Learning to Propagate Labels:

Transductive Propagation Network for Few-shot Learning (ICLR19)

2021.03.06 Yuho Jeong

Contents

Introduction

Few-shot Learning

- AI rely on large datasets for **generalization**
- It is challenging for domains with **scarce data**



novel task with
few examples

Introduction

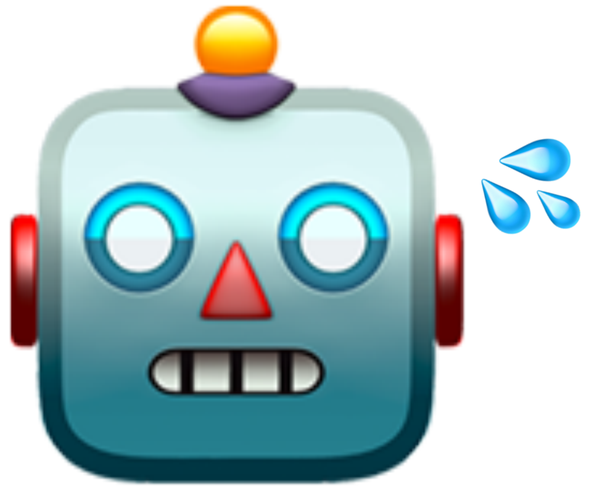
Few-shot Learning

- Machines fail to re-optimize models for novel task

Lesser panda

Cock

Persian cat

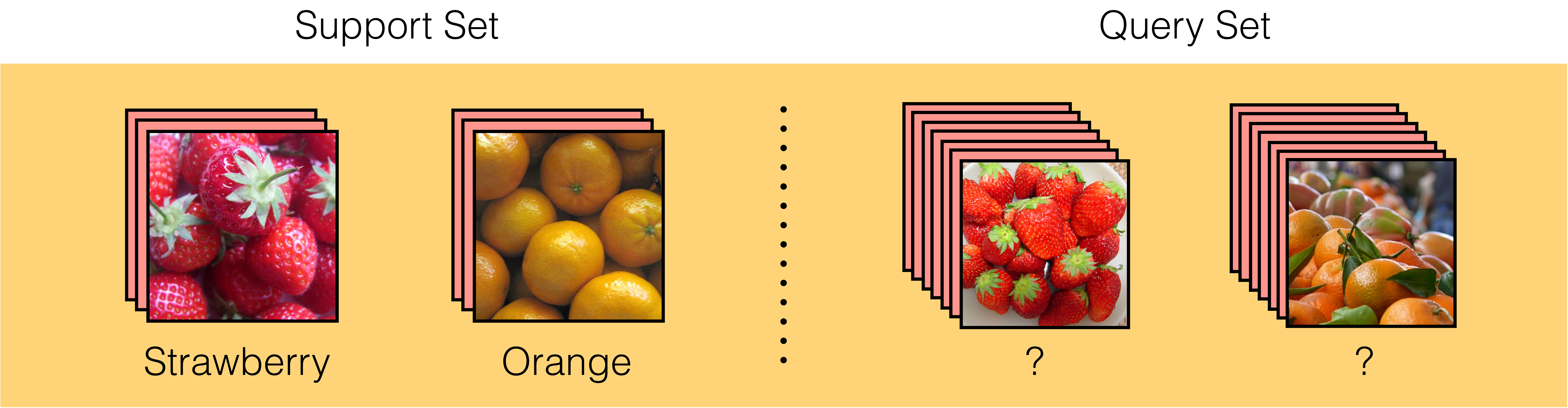


Give me more examples…

Introduction

Few-shot Learning

- **N-way K-shot** episodic learning



Episode with 2-way 3-shot

Introduction

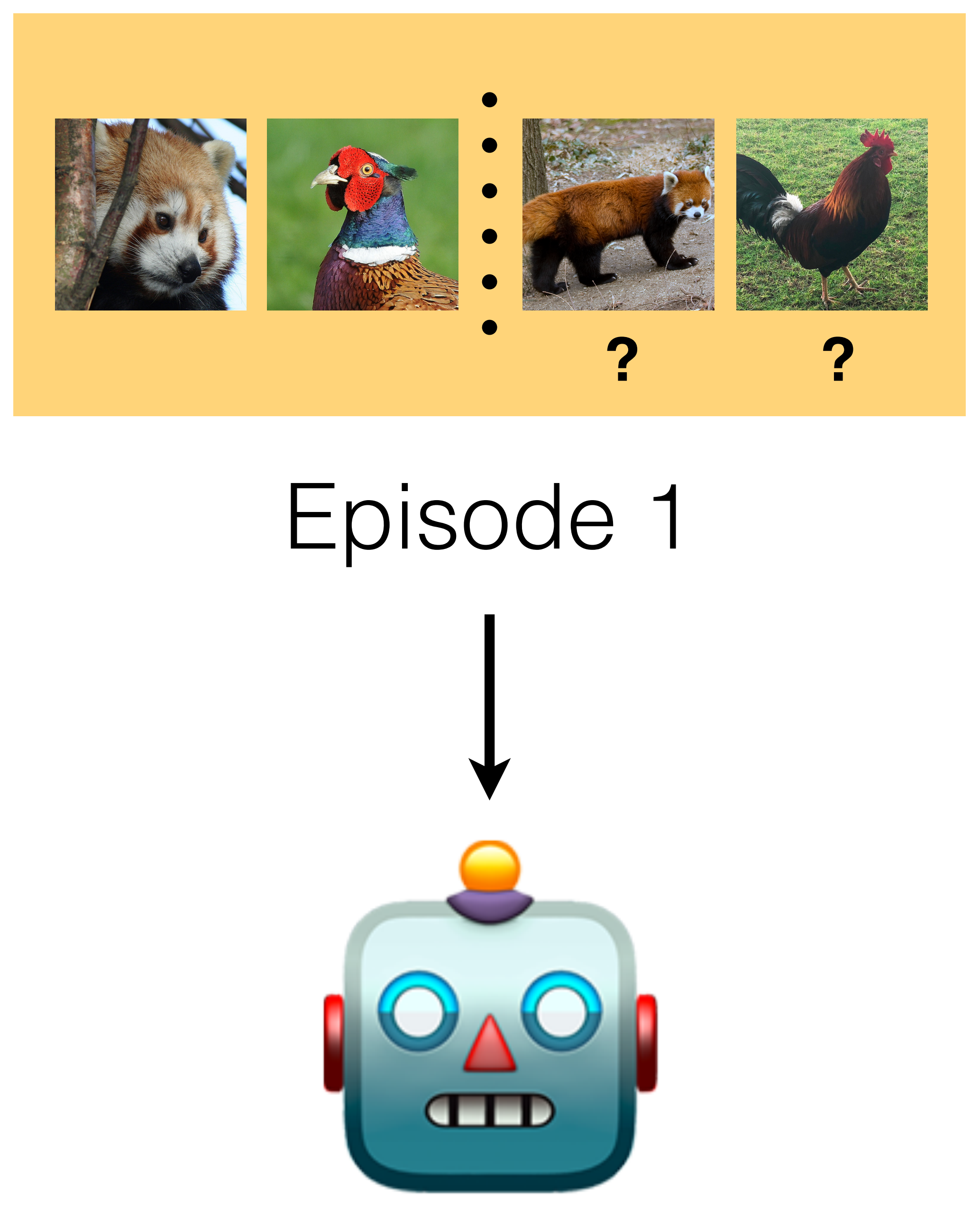
Few-shot Learning

- **N-way K-shot** episodic learning
  - **Support set is considered a clue** for query set
  - **Loss is calculated with query set** (CE in classification task)
  - There are generally more query examples than support examples(*shot*)
  - 5-way 5-shot, 5-way 1-shot setting in general

Introduction

Few-shot Learning

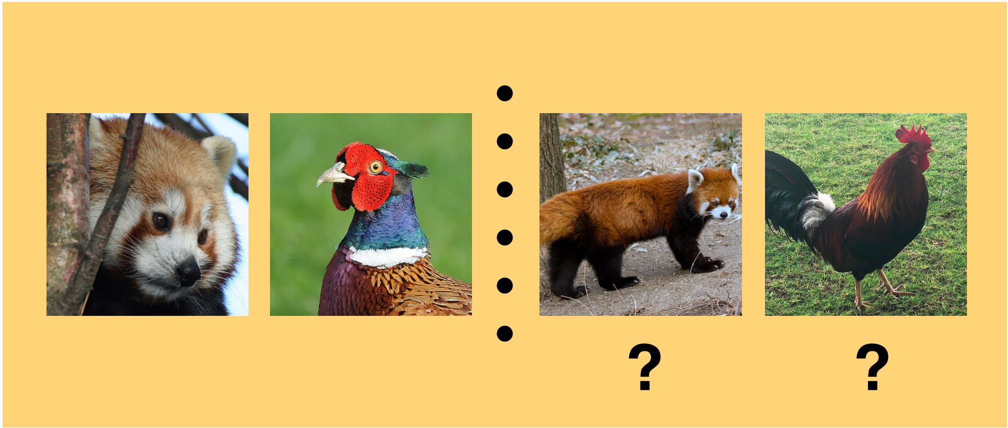
- **N-way K-shot** episodic learning



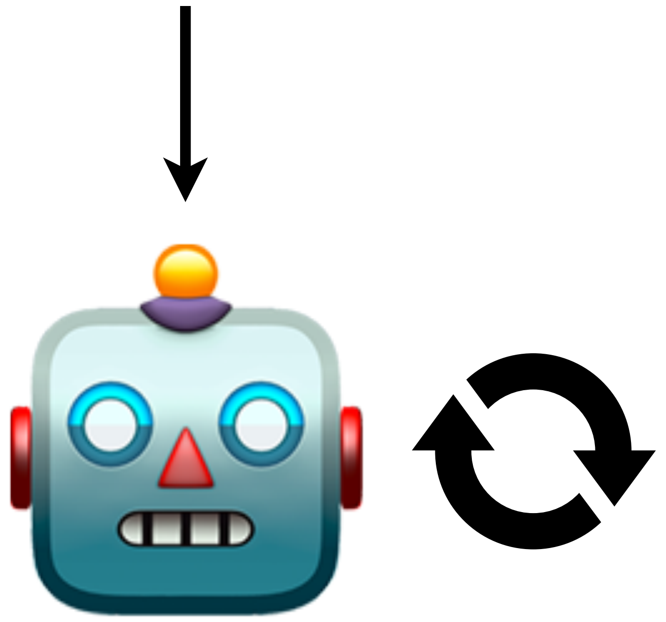
Introduction

Few-shot Learning

- **N-way K-shot** episodic learning



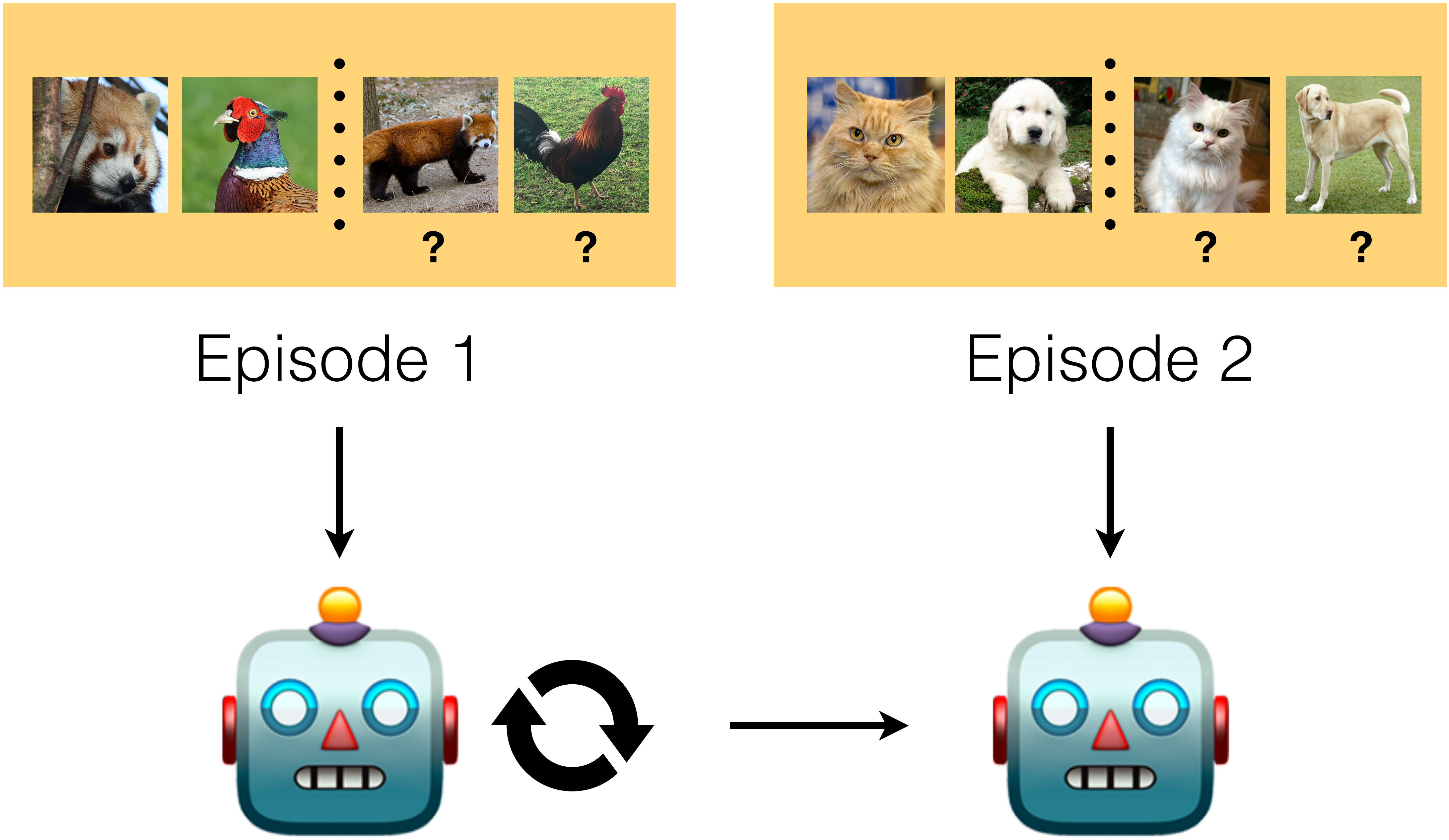
Episode 1



Introduction

Few-shot Learning

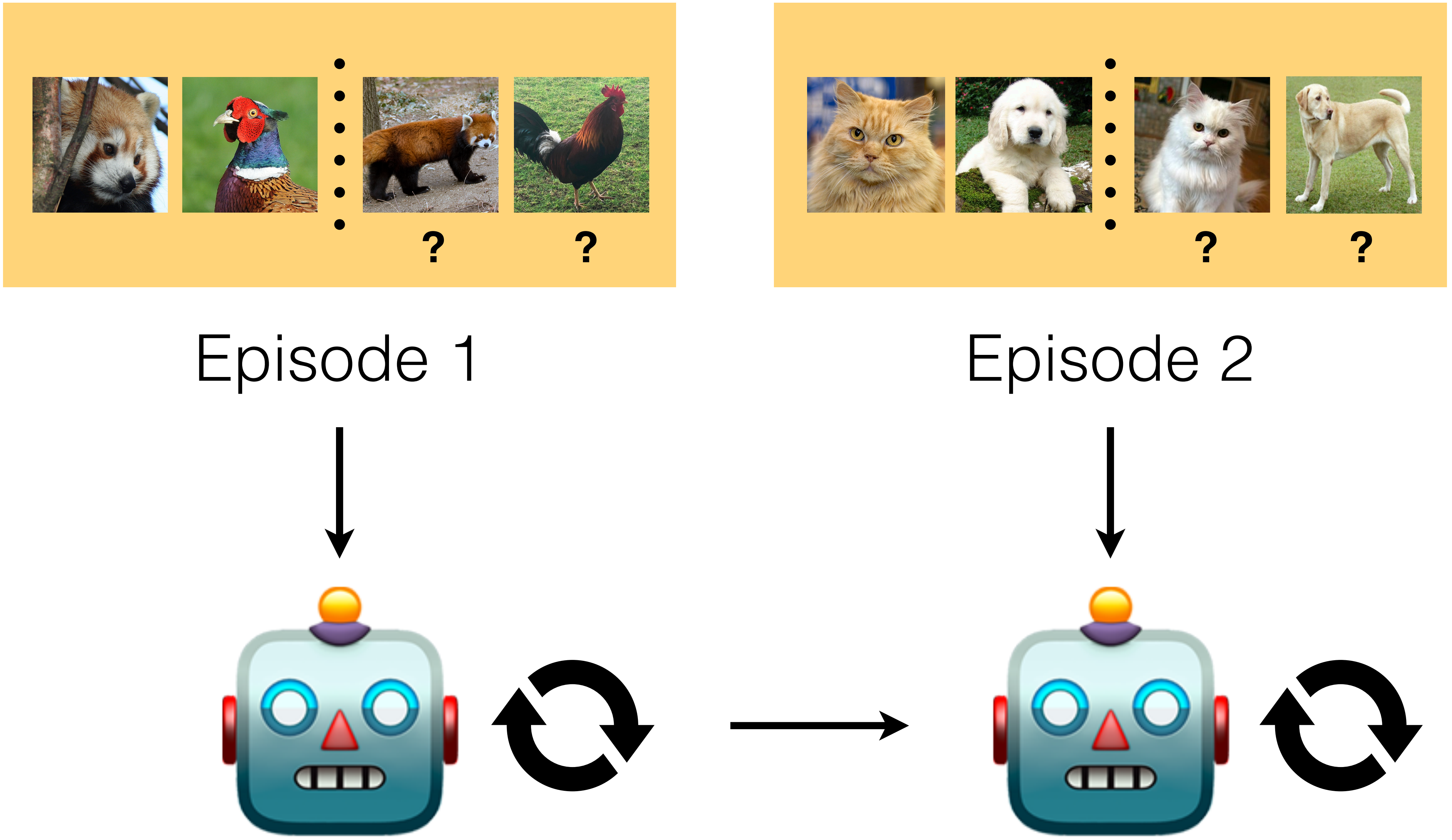
- **N-way K-shot** episodic learning



Introduction

Few-shot Learning

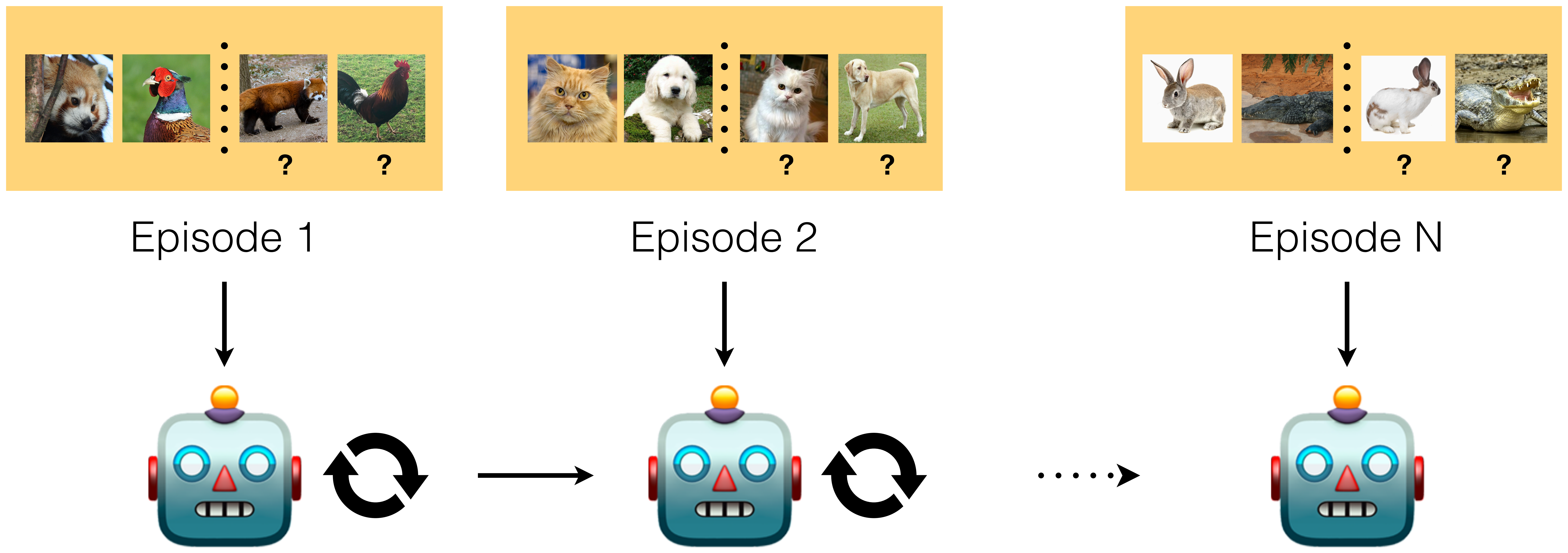
- **N-way K-shot** episodic learning



Introduction

Few-shot Learning

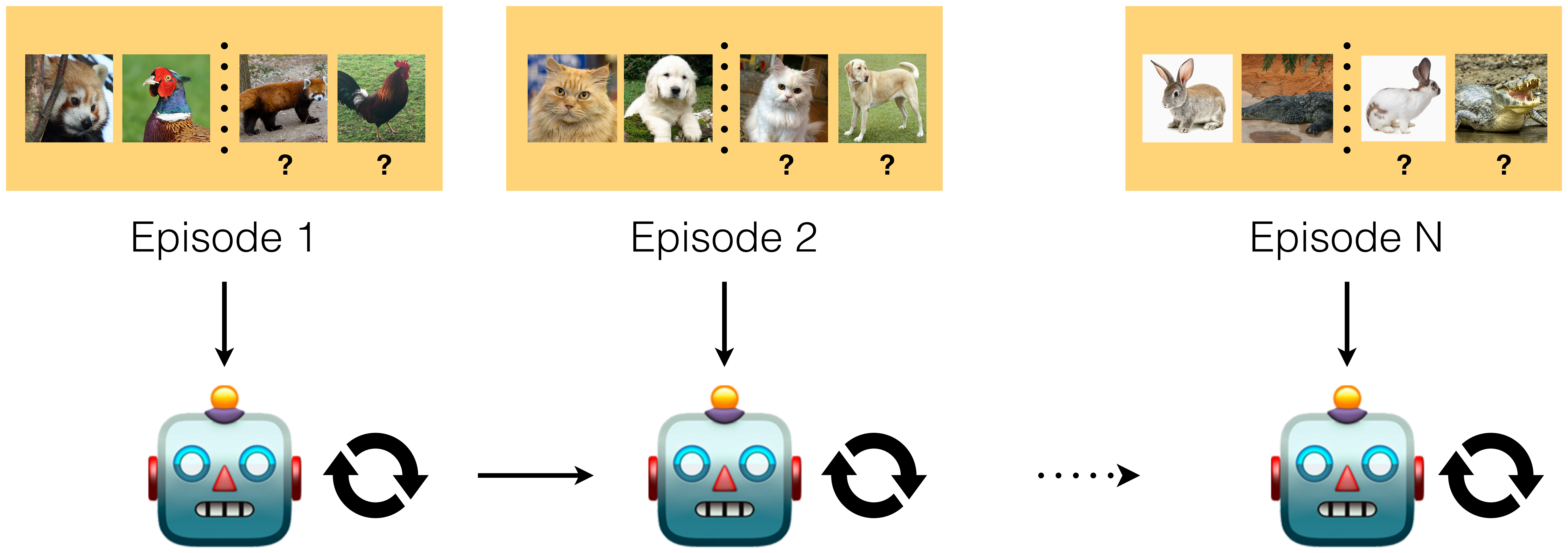
- **N-way K-shot** episodic learning



Introduction

Few-shot Learning

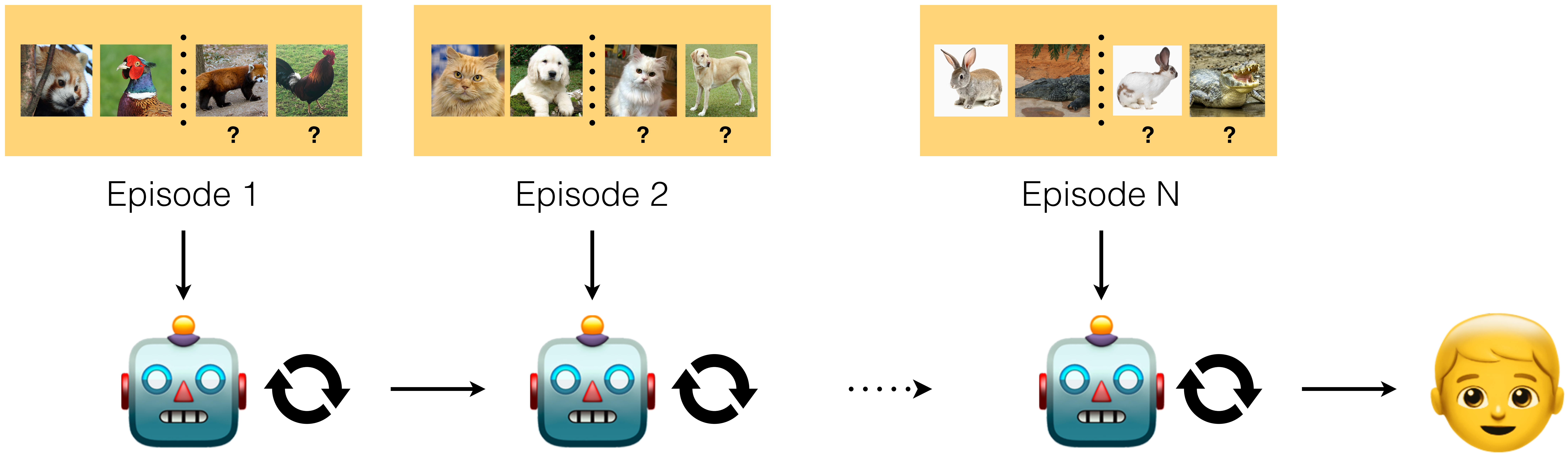
- **N-way K-shot** episodic learning



Introduction

Few-shot Learning

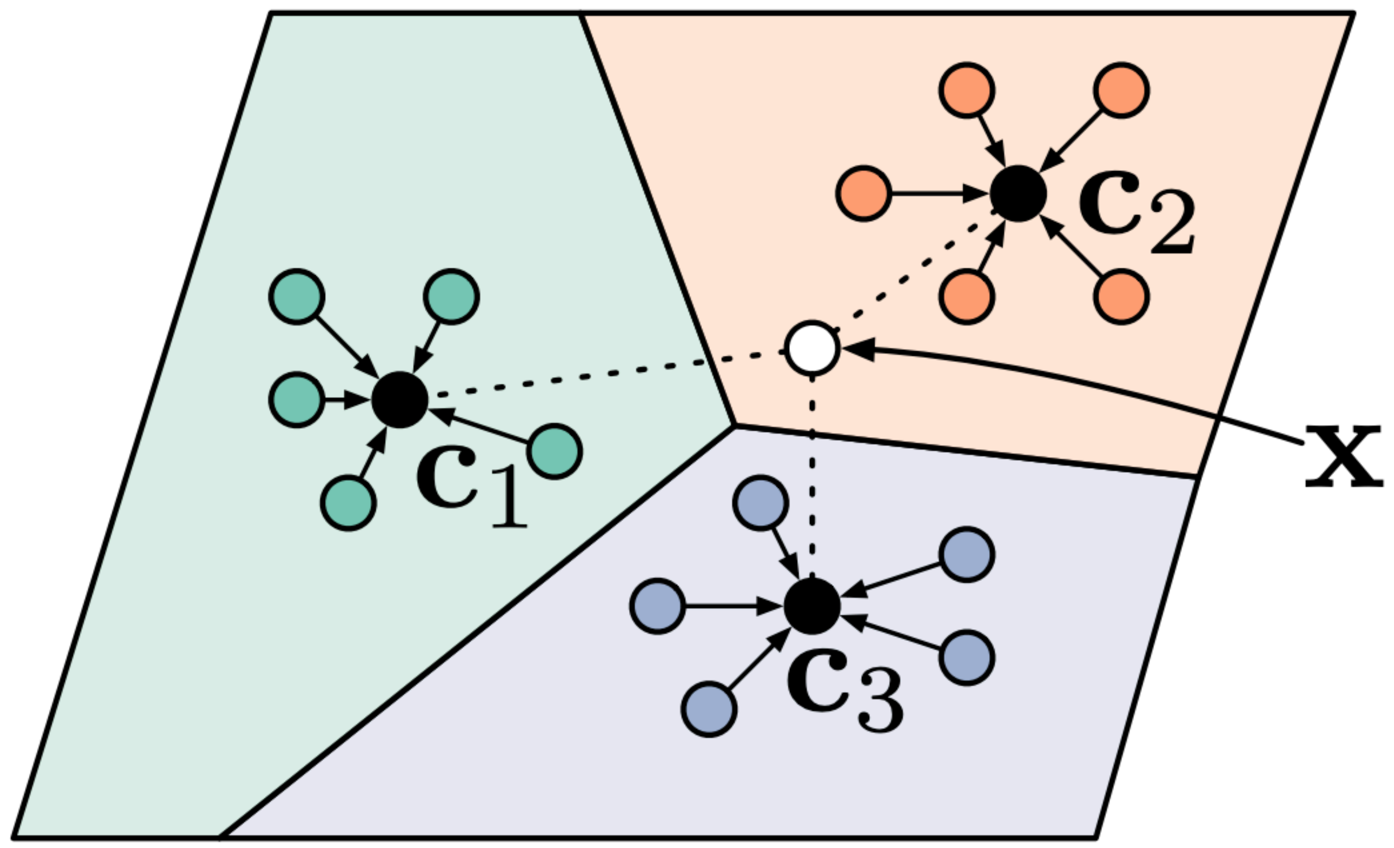
- **N-way K-shot** episodic learning



Introduction

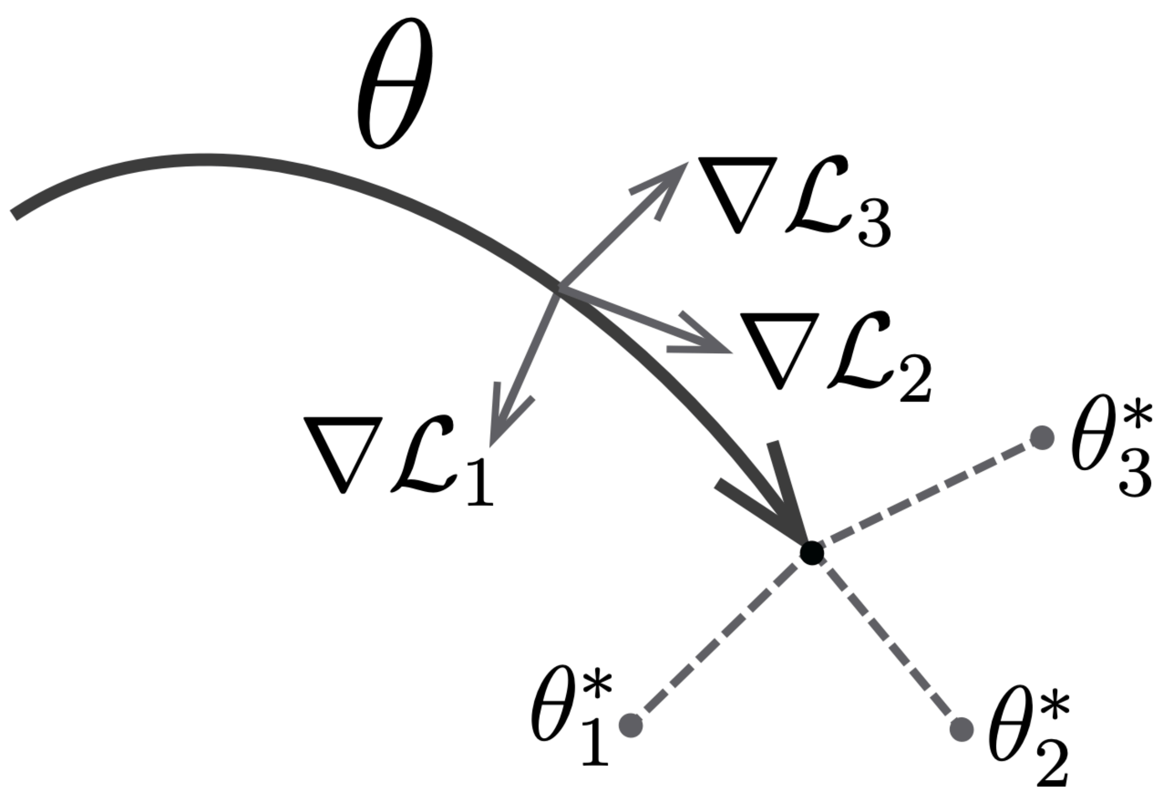
Prior works

Metric-based



Taken from [Snell, 2017]

Optimization-based



Taken from [Finn, 2017]

[Snell, 2017] Snell et al. "Prototypical networks for few-shot learning," NIPS 2017.
[Finn, 2017] C Finn et al. "Model-agnostic meta-learning for fast adaptation of deep networks," ICML 2017.

Introduction

Prior works

- Metric-based model

Support Set Query Set

Episode with 2-way 3-shot

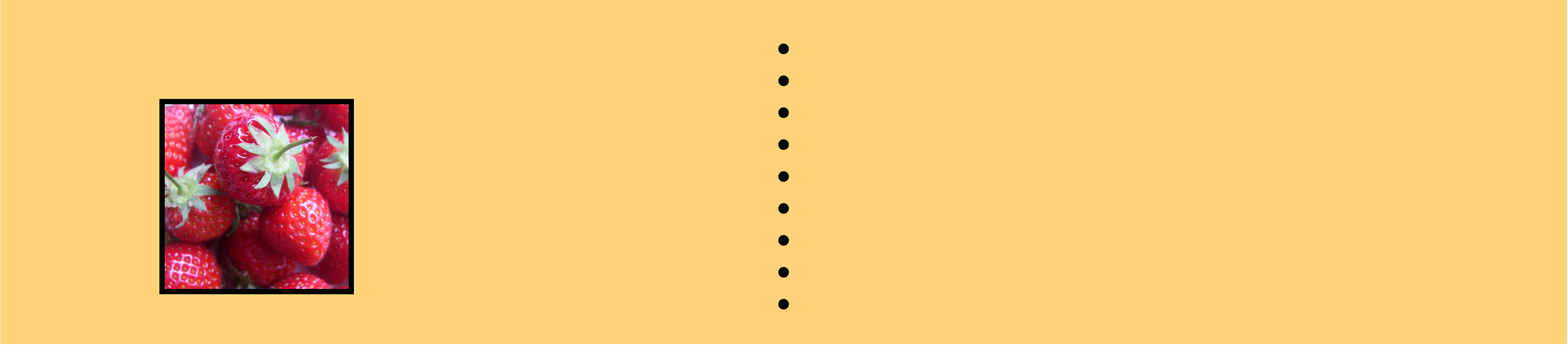
Embedding space

Introduction

Prior works

- Metric-based model

Support Set Query Set



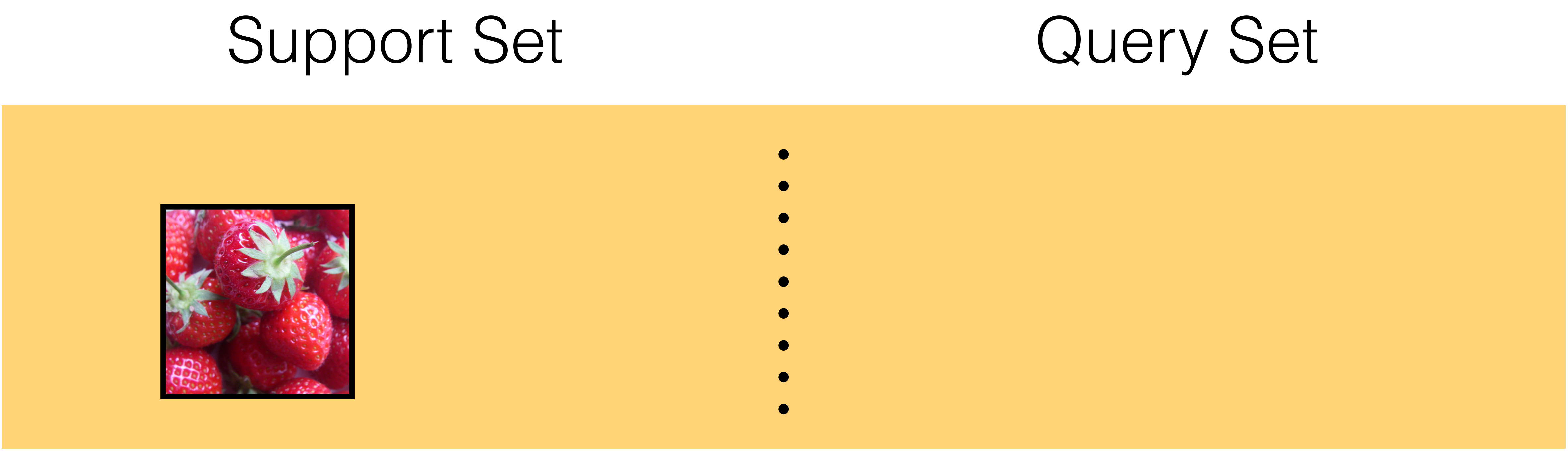
Episode with 2-way 3-shot

Embedding space

Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

Embedding space

Introduction

Prior works

- Metric-based model

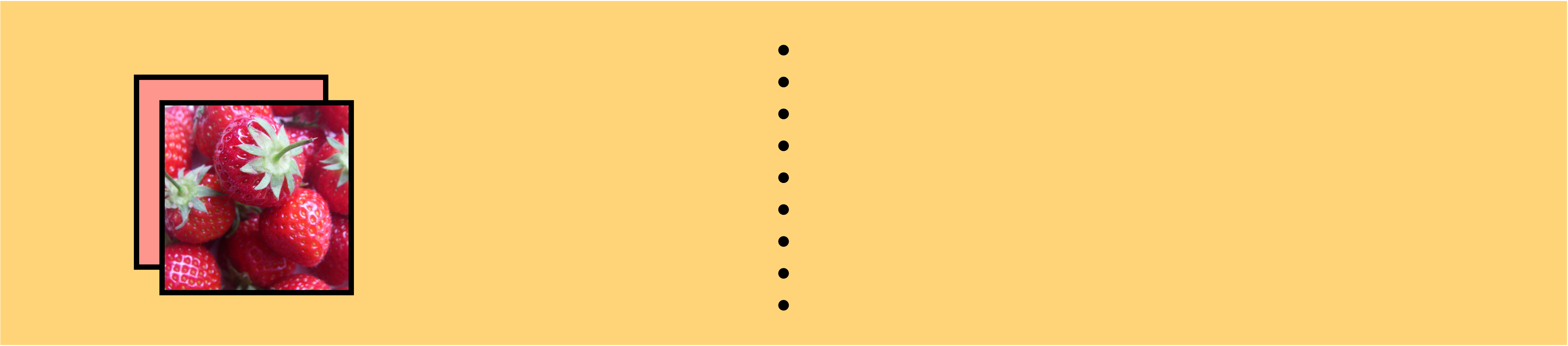
Episode with 2-way 3-shot

Embedding space

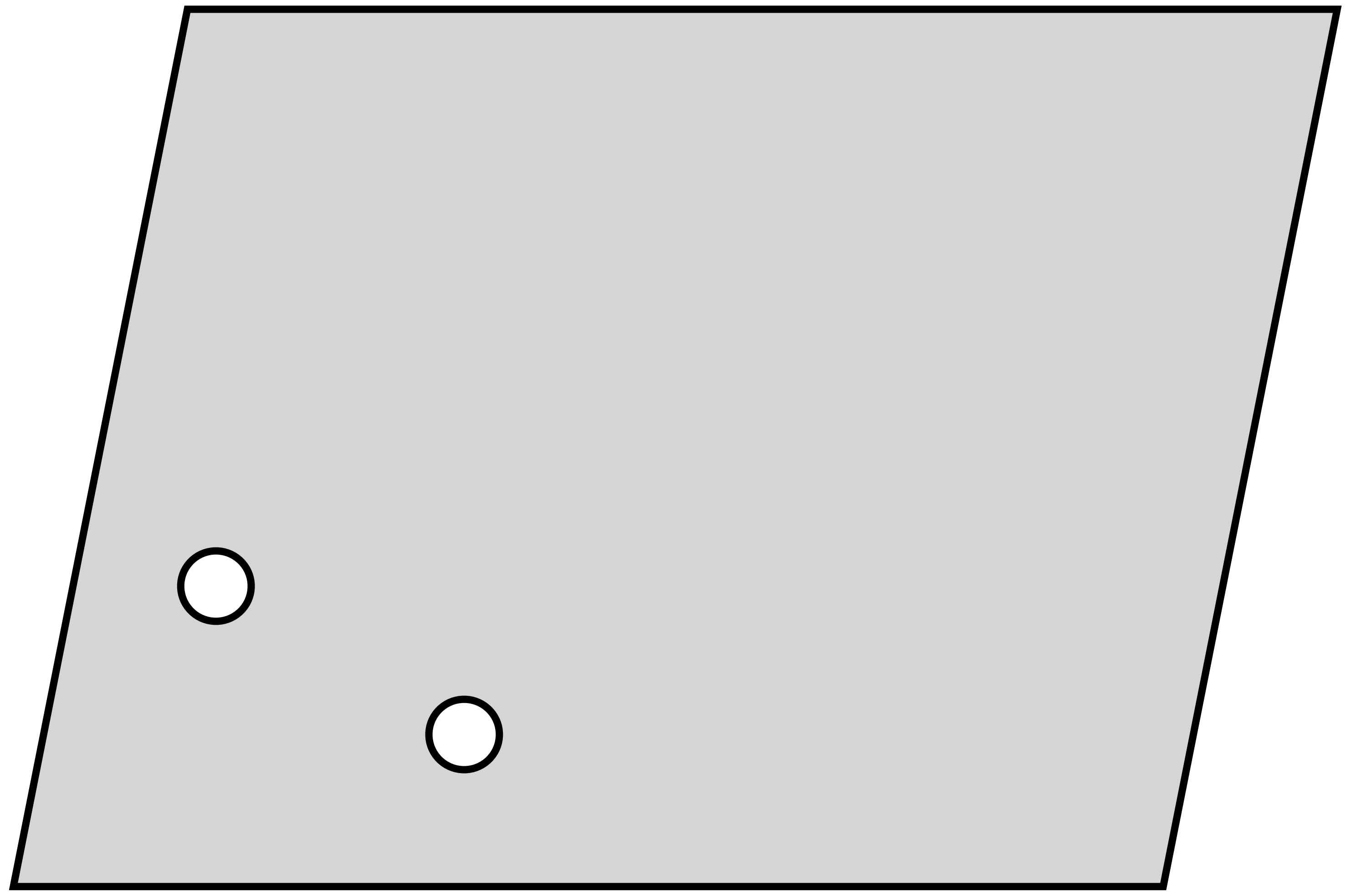
Introduction
Prior works

- Metric-based model

Support Set Query Set



Episode with 2-way 3-shot



Embedding space

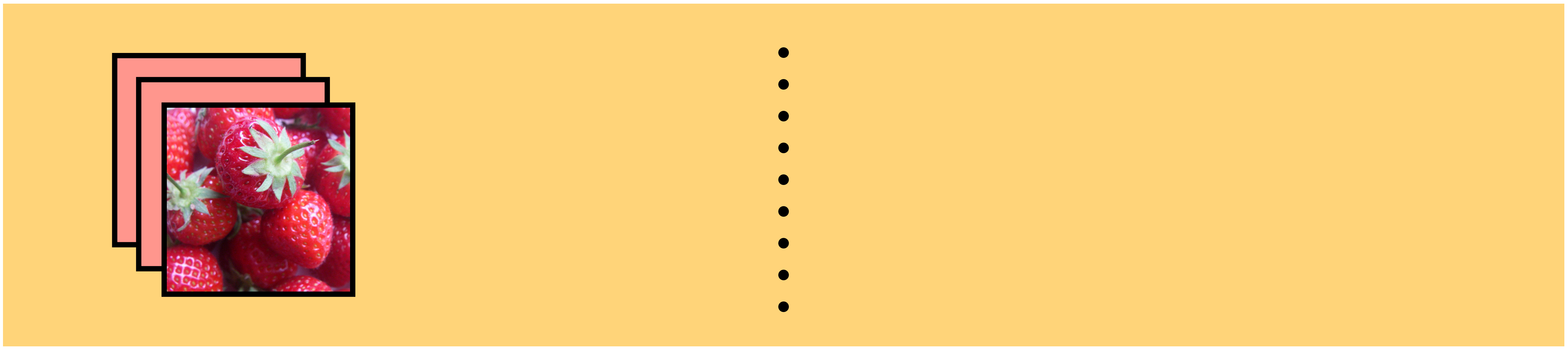
Introduction

Prior works

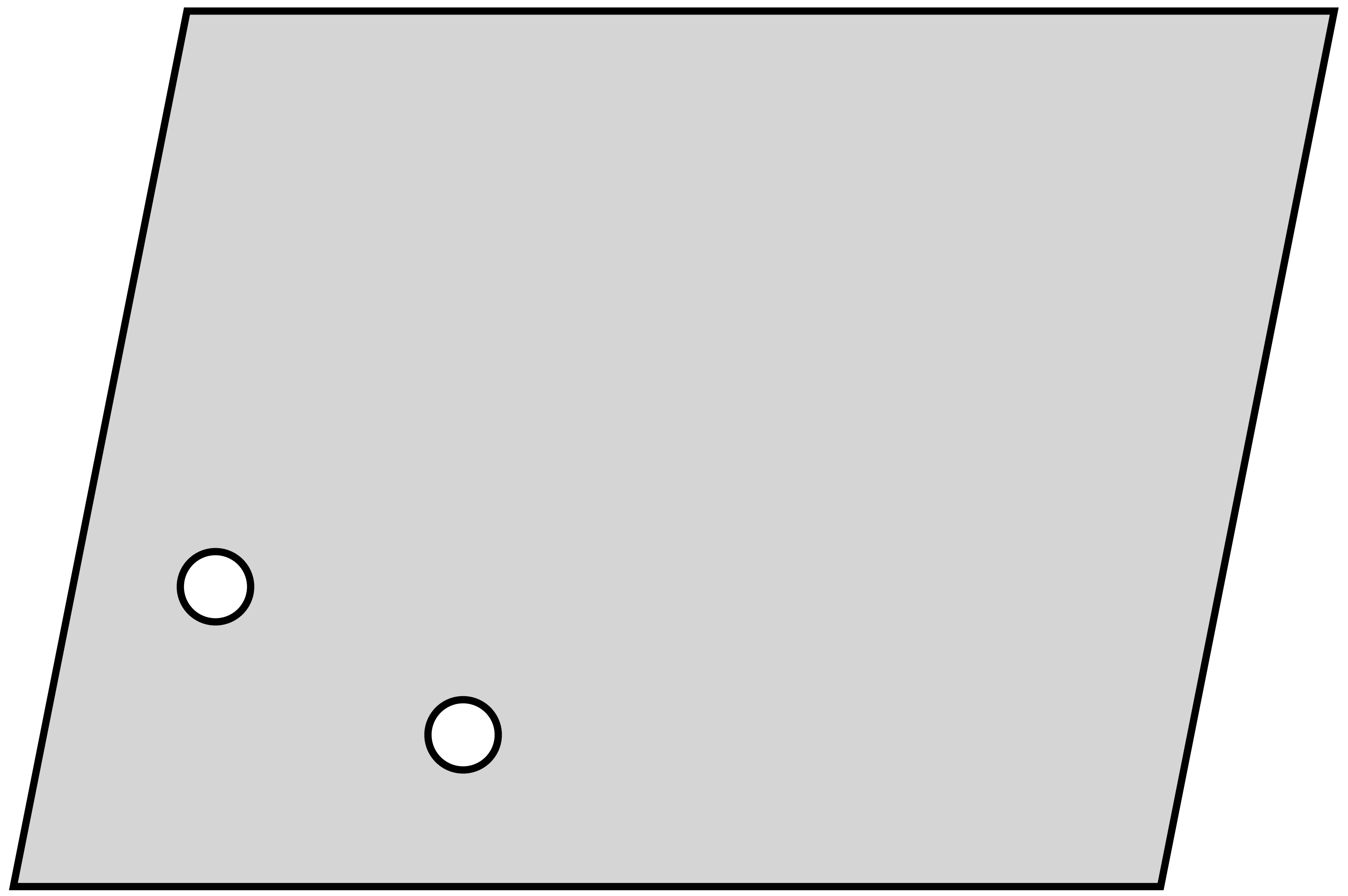
- Metric-based model

Support Set

Query Set



Episode with 2-way 3-shot

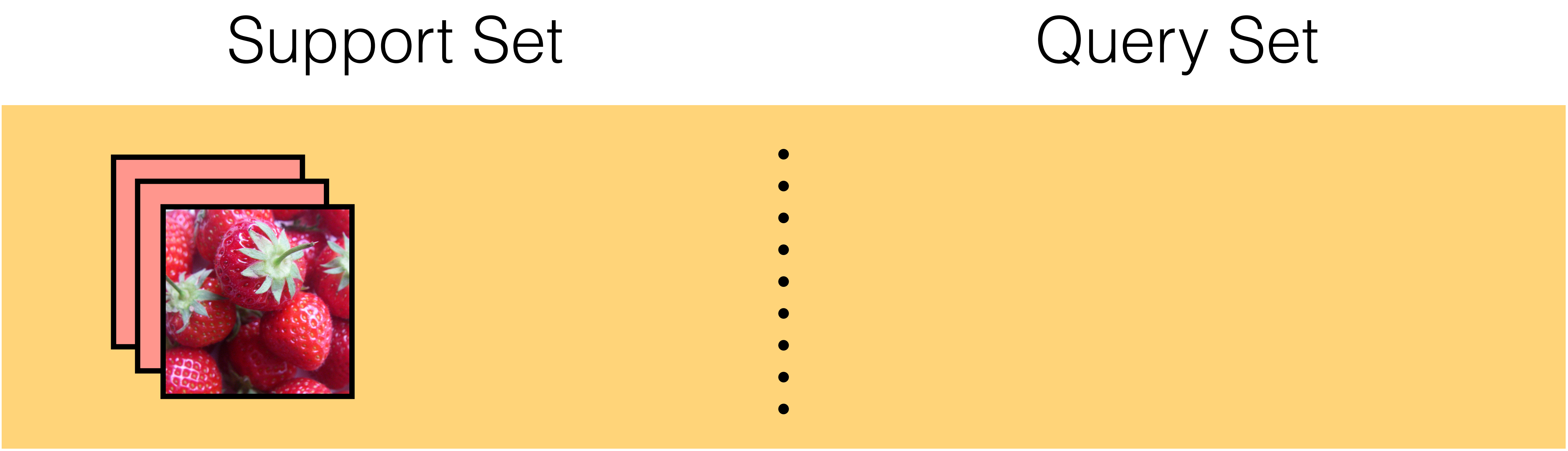


Embedding space

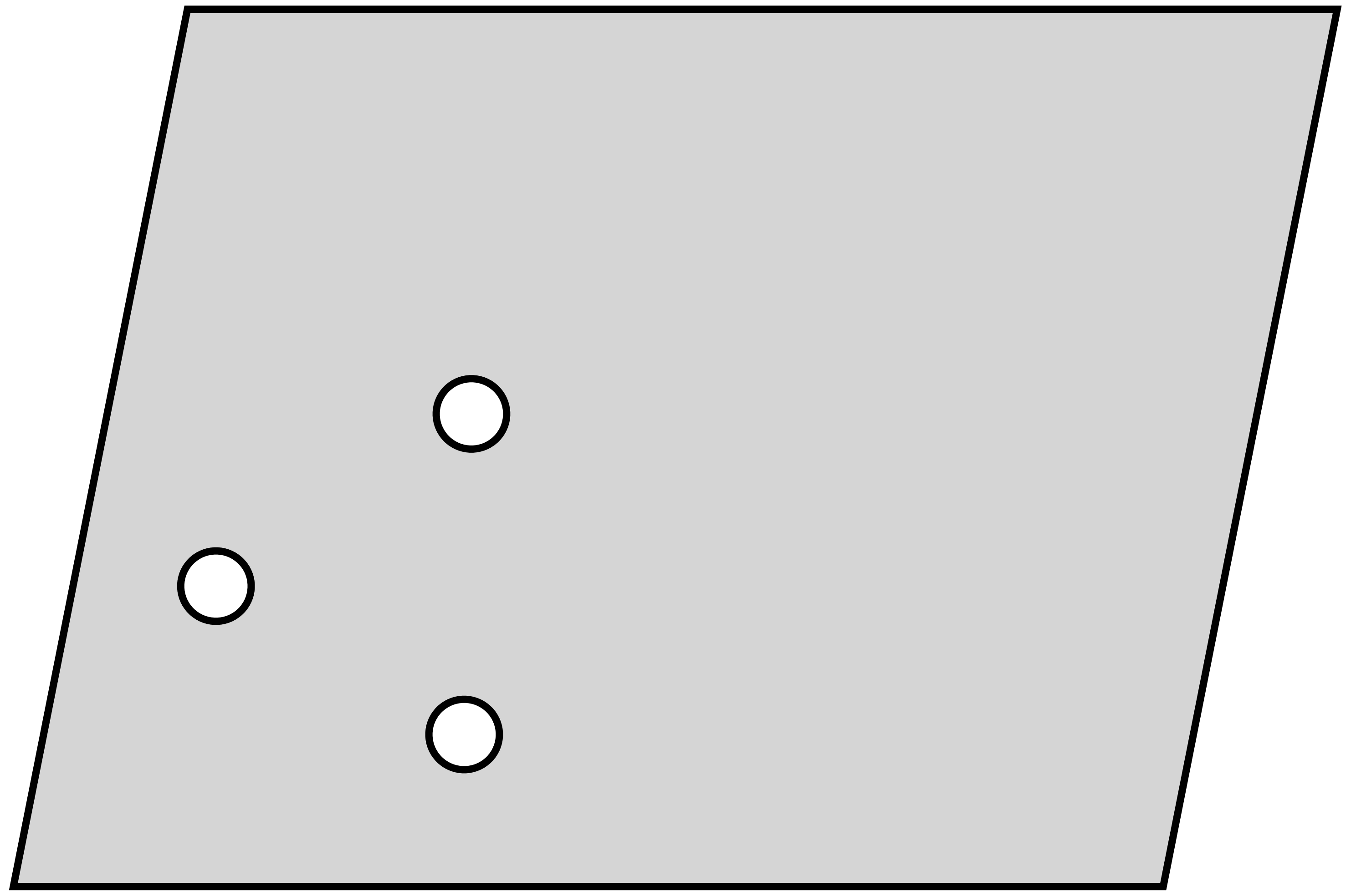
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

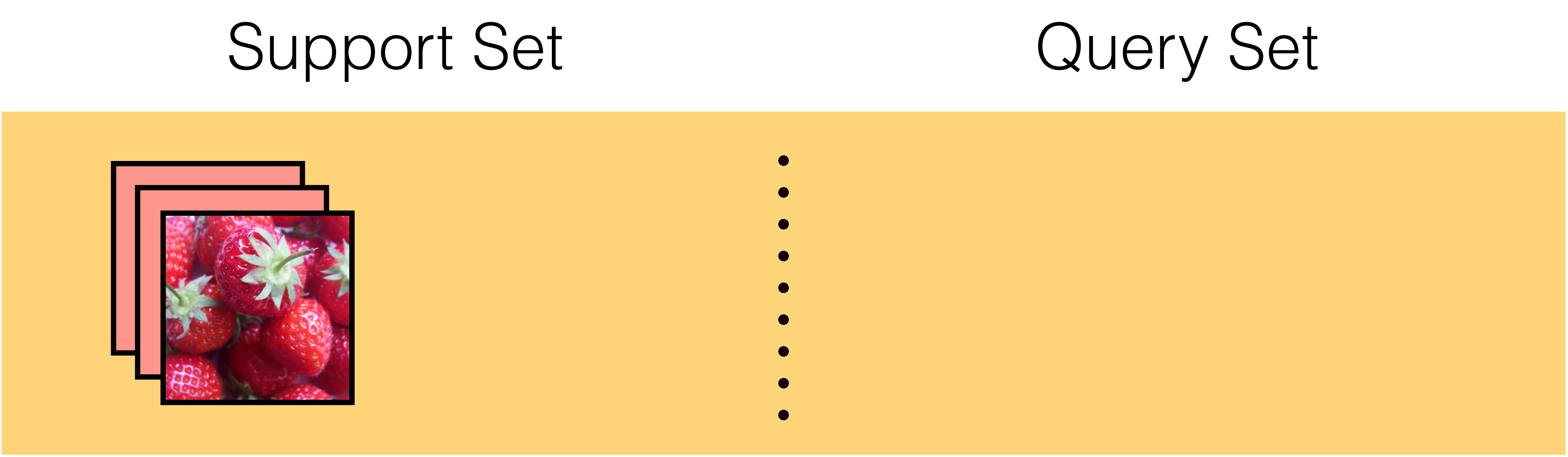


Embedding space

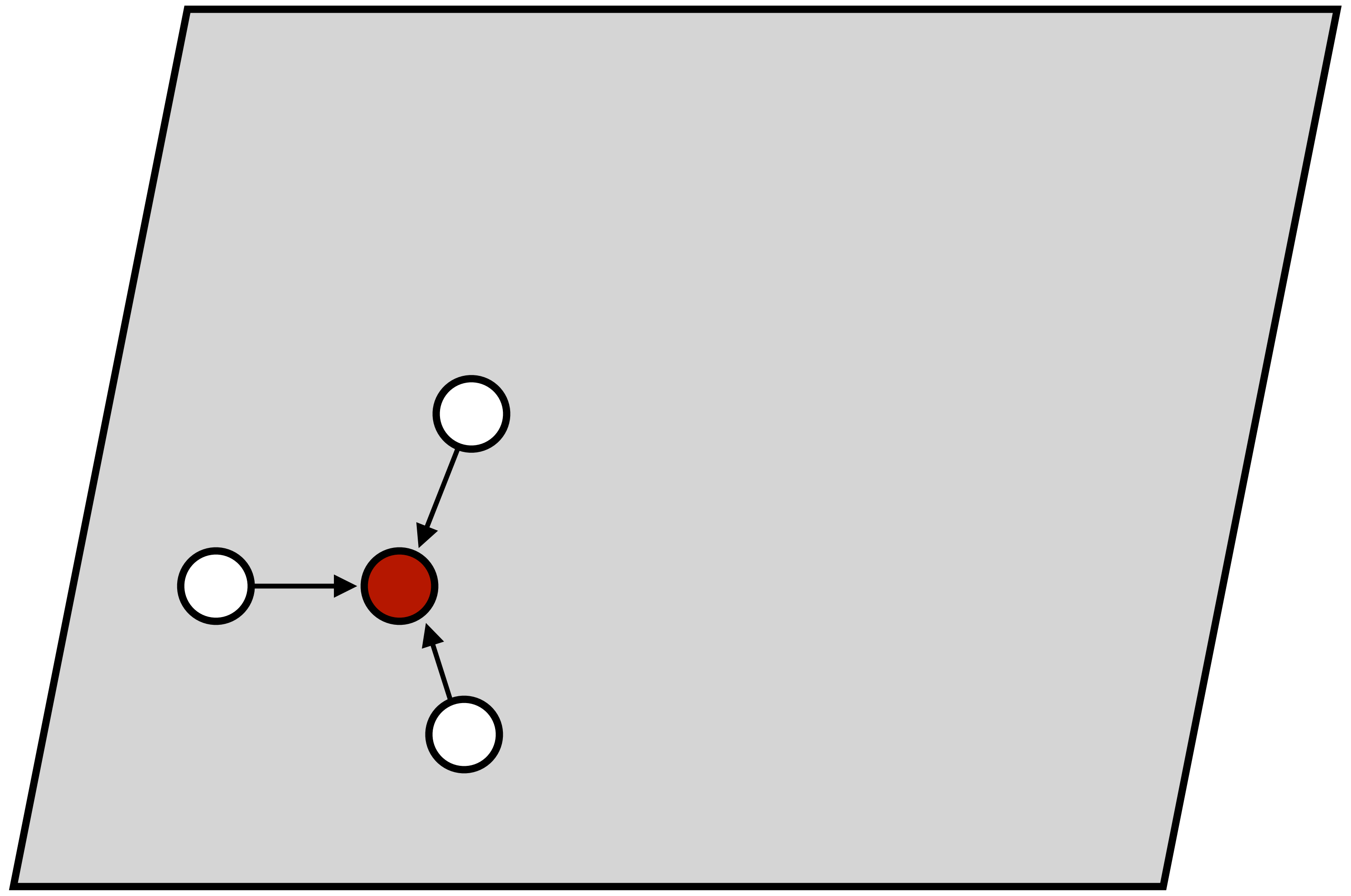
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

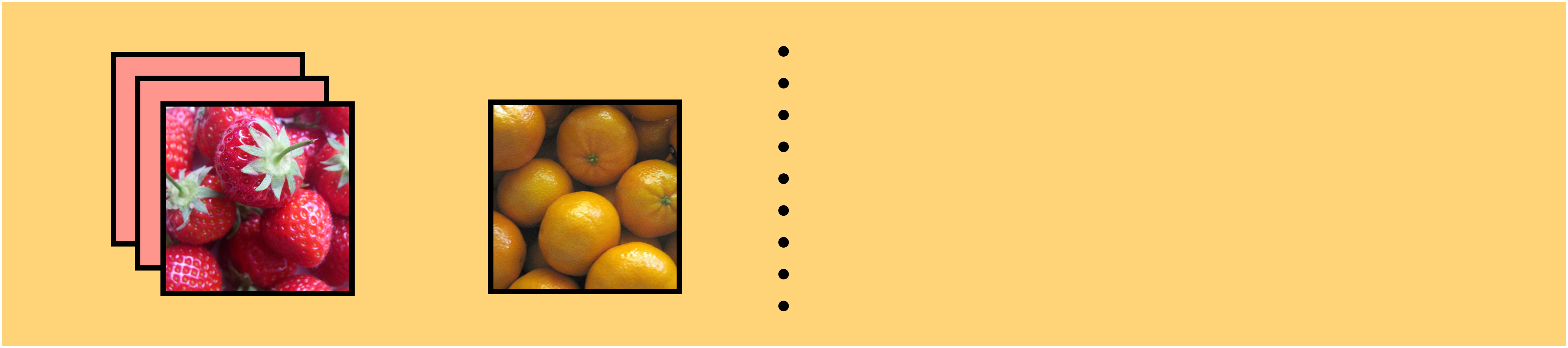


Embedding space

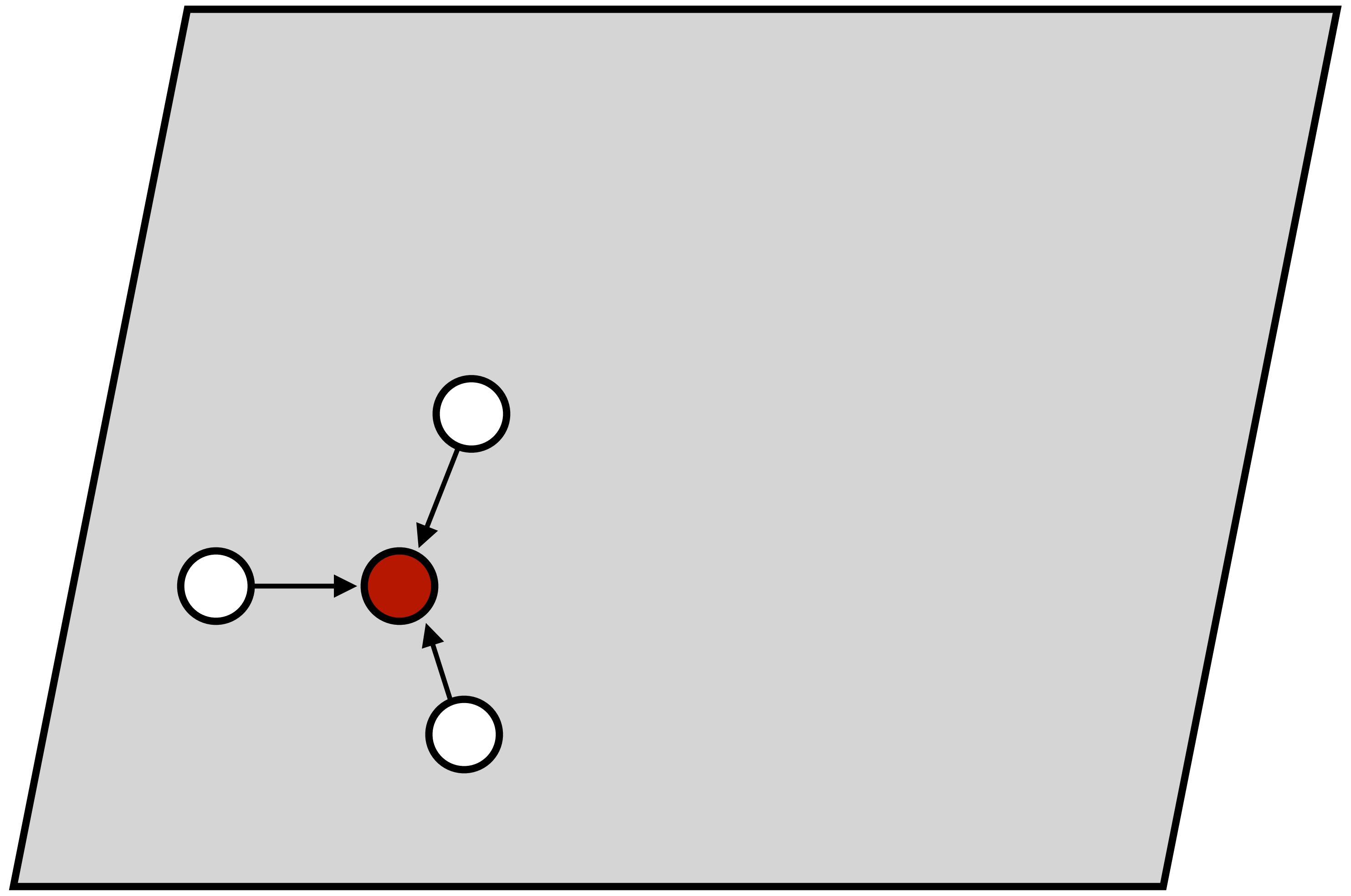
Introduction
Prior works

- Metric-based model

Support Set Query Set



Episode with 2-way 3-shot



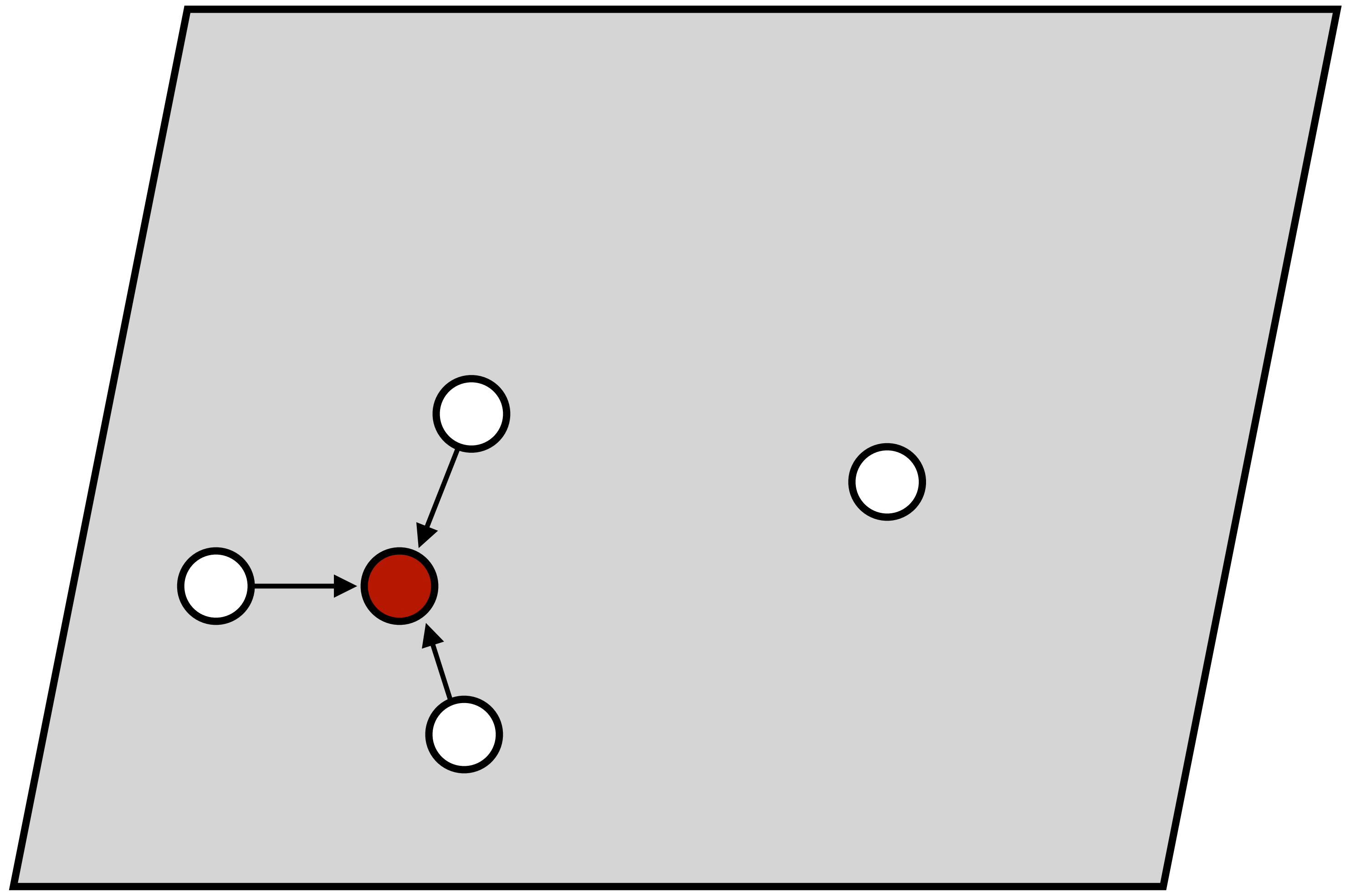
Embedding space

Introduction

Prior works

- Metric-based model

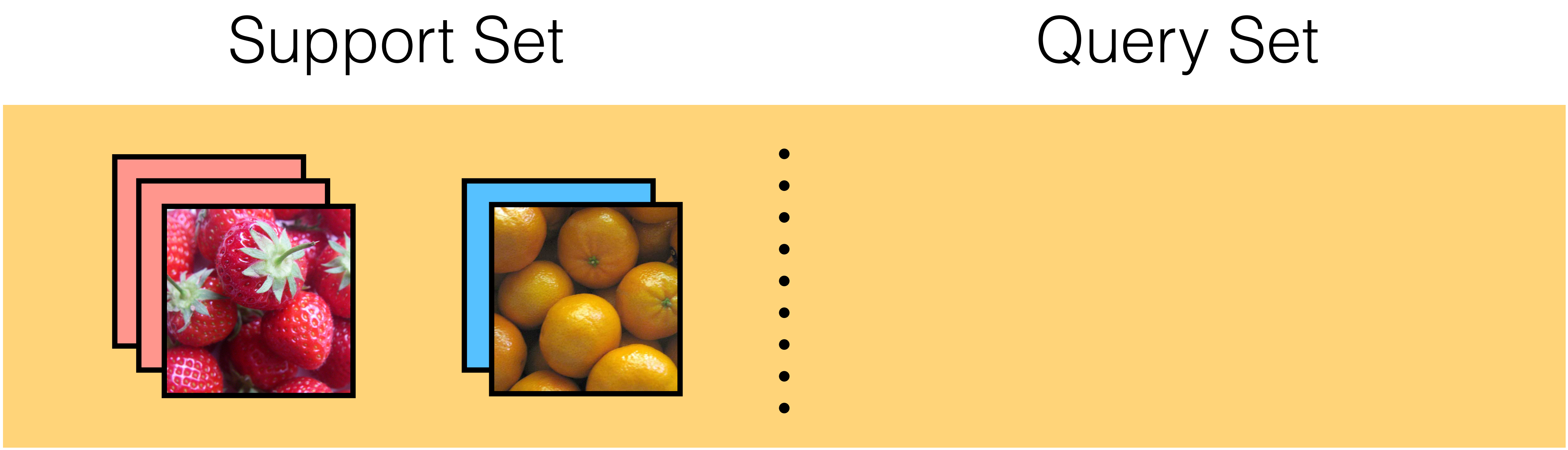
Episode with 2-way 3-shot



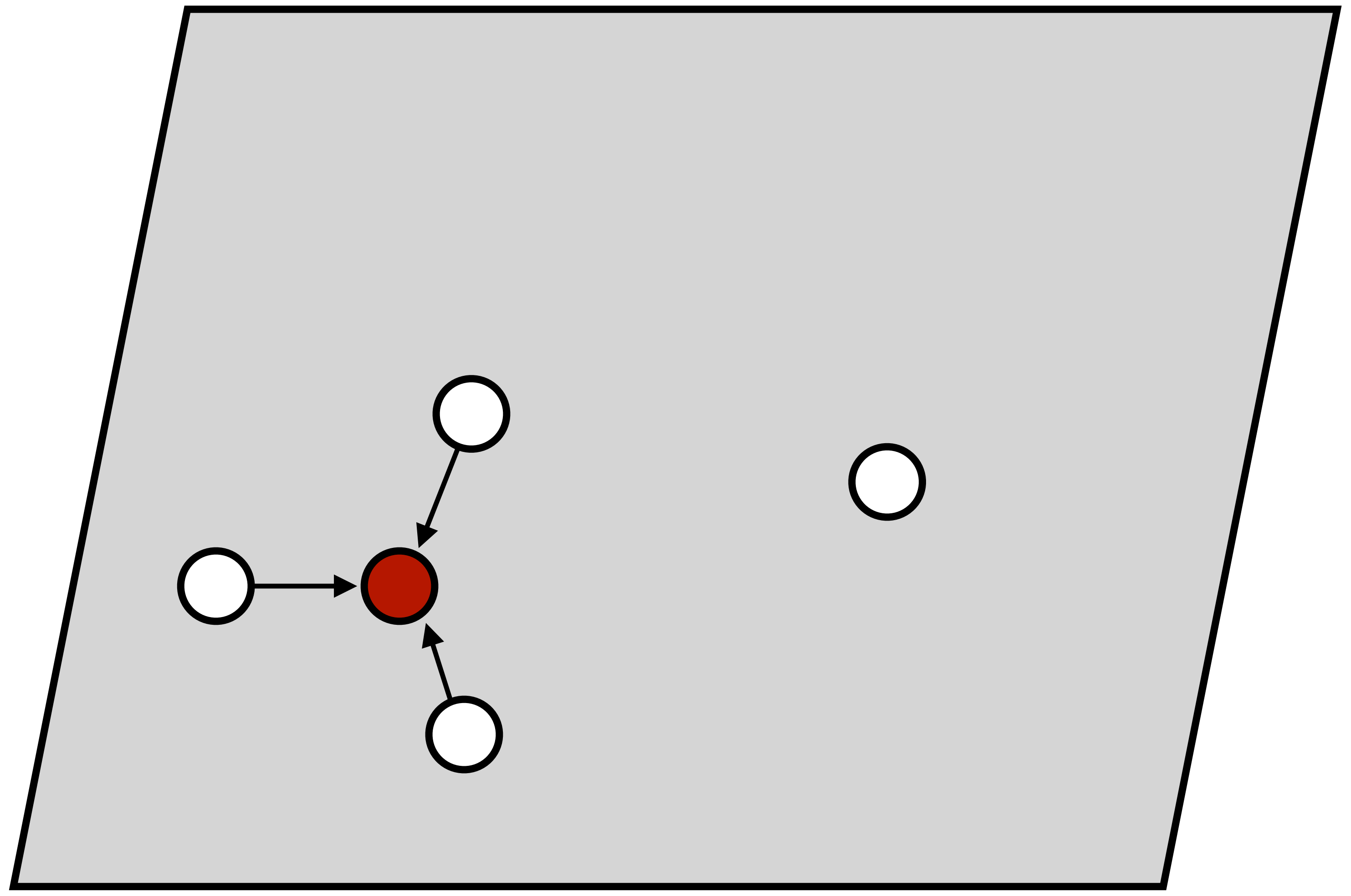
Embedding space

Introduction
Prior works

- Metric-based model



Episode with 2-way 3-shot



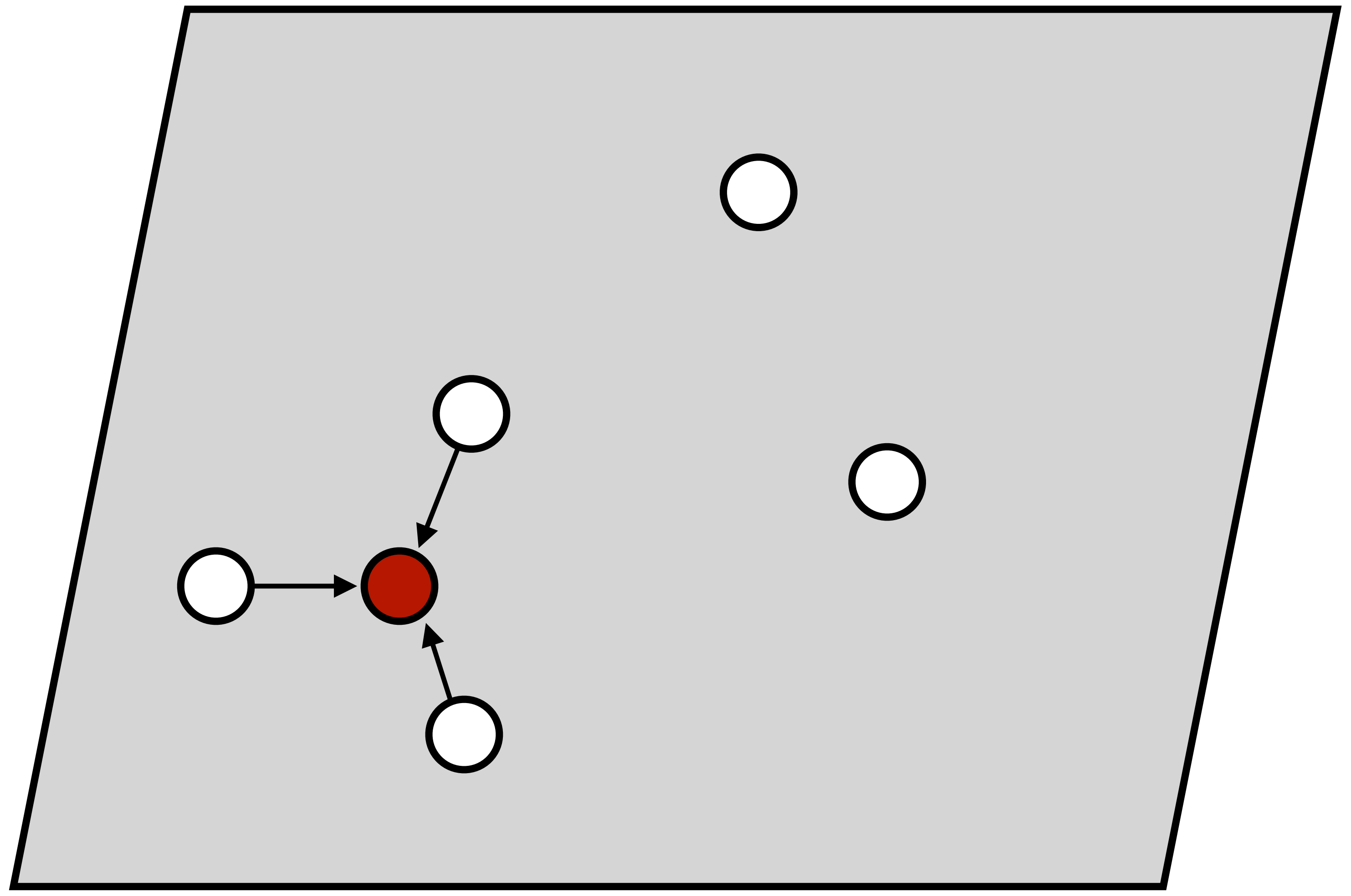
Embedding space

Introduction

Prior works

- Metric-based model

Episode with 2-way 3-shot

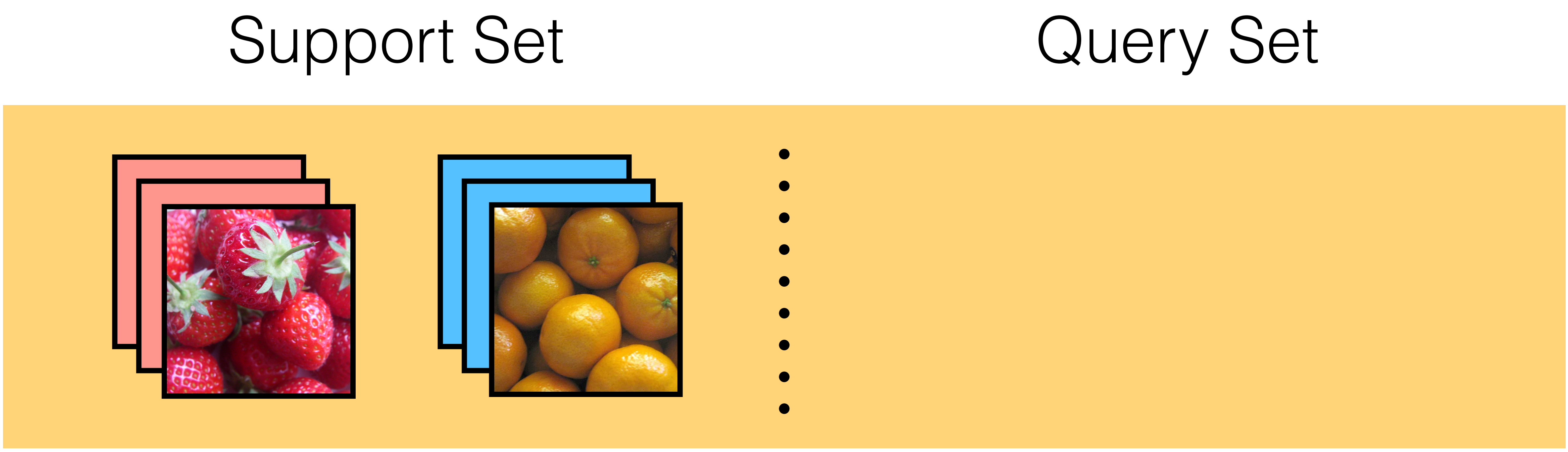


Embedding space

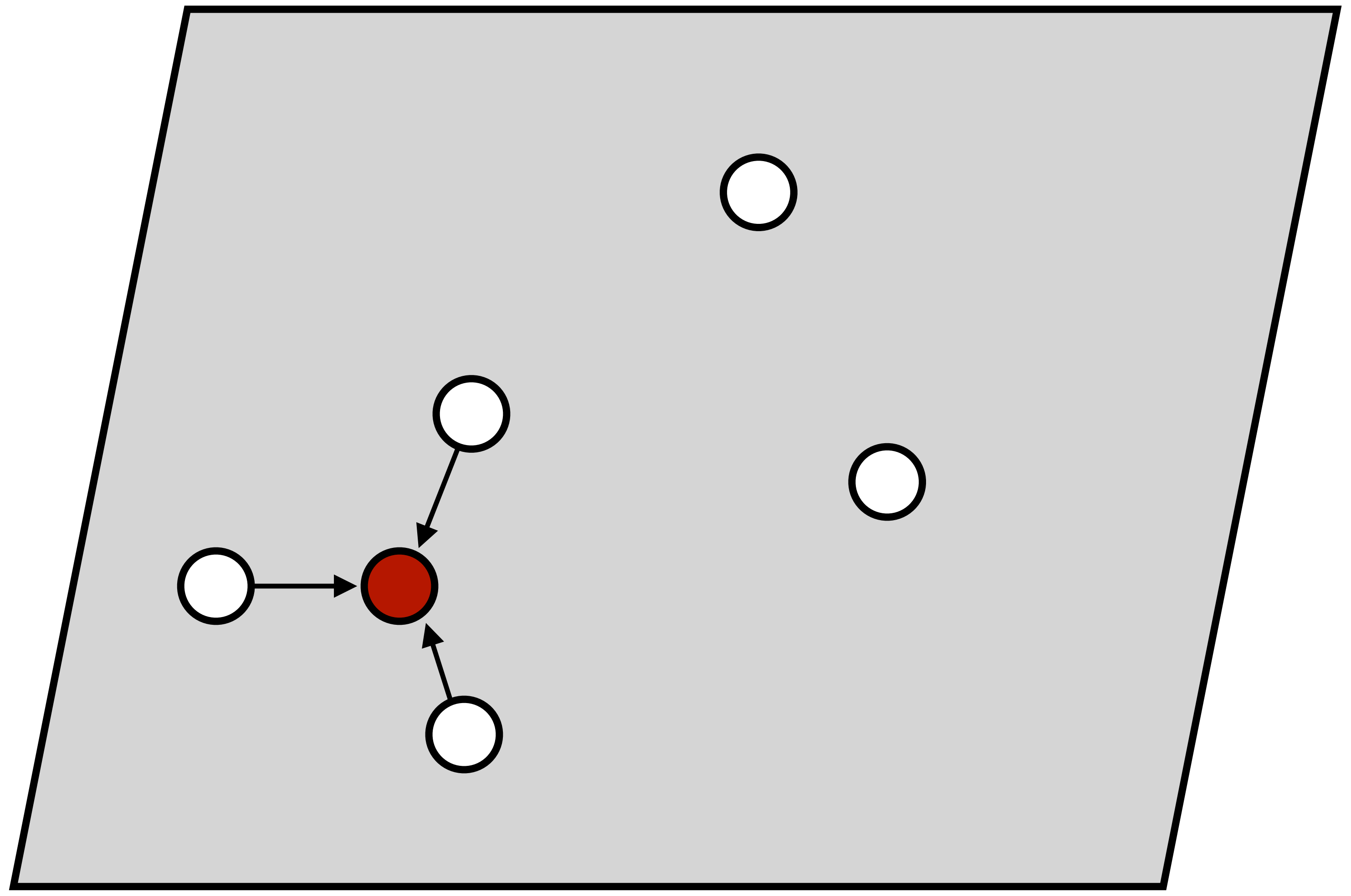
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

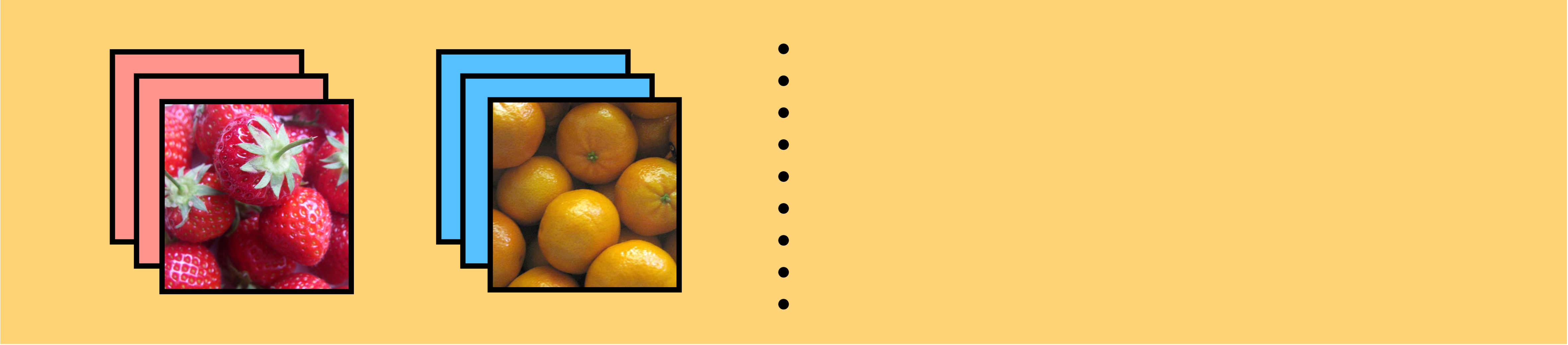


Embedding space

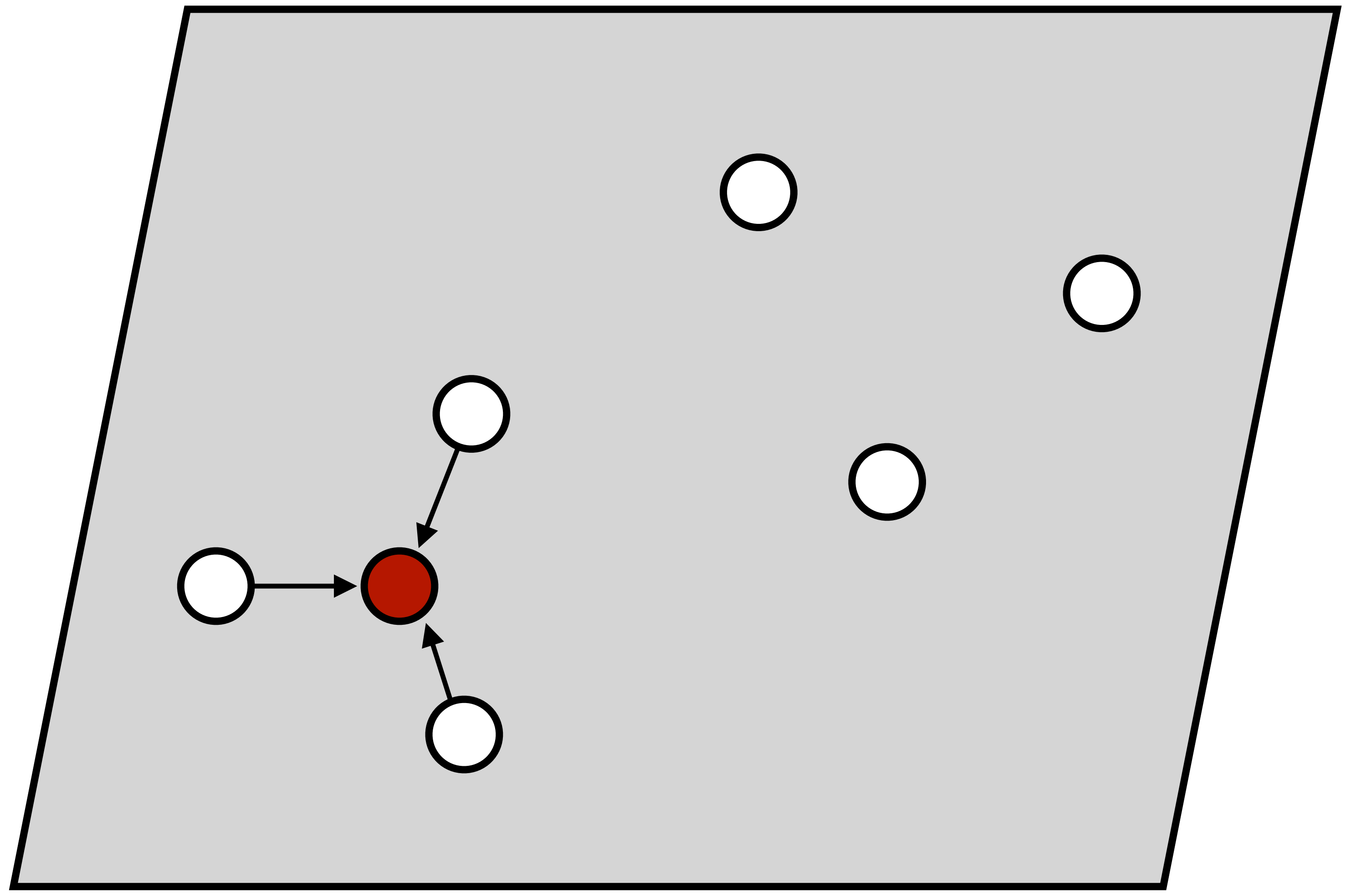
Introduction
Prior works

- Metric-based model

Support Set   Query Set



Episode with 2-way 3-shot

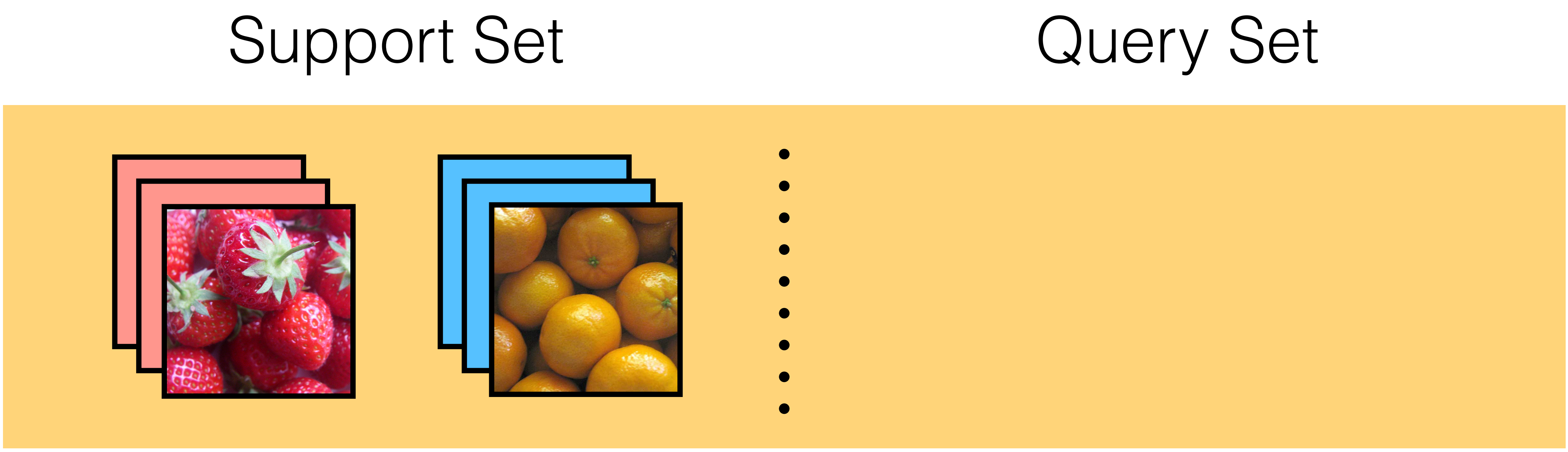


Embedding space

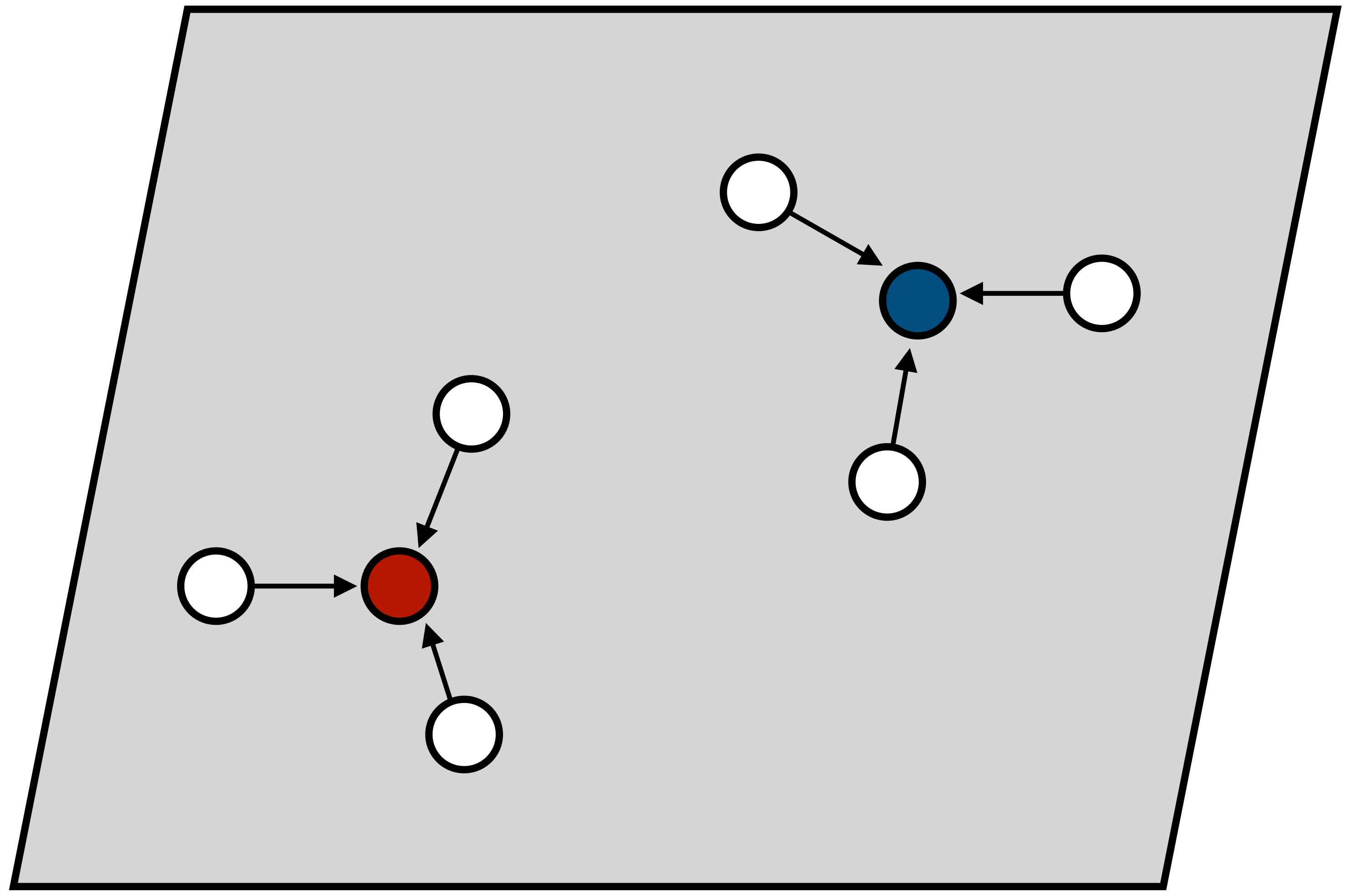
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

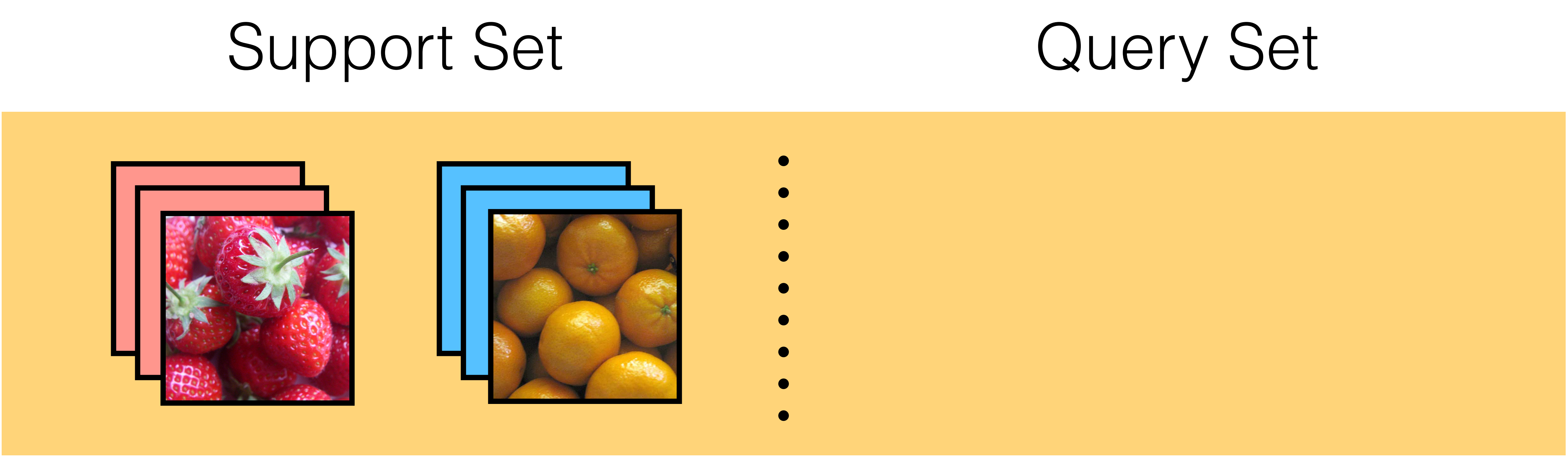


Embedding space

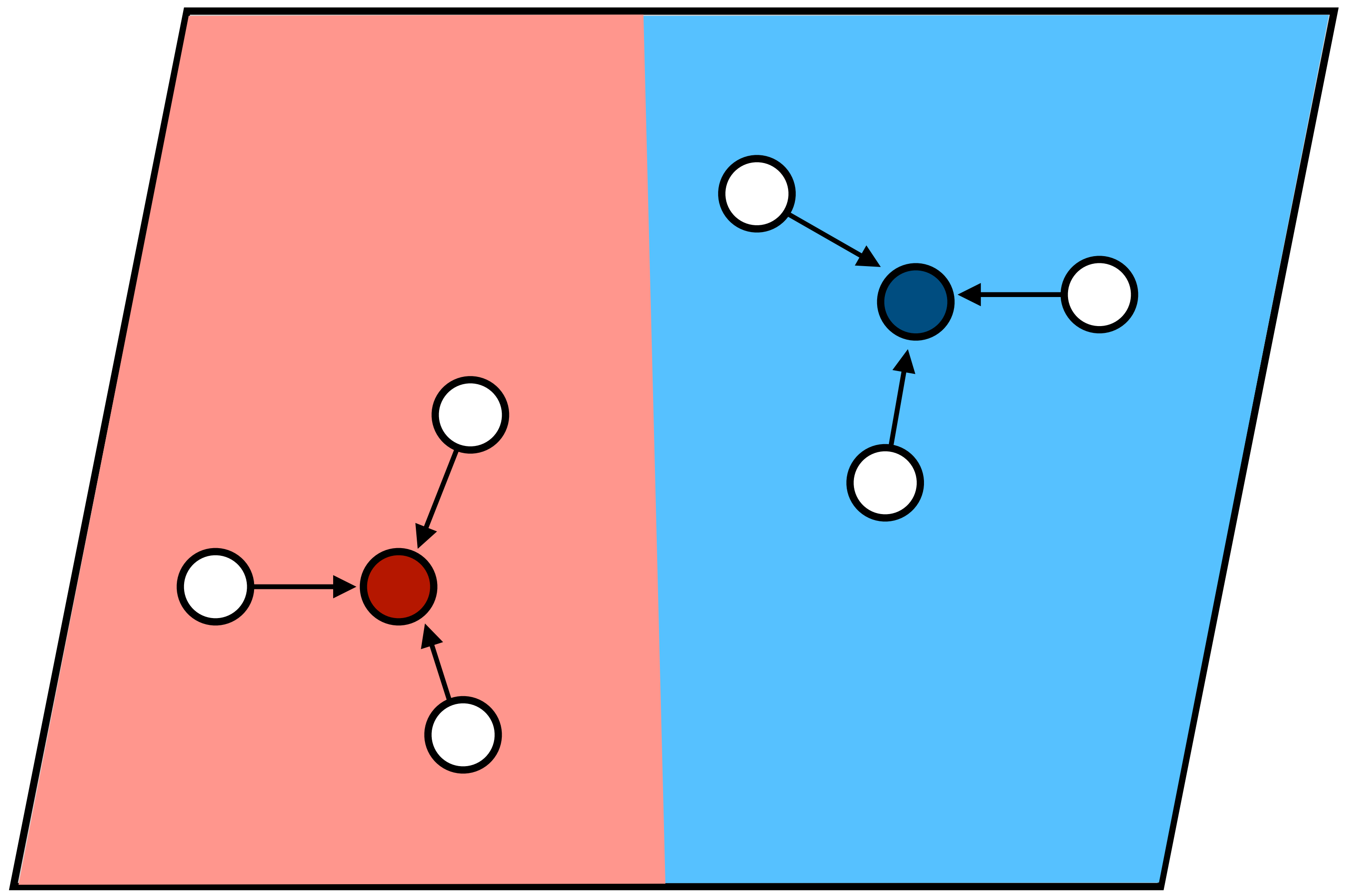
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

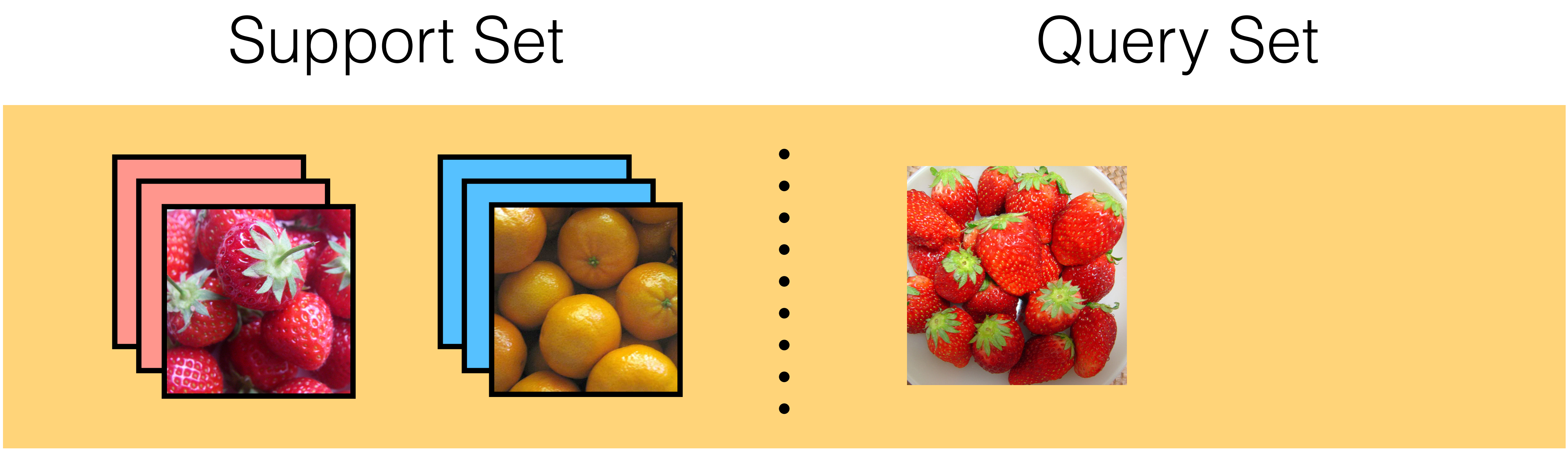


Embedding space

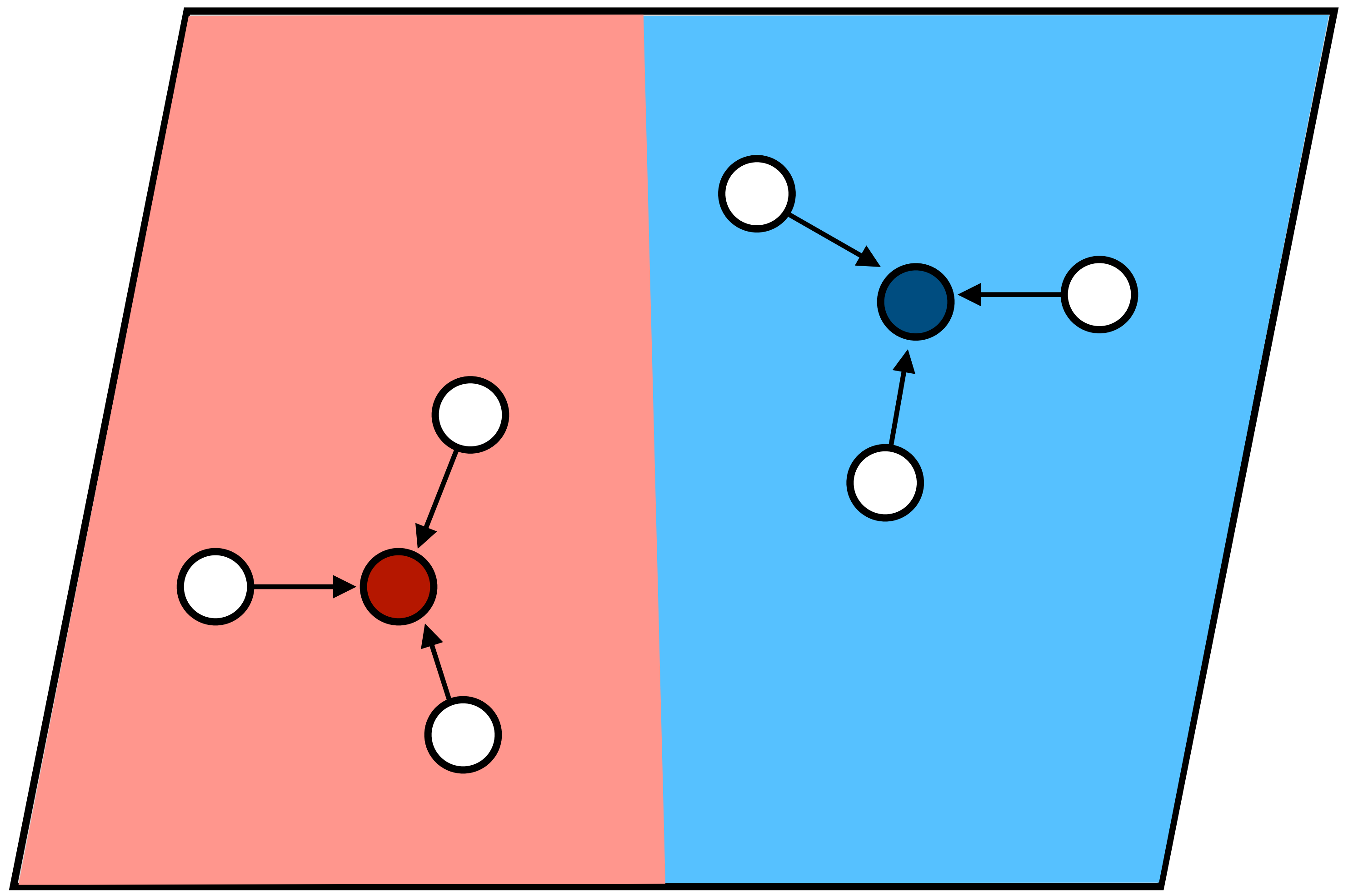
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

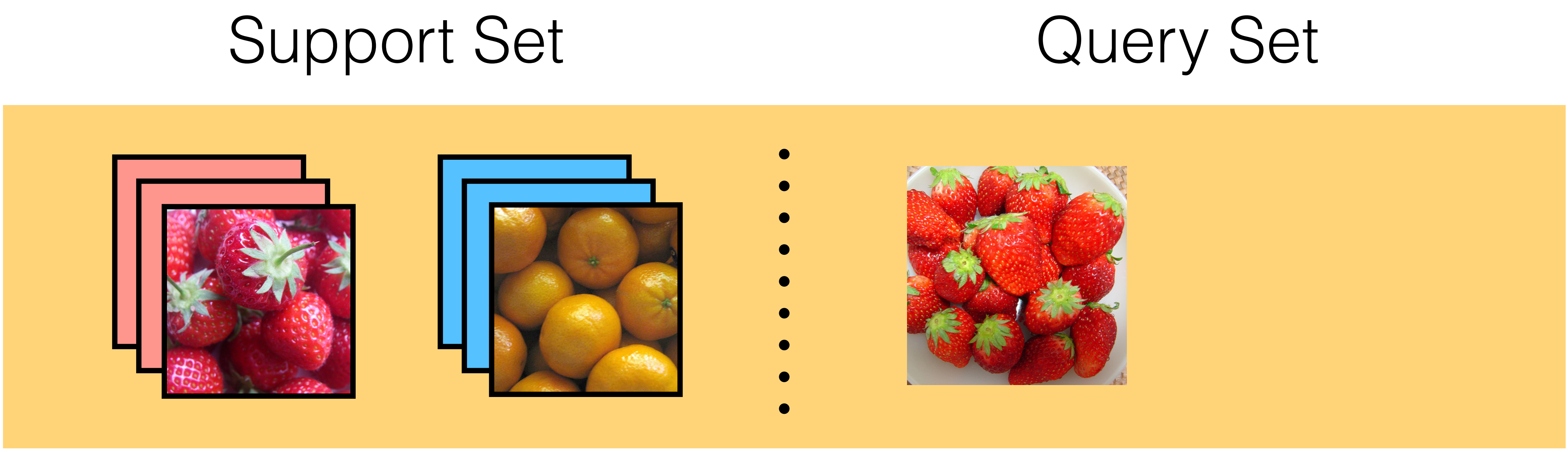


Embedding space

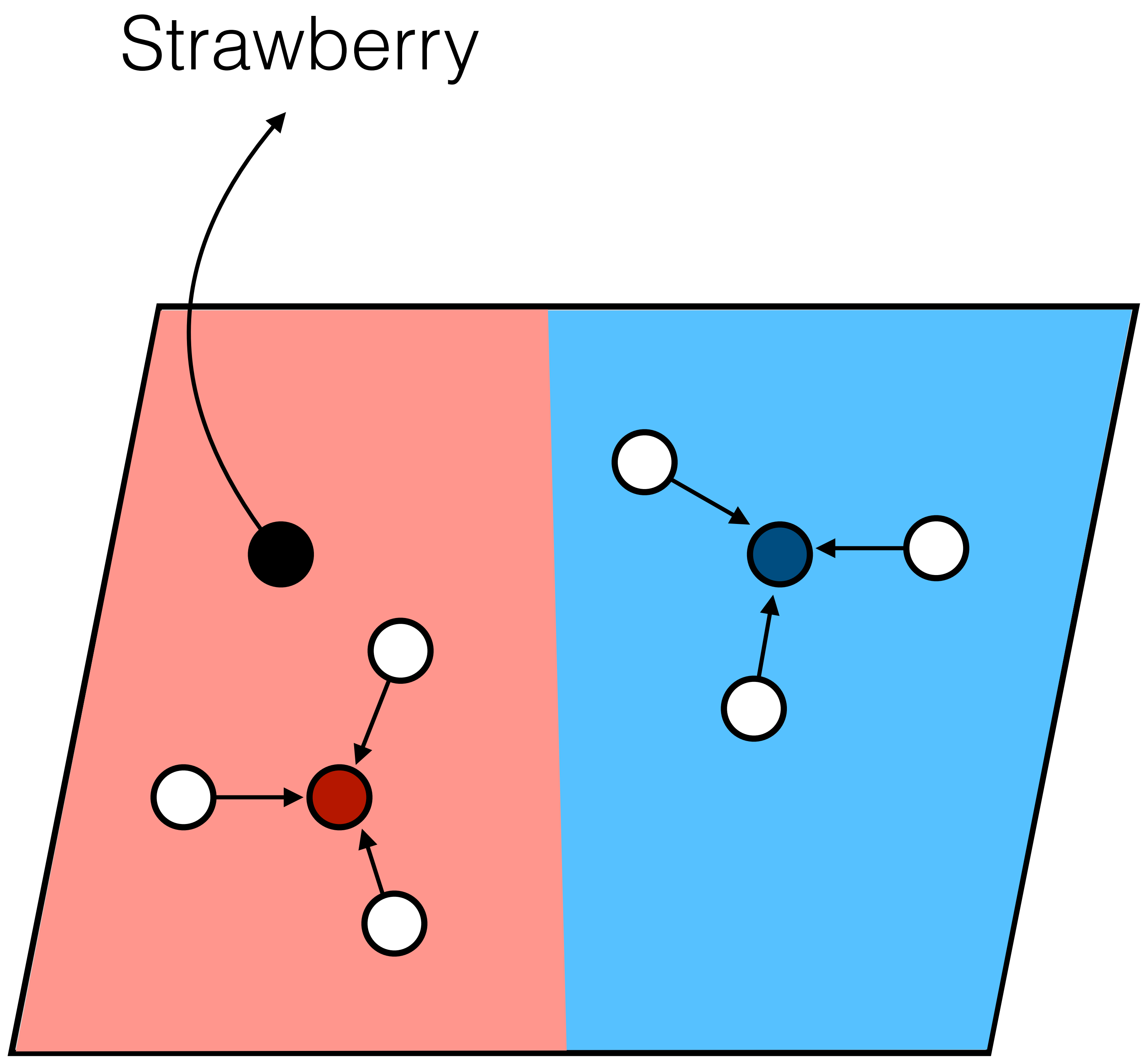
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot

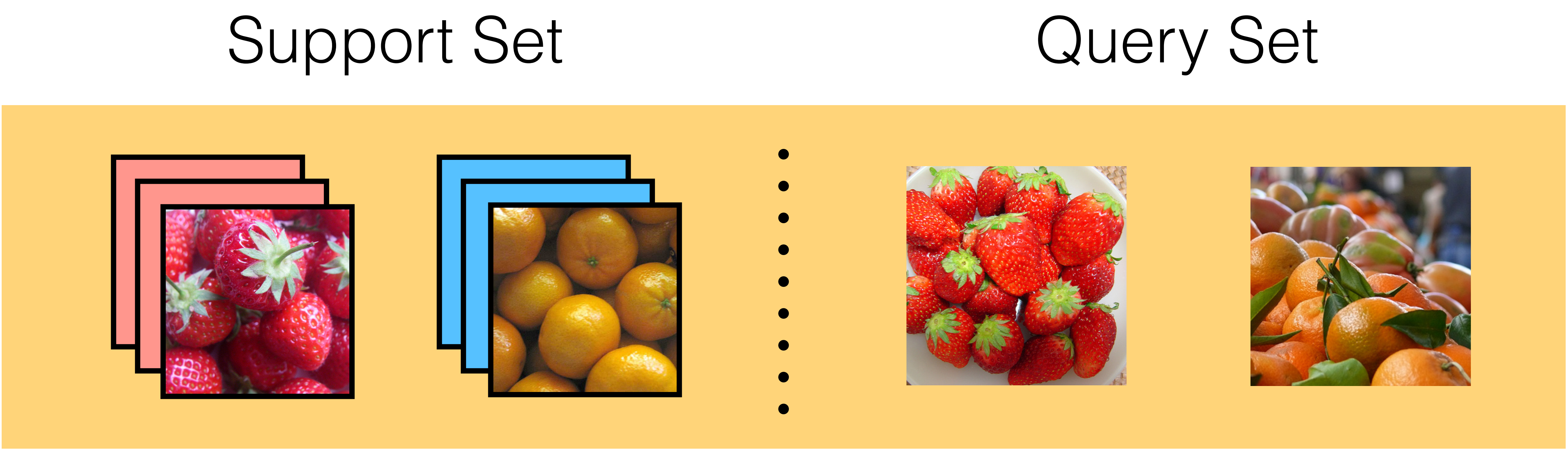


Embedding space

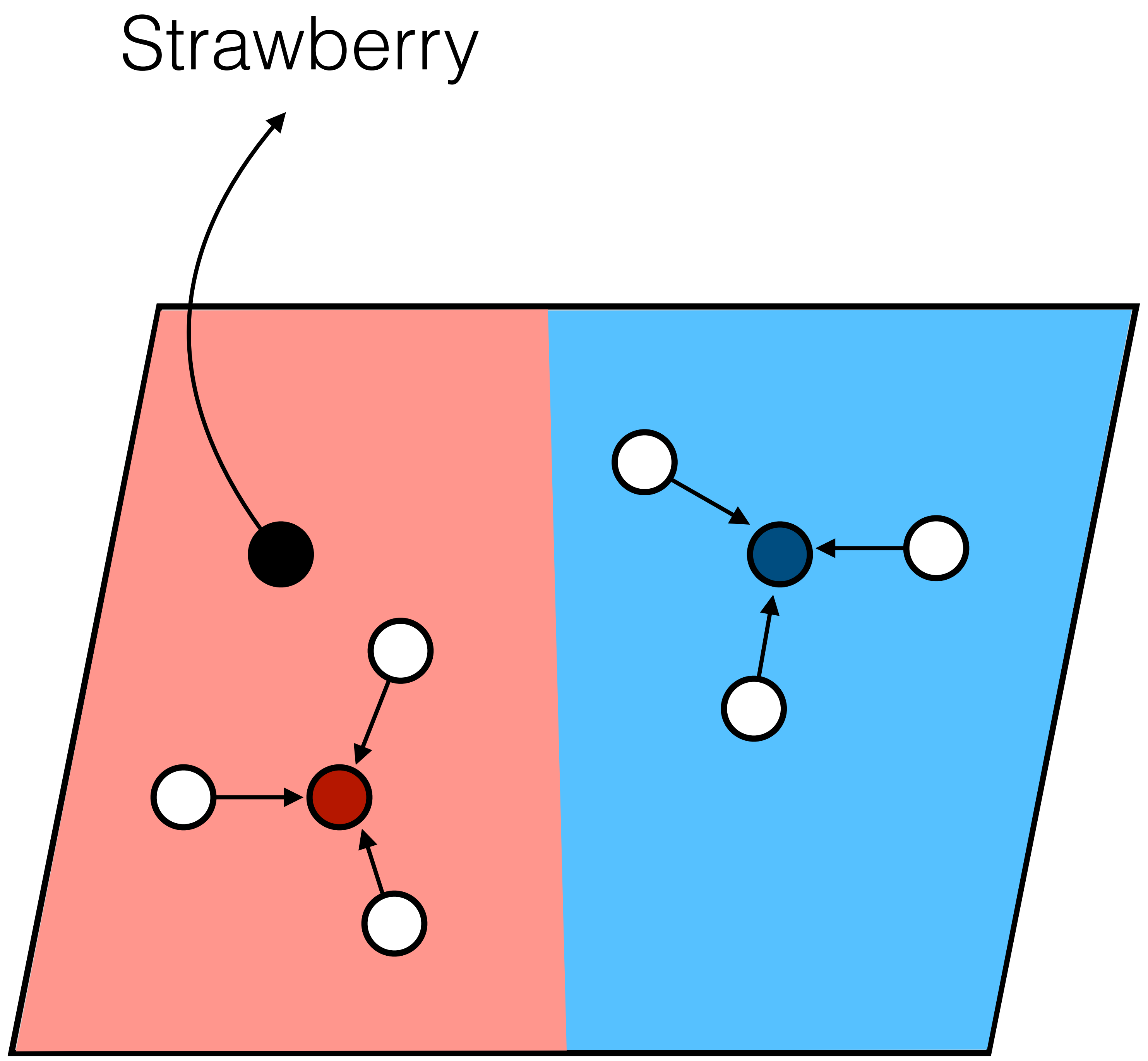
Introduction

Prior works

- Metric-based model



Episode with 2-way 3-shot



Embedding space

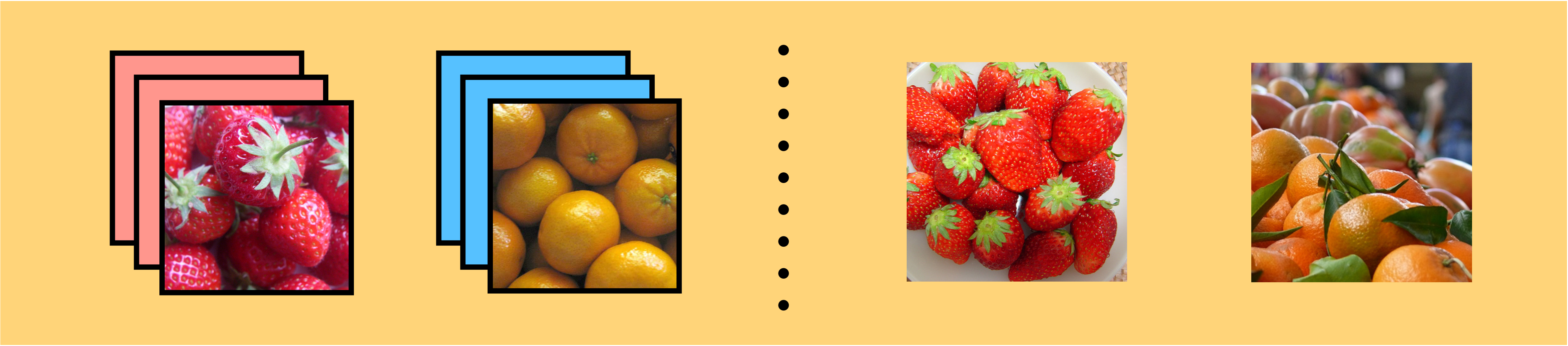
Introduction

Prior works

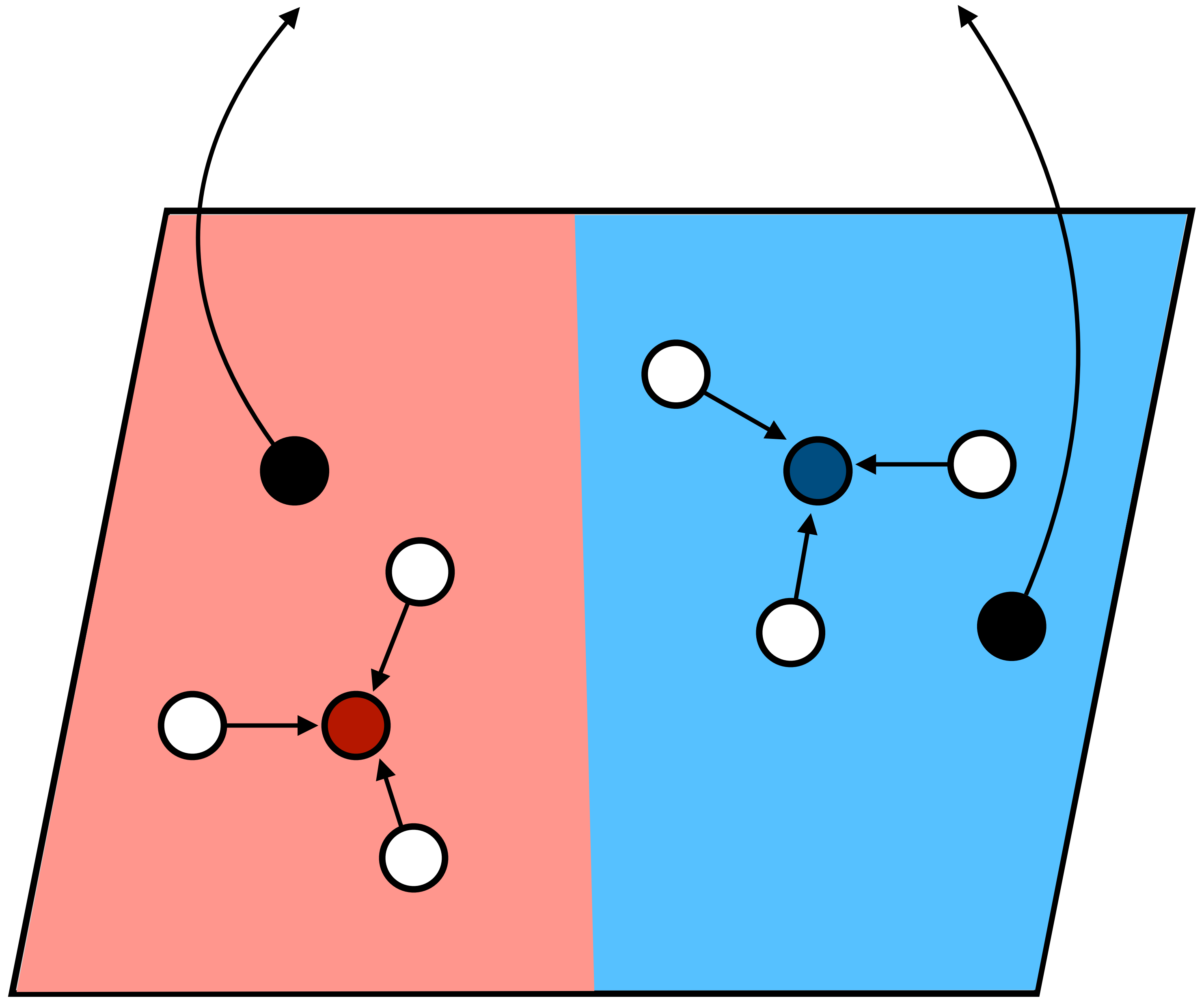
- Metric-based model

Support Set

Query Set



Episode with 2-way 3-shot



Embedding space

Introduction

Prior works

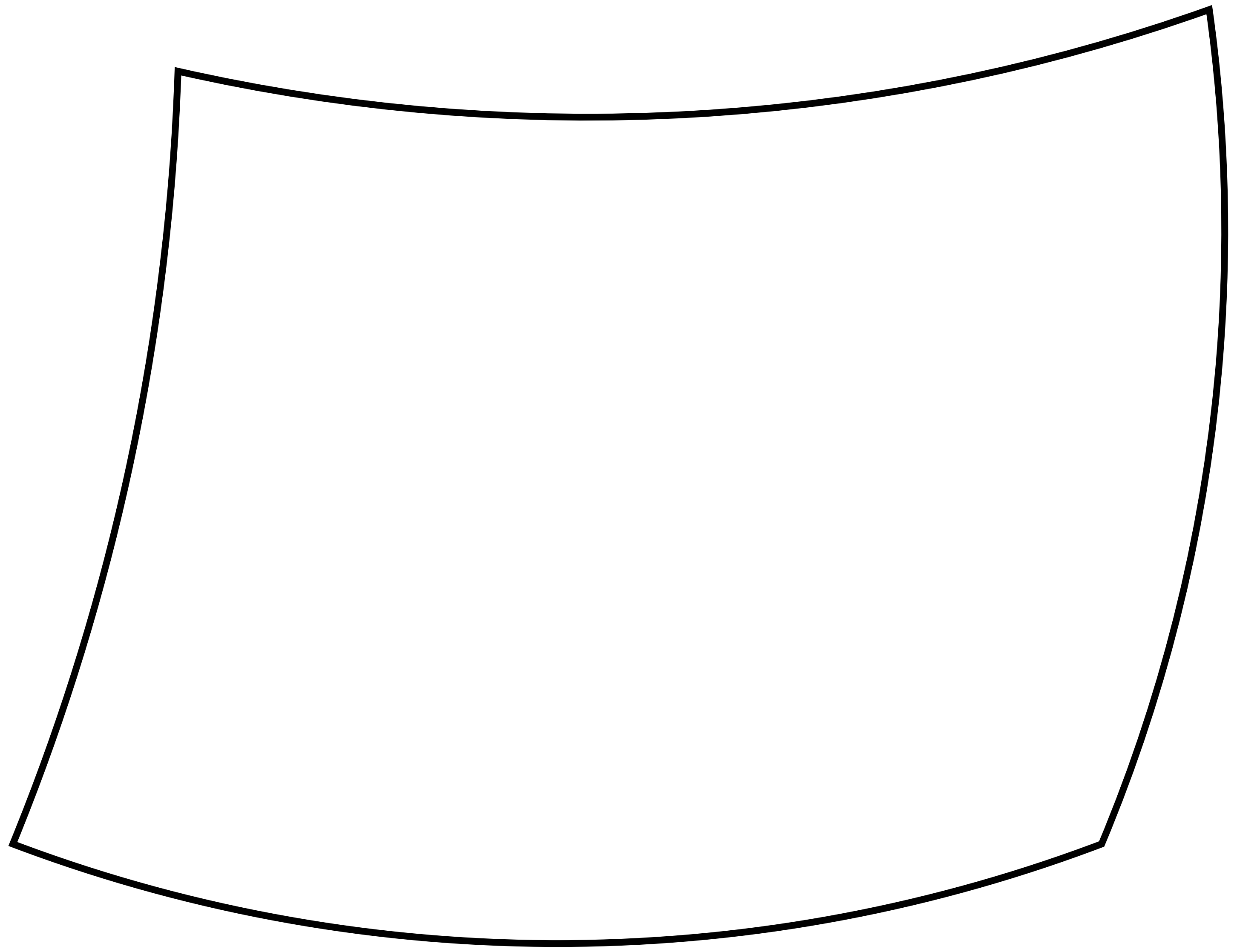
- Optimization-based model

E_1

E_2

E_3

*called ‘meta-batch’

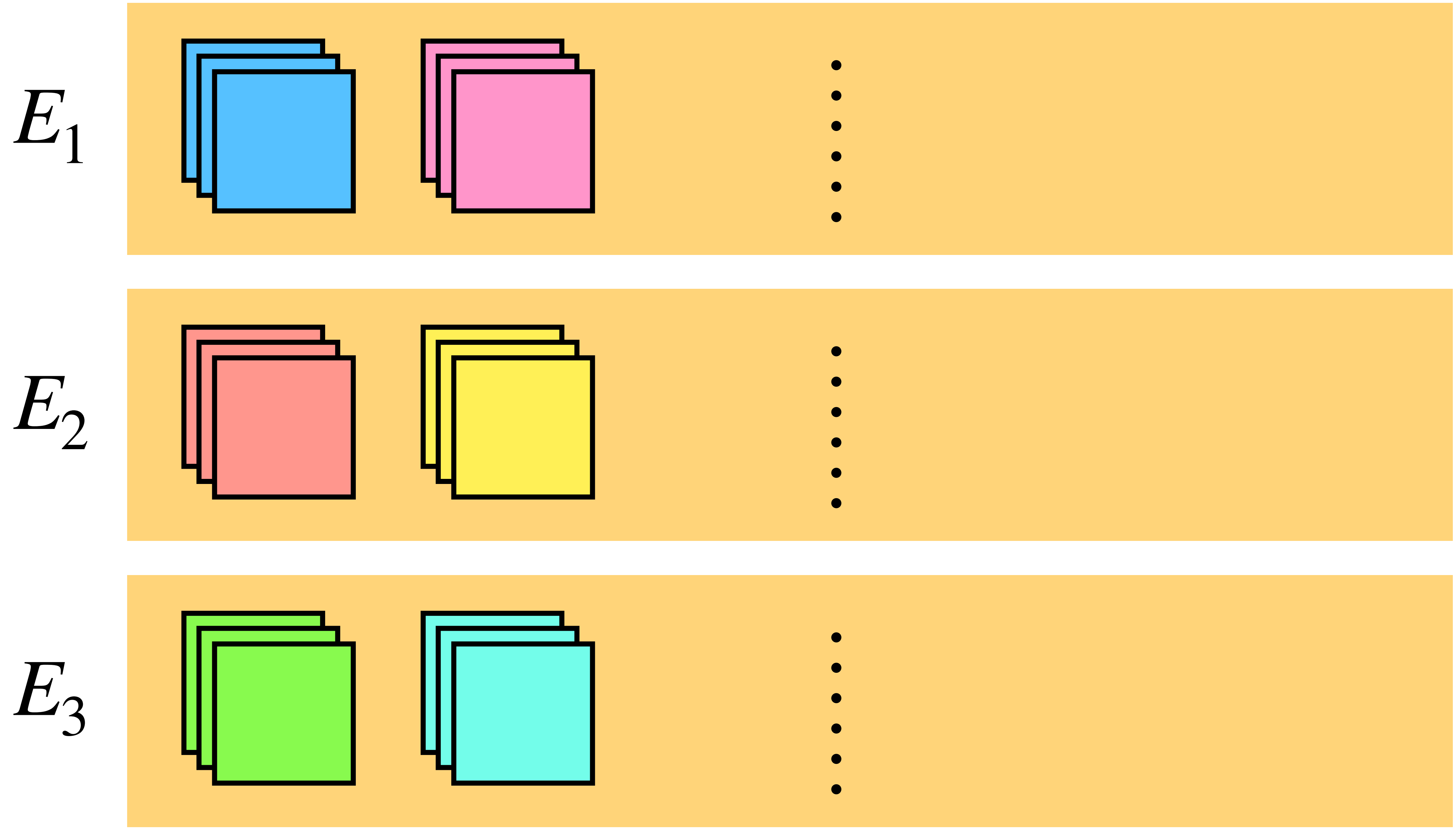


Parameter space

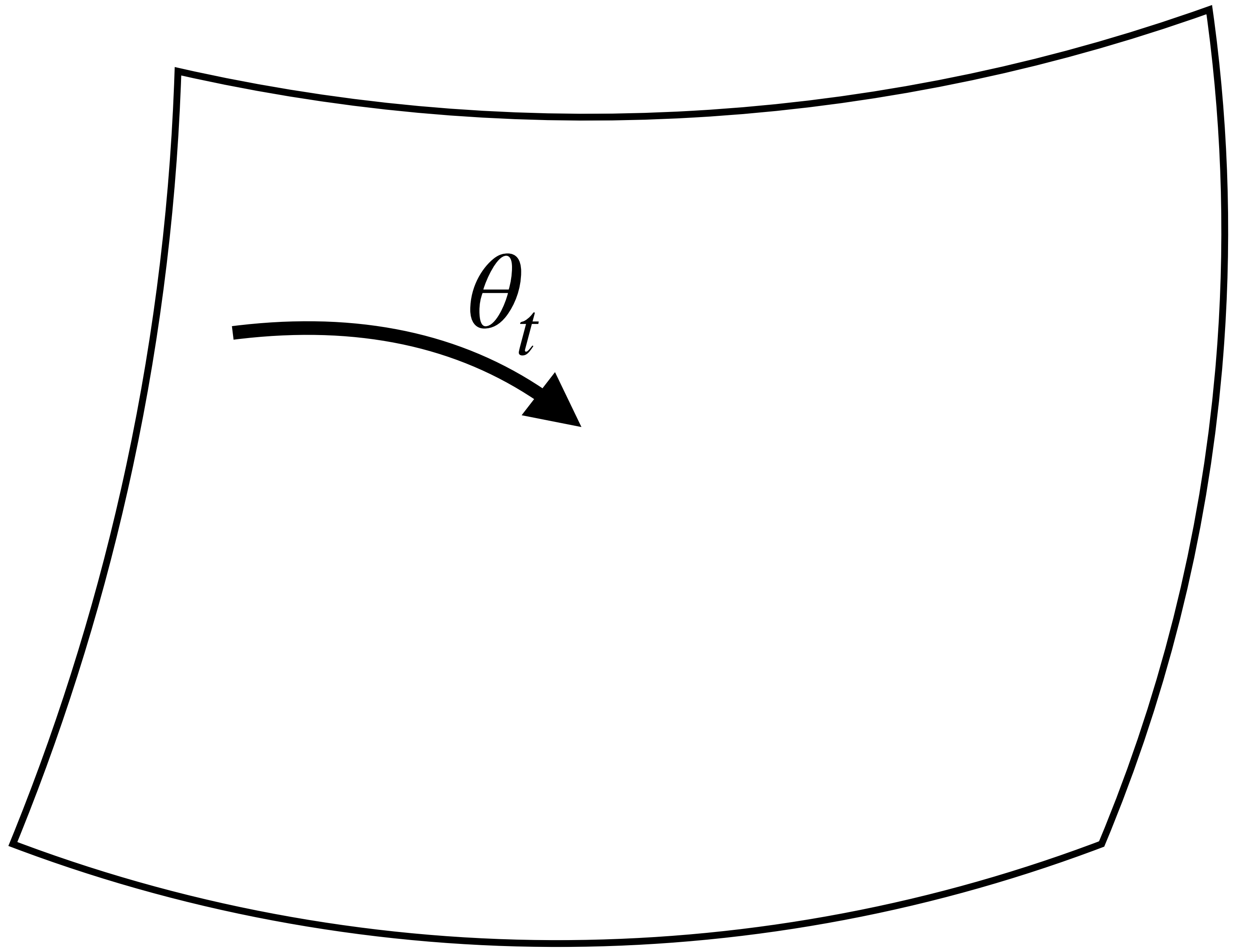
Introduction

Prior works

- Optimization-based model



*called ‘meta-batch’

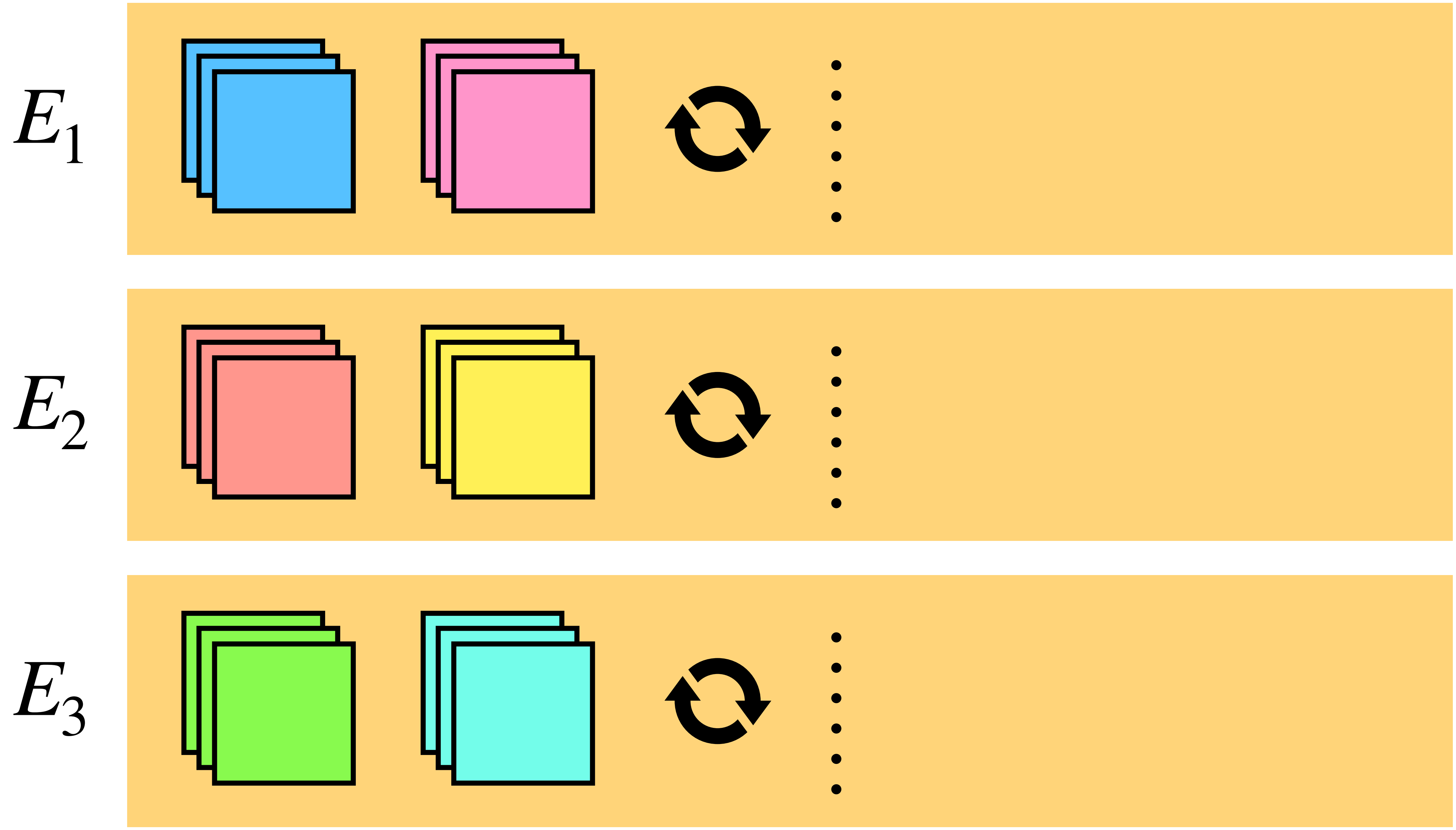


Parameter space

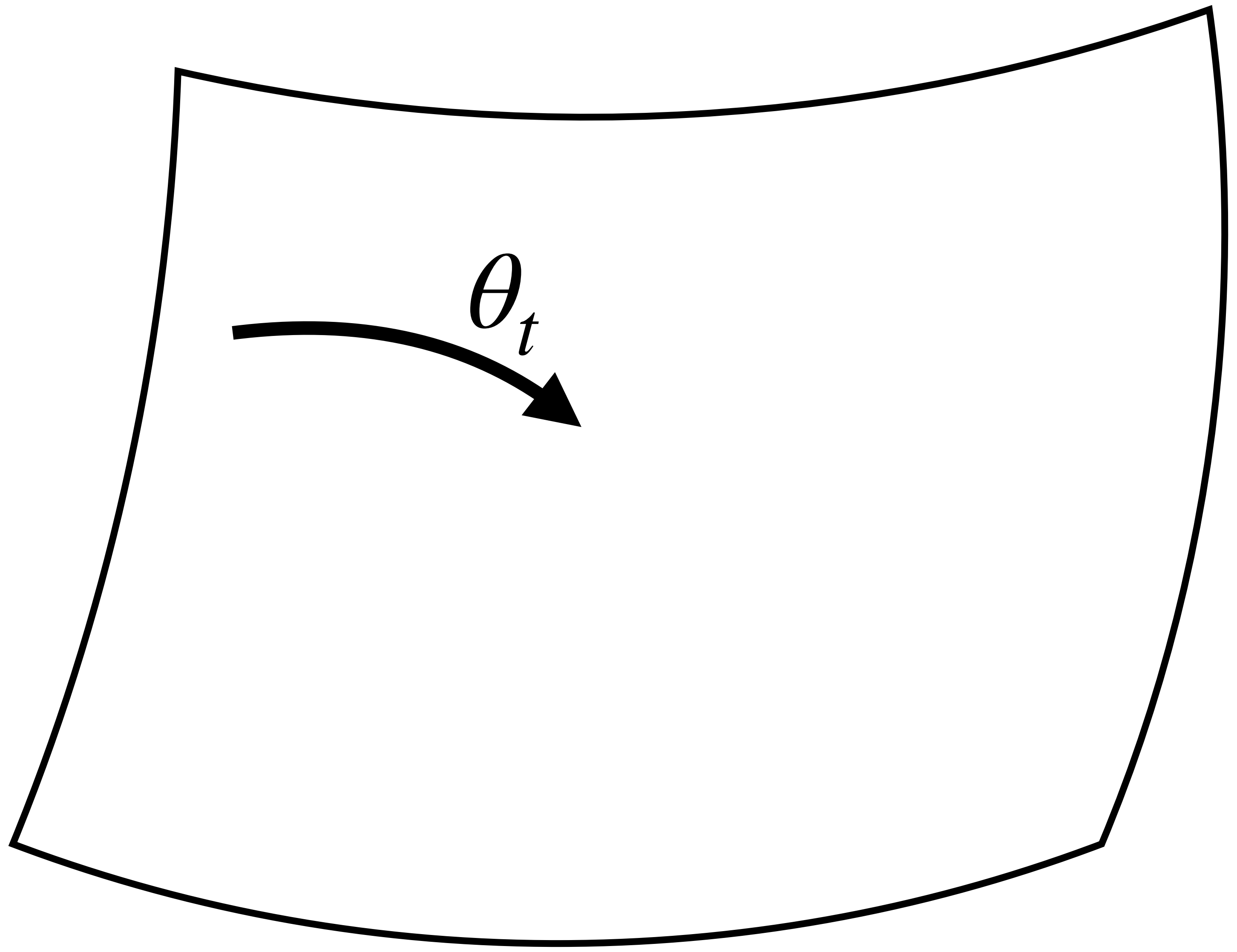
Introduction

Prior works

- Optimization-based model



*called ‘meta-batch’

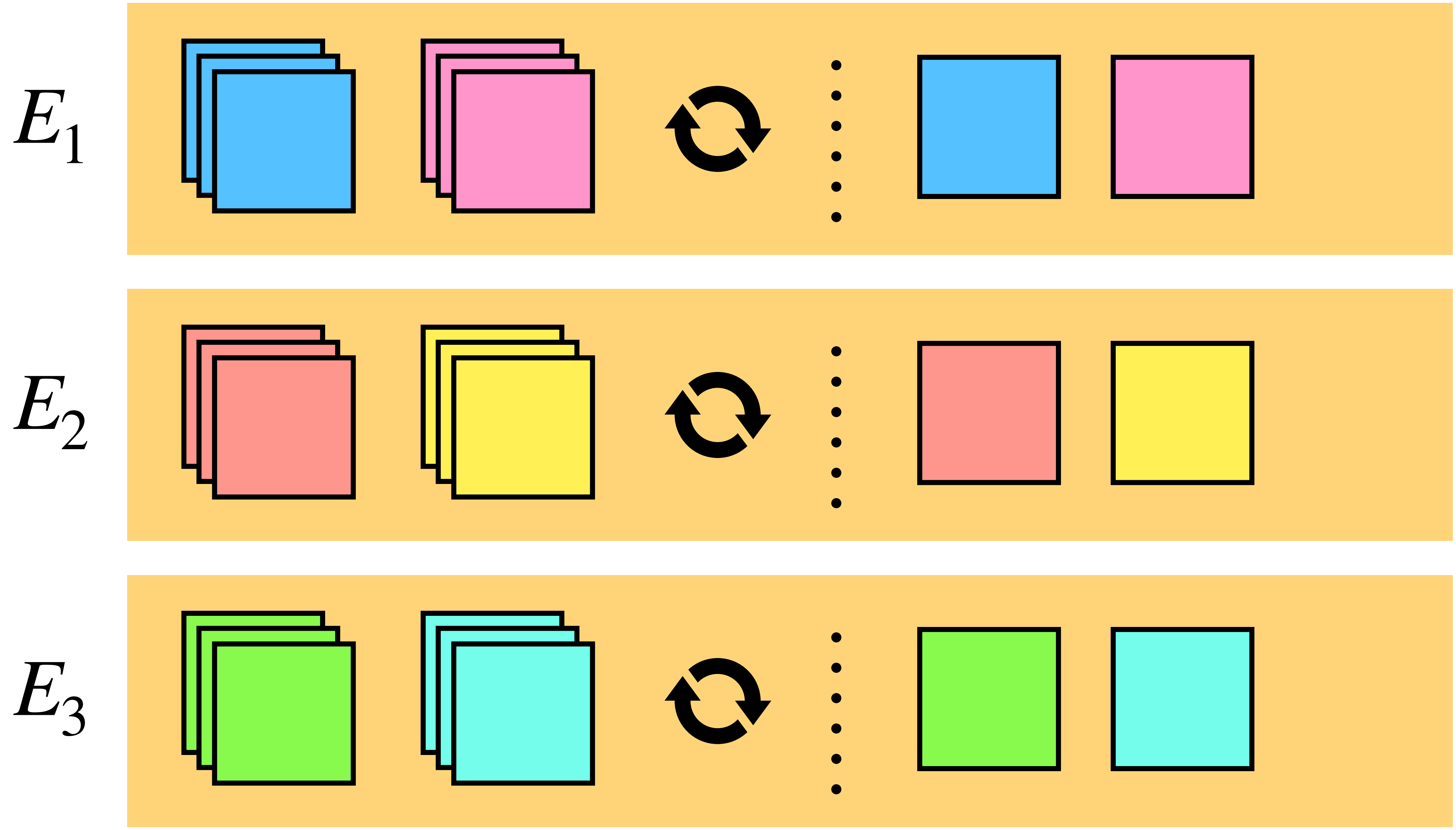


Parameter space

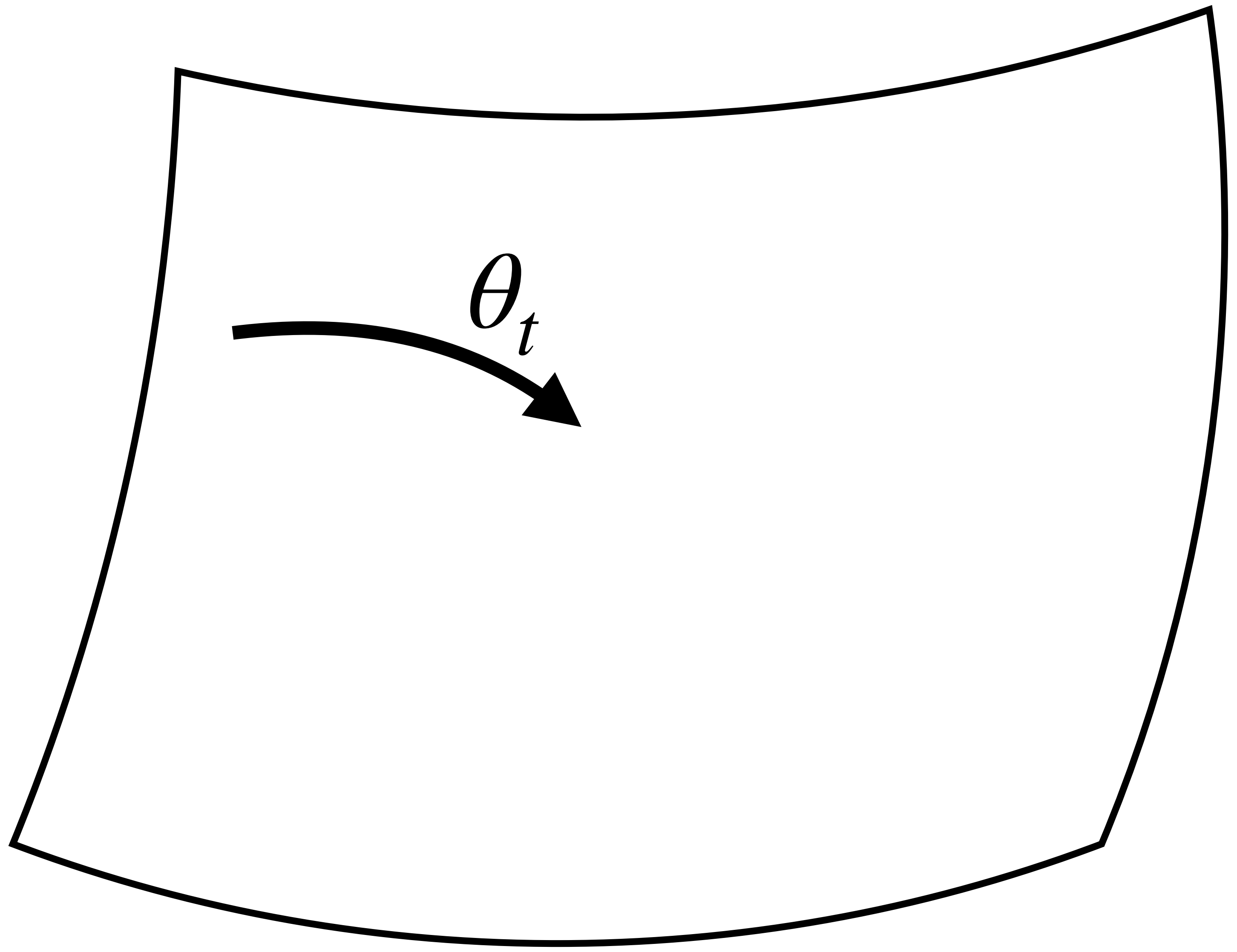
Introduction

Prior works

- Optimization-based model



*called ‘meta-batch’

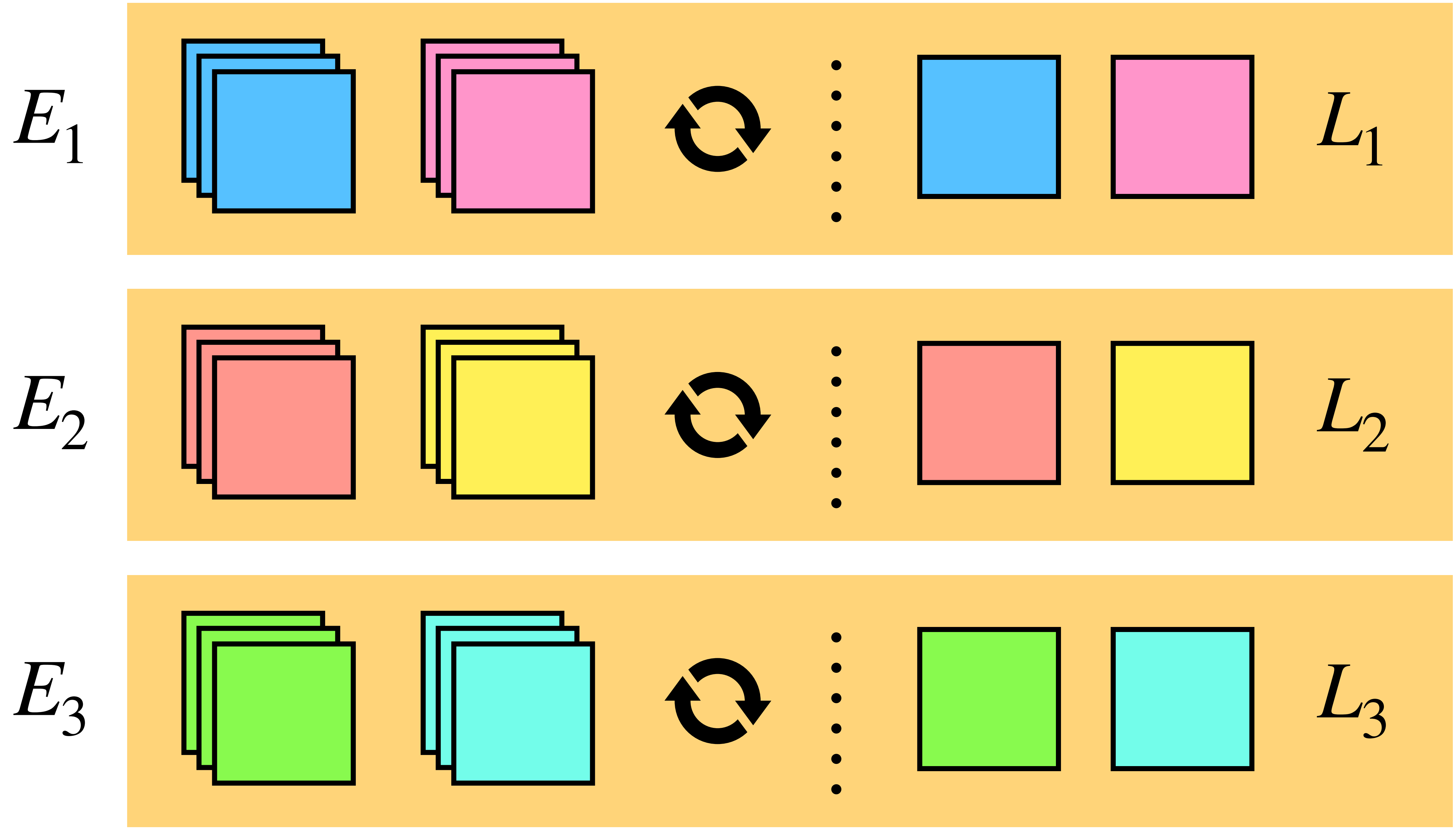


Parameter space

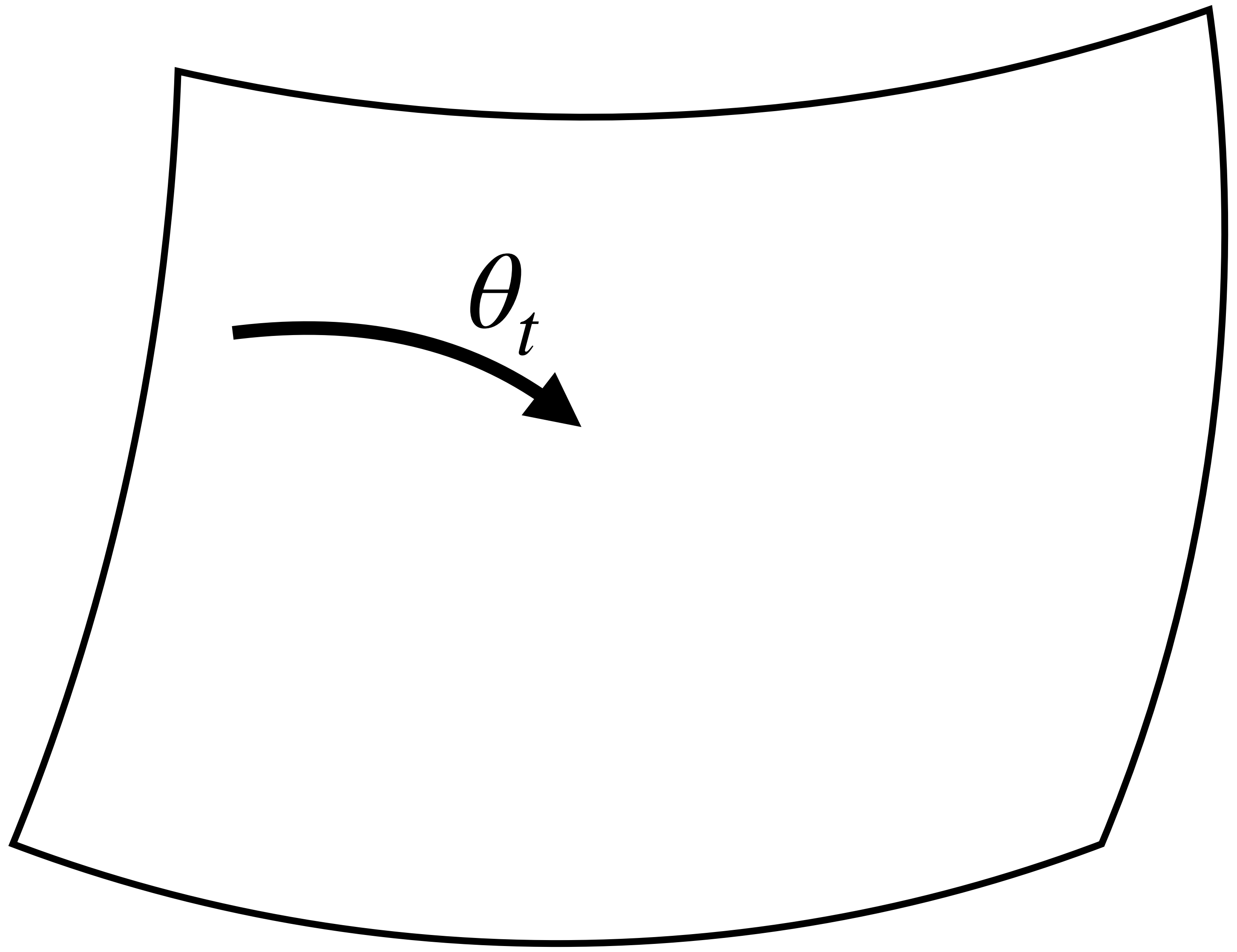
Introduction

Prior works

- Optimization-based model



*called ‘meta-batch’

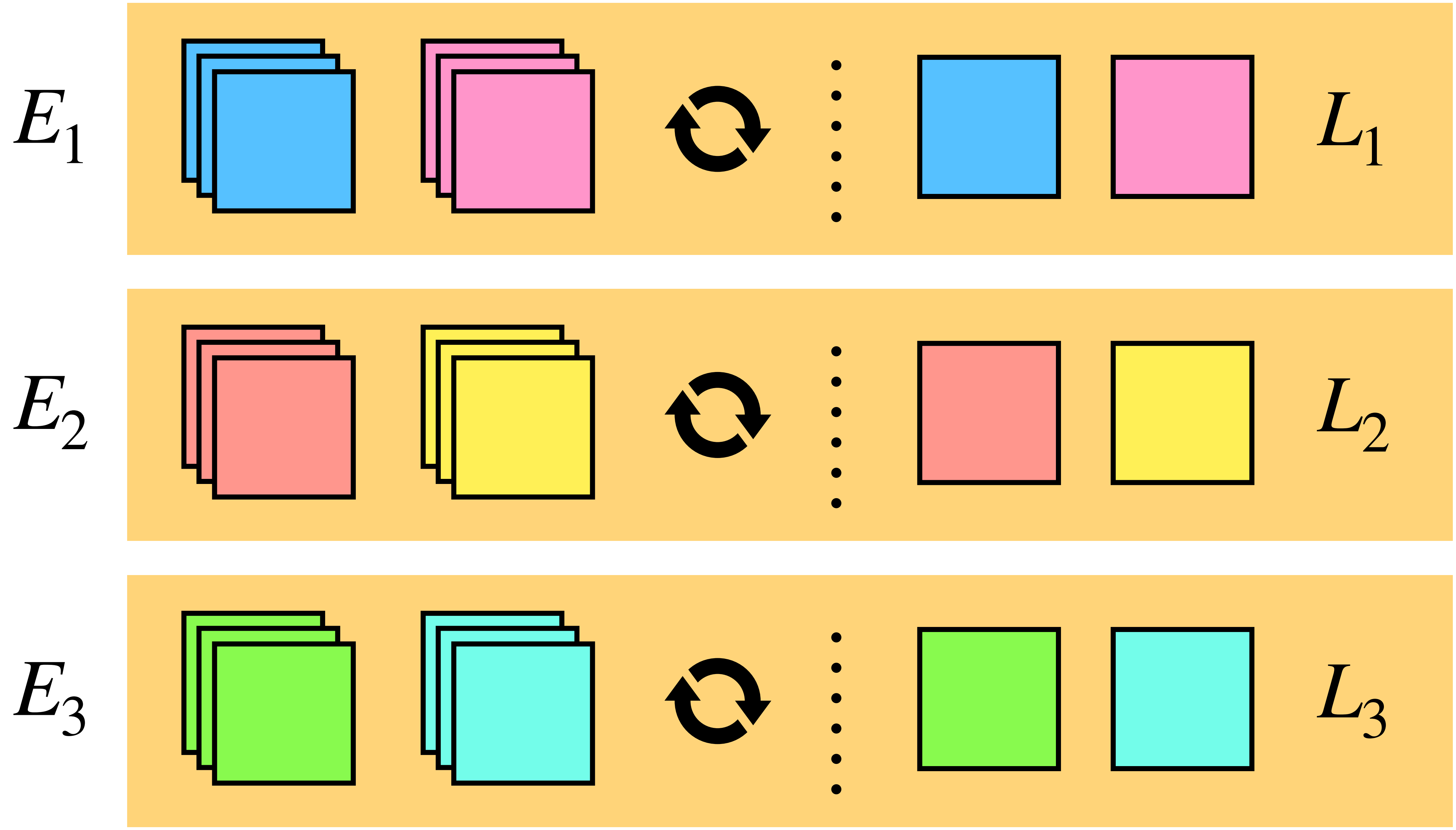


Parameter space

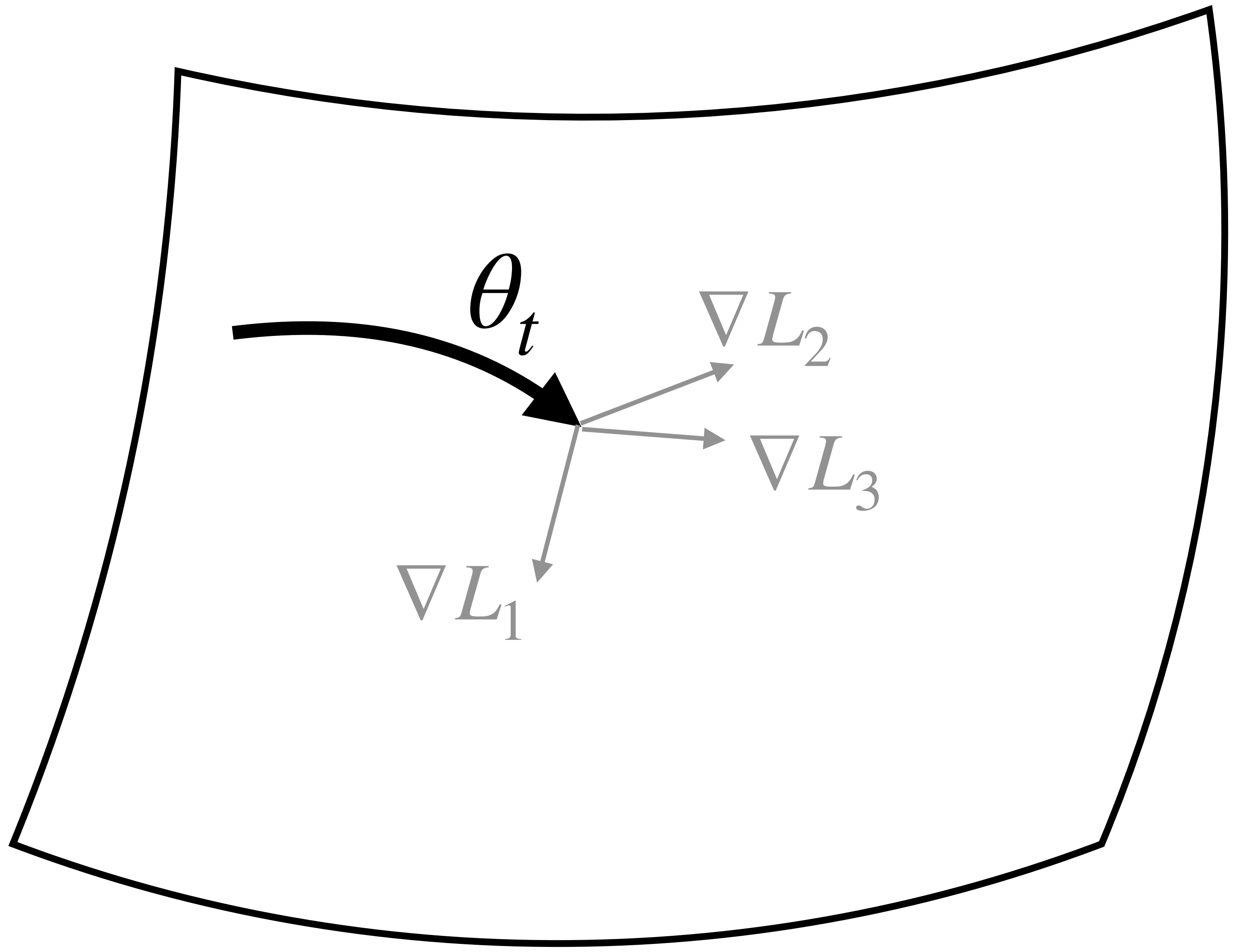
Introduction

Prior works

- Optimization-based model



*called 'meta-batch'

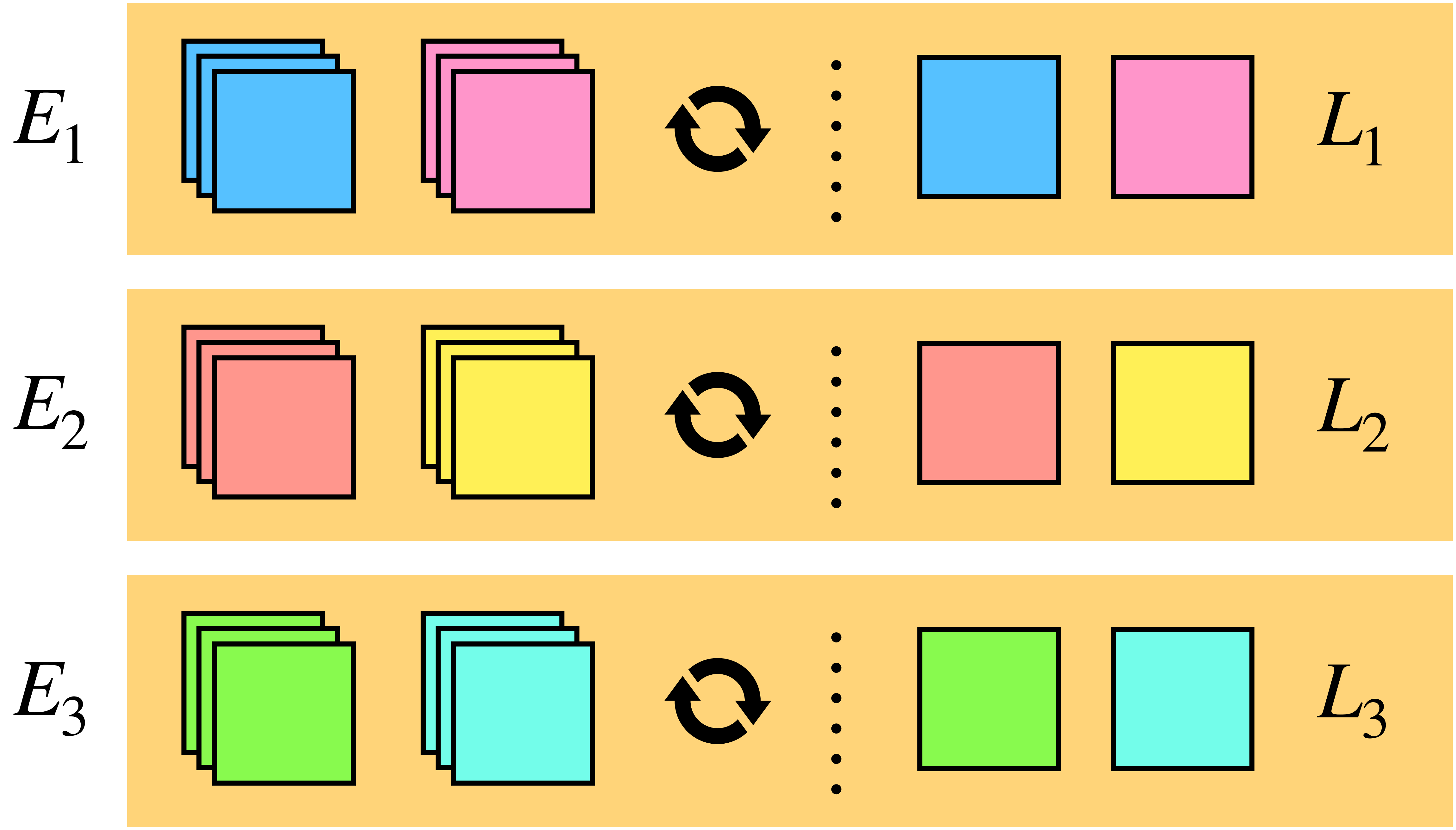


Parameter space

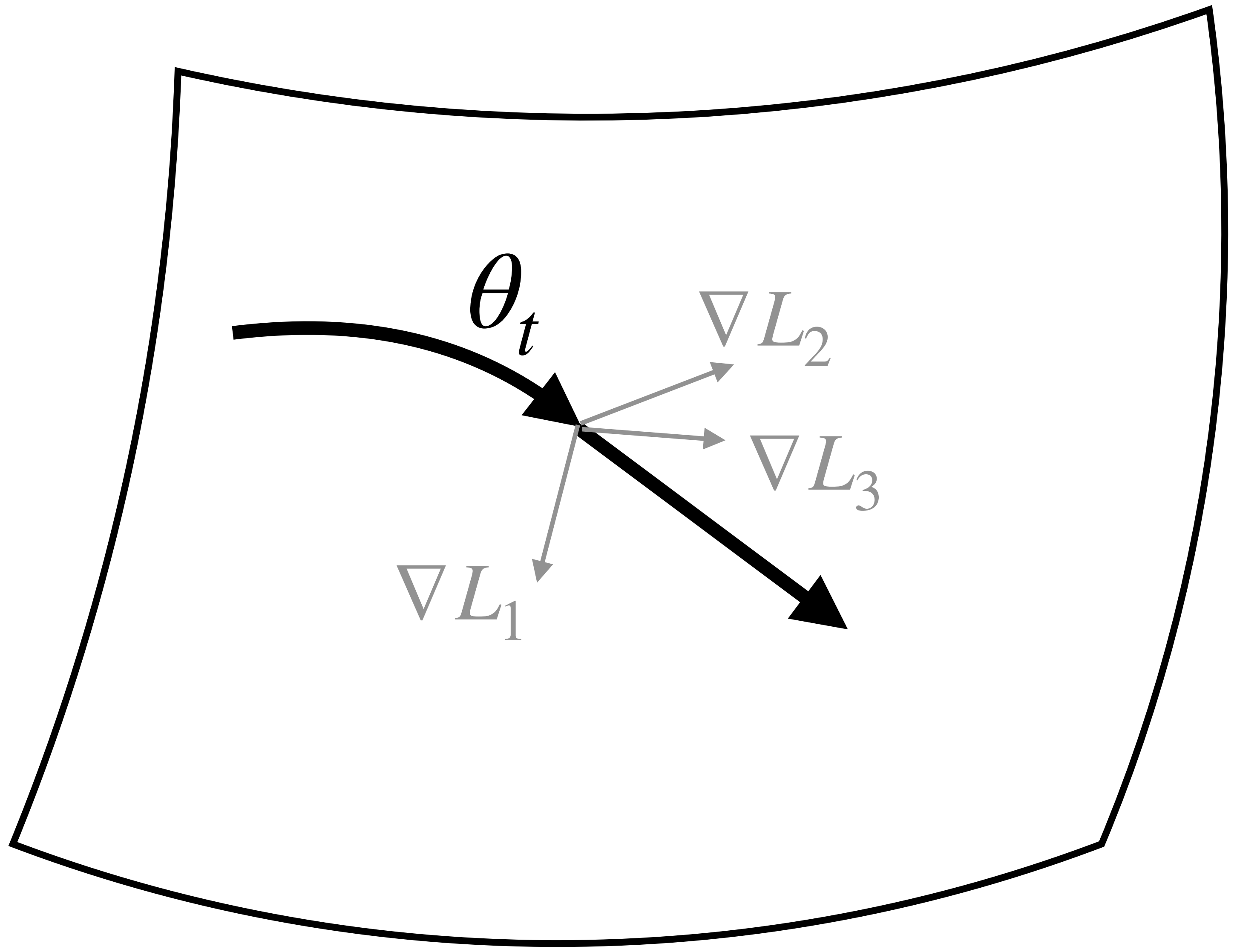
Introduction

Prior works

- Optimization-based model



*called ‘meta-batch’

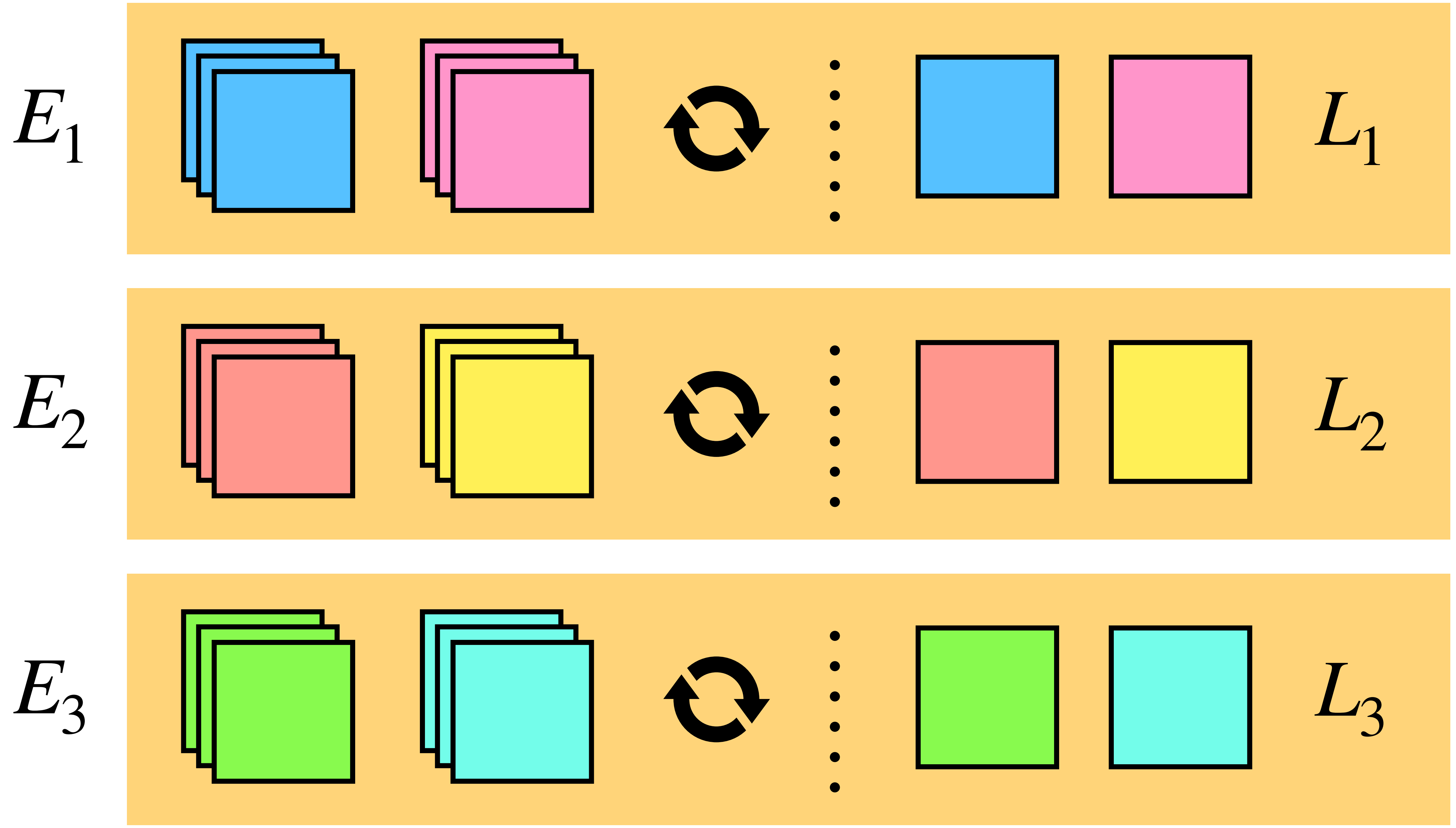


Parameter space

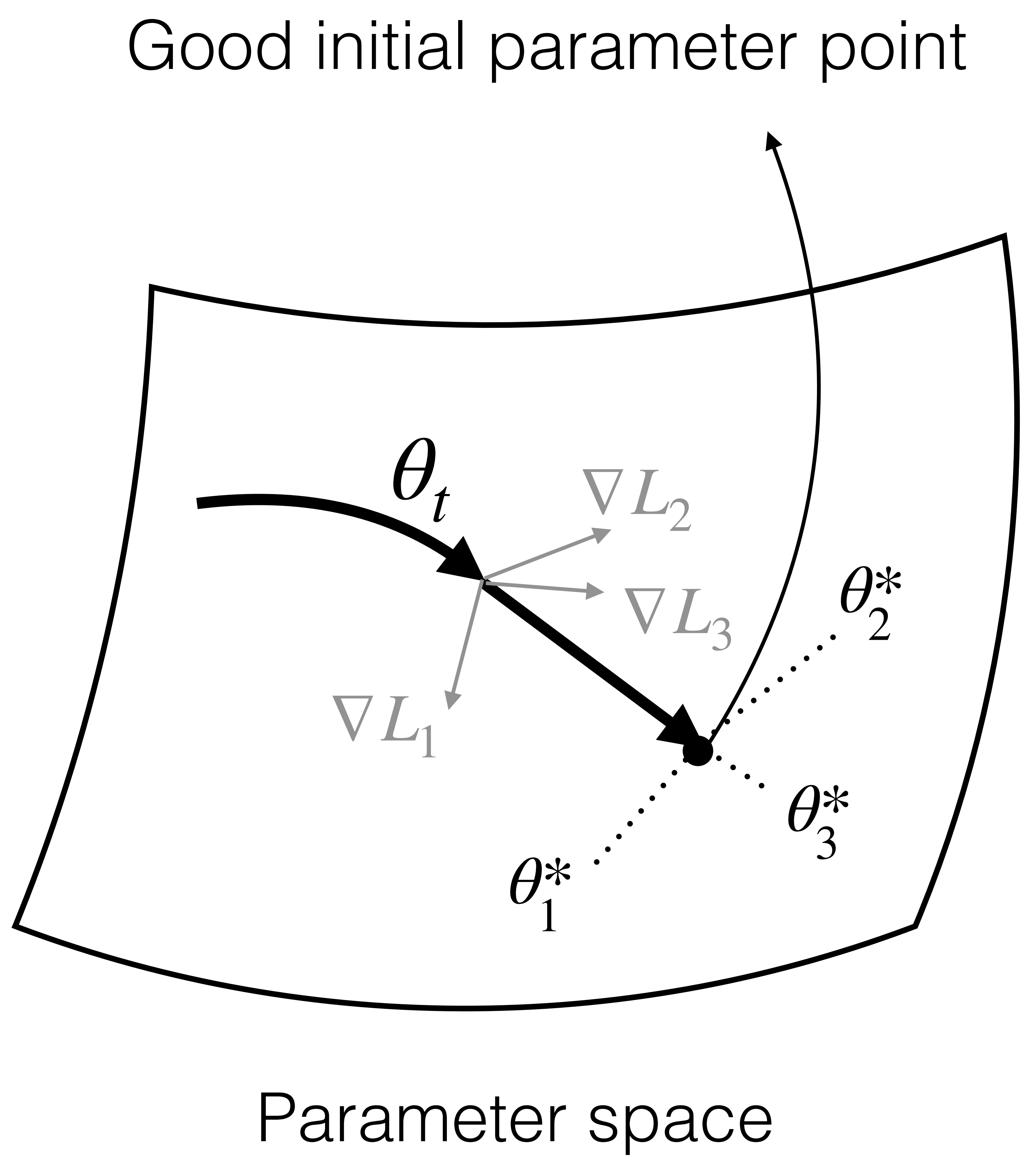
Introduction

Prior works

- Optimization-based model



*called 'meta-batch'



Proposed Model

Motivation

- What can we improve on prior works?
    1. **Relational information** between samples is not explicitly used
    2. Use only **support samples as a clue** and query samples for loss calculation (*Fundamental difficulty*)

Proposed Model

Motivation

- What can we improve on prior works?

    1. **Relational information** between samples is not explicitly used

    2. Use only **support samples as a clue** and query samples for loss calculation (*Fundamental difficulty*)

→ *Statistical measurement, **graph structure, transduction method** ...*

are used in recent researches

Proposed Model

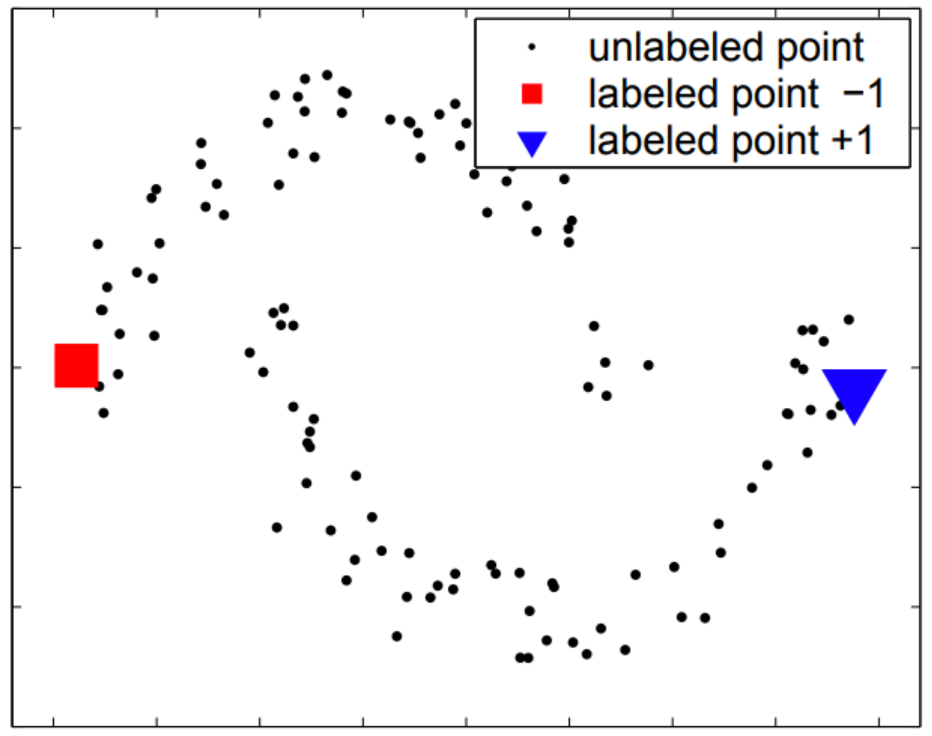
Motivation

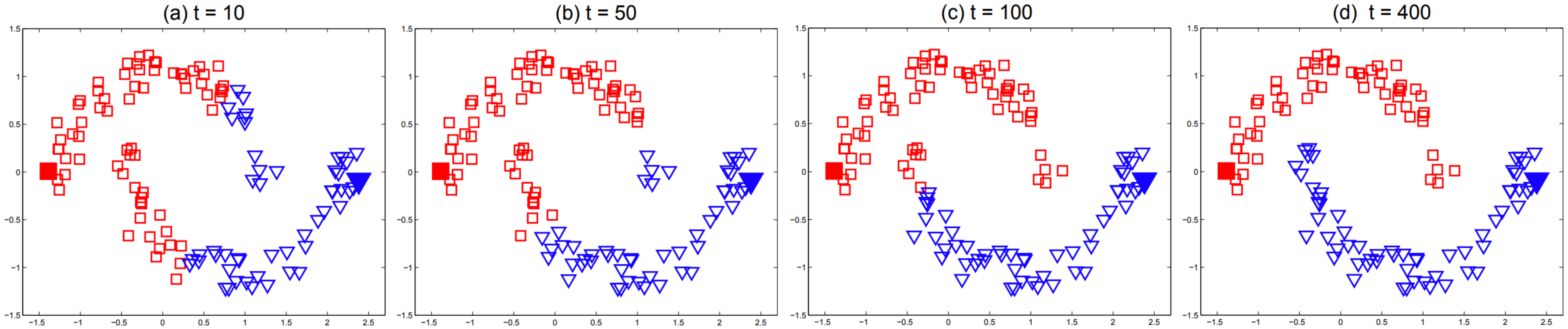
- What can we improve on prior works?
  1. **Relational information** between samples is not explicitly used
  2. Use only **support samples as a clue** and query samples for loss calculation (*Fundamental difficulty*)

→ ***Label propagation algorithm!***

Proposed Model

Motivation





→ ***Label propagation algorithm!***

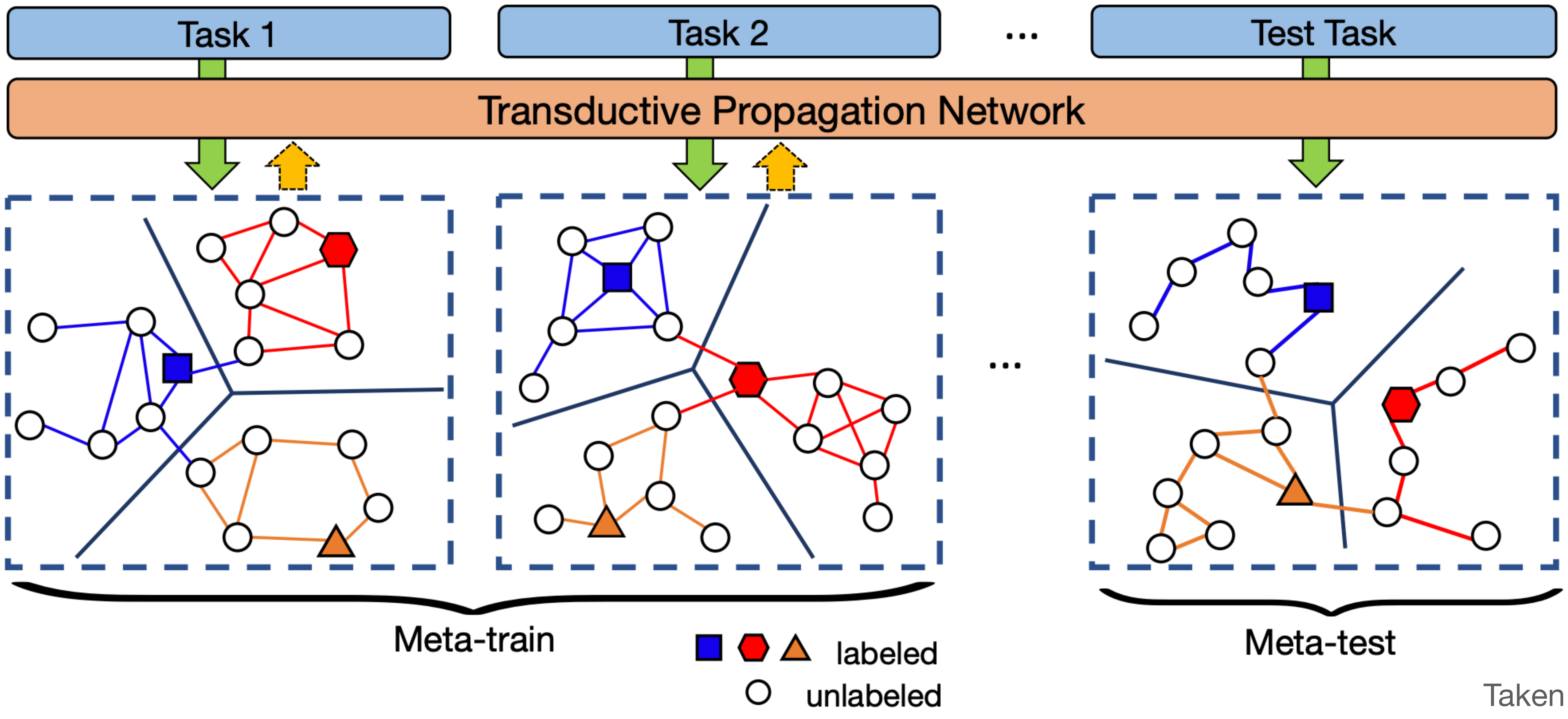
Proposed Model

Motivation

1. Label propagation algorithm?
    - **inapplicable in few-shot** (data is limited and unevenly distributed)
2. What is the appropriate hyper-parameter?
    - Performance of transduction method is **sensitive to the hyper-parameter** (σ in label propagation algorithm)

Proposed Model

Transductive Propagation Networks

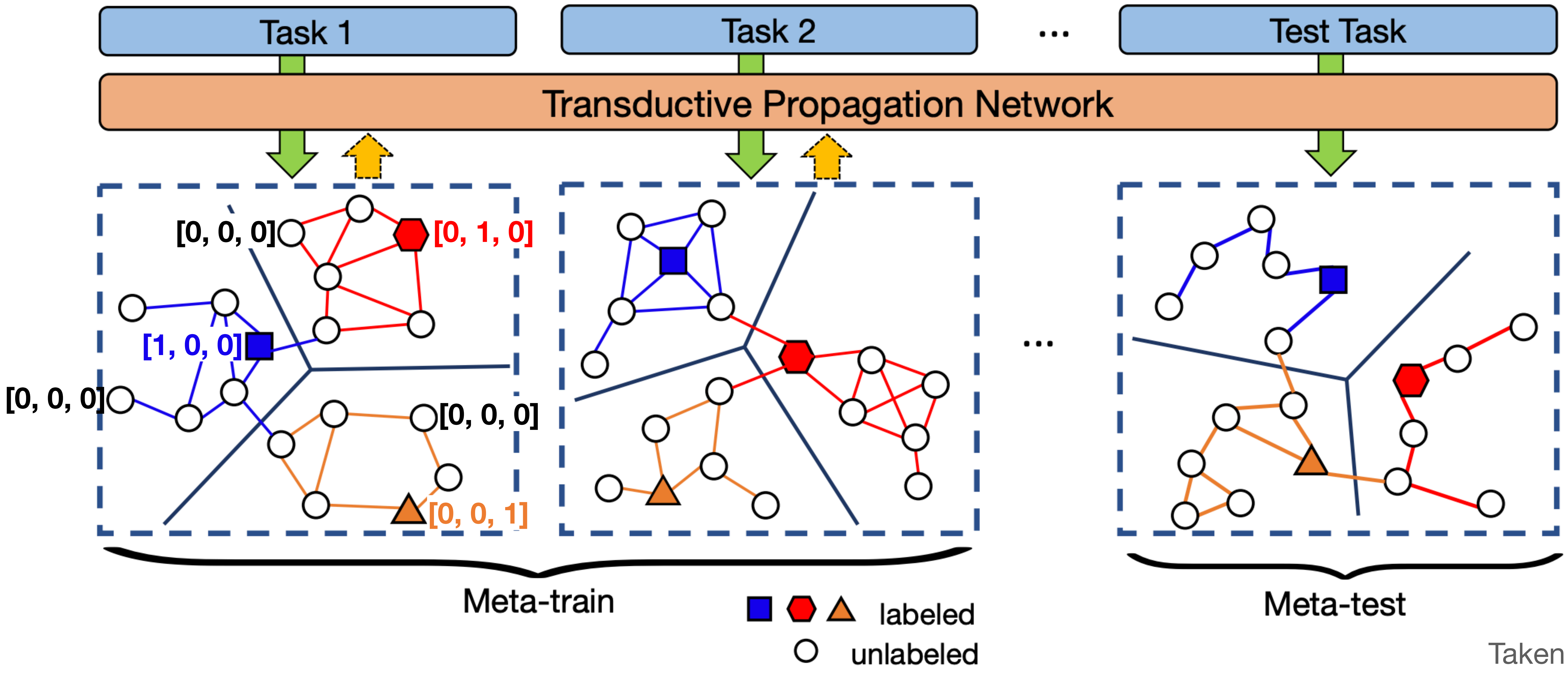


Taken from [Liu, 2019]

[Liu, 2019] Liu, Yanbin, et al. "Learning to propagate labels: Transductive propagation network for few-shot learning," ICLR 2019.

Proposed Model

Transductive Propagation Networks

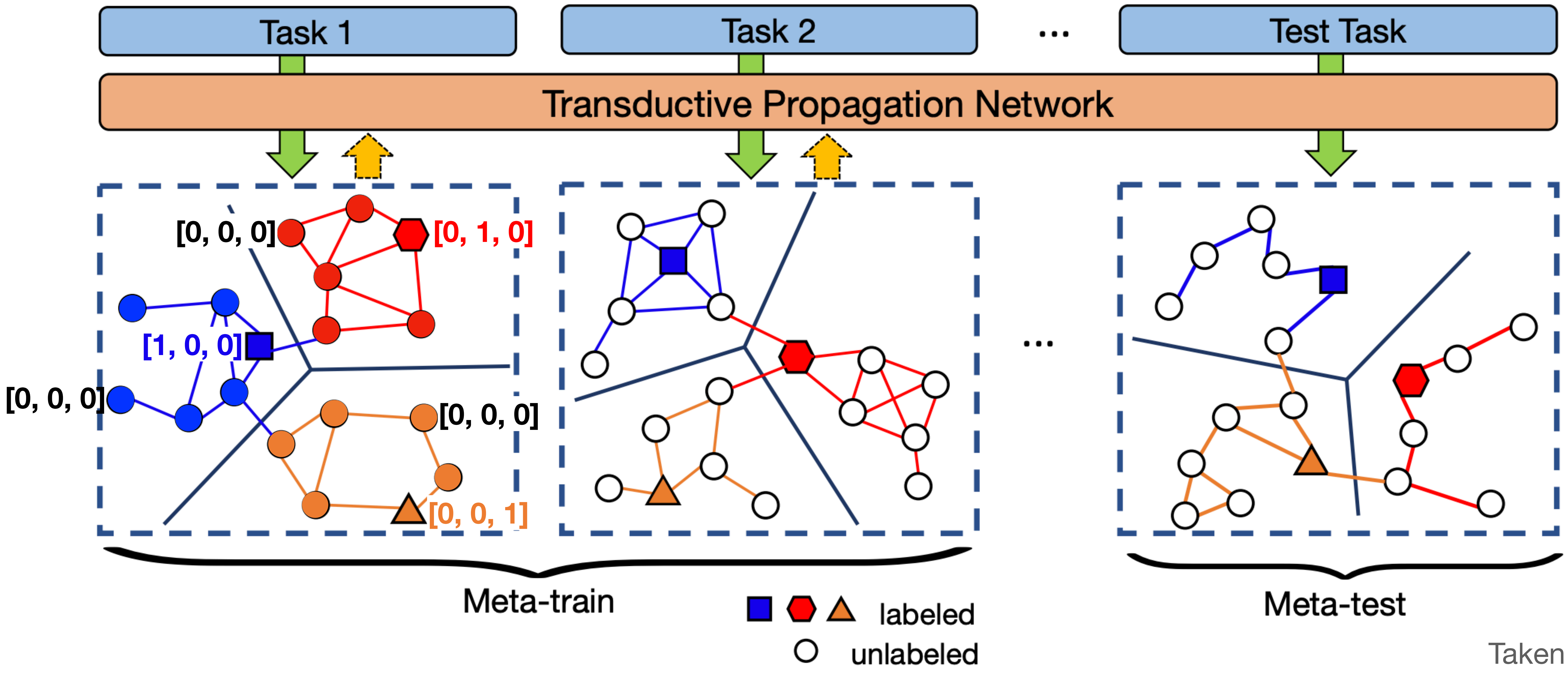


Taken from [Liu, 2019]

[Liu, 2019] Liu, Yanbin, et al. "Learning to propagate labels: Transductive propagation network for few-shot learning," ICLR 2019.

Proposed Model

Transductive Propagation Networks

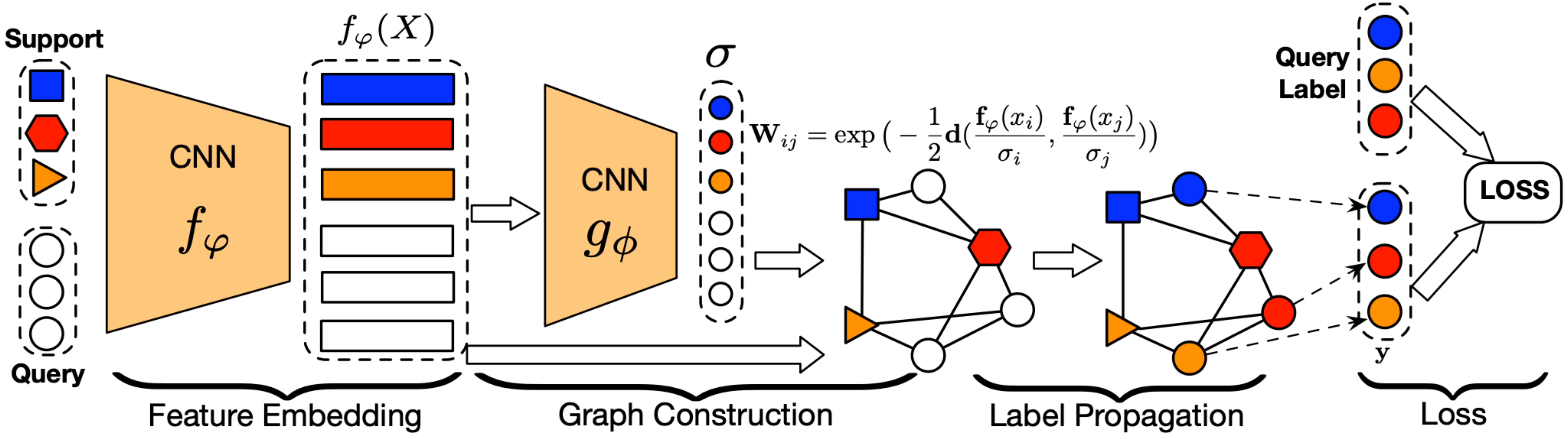


Taken from [Liu, 2019]

[Liu, 2019] Liu, Yanbin, et al. "Learning to propagate labels: Transductive propagation network for few-shot learning," ICLR 2019.

Proposed Model

Transductive Propagation Networks

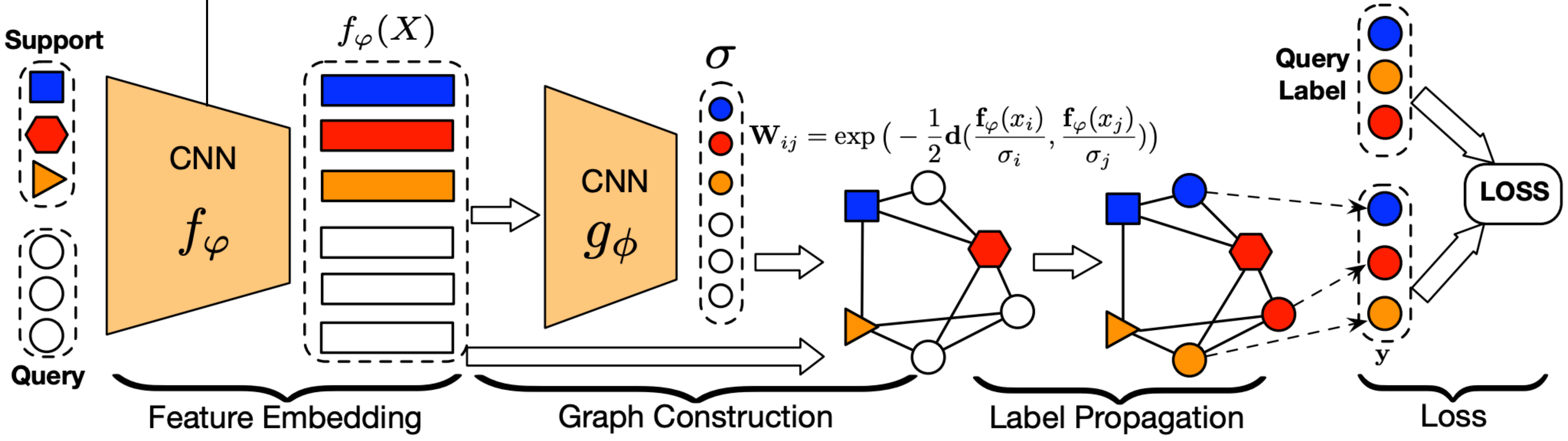


Taken from [Liu, 2019]

[Liu, 2019] Liu, Yanbin, et al. "Learning to propagate labels: Transductive propagation network for few-shot learning," ICLR 2019.

Feature Embedding

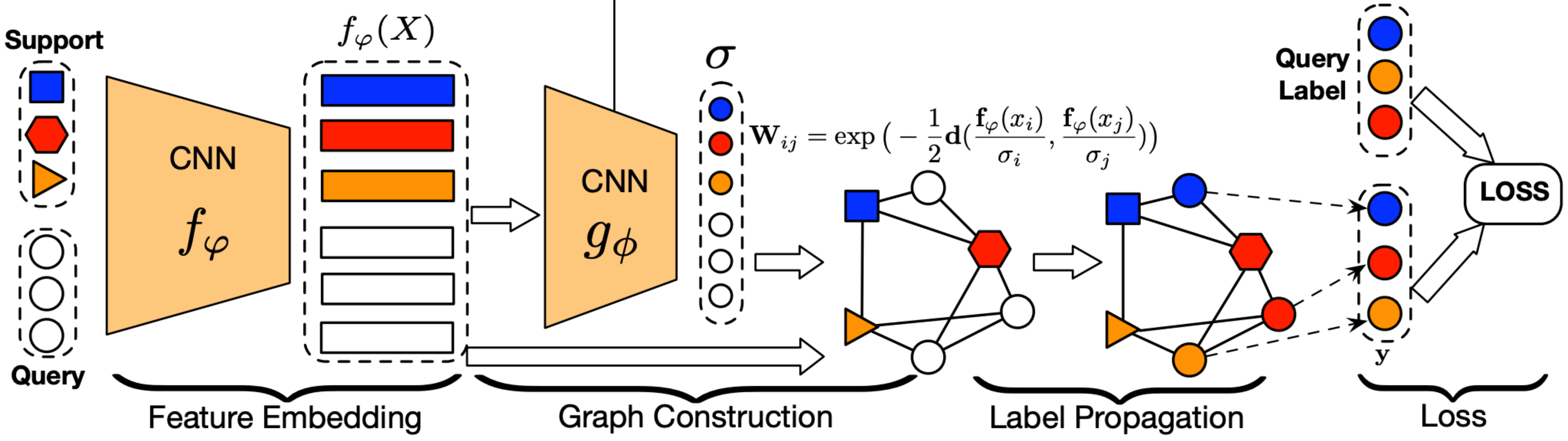
Extract important features of the input images



Same architecture f_φ for fair comparisons (four convolutional blocks)

Graph Construction

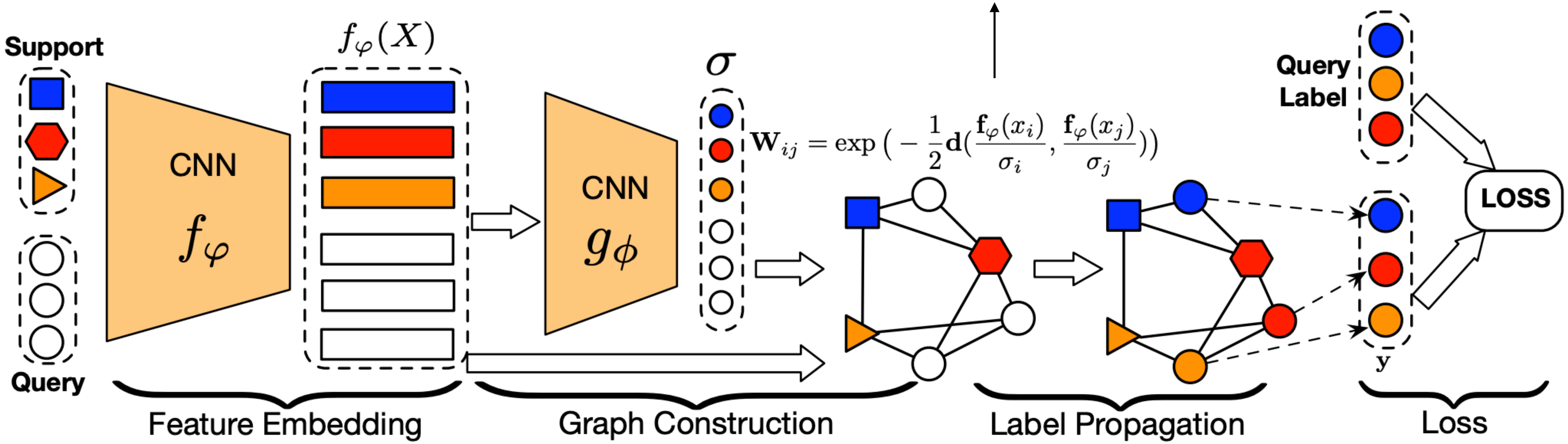
Create **example-wise length-scale parameter** σ



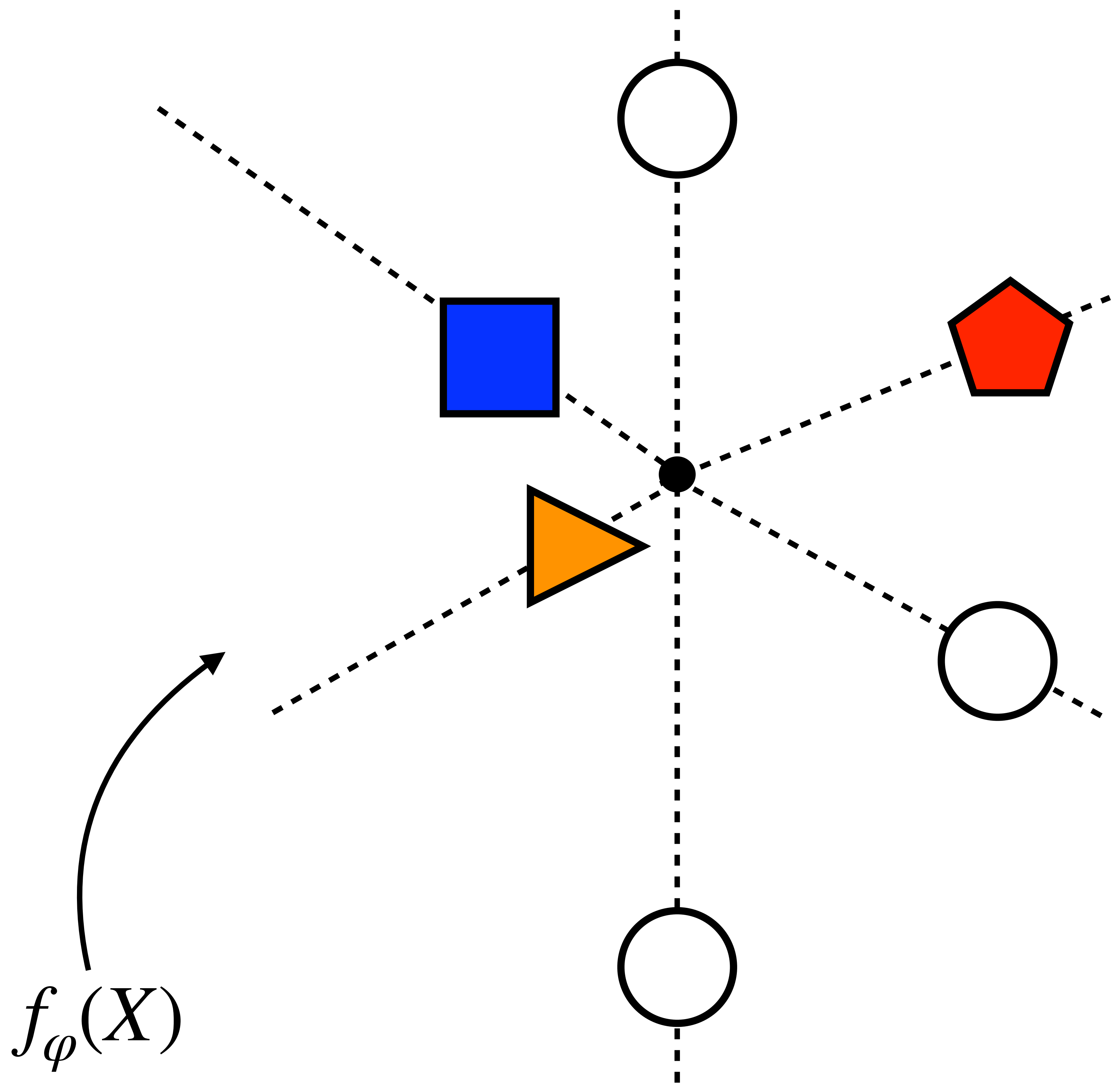
$\sigma = g_\phi(f_\varphi(x_i))$ is used for calculating the similarity function W

Graph Construction

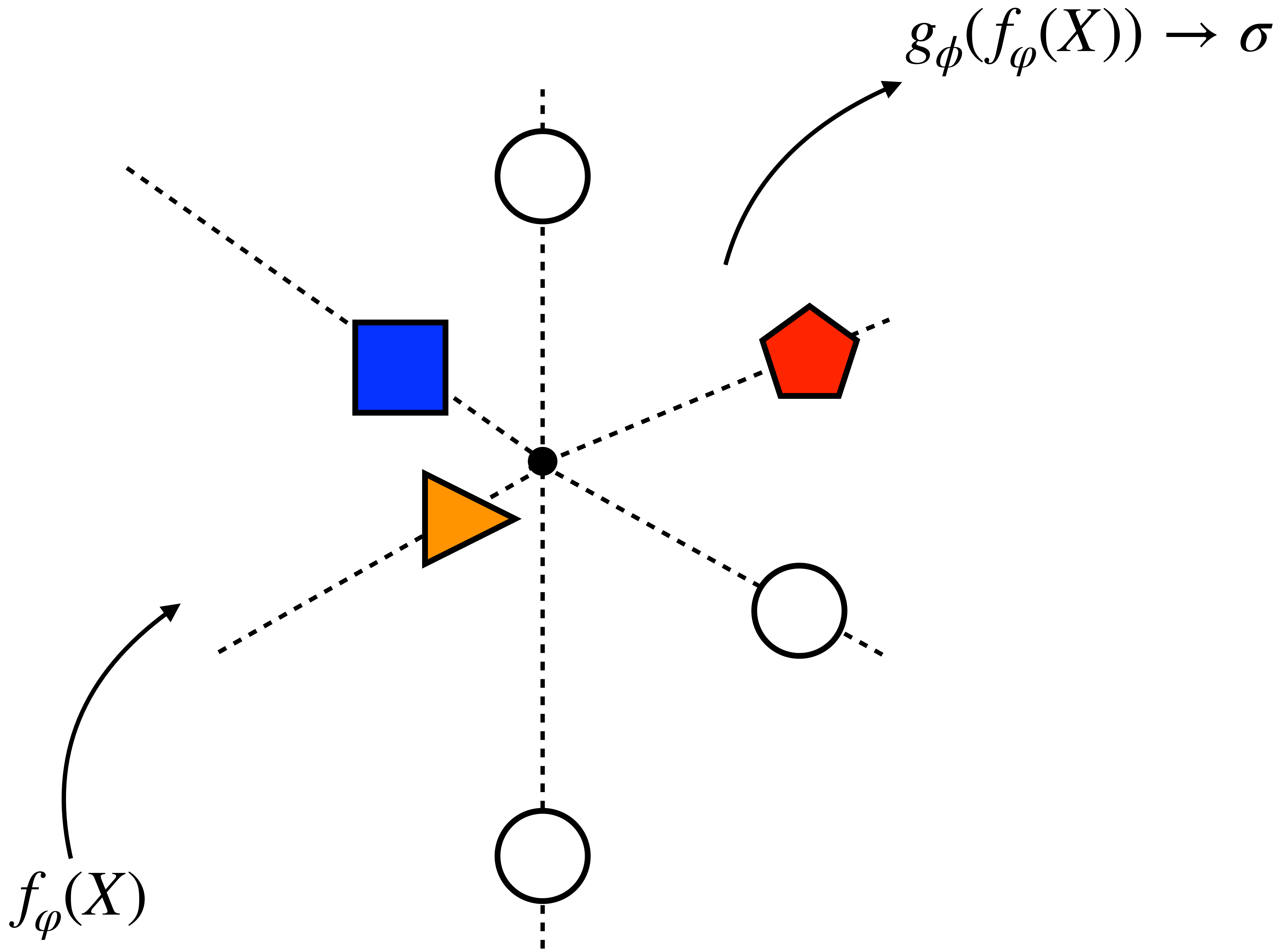
Gaussian similarity function



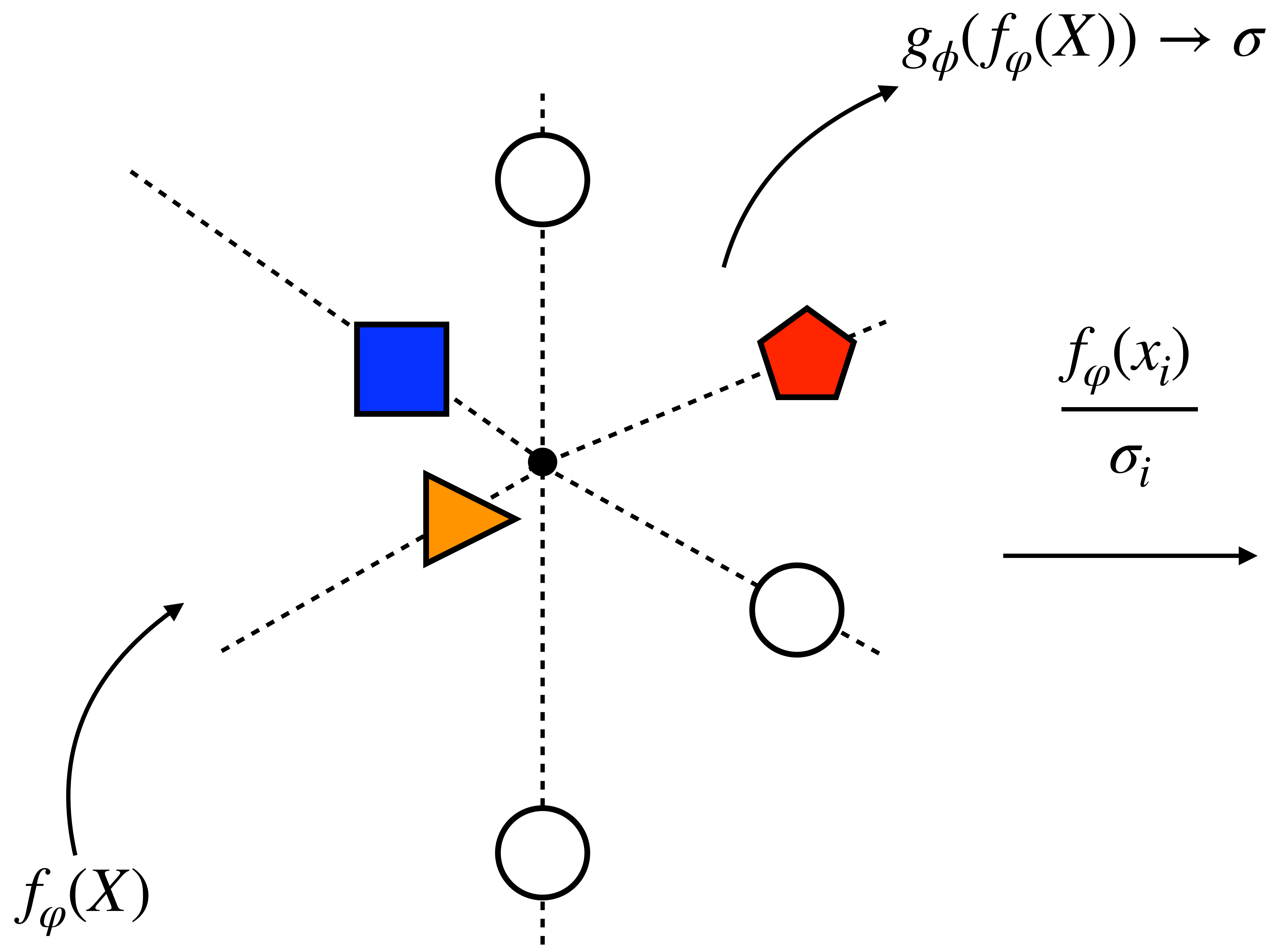
Graph Construction



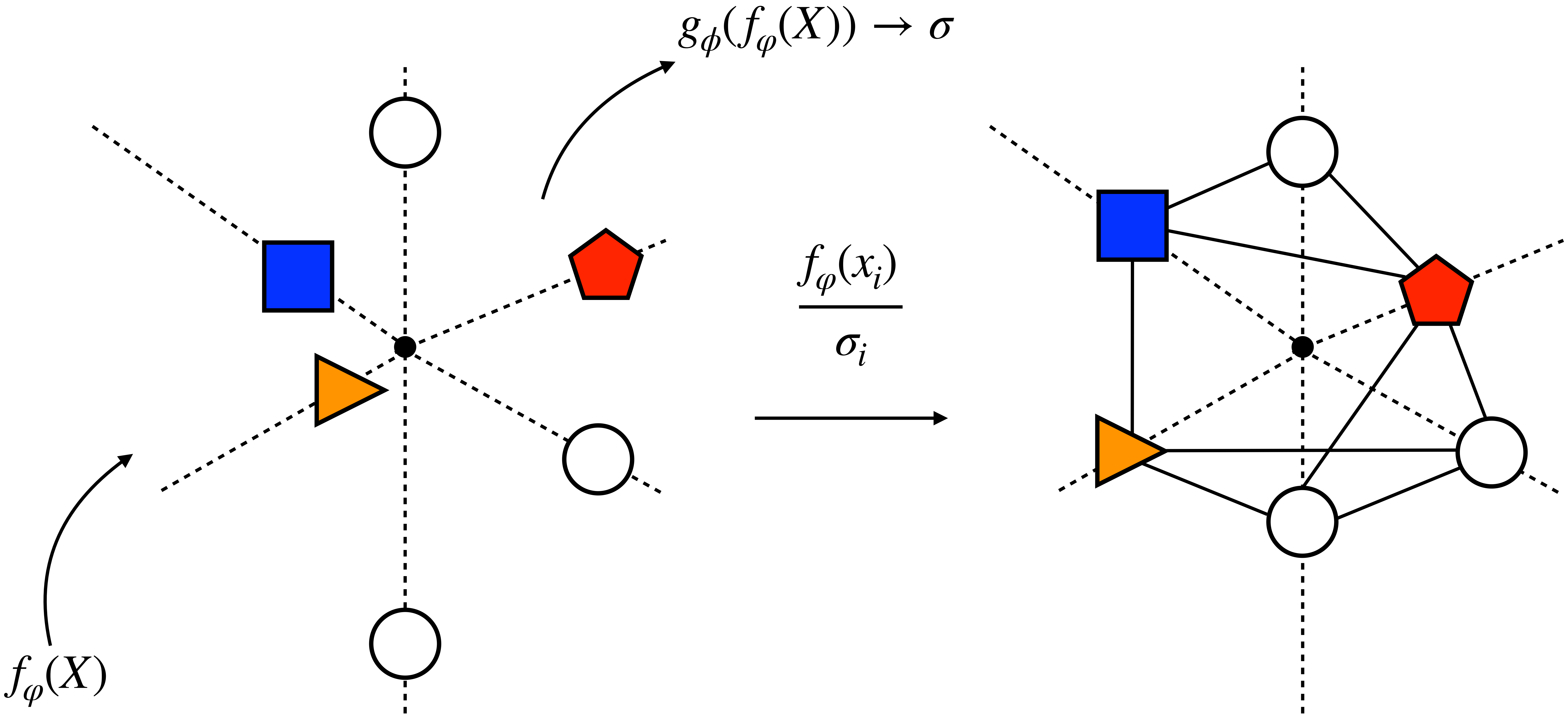
Graph Construction



Graph Construction



Graph Construction



Graph Construction

- A common choice is Gaussian similarity function

$$W_{ij} = \exp(-\frac{d(x_i, x_j)}{2\sigma^2}) \rightarrow W_{ij} = \exp(-\frac{1}{2}d(\frac{f_\varphi(x_i)}{\sigma_i}, \frac{f_\varphi(x_j)}{\sigma_j}))$$

- Calculate the similarity based on the distance,

  but after adjusting with the scaling parameter σ

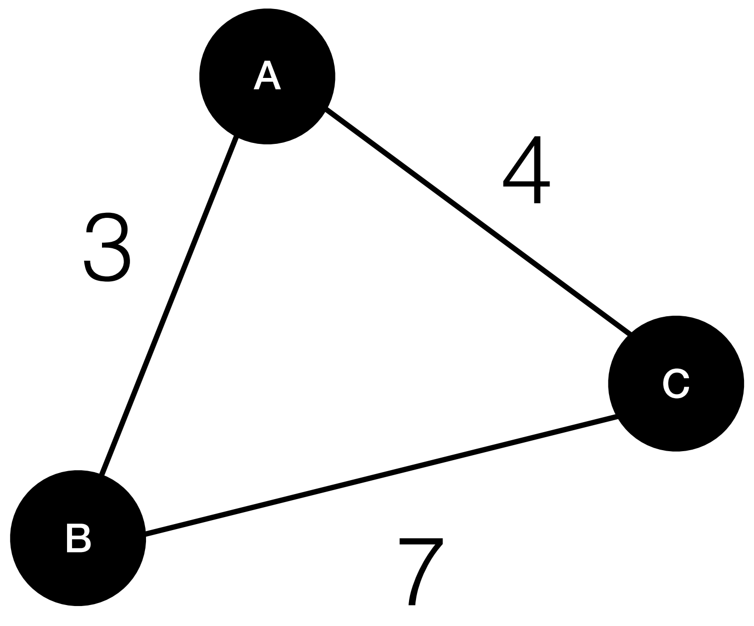
g_ϕ

I will create sigma σ like this.. and adjust the features like this..
Because I learned the general rule for **task-adaptive graph construction**

Graph Construction

- Only keeps the **k-max** values in each row of W (*k-nearest neighbor graph*)
- Apply the **normalized graph Laplacians** on W

$S = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, where D is (i, i)-value to be the sum of the i-th row of W



W

0	3	4
3	0	7
4	7	0

D

7	0	0
0	10	0
0	0	11

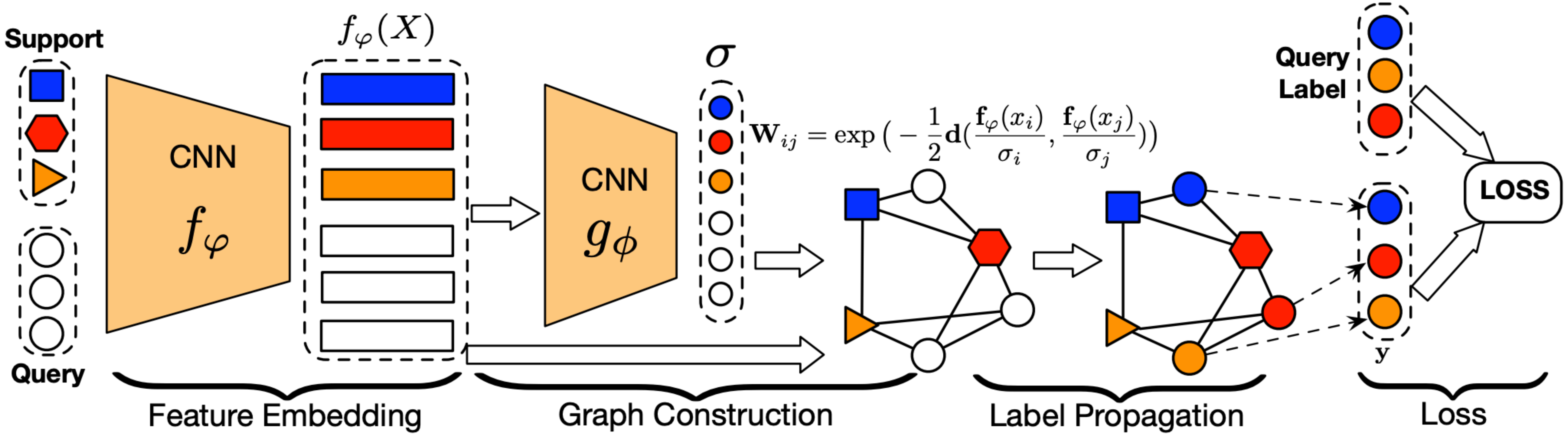
$\longrightarrow$

|eigenvalue| < 1

S

0	0.359	0.456
0.359	0	0.667
0.456	0.667	0

Label Propagation



Label propagation with S

Label Propagation

- No trainable parameters in this stage

$\mathcal{F}$ denote the set of $(N \times K + T) \times N$ matrix and $Y, F_t \in \mathcal{F}$

$\alpha \in (0,1)$ controls the amount of propagated information

$$F_{t+1} = \alpha S F_t + (1 - \alpha) Y$$

Label Propagation

- No trainable parameters in this stage

$\mathscr{F}$ denote the set of $(N \times K + T) \times N$ matrix and $Y, F_t \in \mathscr{F}$

$\alpha \in (0,1)$ controls the amount of propagated information

$$F_{t+1} = \alpha S F_t + (1 - \alpha) Y$$

SF_t

0	0.47
0.36	0.67
0.47	0

Y

1	0
0	0
0	1

Label Propagation

- No trainable parameters in this stage

$\mathcal{F}$ denote the set of $(N \times K + T) \times N$ matrix and $Y, F_t \in \mathcal{F}$

$\alpha \in (0,1)$ controls the amount of propagated information

$$F_{t+1} = \alpha S F_t + (1 - \alpha) Y$$

Label Propagation

- No trainable parameters in this stage

$\mathcal{F}$ denote the set of $(N \times K + T) \times N$ matrix and $Y, F_t \in \mathcal{F}$

$\alpha \in (0,1)$ controls the amount of propagated information

$$F_{t+1} = \alpha S F_t + (1 - \alpha) Y$$

F_t converges $\rightarrow F^* = (1 - \alpha)(I - \alpha S)^{-1} Y$ **closed form (no iteration)*

Label Propagation

- No trainable parameters in this stage

$\mathcal{F}$ denote the set of $(N \times K + T) \times N$ matrix and $Y, F_t \in \mathcal{F}$

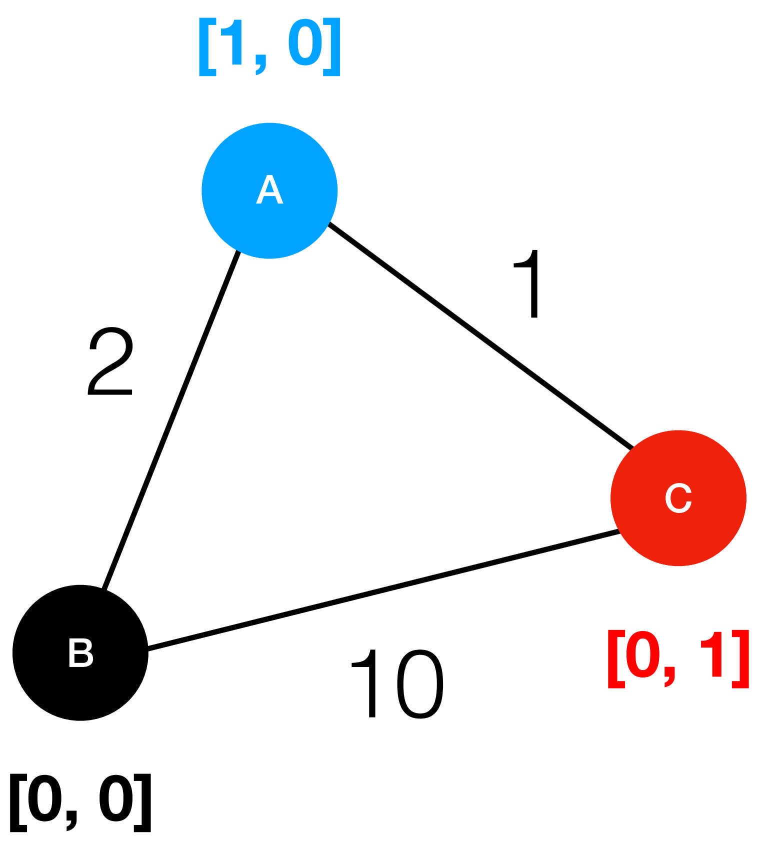
$\alpha \in (0,1)$ controls the amount of propagated information

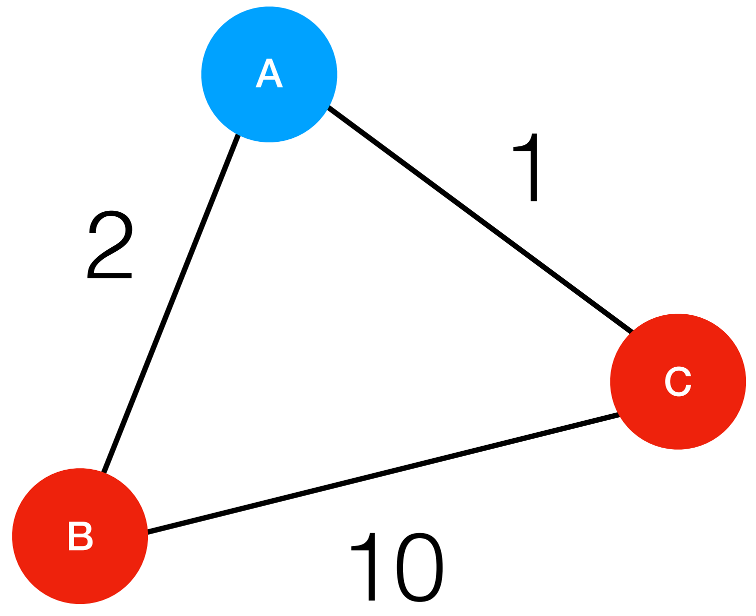
$$F_{t+1} = \alpha S F_t + (1 - \alpha) Y$$

F_t converges $\rightarrow F^* = (1 - \alpha)(I - \alpha S)^{-1} Y$ **closed form (no iteration)*

For classification $\rightarrow F^* = (I - \alpha S)^{-1} Y$

Label Propagation



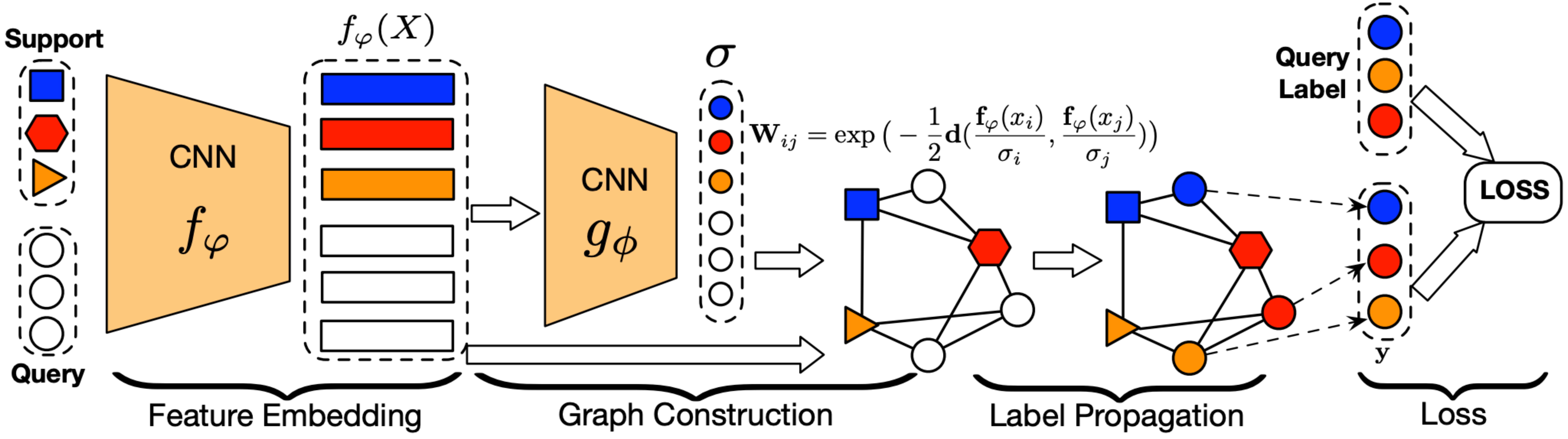


$$S \begin{bmatrix} 0 & 0.359 & 0.456 \\ 0.359 & 0 & 0.667 \\ 0.456 & 0.667 & 0 \end{bmatrix} \quad Y \begin{bmatrix} 1 & 0 \\ \boxed{0 \quad 0} \\ 0 & 1 \end{bmatrix} \rightarrow F^* = (I - \alpha S)^{-1} Y \quad F^* \begin{bmatrix} - & - \\ 22.9 & \boxed{44.0} \\ - & - \end{bmatrix}$$

When $\alpha = 0.99$

Loss

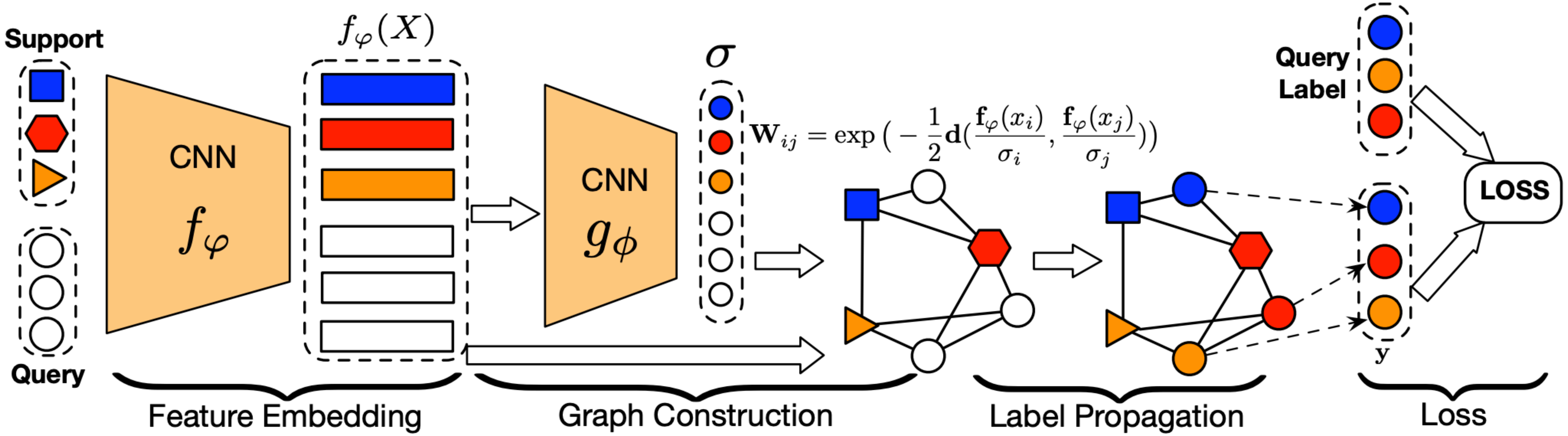
$$J(\varphi, \phi) = \Sigma_{i=1}^{N\times K+T}\Sigma_{j=1}^{N} - \mathbb{I}(y_i == j)log(P(\tilde{y}_i = j \,|\, \mathrm{x}_i))$$



Compute cross-entropy loss between F^* and ground-truth labels from $S \cup Q$

Proposed Model

Transductive Propagation Networks



Contribution

Main contribution

1. **First to model transduction inference** explicitly in few-shot learning
2. In transduction inference, propose to *learn to propagate labels* between data instances **for unseen classes via episodic meta-learning**
3. TPN outperforms the **state-of-the-art** method on both benchmark dataset (miniImageNet and tieredImageNet)

Contribution

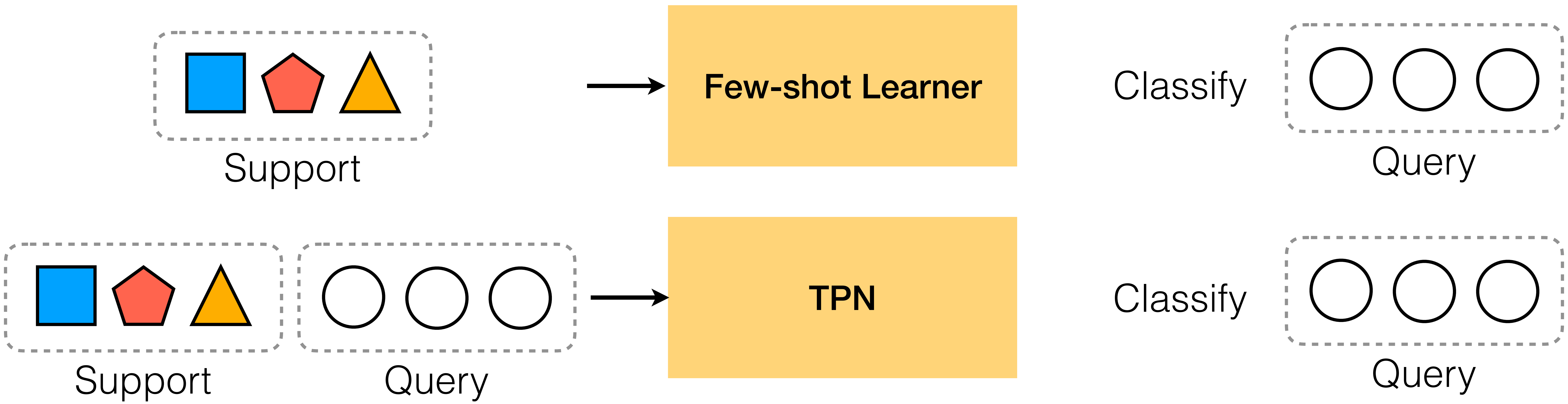
Inductive vs. Transductive

- **Induction** is reasoning from observed *training cases to general rules*, which are then applied to the test cases.
- **Transduction** is reasoning from observed, *specific(training) cases to specific(test) cases*.

Contribution

Inductive vs. Transductive

- Example) 5-way 5-shot, T = 75



75 more examples for inference!

Experiment

- **miniImageNet**: 100 classes, each class containing 600 examples
- **tieredImageNet**: 600 classes, each class containing 1281(avg.) examples

- Transduction - **No**: Inference of query sample is performed **individually**
- Transduction - **Yes**: Inference of query sample is performed **at once** *(TPN)*
- Transduction - **BN**: query samples information is **shared using BN**

Experiment

Table 1: Few-shot classification accuracies on *mini*ImageNet. All results are averaged over 600 test episodes. Top results are highlighted.

Model	Transduction	5-way Acc 1-shot	5-way Acc 5-shot	10-way Acc 1-shot	10-way Acc 5-shot
MAML (Finn *et al.*, 2017)	BN	48.70	63.11	31.27	46.92
MAML+Transduction	Yes	50.83	66.19	31.83	48.23
Reptile (Nichol *et al.*, 2018)	No	47.07	62.74	31.10	44.66
Reptile + BN (Nichol *et al.*, 2018)	BN	49.97	65.99	32.00	47.60
PROTO NET (Snell *et al.*, 2017)	No	46.14	65.77	32.88	49.29
PROTO NET (Higher Way) (Snell *et al.*, 2017)	No	49.42	68.20	34.61	50.09
RELATION NET (Sung *et al.*, 2018)	BN	51.38	67.07	34.86	47.94
Label Propagation	Yes	52.31	68.18	35.23	51.24
TPN	Yes	**53.75**	**69.43**	**36.62**	**52.32**
TPN (Higher Shot)	Yes	**55.51**	**69.86**	**38.44**	**52.77**

Experiment

Table 2: Few-shot classification accuracies on *tiered*ImageNet. All results are averaged over 600 test episodes. Top results are highlighted.

Model	Transduction	5-way Acc 1-shot	5-way Acc 5-shot	10-way Acc 1-shot	10-way Acc 5-shot
MAML (Finn *et al.*, 2017)	BN	51.67	70.30	34.44	53.32
MAML + Transduction	Yes	53.23	70.83	34.78	54.67
Reptile (Nichol *et al.*, 2018)	No	48.97	66.47	33.67	48.04
Reptile + BN (Nichol *et al.*, 2018)	BN	52.36	71.03	35.32	51.98
PROTO NET (Snell *et al.*, 2017)	No	48.58	69.57	37.35	57.83
PROTO NET (Higher Way) (Snell *et al.*, 2017)	No	53.31	72.69	38.62	58.32
RELATION NET (Sung *et al.*, 2018)	BN	54.48	71.31	36.32	58.05
Label Propagation	Yes	55.23	70.43	39.39	57.89
TPN	Yes	**57.53**	**72.85**	**40.93**	**59.17**
TPN (Higher Shot)	Yes	**59.91**	**73.30**	**44.80**	**59.44**

Experiment

Table 3: Semi-supervised comparison on *mini*ImageNet.

Model	1-shot	5-shot	1-shot w/D	5-shot w/D
Soft k-Means (Ren *et al.*, 2018)	50.09	64.59	48.70	63.55
Soft k-Means+Cluster (Ren *et al.*, 2018)	49.03	63.08	48.86	61.27
Masked Soft k-Means (Ren *et al.*, 2018)	50.41	64.39	49.04	62.96
TPN-semi	**52.78**	**66.42**	**50.43**	**64.95**

Table 4: Semi-supervised comparison on *tiered*ImageNet.

Model	1-shot	5-shot	1-shot w/D	5-shot w/D
Soft k-Means (Ren *et al.*, 2018)	51.52	70.25	49.88	68.32
Soft k-Means+Cluster (Ren *et al.*, 2018)	51.85	69.42	51.36	67.56
Masked Soft k-Means (Ren *et al.*, 2018)	52.39	69.88	51.38	69.08
TPN-semi	**55.74**	**71.01**	**53.45**	**69.93**

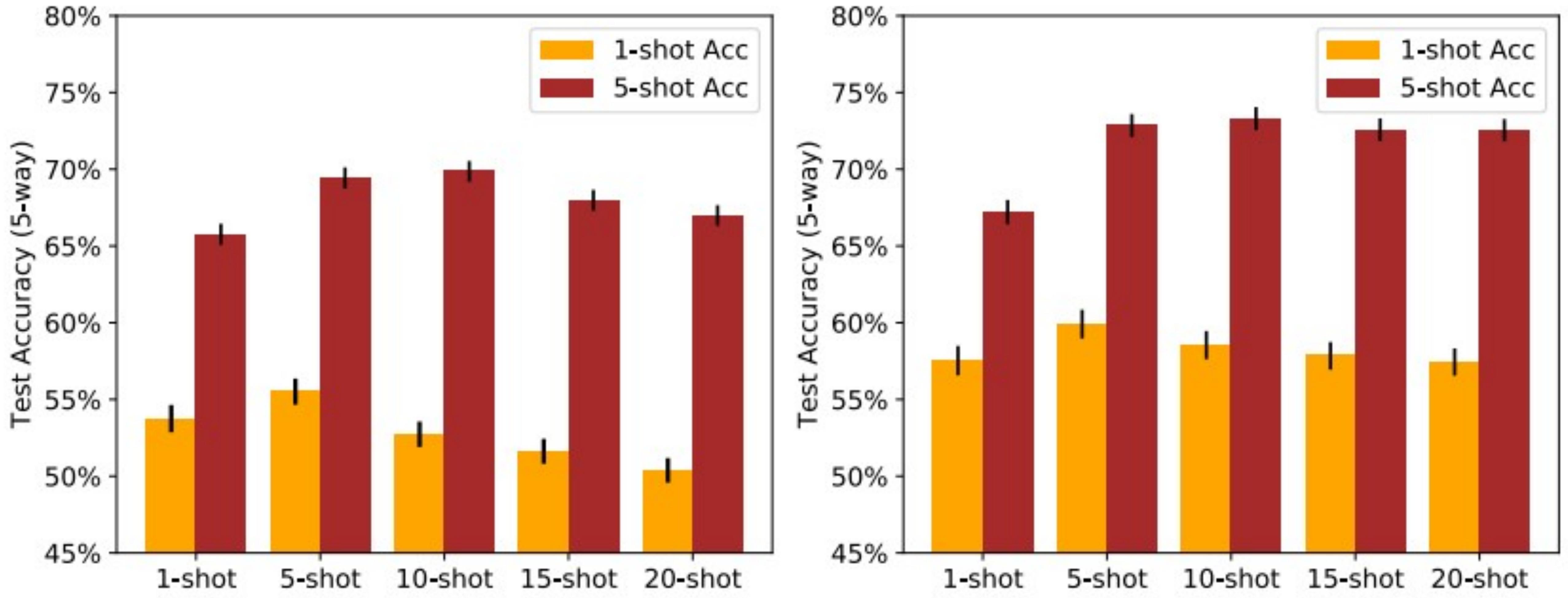
Conclusion

- Transductive + Few-shot
  - **Applicable to Few-shot Leaning**
  - Propose the possibility of follow-up research
- σ is not hyper-parameter
  - Key-point: **Task-adaptive scaling parameter**
  - Largely ameliorating the uneven data distribution problem
- The state-of-the-art performance

Thank you 🙌

Appendix

- Higher shot / Query number



(a) *mini*Imagenet (b) *tiered*Imagenet

Table 5: Accuracy with various query numbers

	*mini*ImageNet 1-shot					
	5	10	15	20	25	30
Train=15	52.29	52.95	53.75	53.92	54.57	54.47
Test=15	53.53	53.72	53.75	52.79	52.84	52.47
Train=Test	51.94	53.47	53.75	54.00	53.59	53.32
	*mini*ImageNet 5-shot					
	5	10	15	20	25	30
Train=15	66.97	69.30	69.43	69.92	70.54	70.36
Test=15	68.50	68.85	69.43	69.26	69.12	68.89
Train=Test	67.55	69.22	69.43	69.85	70.11	69.94

Appendix

- Loss
  - Softmax using F^* and negative log-likelihood cross-entropy

$$P(\tilde{y}_i = j|\mathrm{x}_i) = \frac{\exp(F^*_{ij})}{\Sigma_{j=1}^{N} \exp(F^*_{ij})}$$

$$J(\varphi, \phi) = \Sigma_{i=1}^{N\times K+T} \Sigma_{j=1}^{N} - \mathbb{I}(y_i == j) log(P(\tilde{y}_i = j)|\mathrm{x}_i)$$

Appendix

- $|$eigenvalue of S$| < 1$

1. Similar matrix in Linear algebra

$B = P^{-1}AP \iff PBP^{-1} = A$. If $Av = \lambda v$, then $PBP^{-1}v = \lambda v \implies BP^{-1}v = \lambda P^{-1}v$

2. S is similar with markov(stochastic) matrix

$$S \text{ is similiar with } A = D^{-1}W$$
$$\text{Meaning of similarity: } B = P^{-1}AP$$
$$A = D^{-1}W = D^{-1/2}SD^{1/2}$$

Appendix

- Convergence of F_t

$$F(0) = Y, \text{ and } F(t+1) = \alpha S F(t) + (1-\alpha) Y$$
$$F(t) = (\alpha S)^{t-1} Y + (1-\alpha) \Sigma_{i=0}^{t-1} (\alpha S)^i Y$$

If $0 < \alpha < 1$ and |eigenvalue of S| < 1,

$$\lim_{t \to \infty} (\alpha S)^{t-1} = 0, \text{ and } \lim_{t \to \infty} \Sigma_{i=0}^{t-1} (\alpha S)^i = (1 - \alpha S)^{-1}$$

$$F^* = lim_{t \to \infty} F_t = (1-\alpha)(1-\alpha S)^{-1} Y$$

Appendix

- Transductive setting
  - **MAML**: All support & query info were used for calculating BN statistics
  - **MAML+Transduction**: Add *transduction regularization* term
  - **Reptile**: All support and only one query info were used for calculating BN statistics
  - **Reptile+BN**: All support & query info were used for calculating BN statistics
  - **ProtoNet**: All support and only one query info were used for calculating BN statistics

Appendix

- Normalized graph Laplacian

$$
\begin{aligned}
L &= D - A \\
D^{-\frac{1}{2}} L D^{-\frac{1}{2}} &= D^{-\frac{1}{2}} (D - A) D^{-\frac{1}{2}} \\
&= D^{-\frac{1}{2}} (D^{\frac{1}{2}} - A D^{-\frac{1}{2}}) \\
&= I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}
\end{aligned}
$$

Appendix

- MAML+Transduction

$$\mathcal{J}(\theta) = \sum_{i=1}^{T} \mathbf{y}_i \log \mathbb{P}(\widehat{\mathbf{y}}_i|\mathbf{x}_i) + \sum_{i,j=1}^{N\times K+T} W_{ij}\|\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j\|_2^2$$

- Sigma
  - conv - conv - FC(out 8) - FC(out 1)