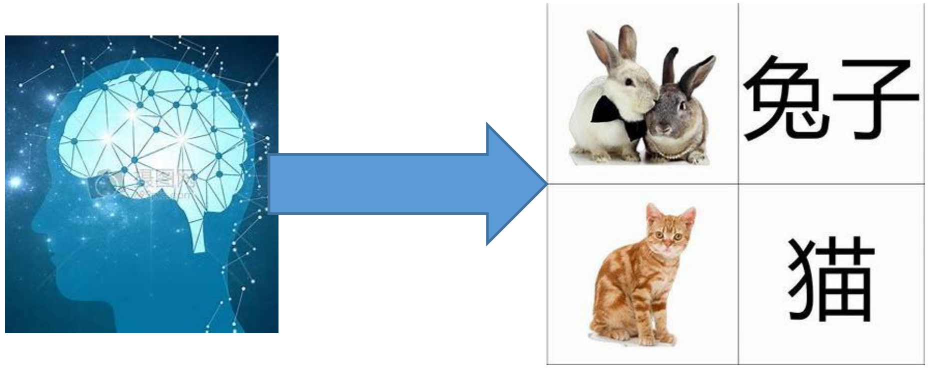
Few-Shot Learning

in views of
Meta, Metric and Prompt Learning

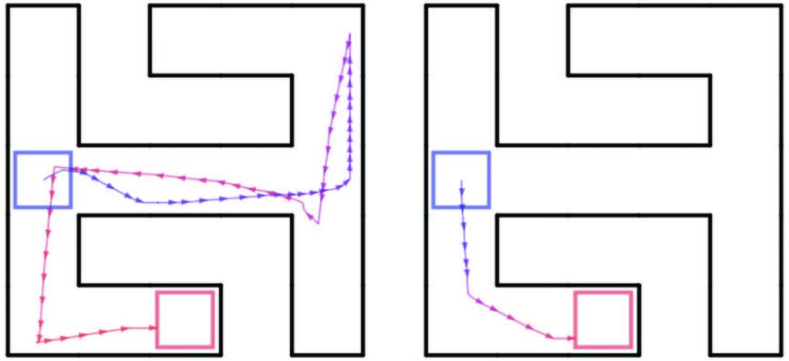
Chenyou Fan
School of Artificial Intelligence, SCNU

Few-Shot Learning - 少样本(小样本)学习

- Humans can recognize new concepts very quickly



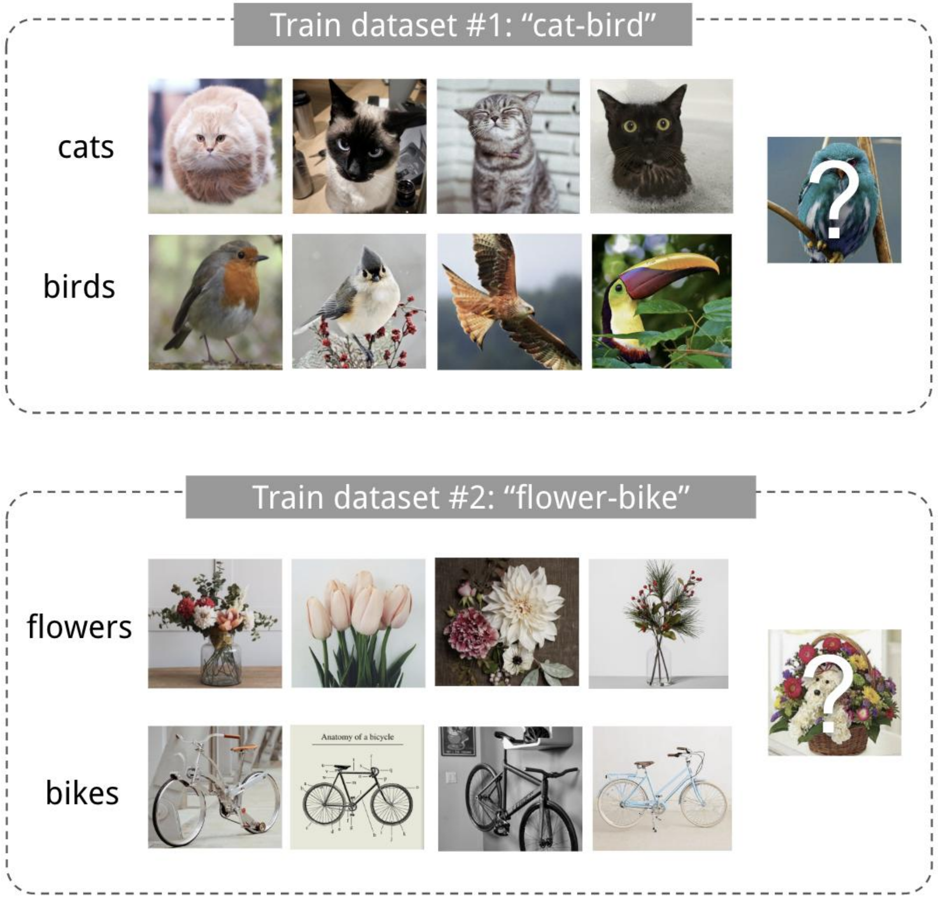
- Humans can learn to play a game by just observing a few game replays.



learning to quickly navigate new mazes
Duan et al.'16

Few-Shot Learning

Training (huge)



Support (few)

N-way K-shot classification task

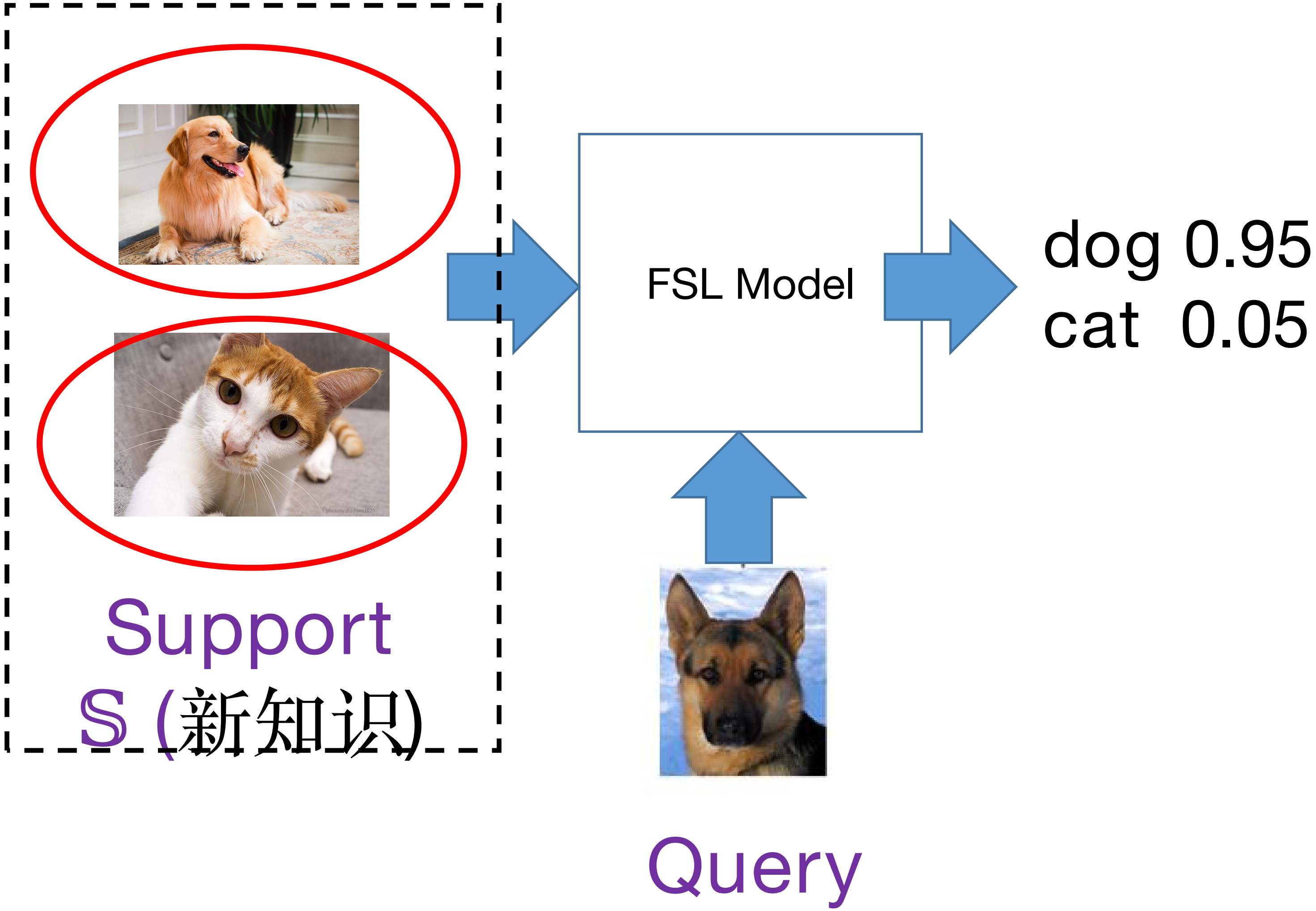
- The Support set (𝕊) contains N*K data instances (新知识)
- N-way： there are N new data classes
- K-shot： each class has K data instances. (K = 1, 5, etc)
- Task: for a new query data, find its correct label (1-of-N classification)

- Training set (𝕋) has non-overlapping classes with support set. (一般知识)
- Learn a model on 𝕋(大样本) which can quickly apply to 𝕊 (少样本).

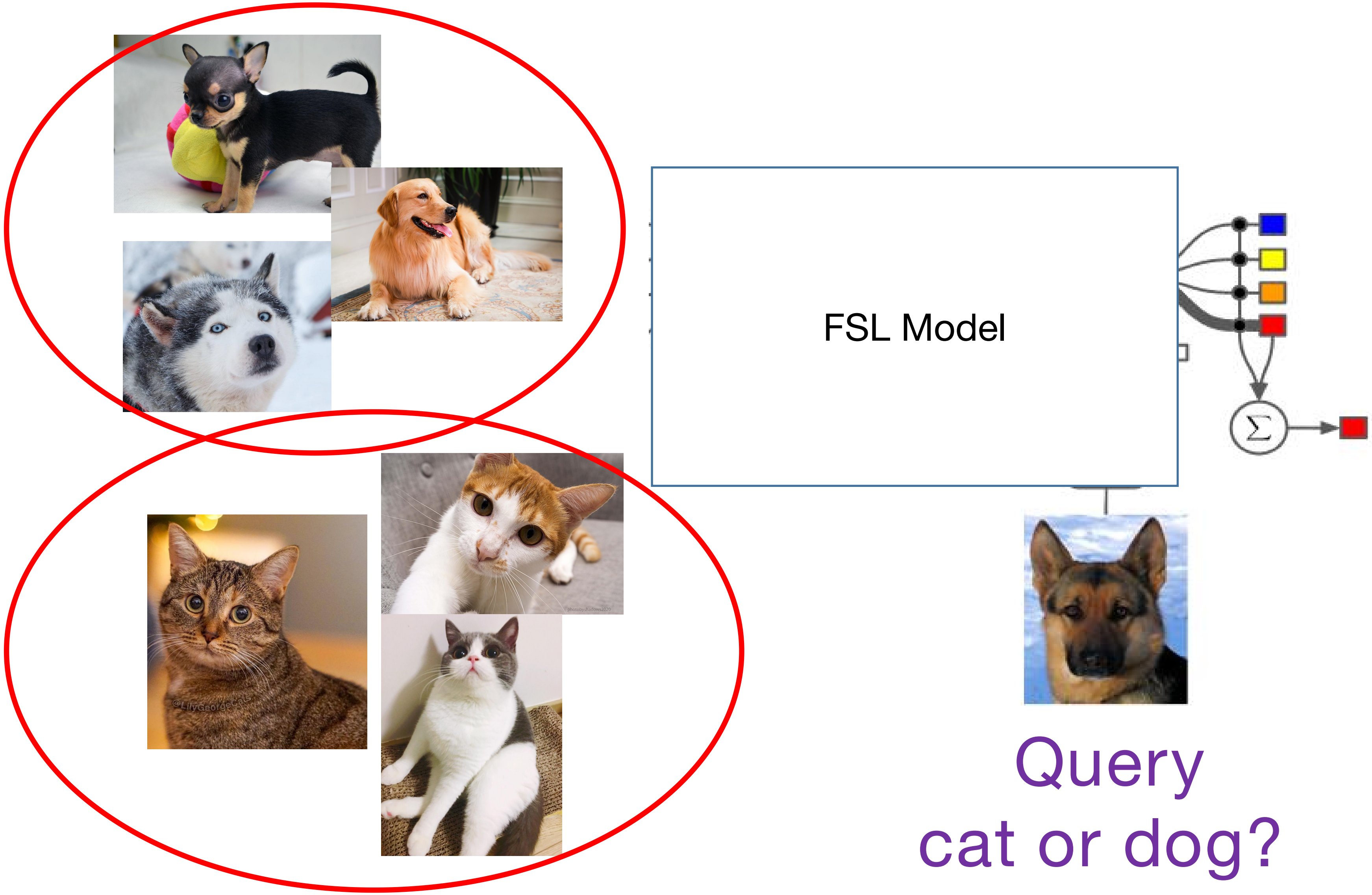
Demo of 2-Way 1-shot FSL

2 animal classes (2-way)

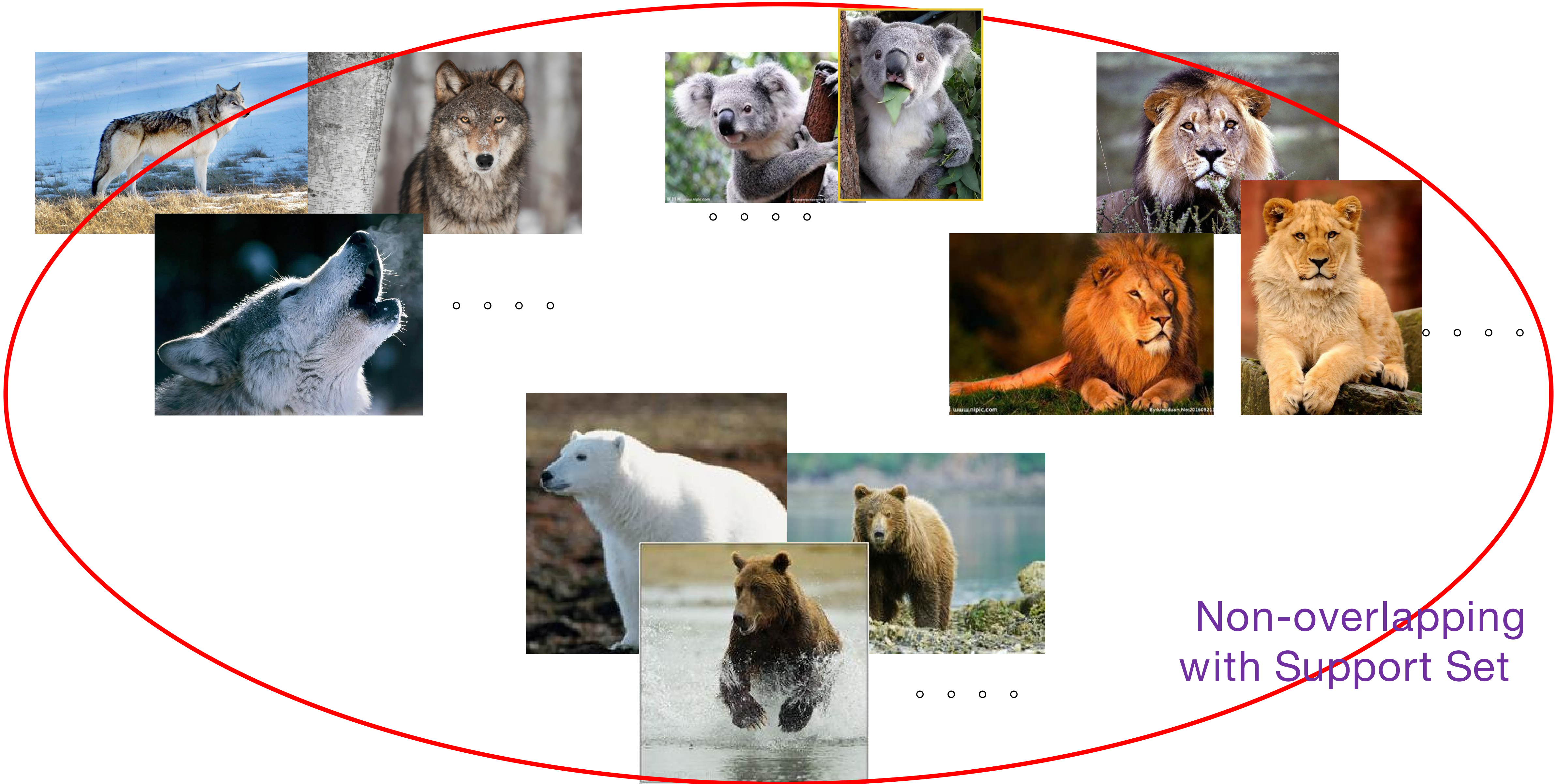
Each class has ONE data (1-shot)

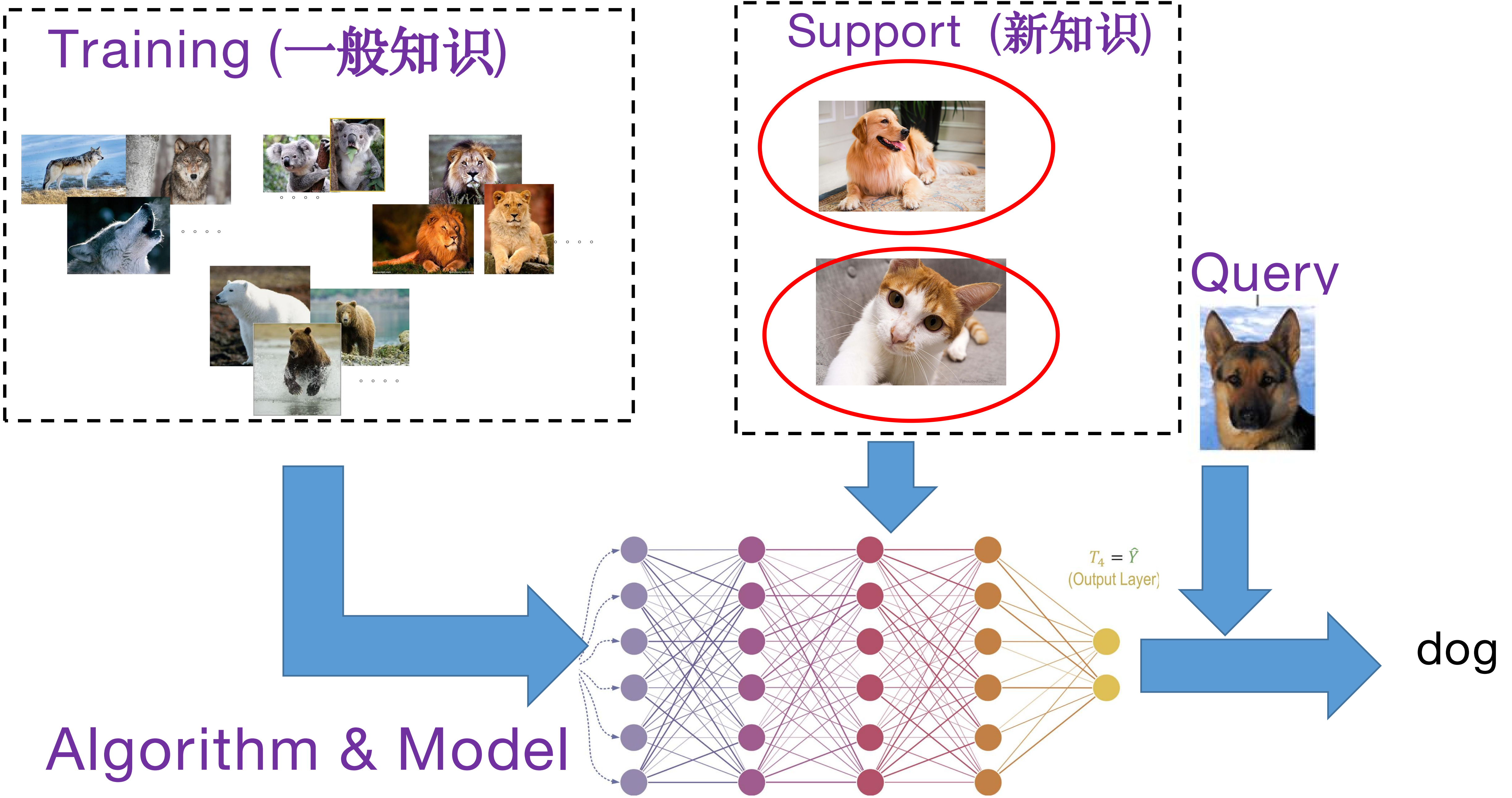


Demo of 2-Way 3-shot FSL



Training Set (common knowledge)




Training (一般知识)
Support（新知识)
Query
$T_4 = \hat{Y}$
(Output Layer)
dog
Algorithm & Model

Topic-1 - FSL with Meta Learning (元学习)

- Meta-Learning is learning to learn;
- Meta-Learning searches a set of optimal parameter
- which can adapt to N-way K-shot FSL tasks.

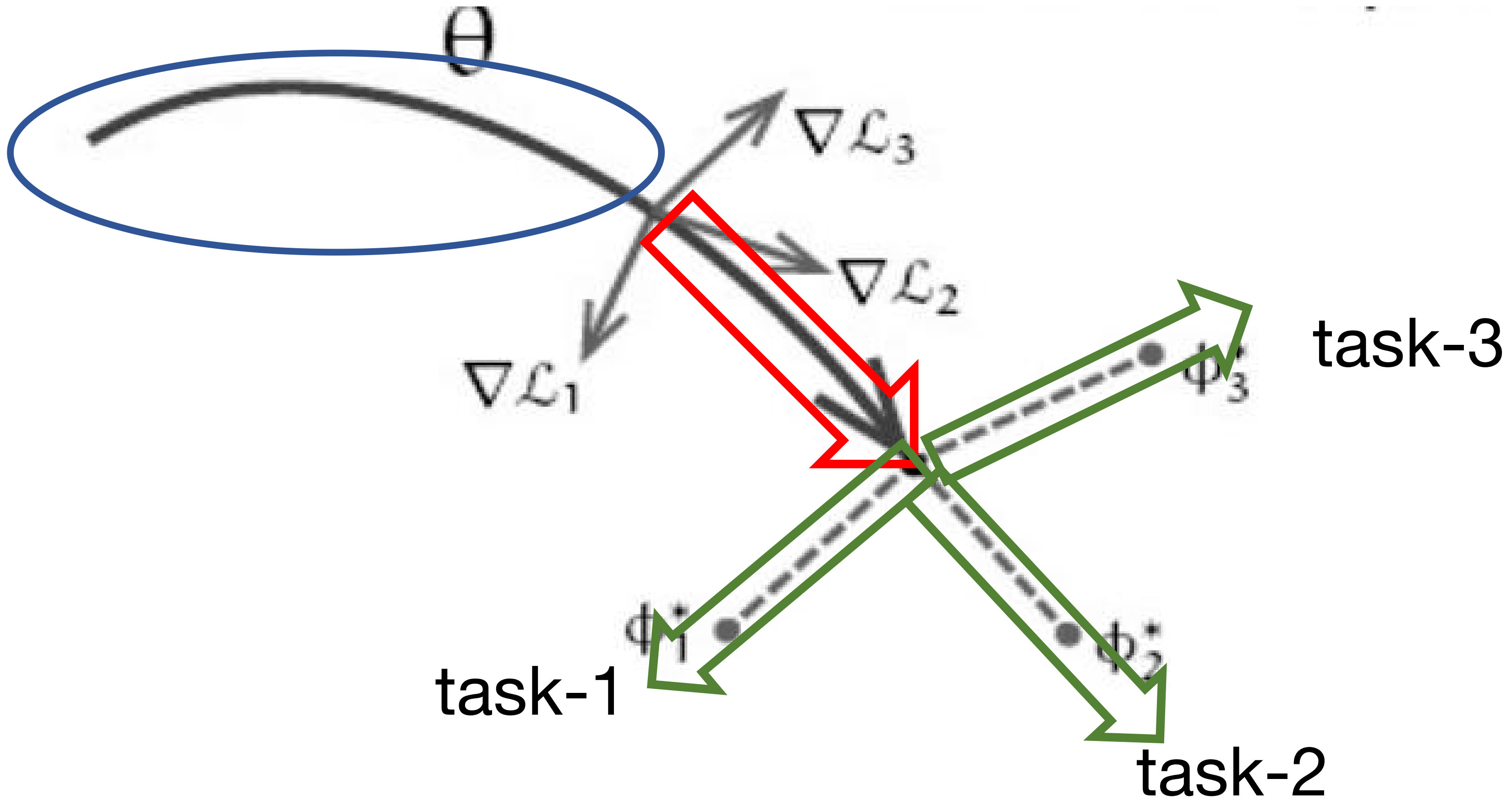
Topic-1 - FSL with Meta Learning (元学习)

- In DL context, Meta-learning constructs a base-NN

- The base-NN can tweak its parameters on a new task (with a few data) with SGD to form a target-NN.

- The target-NN performs significantly better than base-NN on the new task.

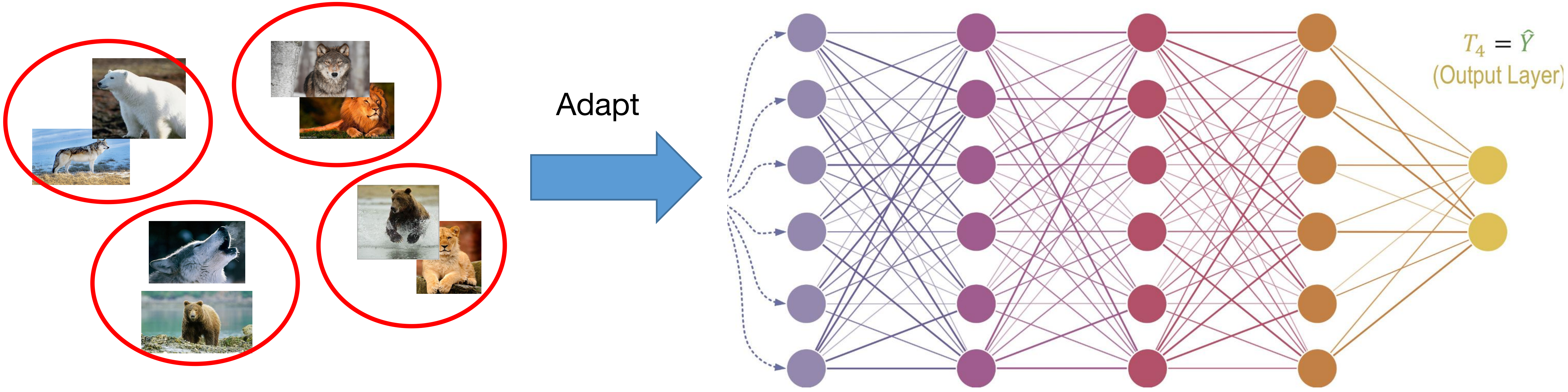
Topic-1 - FSL with Meta Learning (元学习)

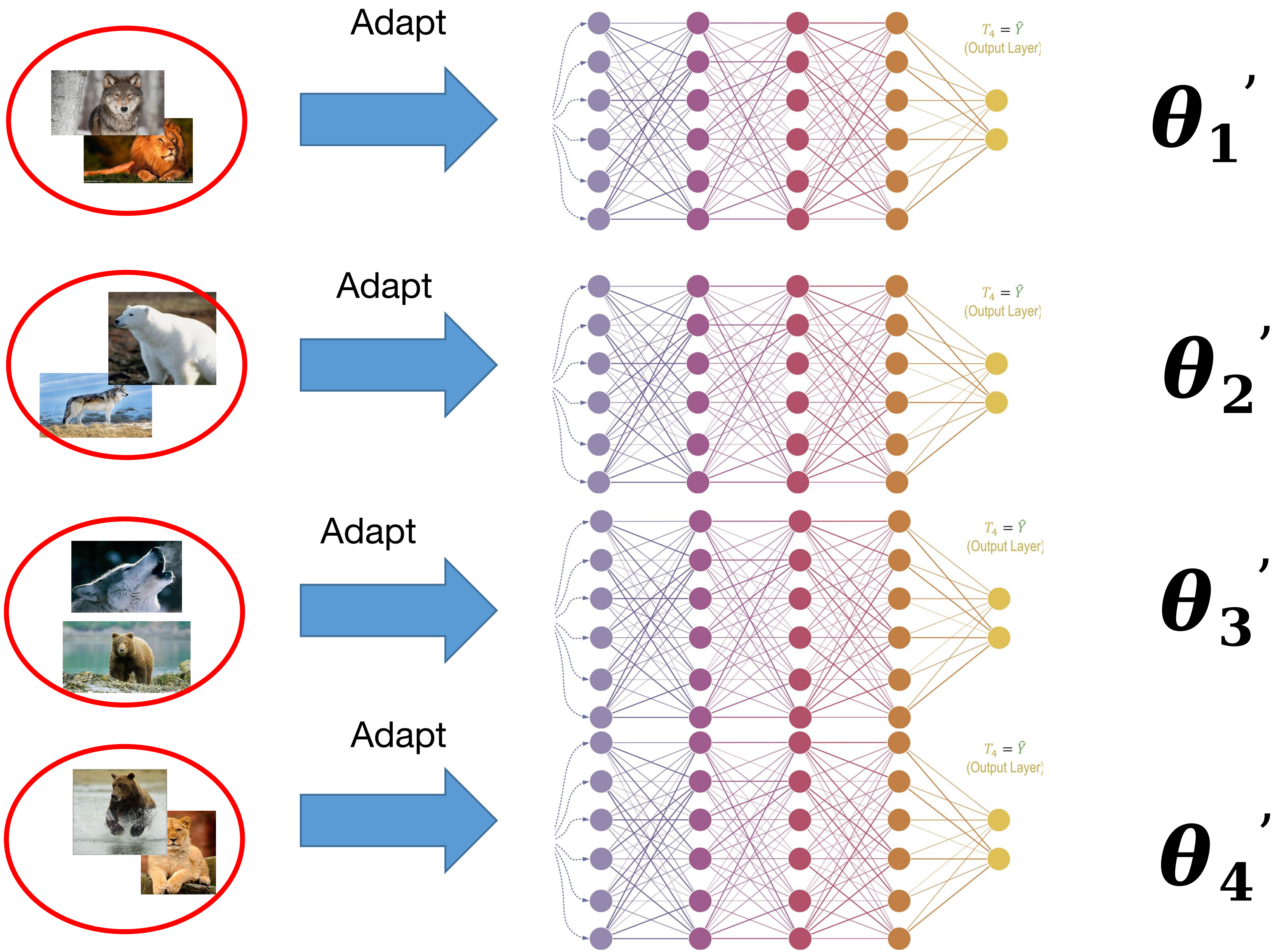


Meta-training stage-1: construct base-NN

- Sample batches of (K-way N-shot) tasks as support data

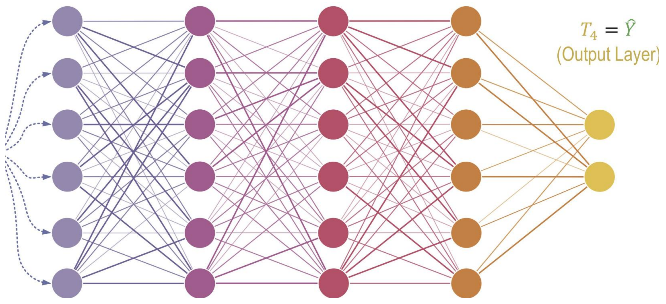
- Adapt the base-NN θ to each task with SGD $\theta' = \theta - \alpha \nabla L(f_\theta)$




Adapt
$T_4 = \hat{Y}$
(Output Layer)
θ_1'
Adapt
$T_4 = \hat{Y}$
(Output Layer)
θ_2'
Adapt
$T_4 = \hat{Y}$
(Output Layer)
θ_3'
Adapt
$T_4 = \hat{Y}$
(Output Layer)
θ_4'

Meta-training stage-2: optimize target-NN

- Given a query image and its correct label
- Optimize the target-NN by another SGD $\theta'' = \theta' - \beta \nabla L(f_{\theta'})$

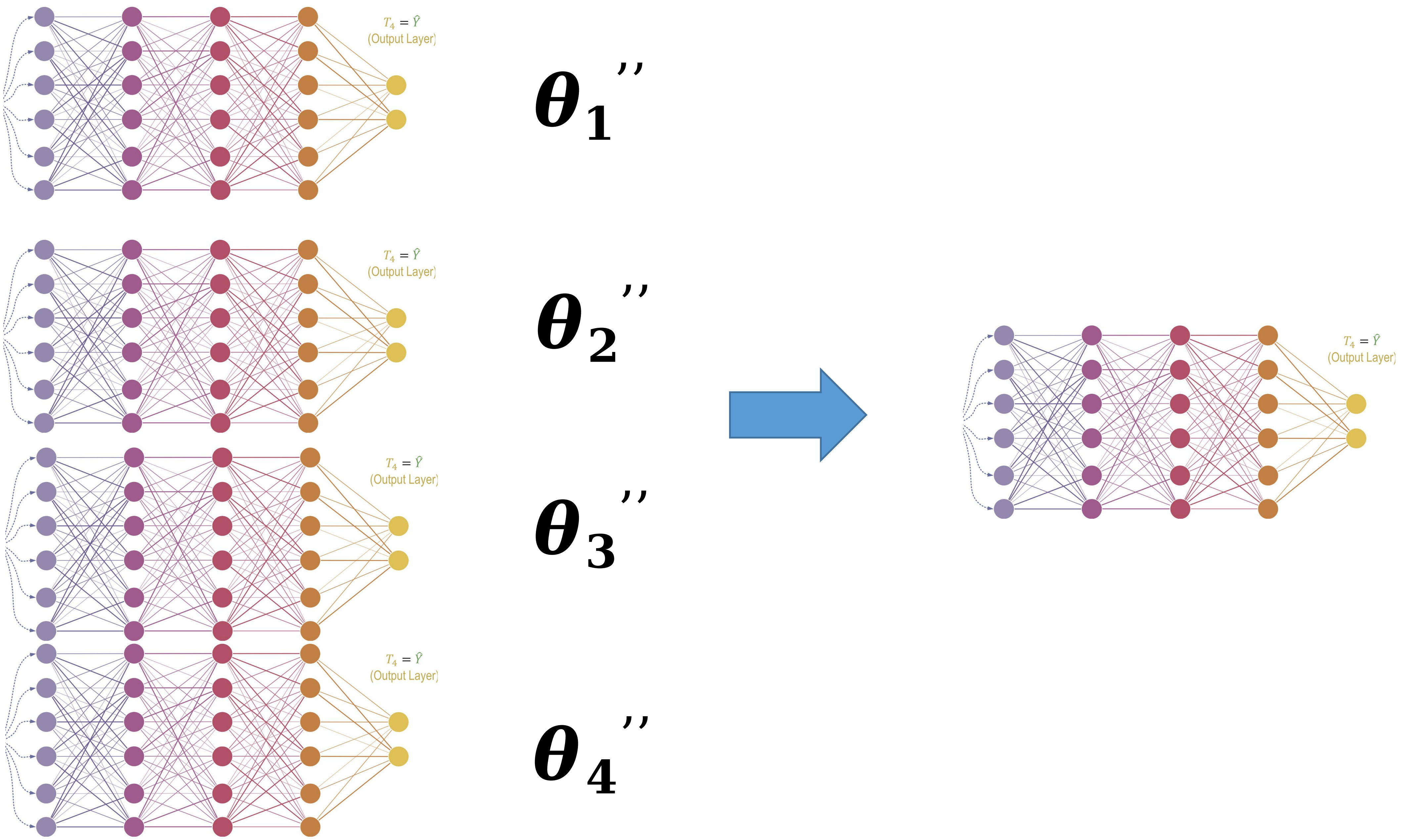


$\boldsymbol{\theta_1}'$

classify

wolf 0.6
lion 0.4

$\boldsymbol{\theta_1}''$

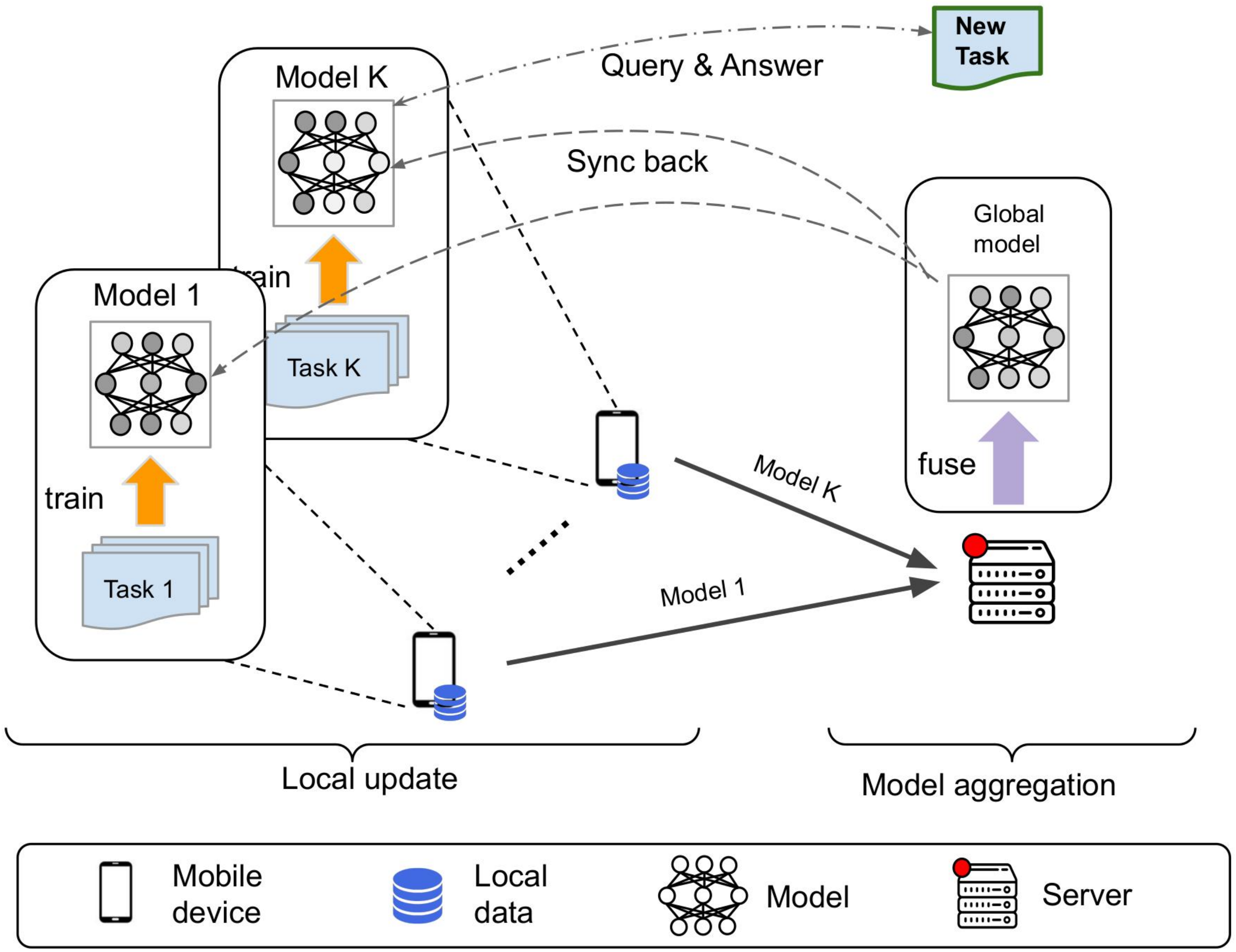

$T_4 = \hat{Y}$
(Output Layer)
θ_1''
$T_4 = \hat{Y}$
(Output Layer)
θ_2''
$T_4 = \hat{Y}$
(Output Layer)
θ_3''
$T_4 = \hat{Y}$
(Output Layer)
θ_4''
$T_4 = \hat{Y}$
(Output Layer)

Overview of Few-Shot Federated Learning (FedFSL)

Few-shot image classification task

- Stage-1: [Local Update] multiple clients are training with local data and perform local update in parallel
- Stage-2: [Model Aggregation] each client sends its model to a central server for aggregation to a global model
- Stage-3: [Synchronize] the global model is synchronized back to each client for next round of training

Federated Few-Shot Learning with Adversarial Learning. Fan et al., 2021

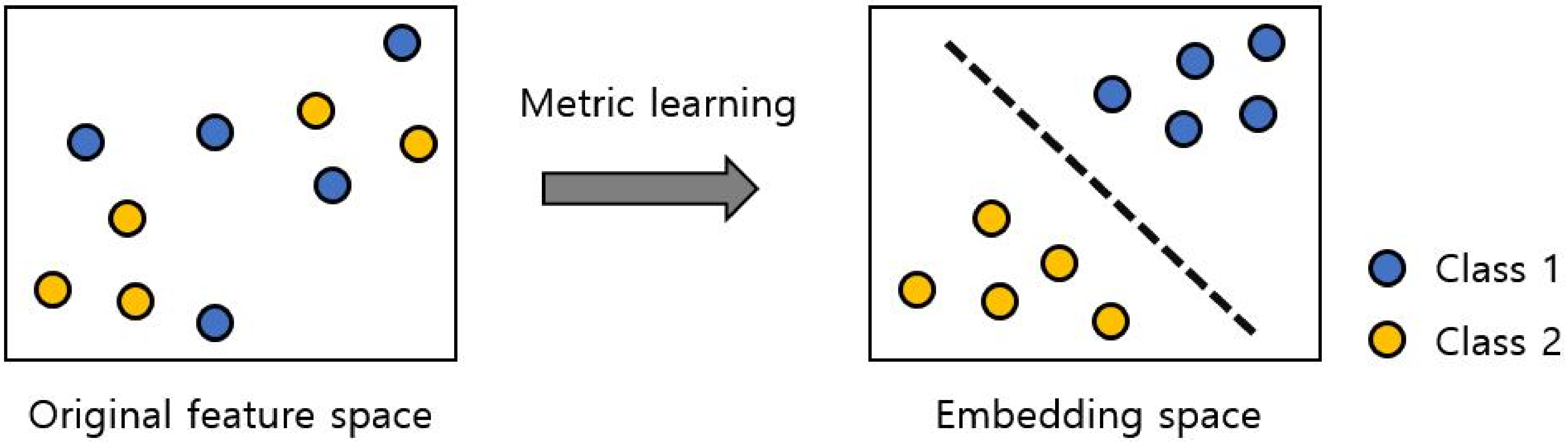


Topic-2 - FSL based on Metric Learning

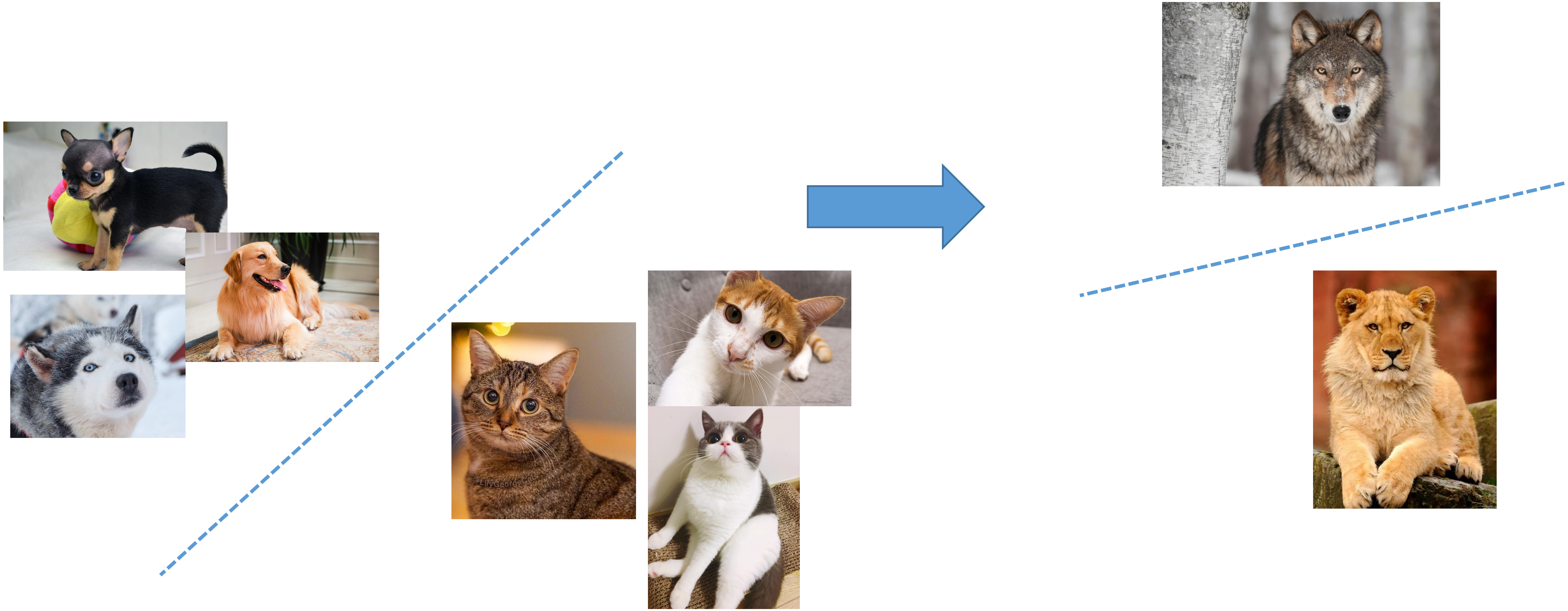
- 度量学习
- Matching Net [Vinyals et al., 2016]
- Prototypical [Snell et al., 2017]
- DeepEMD [Zhang et al., 2020]

Topic-2 - FSL based on Metric Learning

- Learning a feature space in which
    - Data samples of a same class are close;
    - Data samples of different classes are far away.

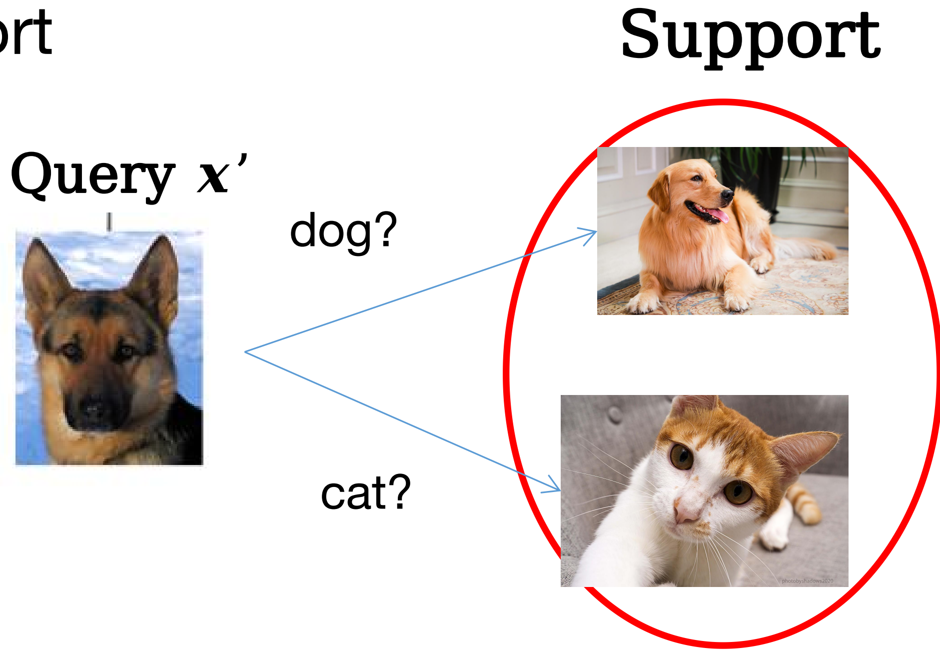


Demo of Metric Learning



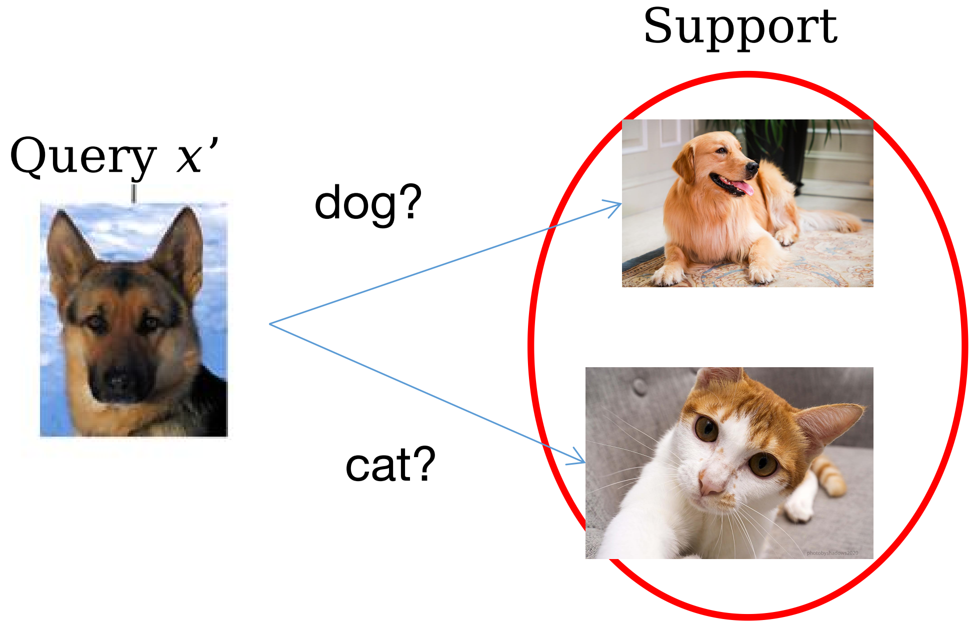
Matching Net [Vinyals et al., 2016]

- For a query image, find the most similar support image in support set and assign that label.



- Learn the distance metric $\boldsymbol{A}$
- p(dog) = $\boldsymbol{A}$( , ) $\binom{1}{0}$
- p(dog) = $\boldsymbol{A}$( , ) $\binom{0}{1}$

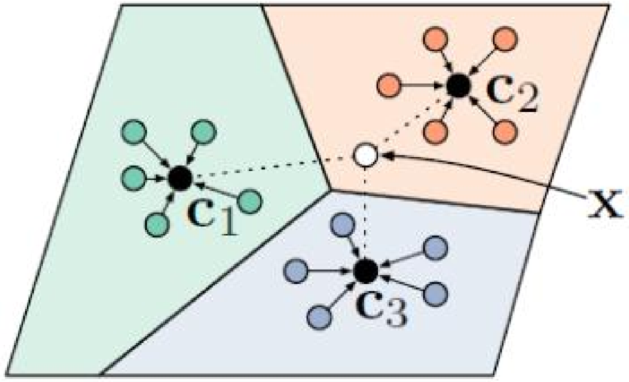
Matching Net [Vinyals et al., 2016]



- Let training set $S = \{(x_i, y_i)\}_{i=1}^{k}$
- $y' = \sum_{i=1}^{k} A(x', x_i)\, y_i$
- p(dog) = A( , ) $\binom{1}{0}$
- p(dog) = A( , ) $\binom{0}{1}$

Prototypical Network [Snell et al., 2017]

- Define c_n as the centroid of n-th class.



$$c_n = \frac{1}{|S_k|} \sum_{(x_i, z_i) \in S_k} f_w(x_i)$$

- The metric-based FSL objective is the cross-entropy loss

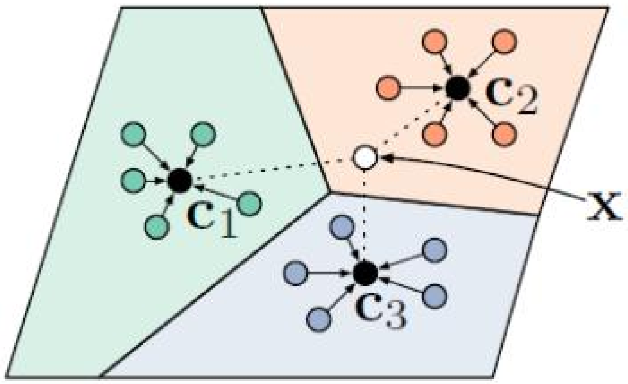
$$\min_{w,\pi} \mathcal{L}(w, \pi) = \sum_{i=1}^{|S|} \log \frac{\exp(-d(f_w(x_i), \boldsymbol{c}_n))}{\sum_{n'} \exp(-d(f_w(x_i), \boldsymbol{c}_{n'})}$$

in which d is a distance function such as Euclidean.

Prototypical Network [Snell et al., 2017]

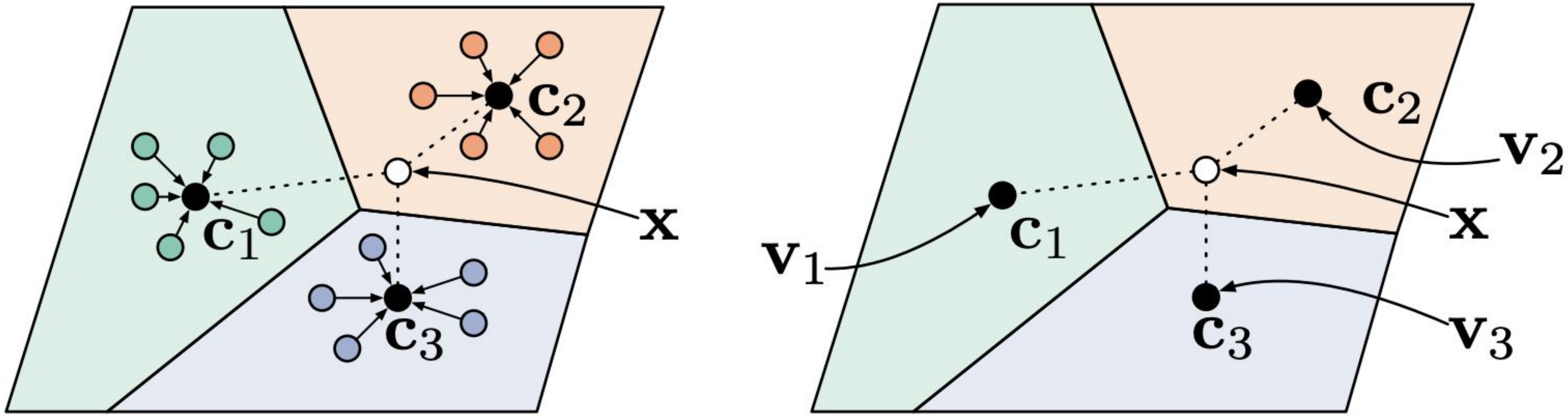
- Conventional wisdom: iteratively and alternatively update w, π.
- Update π by its K nearest neighbors $z_\pi(i) = \arg\min_{n'} d(f(x_i), \boldsymbol{c}_{n'})$.
- Update c_n by the average features from its K nearest neighbors

$$c_n = \frac{1}{|S_k|} \sum_{(x_i, z_i) \in S_k} f_w(x_i)$$



- Update w with gradient steps

$$w \leftarrow w - \eta \nabla_w \mathcal{L}(w, \pi)$$



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

Figure 1: Prototypical Networks in the few-shot and zero-shot scenarios. **Left**: Few-shot prototypes $\mathbf{c}_k$ are computed as the mean of embedded support examples for each class. **Right**: Zero-shot prototypes $\mathbf{c}_k$ are produced by embedding class meta-data $\mathbf{v}_k$. In either case, embedded query points are classified via a softmax over distances to class prototypes: $p_{\boldsymbol{\phi}}(y = k|\mathbf{x}) \propto \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k))$.

Distributed Metric-Learning

"Few-Shot Multi-Agent Perception." Fan et al., 2021, ACM MM.

Topic-3 - FSL with Prompt Learning (提示学习)

Definition

- specify an NLP task as a template
- then use a pre-trained language model (PLM) to interpret and answer
- convert the answers to task labels

Zero-Shot

```
Translate English to French:     ← task description
cheese =>                        ← prompt
```

prompt (提示)

One-Shot

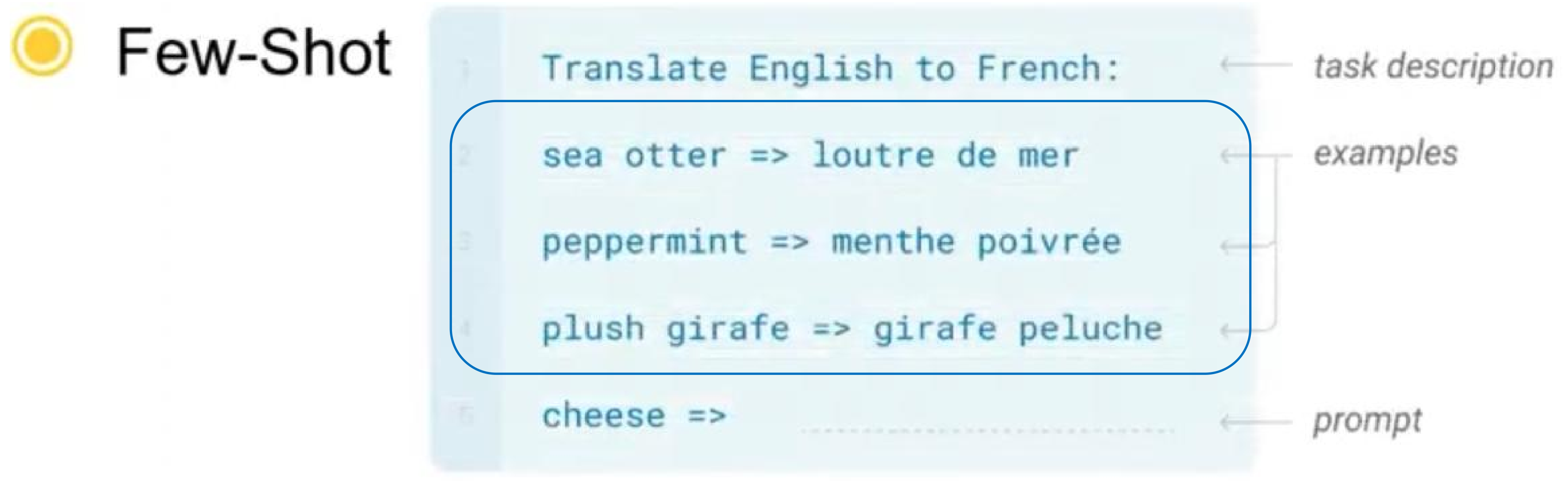
```
Translate English to French:     ← task description
sea otter => loutre de mer       ← example
cheese =>                        ← prompt
```

Few-Shot

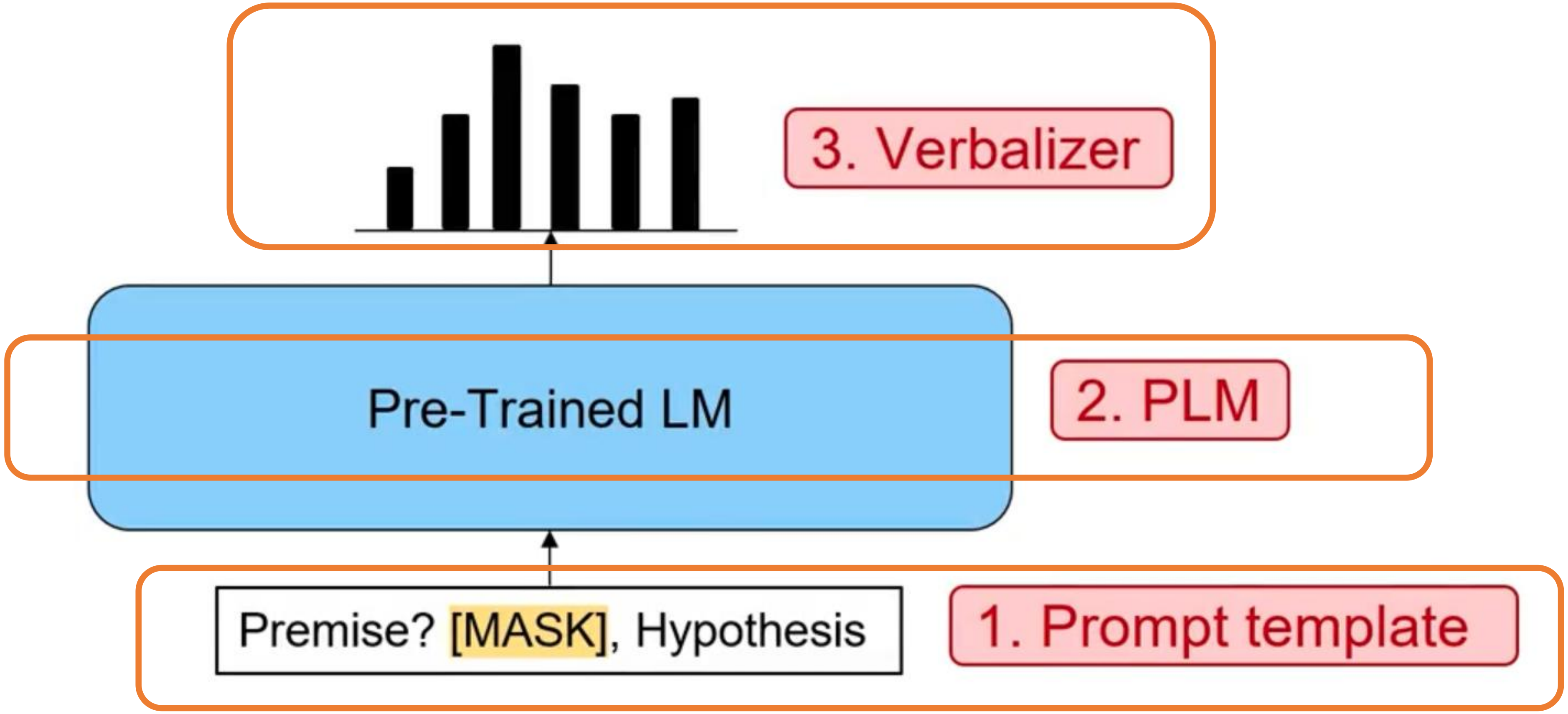
```
Translate English to French:           ← task description
sea otter => loutre de mer             ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                              ← prompt
```

- Zero-shot Prompt learning

- Assume a powerful PLM to solve your tasks



- Prompt-finetuning
- Provide a few examples to better tune the model
- Activate the PLM to do better on your tasks



Methodology

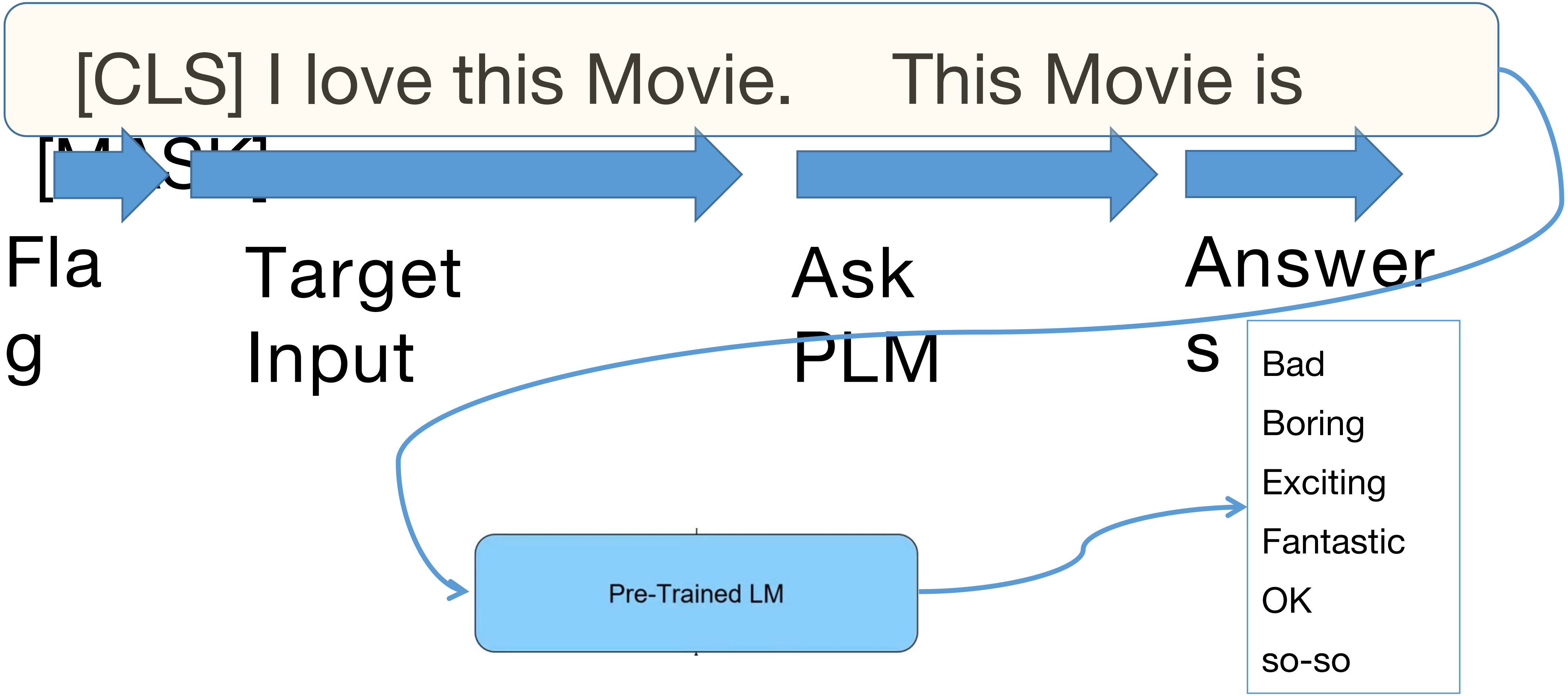
- Reformulate a task by constructing a language template ;
- Use Pre-trained Language Model (PLM) to produce NLg answers
- Map the NLg answers to the task lables (0, 1)

Example -- Sentiment Analysis

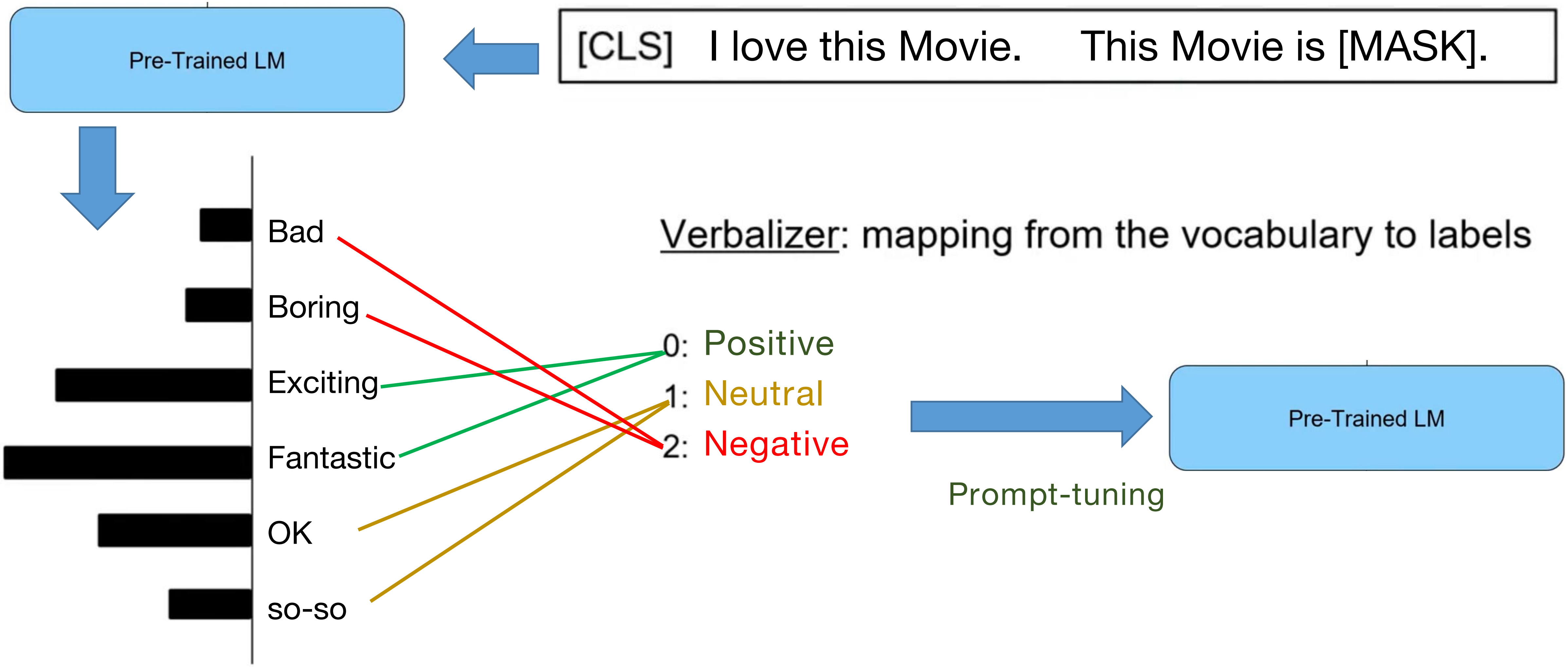
- Sentence in NLg: I love this Movie
- Is my attitude Positive or Negative or Neutral ?
- The class is a 3-way classification

0. Positive
1. Neutral
2. Negative

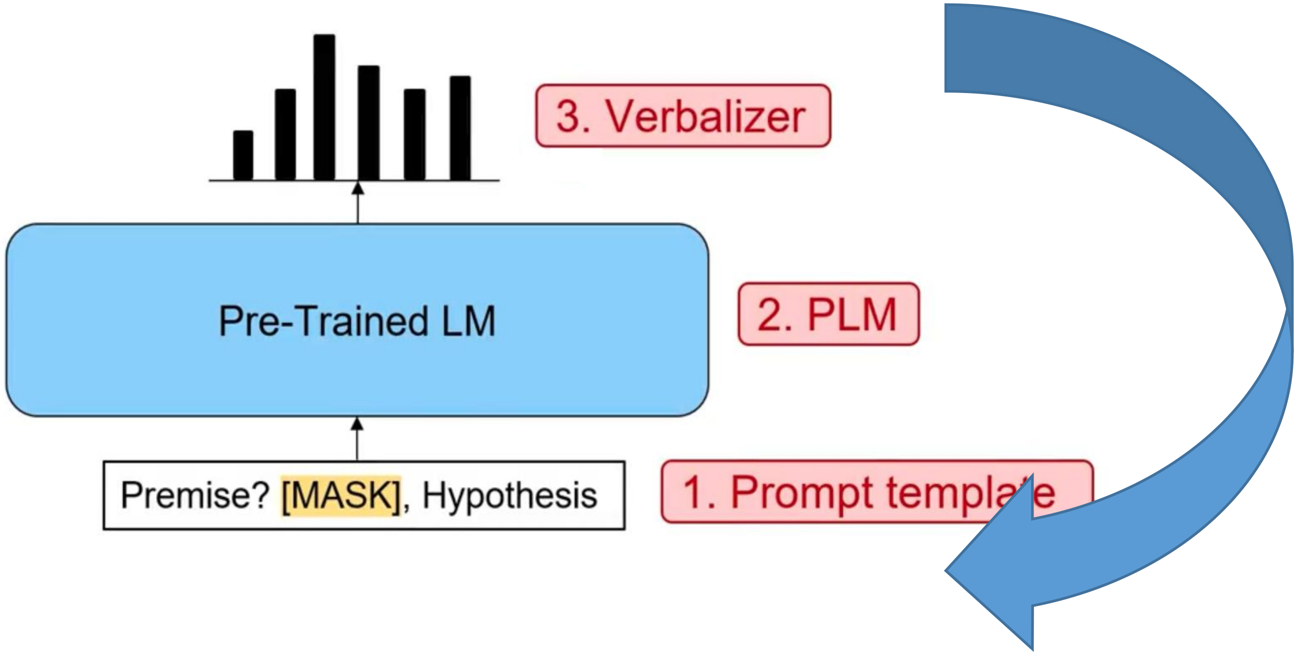
1. Prompt Template



2. PLM anwsers + 3. Verbalizer



Prompt-tuning with few-shot data



No reason to watch. It was [terrible].

You cannot miss it. I think it was [great].

This is best-seller. A [fun] one.

Prompt-tuning v.s. Fine-tuning

- PL is parameter efficient

(no additional layers to finetune)

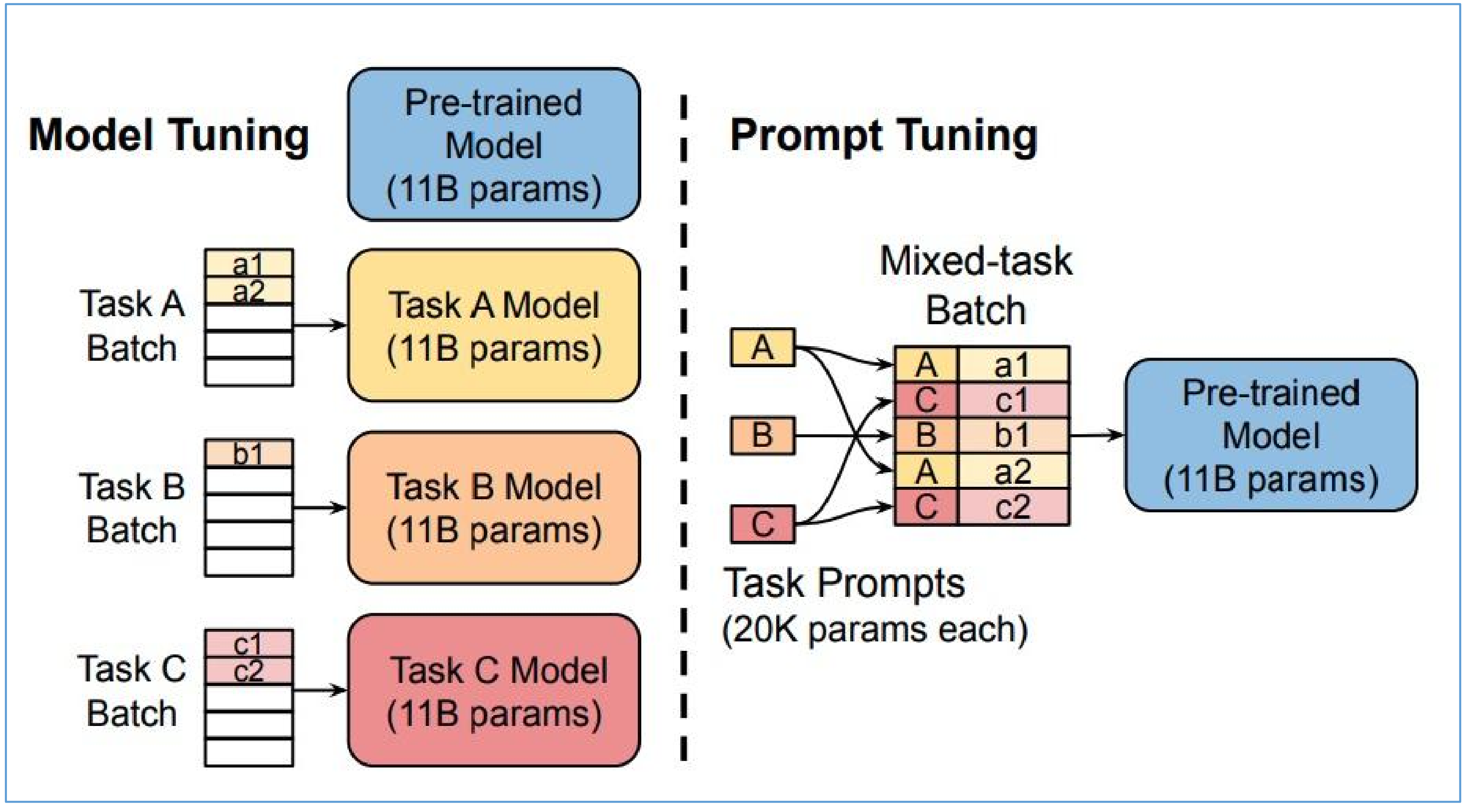
-- zero-shot learning

-- few-shot learning

- PL alleviates over-fitting

-- similar inputs (as templates)

-- won't affect pre-trained knowledge



The Power of Scale for Parameter-Efficient Prompt Tuning, 2021.

Thank you !

Thank you !   Q & A

- Which methodology to choose to use ?
- What are the interesting tasks ?

- https://lilianweng.github.io/posts/2018-11-30-meta-learning/
- https://mp.weixin.qq.com/s/iDXAdmheiJfZyCSdfQxodQ