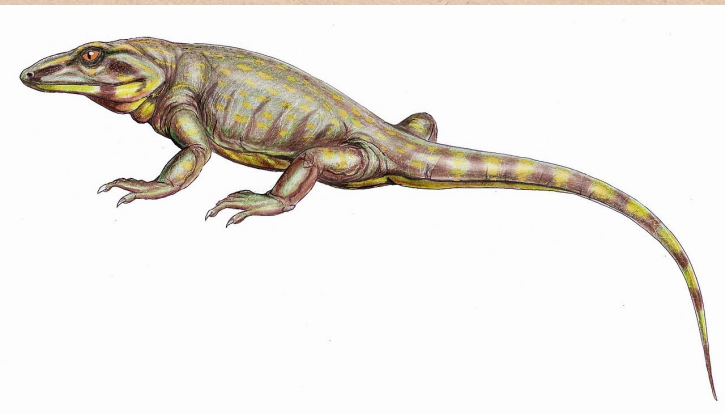
Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

Optimization-Based Meta-Learning

Learning to Initialize

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

- **Reptile**
  - Alex Nichol, Joshua Achiam, John Schulman, On First-Order Meta-Learning Algorithms, arXiv, 2018



Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

MAML

$\hat{\theta}^n$: model learned from task n

$\hat{\theta}^n$ depends on ϕ

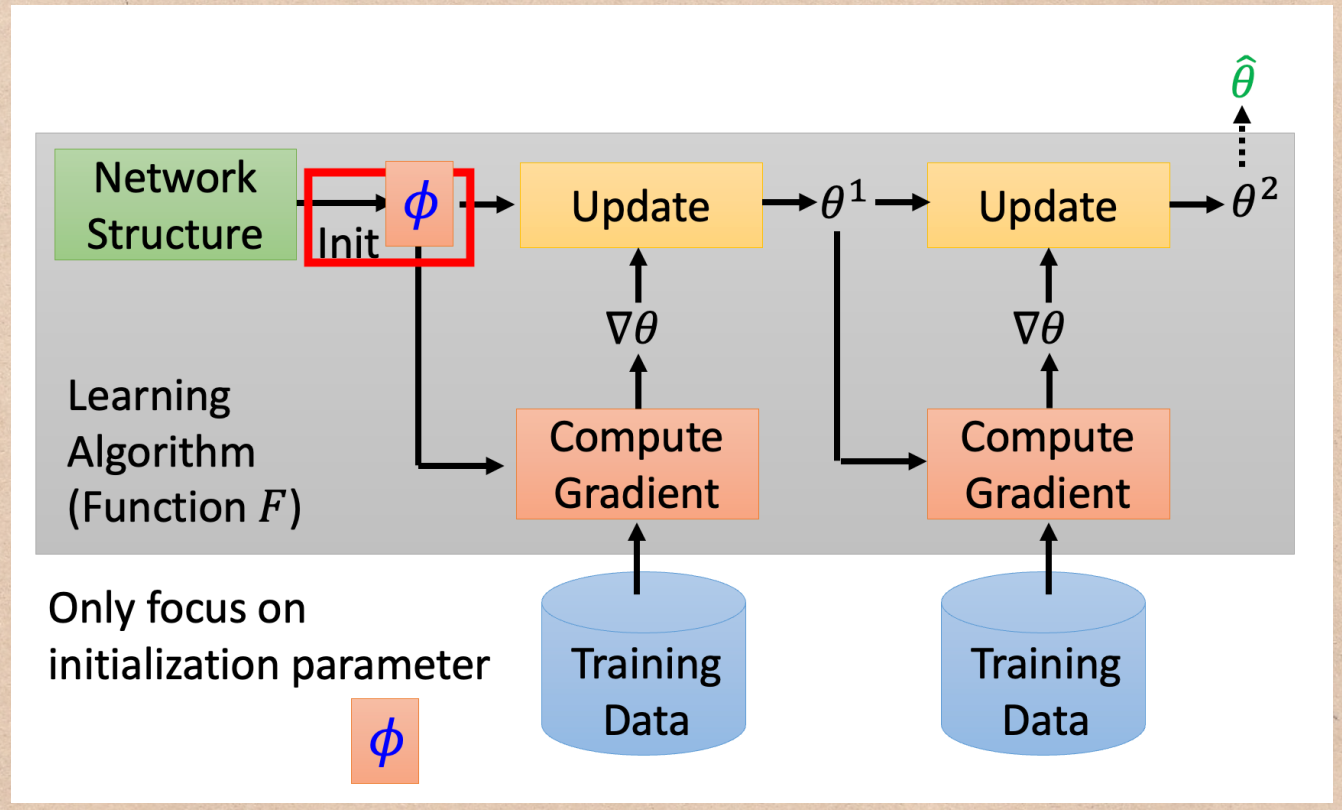
$l^n(\hat{\theta}^n)$: loss of task n on the testing set of task n

Loss Function:

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

Use Gradient Descent to minimize $L(\phi)$

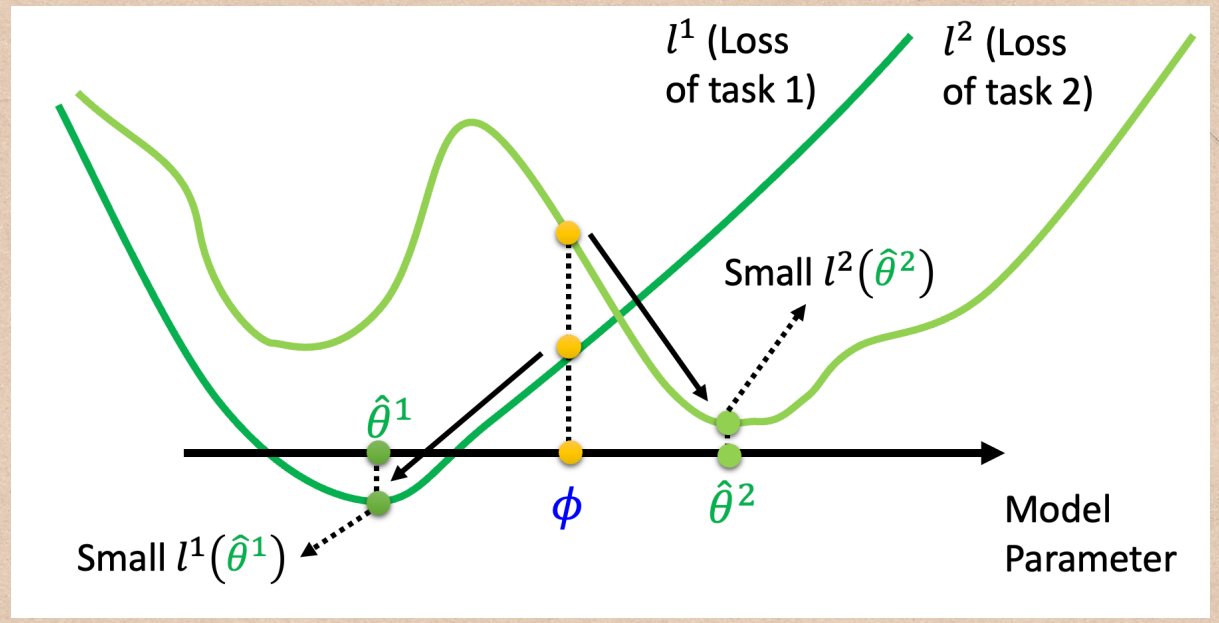
$$\phi \leftarrow \phi - \eta \nabla_\phi L(\phi)$$



Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

MAML vs Pre-training

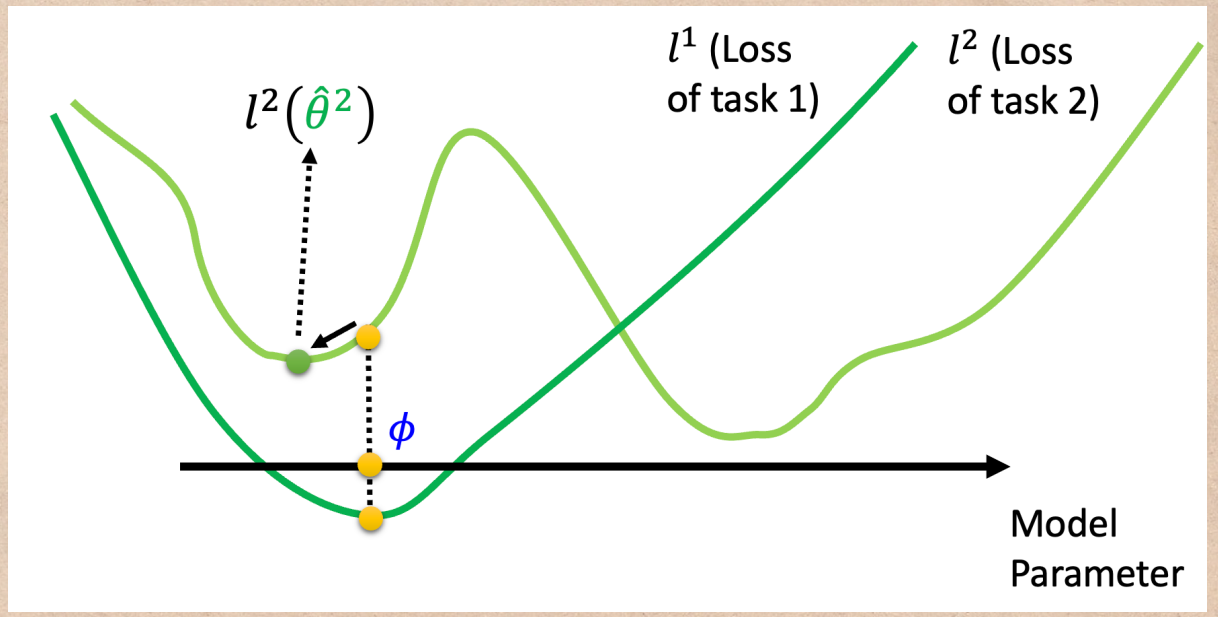
$$L(\phi) = \sum_{n=1}^{N} l^n(\hat{\theta}^n)$$



MAML

Find ϕ that can achieve good performance **after training**

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



Pre-training

Find ϕ that can achieve good performance currently

Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

MAML: Toy Model

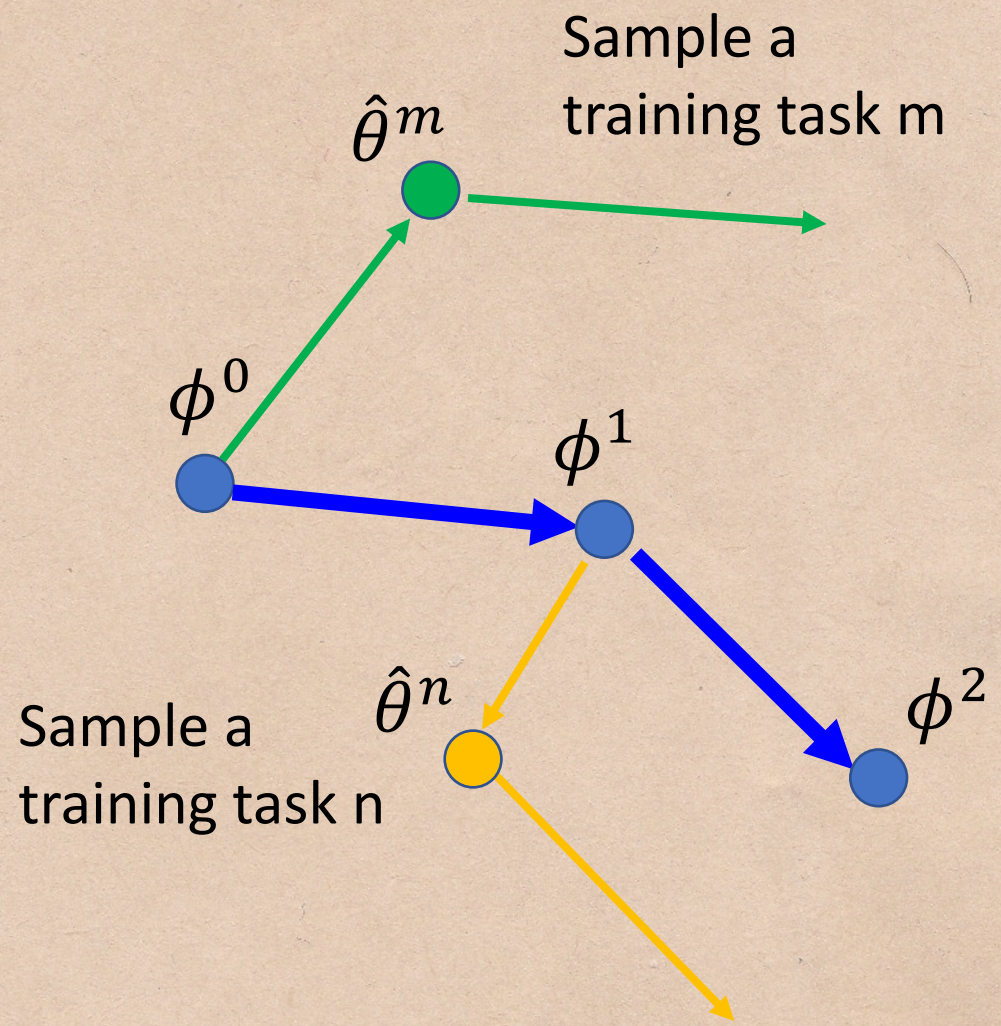
Each task:

- Given a target sine function $y = a\ sin(x + b)$
- Sample K points from the target function
- Use the samples to estimate the target function

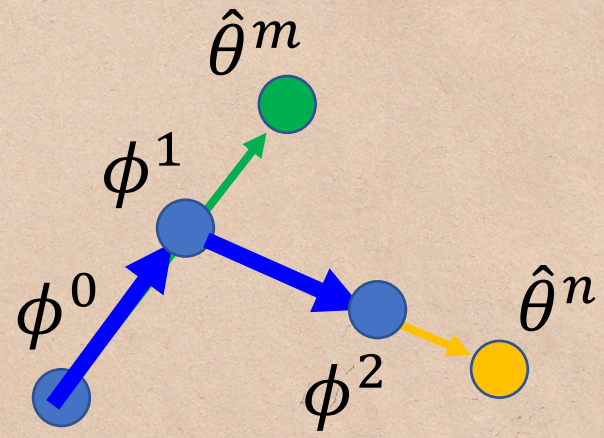
Sample a and b to form a task

Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

MAML

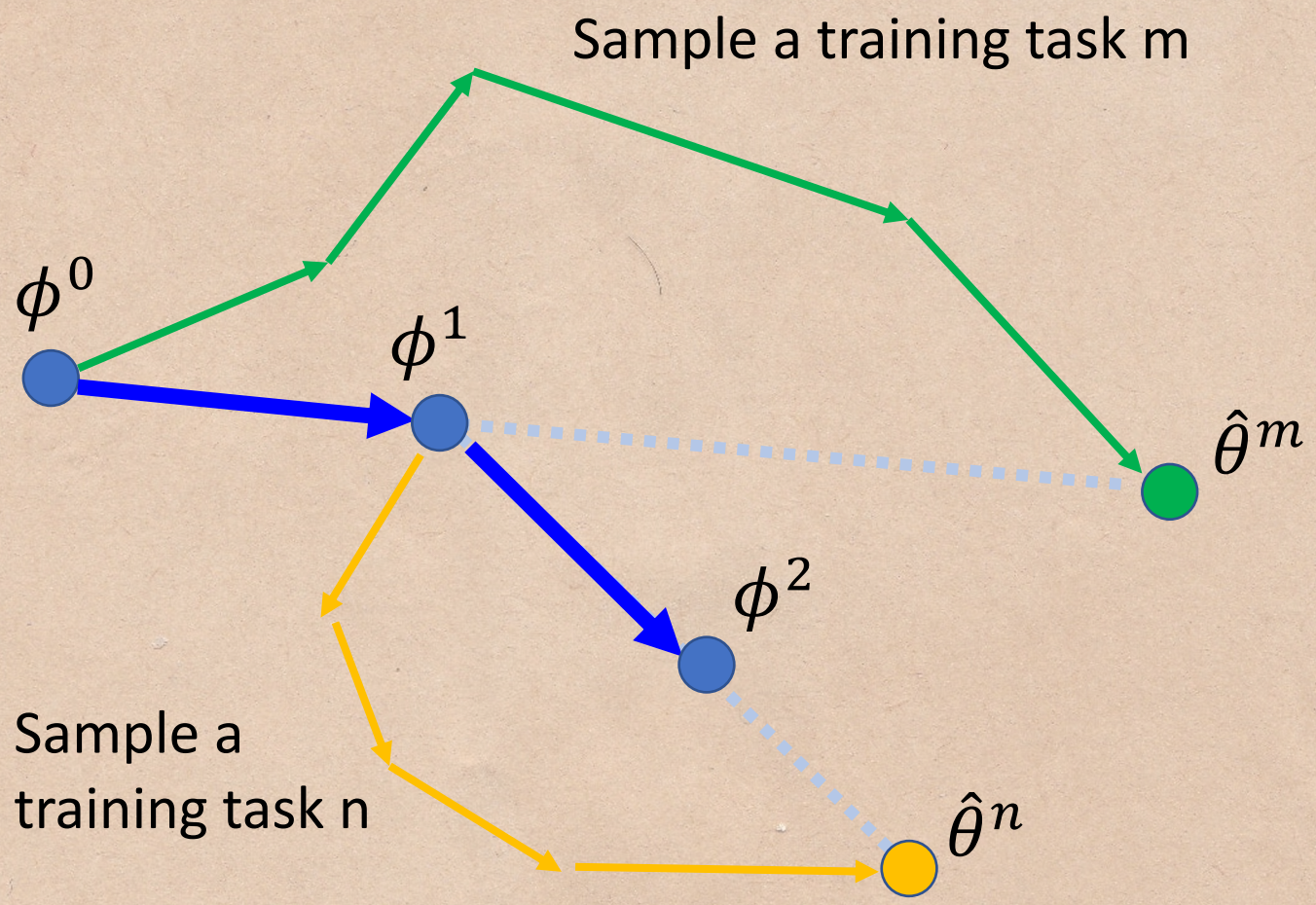


Model Pre-training



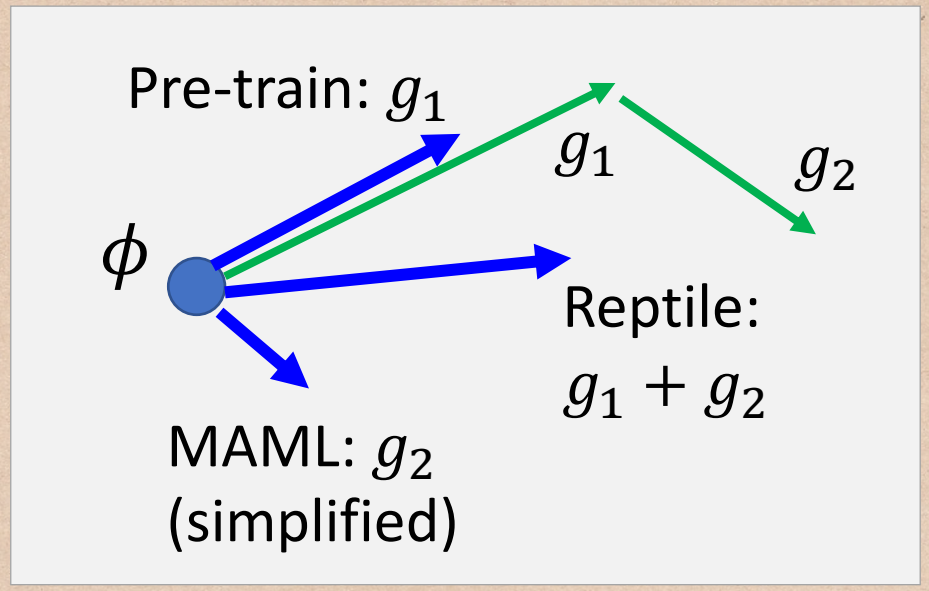
Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

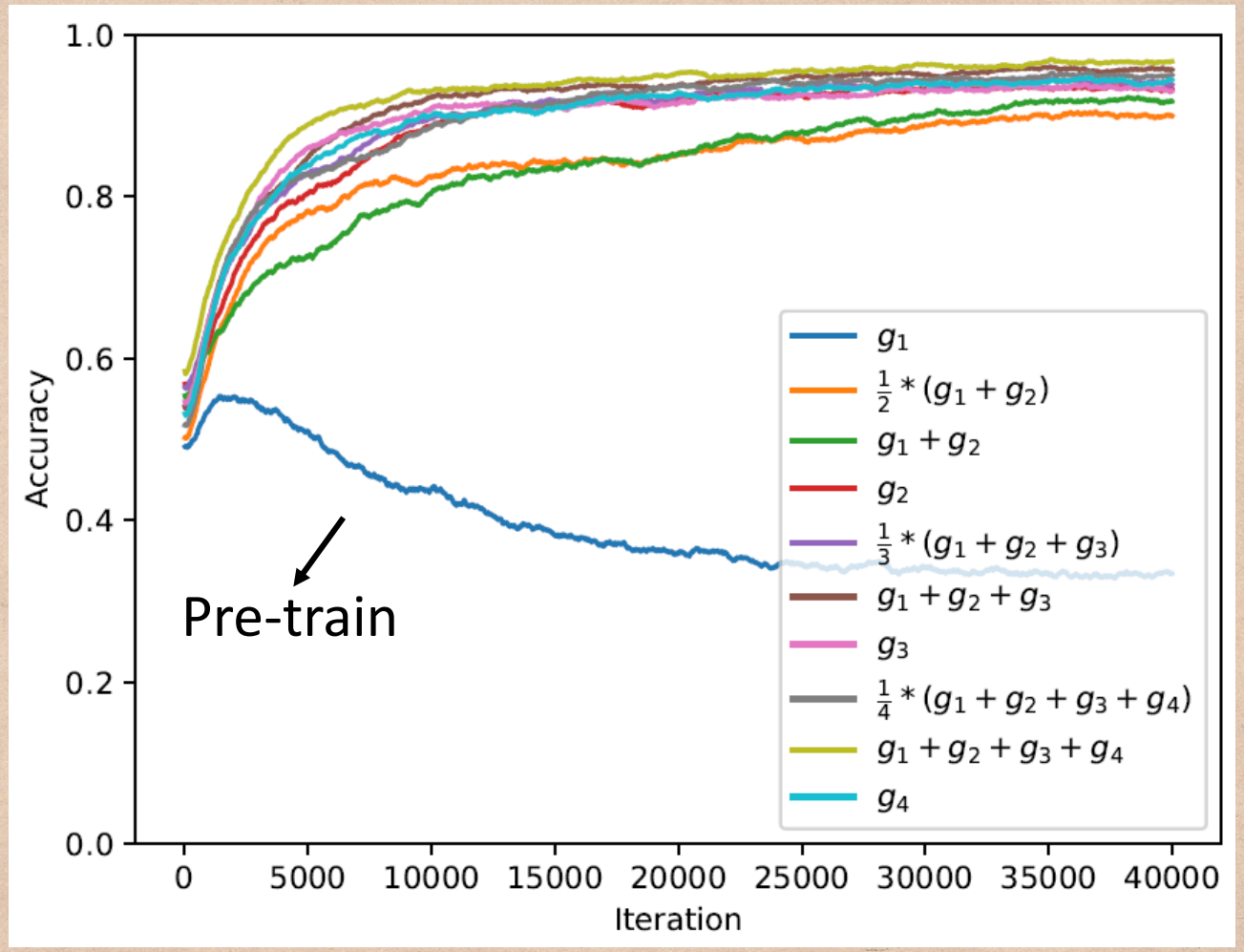
Reptile



Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

Reptile



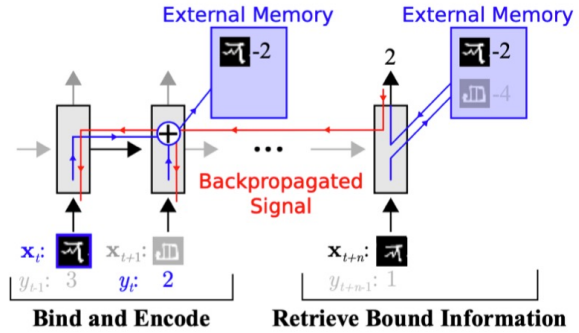


Adapted from https://www.youtube.com/watch?v=vUwOA3SNb_E

Model-Based Meta-Learning

Meta-Learning as a sequence-to-sequence problem

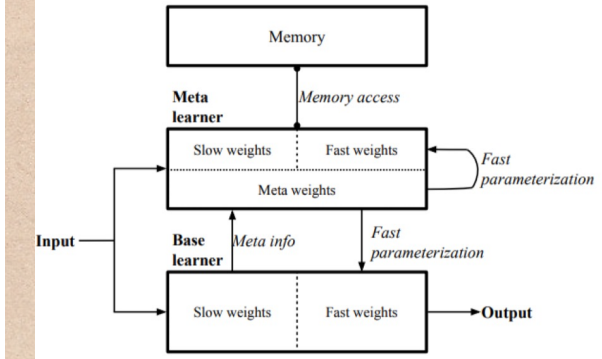
LSTMs or Neural Turing Machine



Meta-Learning with Memory-Augmented Neural Networks

Santoro, Bartunov, Botvinick, Wierstra, Lillicrap. ICML '16

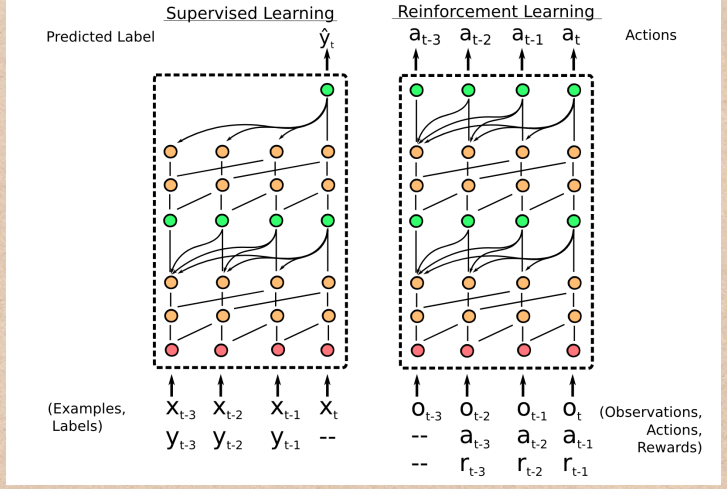
Other External Memory Mechanisms



Meta Networks

Munkhdalai, Yu. ICML '17

Convolutions + Attention



A Simple Neural Attentive Meta-Learner, 2018

In fact, model-based approaches remain heuristic and leave ample space, and most strategies may need further modifications to improve their applicability. Current approaches still have drawbacks, and performances drop for out-of-distribution tasks compared to optimization-based methods (explored in Chapter 4). While the integration of slow weights and fast weights can be achieved through layer augmentation in Meta-Net, the current solution will cause trouble if additional types of parameters need to be processed at many distinct time scales. If the memory buffer is not cleared among tasks, unexpected proactive interference (Underwood, 1957) will occur in MANN, so that the previous knowledge produces an interference effect on the newer knowledge, which is an effect that often happens in human memory.

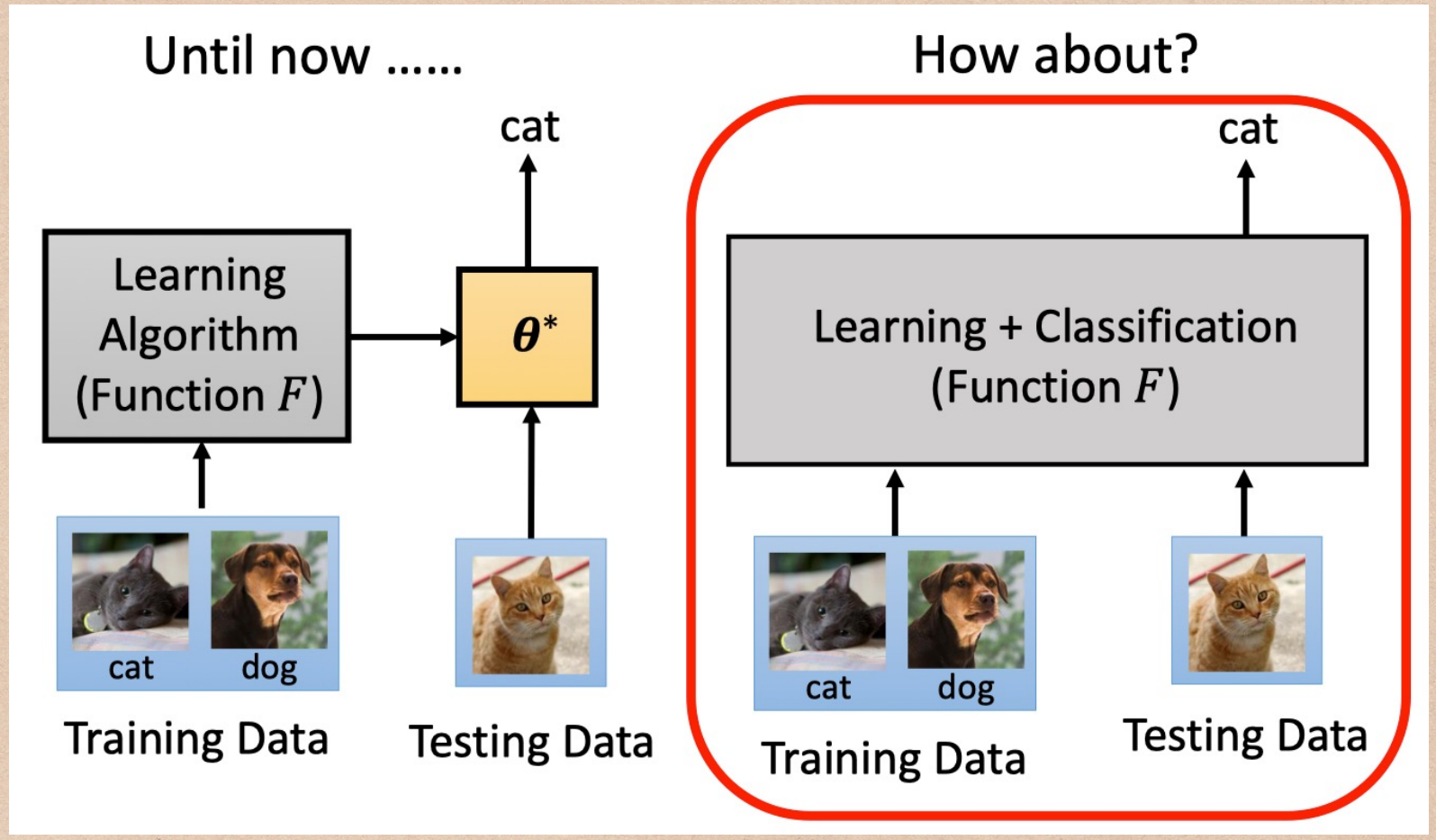
From Meta-Learning: Theory, Algorithms and Applications Chapter 2

Method	5-Way Omniglot		20-Way Omniglot	
	1-shot	5-shot	1-shot	5-shot
Santoro et al. (2016)	82.8%	94.9%	–	–
Koch (2015)	97.3%	98.4%	88.2%	97.0%
Vinyals et al. (2016)	98.1%	98.9%	93.8%	98.5%
Finn et al. (2017)	**98.7% ± 0.4%**	**99.9% ± 0.3%**	95.8% ± 0.3%	98.9% ± 0.2%
Snell et al. (2017)	97.4%	99.3%	96.0%	98.9%
Munkhdalai & Yu (2017)	**98.9%**	–	97.0%	–
SNAIL, Ours	**99.07% ± 0.16%**	**99.78% ± 0.09%**	**97.64% ± 0.30%**	**99.36% ± 0.18%**

Adapted from https://youtu.be/cjTs7lH5eUs?t=3865

Metric-Based Meta-Learning

Learning to Compare

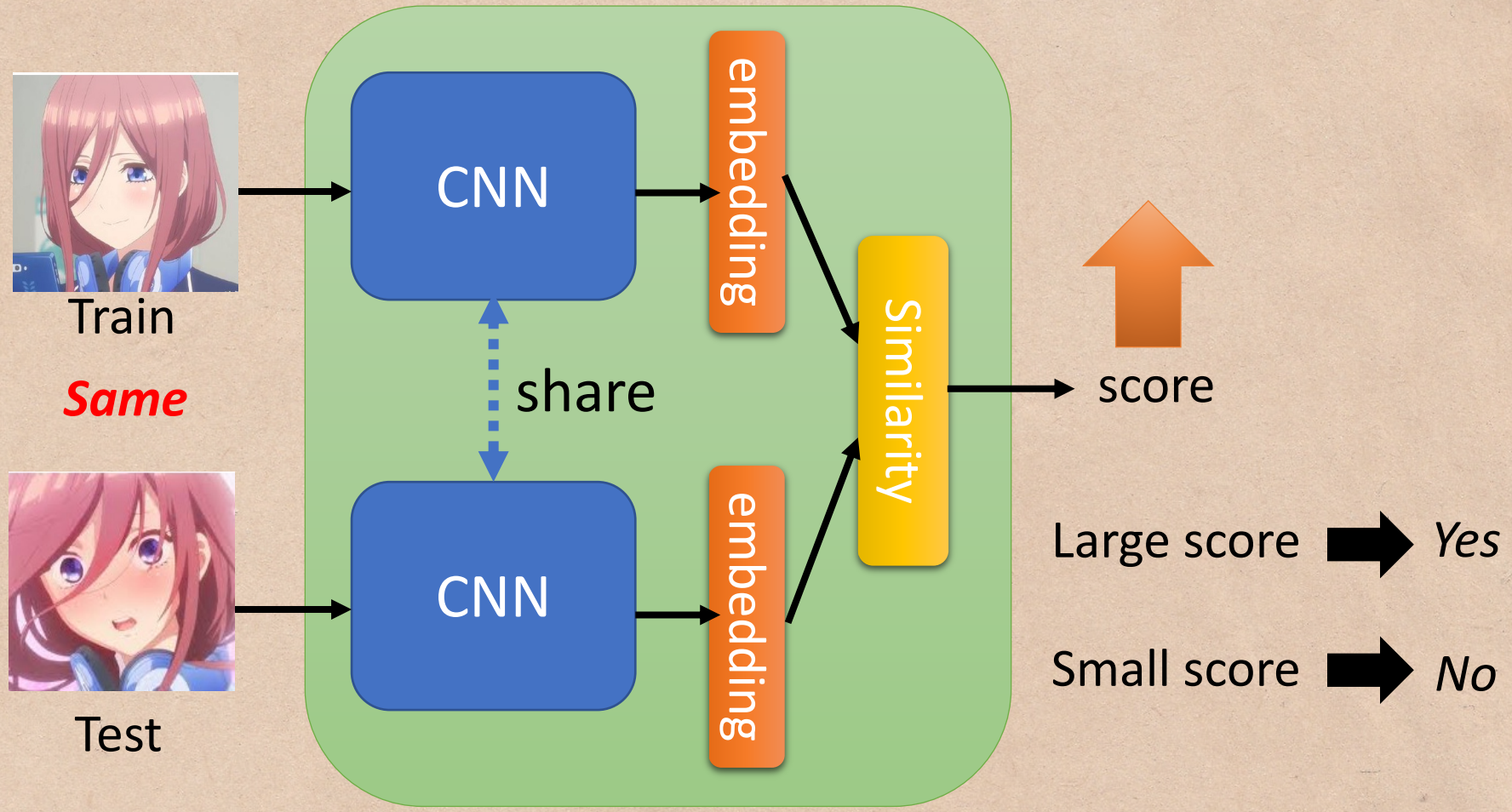


learns similarity or dissimilarity measures using distance measures to tackle tasks such as classification or few-shot learning

Adapted from https://youtu.be/Q68Eh-wm1Ts?t=1441

Metric-Based Meta-Learning

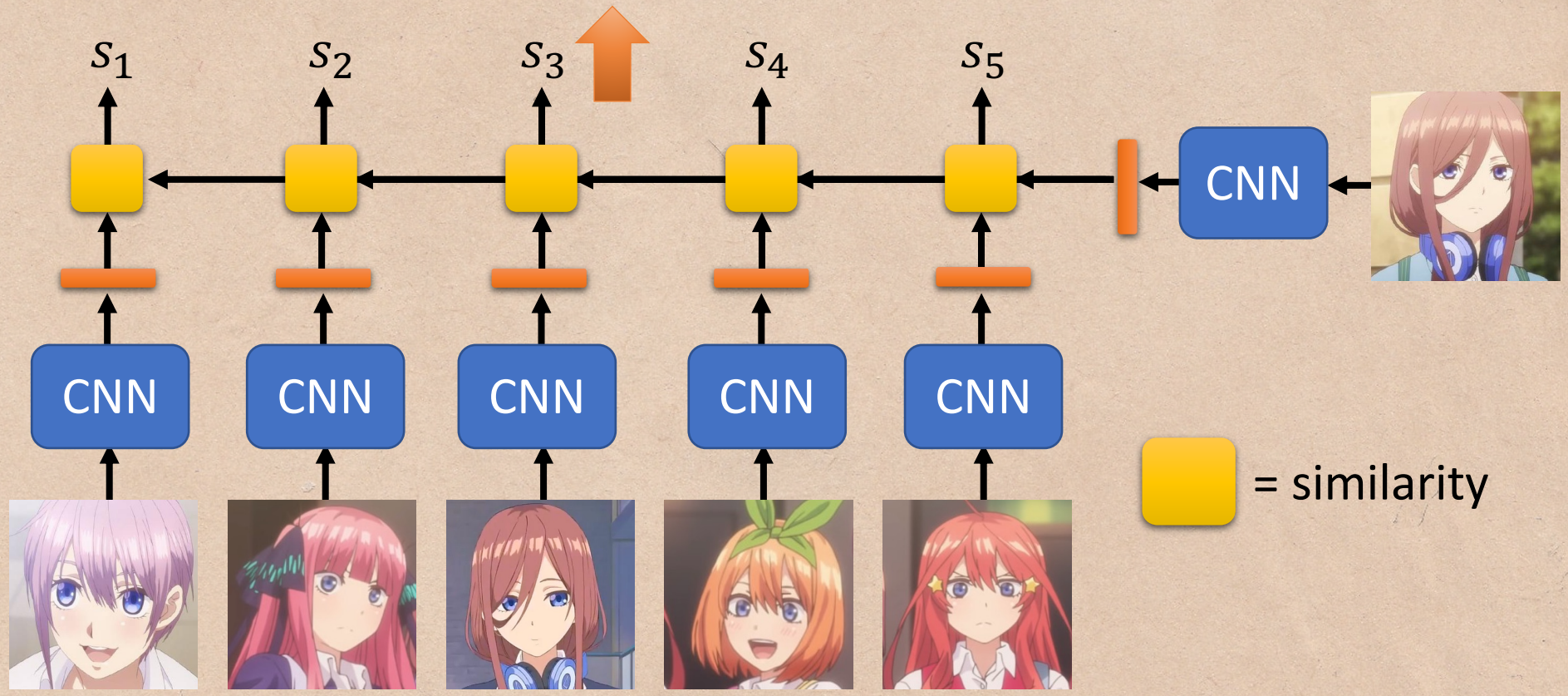
Siamese Network



Adapted from https://www.youtube.com/watch?v=yyKaACh_j3M

Metric-Based Meta-Learning

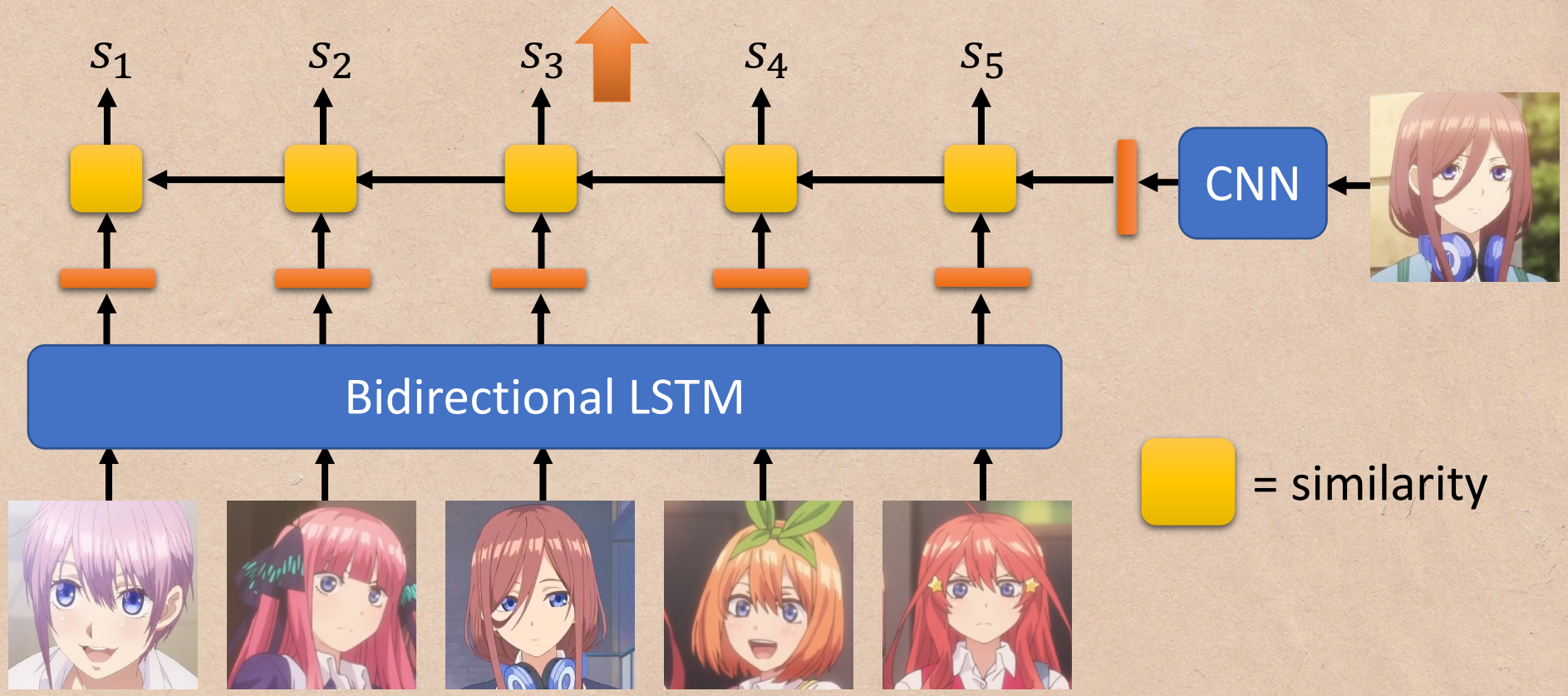
Prototypical Network



Adapted from https://www.youtube.com/watch?v=semSxPP2Yzg

Metric-Based Meta-Learning

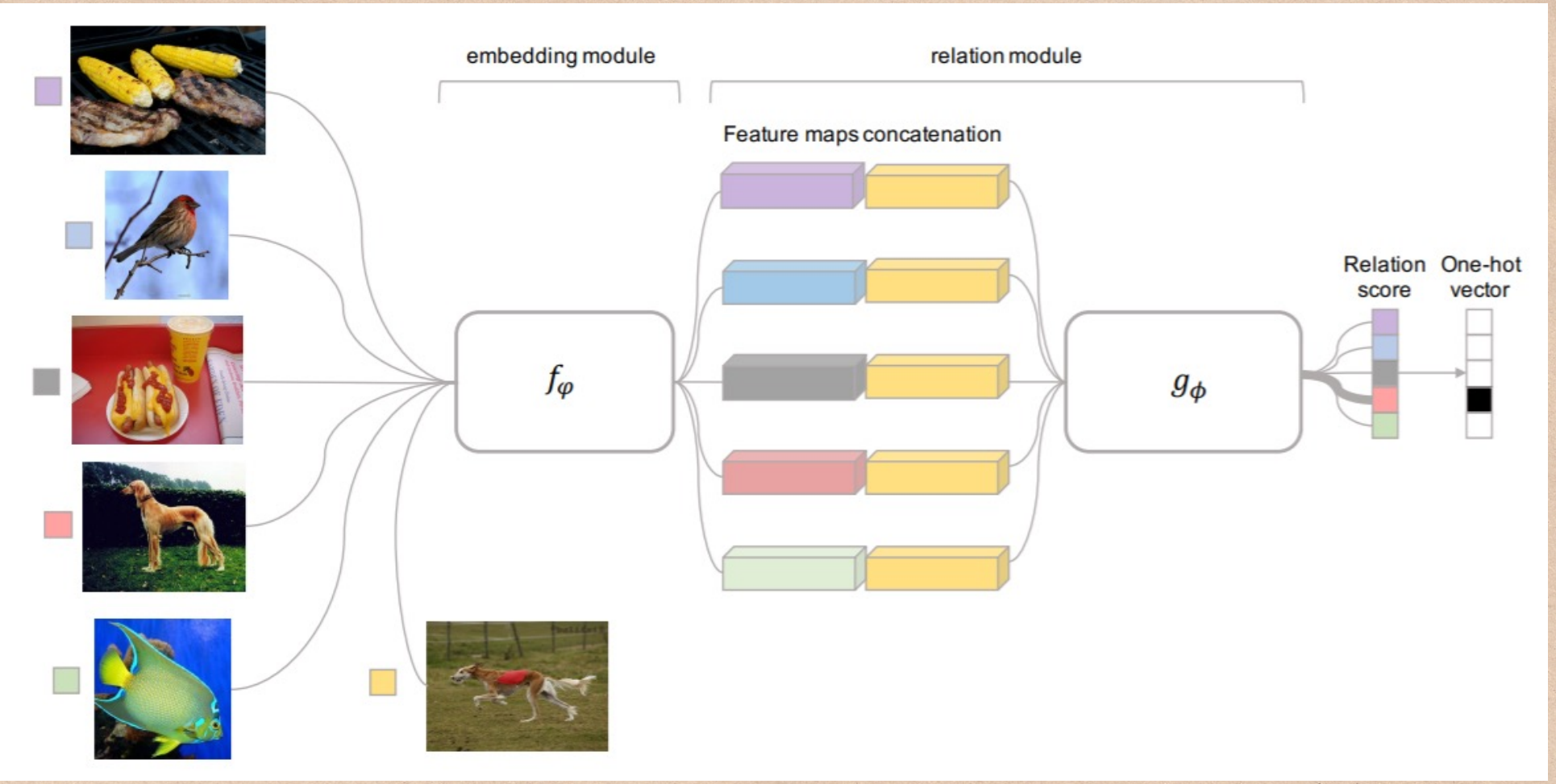
Matching Network



Adapted from https://www.youtube.com/watch?v=semSxPP2Yzg

Metric-Based Meta-Learning

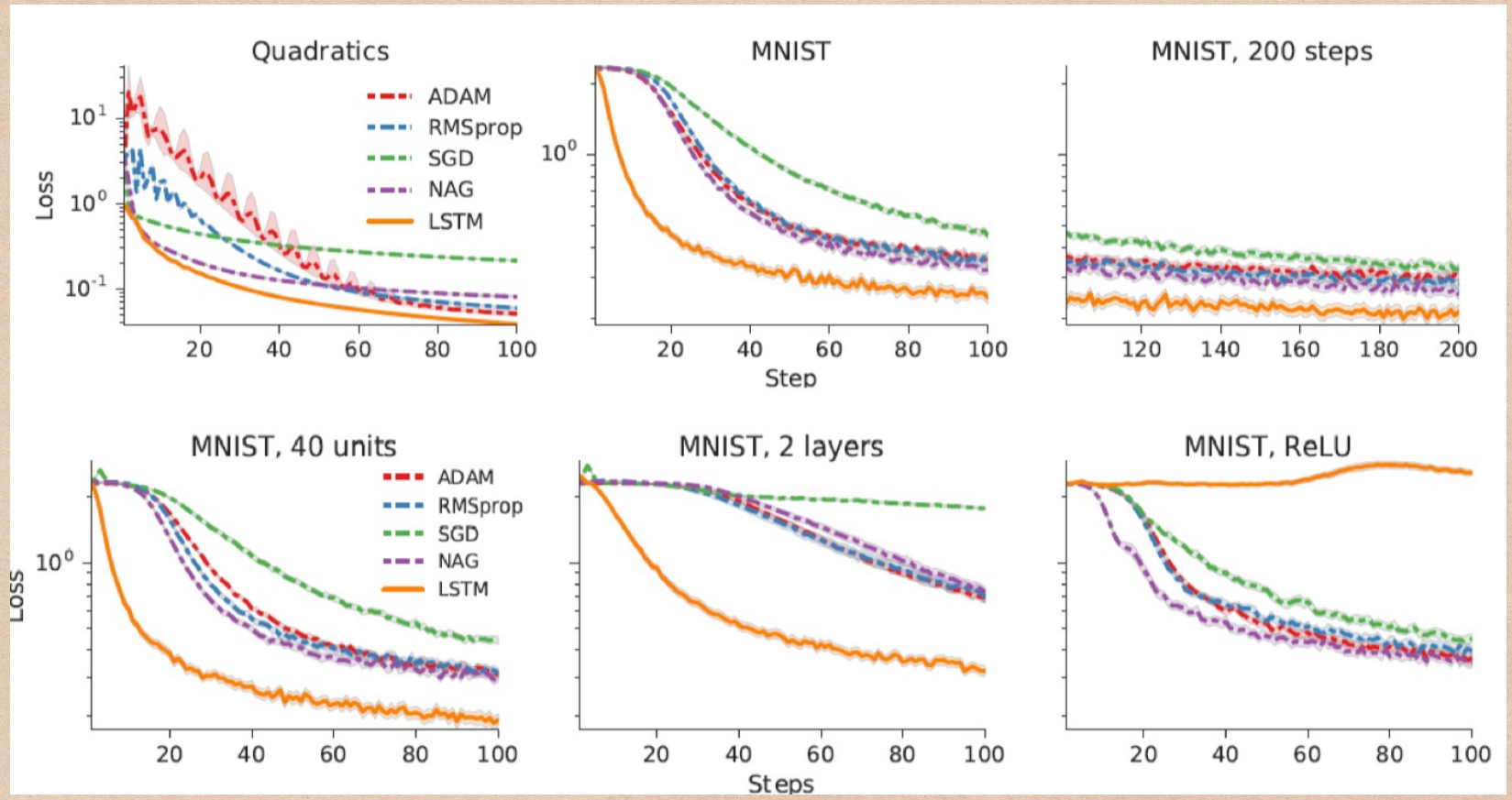
Relation Network



Adapted from https://www.youtube.com/watch?v=semSxPP2Yzg

Application: Optimizer

Learning to learn by gradient descent by gradient descent, 2016



Right: The LSTM optimizer was trained on an MLP with sigmoid activations.

Application: Neural Architectural Search (NAS)

NAS as a sequence-to-sequence problem

Neural Architecture Search with Reinforcement Learning, 2017

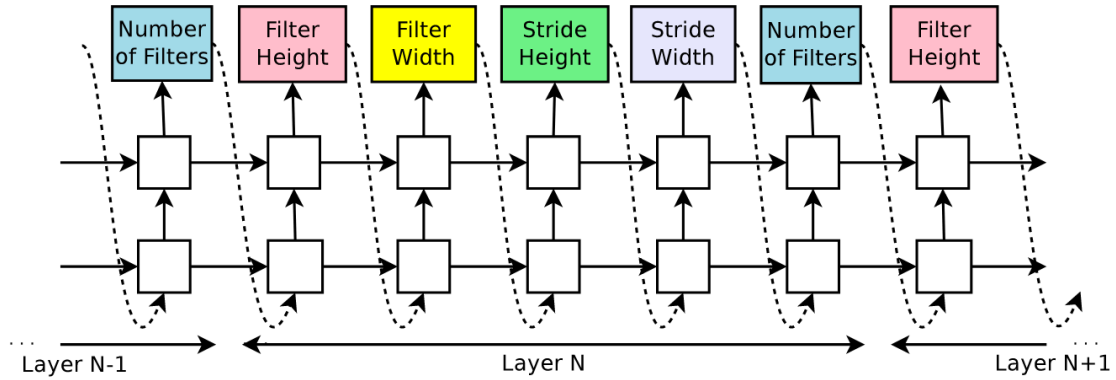


Figure 2: How our controller recurrent neural network samples a simple convolutional network. It predicts filter height, filter width, stride height, stride width, and number of filters for one layer and repeats. Every prediction is carried out by a softmax classifier and then fed into the next time step as input.

Application: Neural Architectural Search (NAS)

Gradient-Based NAS for discovering high-performance CNN for image classification & RNN for language modelling

DARTS: Differentiable Architectural Search, 2019

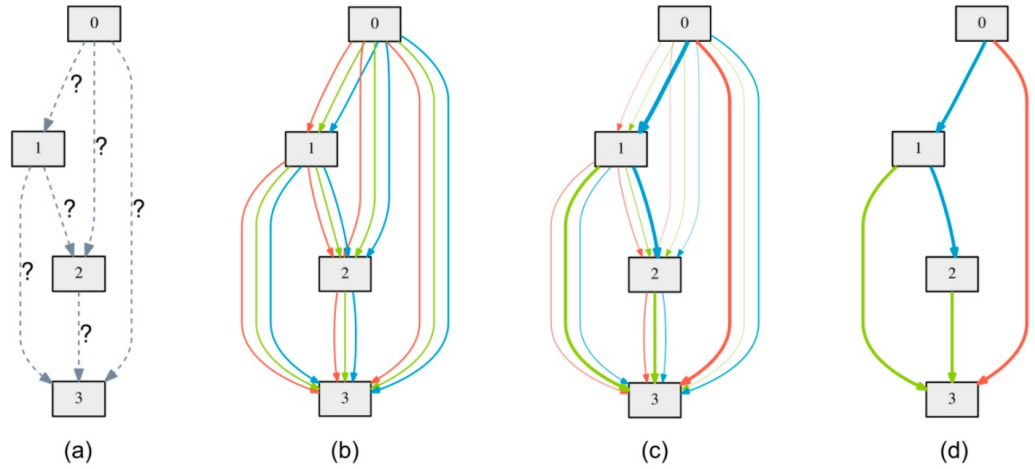


Figure 1: An overview of DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.

Table 3: Comparison with state-of-the-art image classifiers on ImageNet in the mobile setting.

Architecture	**Test Error (%)** top-1	top-5	**Params (M)**	**+× (M)**	**Search Cost (GPU days)**	**Search Method**
Inception-v1 (Szegedy et al., 2015)	30.2	10.1	6.6	1448	–	manual
MobileNet (Howard et al., 2017)	29.4	10.5	4.2	569	–	manual
ShuffleNet 2× ($g = 3$) (Zhang et al., 2017)	26.3	–	∼5	524	–	manual
NASNet-A (Zoph et al., 2018)	26.0	8.4	5.3	564	2000	RL
NASNet-B (Zoph et al., 2018)	27.2	8.7	5.3	488	2000	RL
NASNet-C (Zoph et al., 2018)	27.5	9.0	4.9	558	2000	RL
AmoebaNet-A (Real et al., 2018)	25.5	8.0	5.1	555	3150	evolution
AmoebaNet-B (Real et al., 2018)	26.0	8.5	5.3	555	3150	evolution
AmoebaNet-C (Real et al., 2018)	24.3	7.6	6.4	570	3150	evolution
PNAS (Liu et al., 2018a)	25.8	8.1	5.1	588	∼225	SMBO
DARTS (searched on CIFAR-10)	26.7	8.7	4.7	574	4	gradient-based

Application: Data Augmentation

DADA: Differentiable Automatic Data Augmentation, 2020

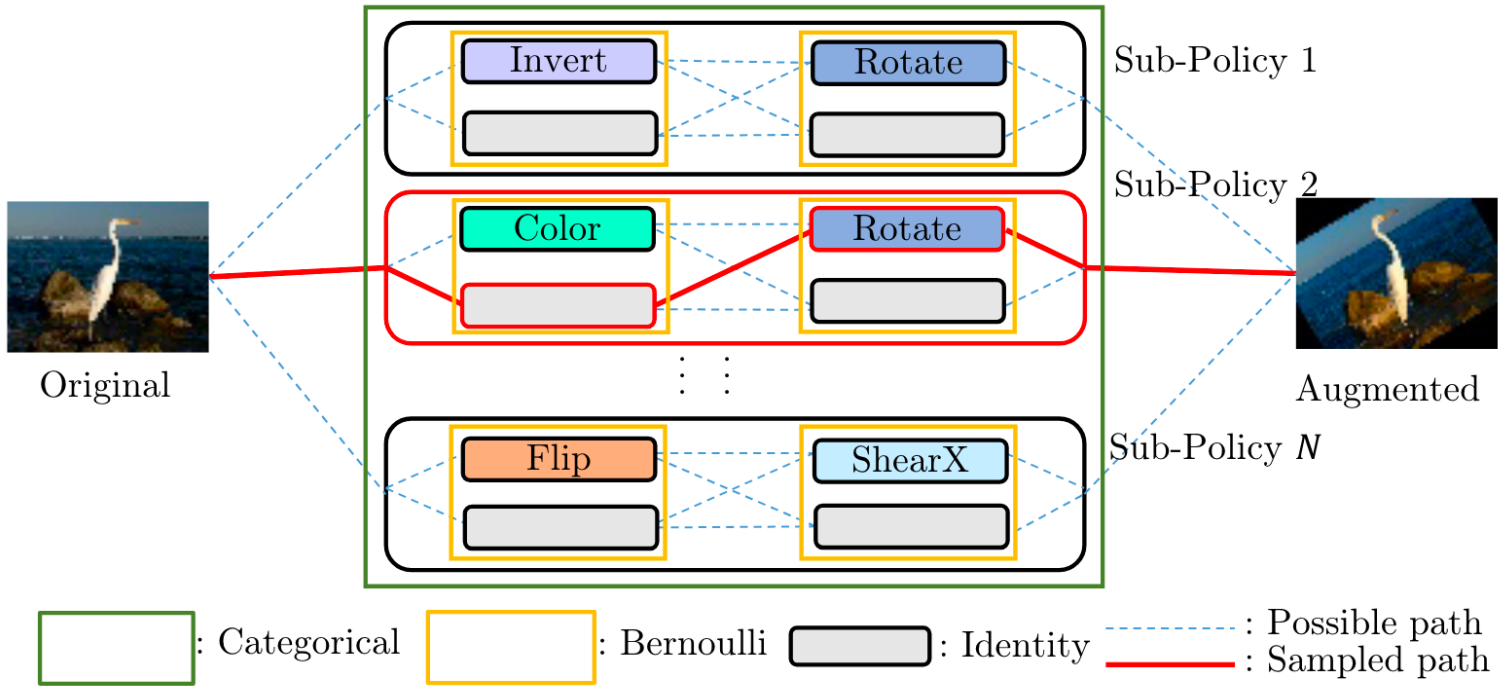
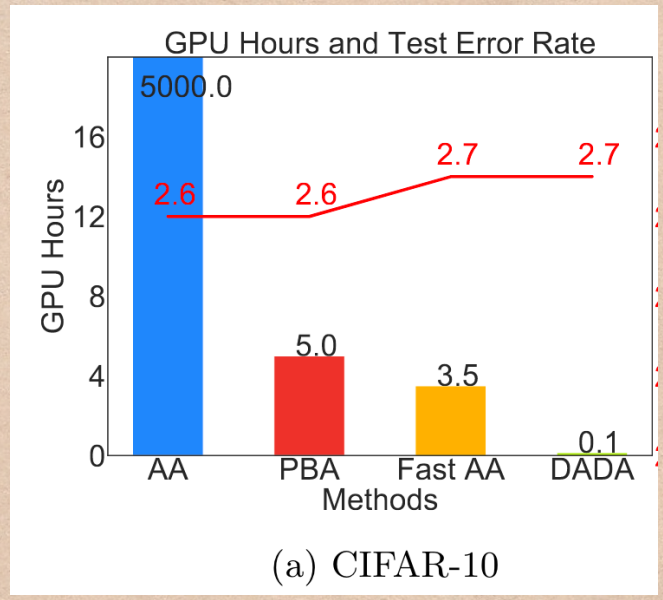


Fig. 2: The framework of DADA. The sub-policies and operations are sampled from Categorical and Bernoulli distributions respectively.

DADA relaxes the discrete DA policy selection to a differentiable optimization problem via Gumbel-Softmax.



(a) CIFAR-10

AA: AutoAugment

PBA: Population Based Augmentation

Fast AA: Fast AutoAugment

Advanced Topics

Unsupervised / Semi-Supervised Meta-Learning

Given unlabeled dataset(s) → Propose tasks → Run meta-learning

Goal:

- Automatically construct tasks from unlabeled data.

For more info:

- Lecture: watch https://youtu.be/4z3ijI9QacE?t=2929

Bayesian Meta-Learning

Incorporating Bayesian methods and probabilistic reasoning with meta-learning.

Goal:

- Explictly model and reason about **uncertainty** in the meta-learning process.

For more info:

- Lecture: watch https://www.youtube.com/watch?v=-y3ufzjgmlY

- Applications in CV: see *Meta-Learning: Theory, Algorithms and Applications: Section 5.2.13*

- Applications in NLP: see *Meta-Learning: Theory, Algorithms and Applications: Section 6.6.3*

Meta Reinforcement Learning

Learning to Explore

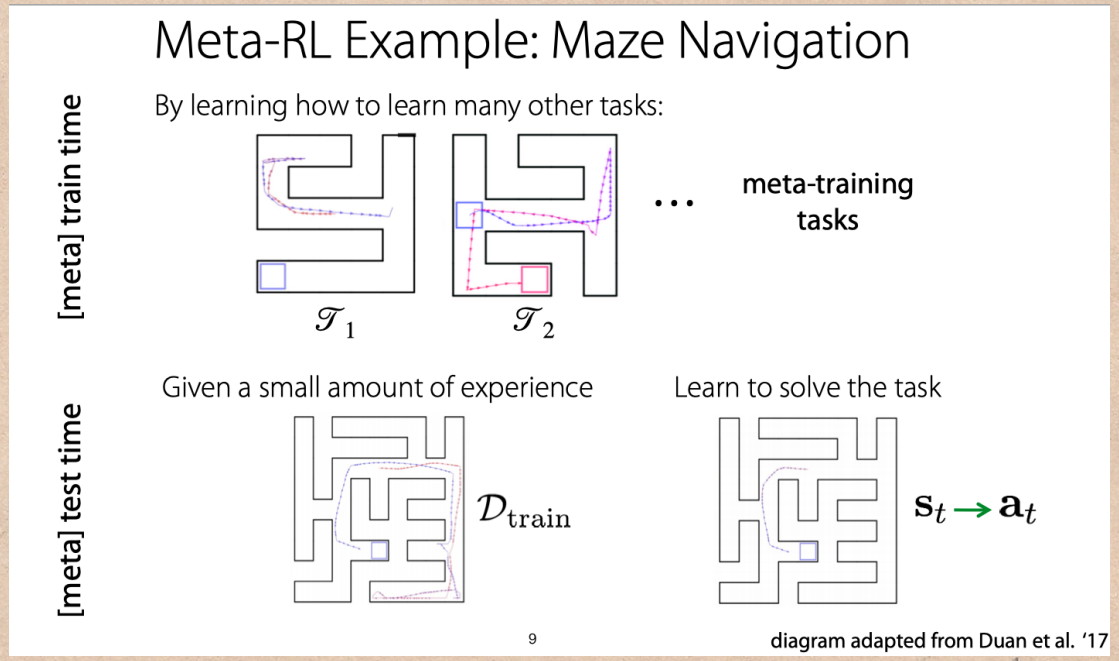
Goal:

- Given a small amount of experience, learn to solve the test task in meta test time;
- by learning how to learn many other task during meta train time

For more info:

- Lecture: watch https://www.youtube.com/watch?v=gUutjhnc2Q4, https://www.youtube.com/watch?v=_KoLFz5BTWw, https://www.youtube.com/watch?v=VGLqzbsOSJY

- see *Meta-Learning: Theory, Algorithms and Applications: Chapter 7*



From **Meta Reinforcement Learning Adaptable Models & Policies, CS330, 2021**

Meta Learning and NLP

Meta-Learning and Self-Supervised Learning

Self-supervised Learning (BERT and pals)

Learn to Init (MAML family)

Adapted from https://youtu.be/QNfymMRUg3M?t=87

Meta-Learning and Self-Supervised Learning

- MAML learns the initialization parameter ϕ by gradient descent
  - In other words, you need a set of initialization parameters to find the initialization parameter for your downstream architecture

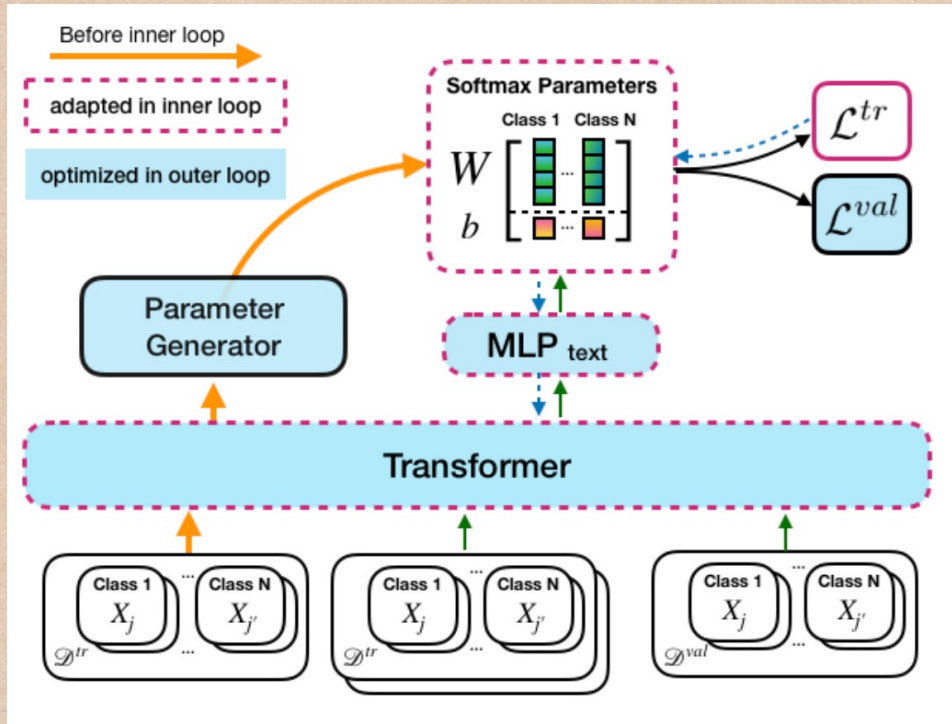
- BERT can serve as ϕ

Adapted from https://youtu.be/QNfymMRUg3M?t=87

Meta-Learning and NLP

Work	Method	How to Initailize the Initailization
(Bansal et al., 2020a)	LEOPARD	BERT
(Li et al., 2020a)	MAML	Word Embedding
(Park et al., 2021)	MAML	XLM
(Gu et al., 2018)	FOMAML	Word Embedding
(Langedijk et al., 2021)	FOMAML	mBERT
(Chen et al., 2020b)	Reptile	BART
(Huang et al., 2020a)	MAML	BERT
(Wang et al., 2021b)	Propose a new method based on Reptile	Word Embedding
(Dingliwal et al., 2021)	Reptile	RoBERTa
(Qian and Yu, 2019)	MAML	Word Embedding
(Qian et al., 2021)	MAML	Word Embedding
(Madotto et al., 2019)	MAML	Word Embedding
(Dai et al., 2020)	MAML	-
(Hsu et al., 2020)	FOMAML	Multilingual ASR
(Xiao et al., 2021)	MAML/FOMAML/Reptile	-
(Winata et al., 2020b)	MAML	Pretrain by Supervised Learning
(Klejch et al., 2019)	FOMAML	-
(Huang et al., 2021)	MAML/FOMAML	-
(Indurthi et al., 2020)	FOMAML	-
(Winata et al., 2020a)	FOMAML	-
(Wu et al., 2021b)	MAML	Pretrain by Multi-task Learning
(Ke et al., 2021)	MAML	BERT
(Xia et al., 2021)	MetaXL	mBERT/XLM-R
(Dou et al., 2019)	MAML/FOMAML/Reptile	BERT
(Obamuyide and Vlachos, 2019b)	FOMAML	Word Embedding
(Lv et al., 2019)	MAML	-
(Holla et al., 2020)	FOMAML/Proto(FO)MAML	Word Embedding/ELMo/BERT
(Huang et al., 2020b)	MAML	Word Embedding
(Mi et al., 2019)	MAML	-
(Wang et al., 2021a)	DG-MAML	BERT
(Conklin et al., 2021)	DG-MAML	-
(M'hamdi et al., 2021)	MAML	mBERT
(Nooralahzadeh et al., 2020)	MAML	BERT/mBERT/XLM-R
(Garcia et al., 2021)	MAML	mBERT
(van der Heijden et al., 2021)	FOMAML/Reptile/Proto(FO)MAML	XLM-R
(Bansal et al., 2020b)	LEOPARD	BERT
(Murty et al., 2021)	FOMAML	BERT
(Hua et al., 2020)	Reptile	-
(Yan et al., 2020)	MAML	BERT/RoBERTa
(Wang et al., 2019b)	Reptile	-
(Bose et al., 2020)	Meta-Graph	-

From **Meta Learning for Natural Language Processing: A Survey, 2022**

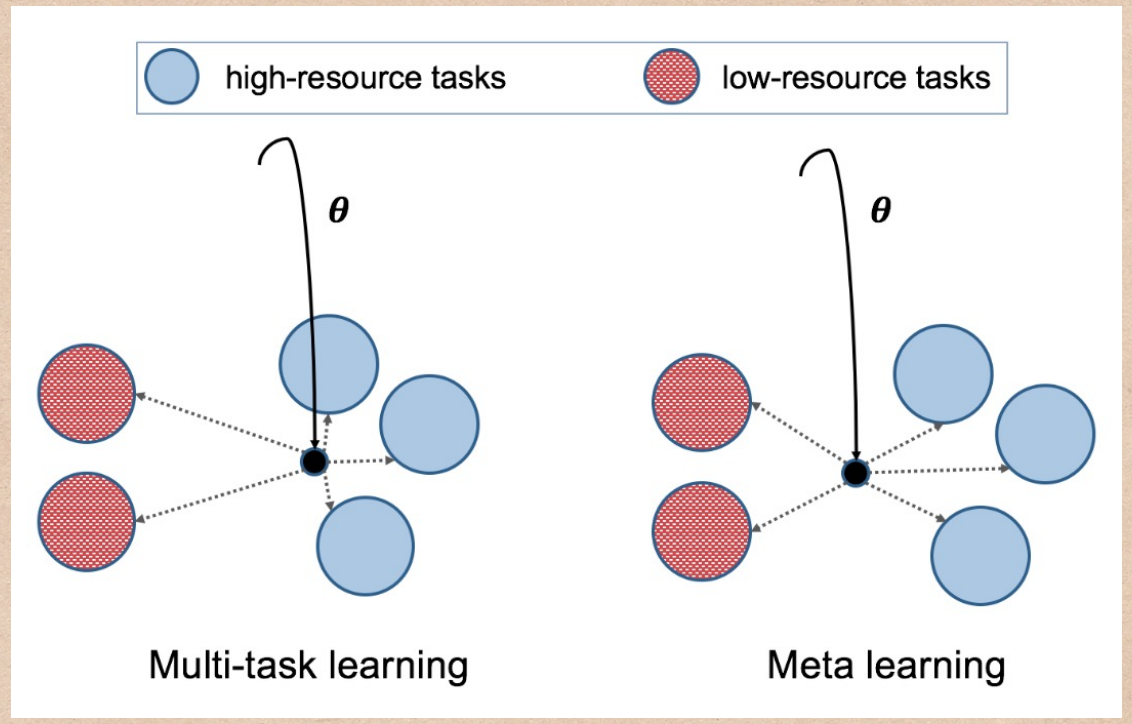


From **Learning to Few-Shot Learn Across Diverse Natural Language Classification Tasks, 2020**

LEOPARD model:
for learning new NLP classification tasks with k-examples and with *different number of classes*

Meta-Learning and NLP

Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks, 2019



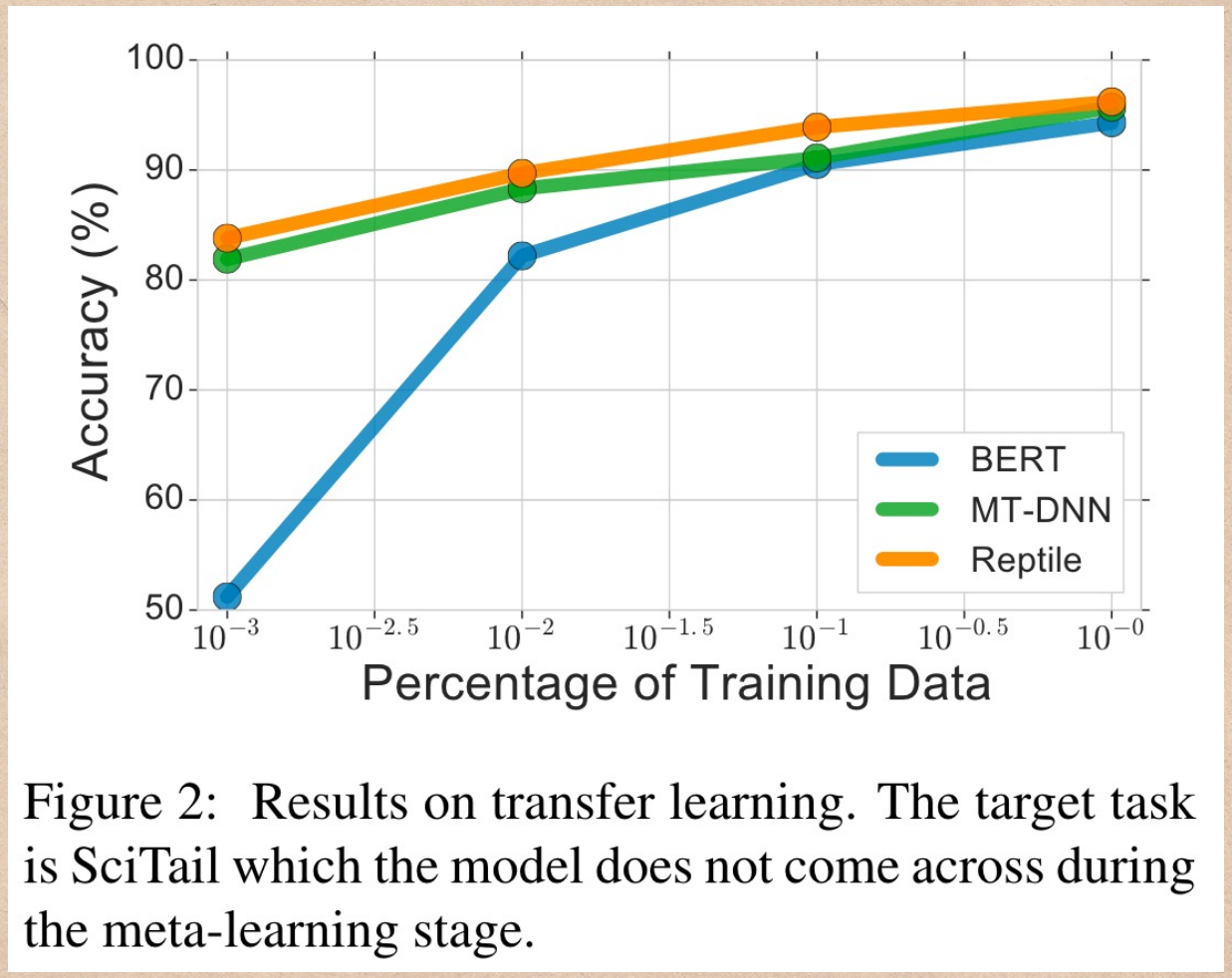


Figure 2: Results on transfer learning. The target task is SciTail which the model does not come across during the meta-learning stage.

Multi-task learning is a base line of Meta-Learning

The SciTail dataset is an entailment dataset created from multiple-choice science exams and web sentences.

Q & A