Few-shot Learning

[Amirhosein Kazemnejad , Mostafa MohammadAli Ebrahim] / IUST

Problem Definition

- Deep learning successes have required a lot of **labeled** training data

- Constructing such huge datasets, Requires significant amount of work and money.

Problem Definition

- Deep learning successes have required a lot of **labeled** training data

- Constructing such huge datasets, Requires significant amount of work and money.

But:

- Humans learn new concepts using just few examples

Problem Definition

- Deep learning successes have required a lot of **labeled** training data

- Constructing such huge datasets, Requires significant amount of work and money.

But:

- Humans learn new concepts using just few examples

- Prior experience plays a key role in human's learning.

Problem Definition

- Deep learning successes have required a lot of **labeled** training data

- Constructing such huge datasets, Requires significant amount of work and money.

But:

- Humans learn new concepts using just few examples

- Prior experience plays a key role in human's learning.

- Humans know how to learn!

Few-shot learning (FSL) refers to

The training of machine learning algorithms using a very small set of training data (e.x. 1 sample per each class)

How to "few-shot learning" ?

How to "few-shot learning" ?

1. Data-level Approach

How to "few-shot learning" ?

1. Data-level Approach

2. Parameter-level Approach

How to "few-shot learning" ?

1. Data-level Approach

2. Parameter-level Approach

How to "few-shot learning" ?

1. Data Augmentation

2. Meta Learning

What is Meta-learning?

What is Meta-learning?

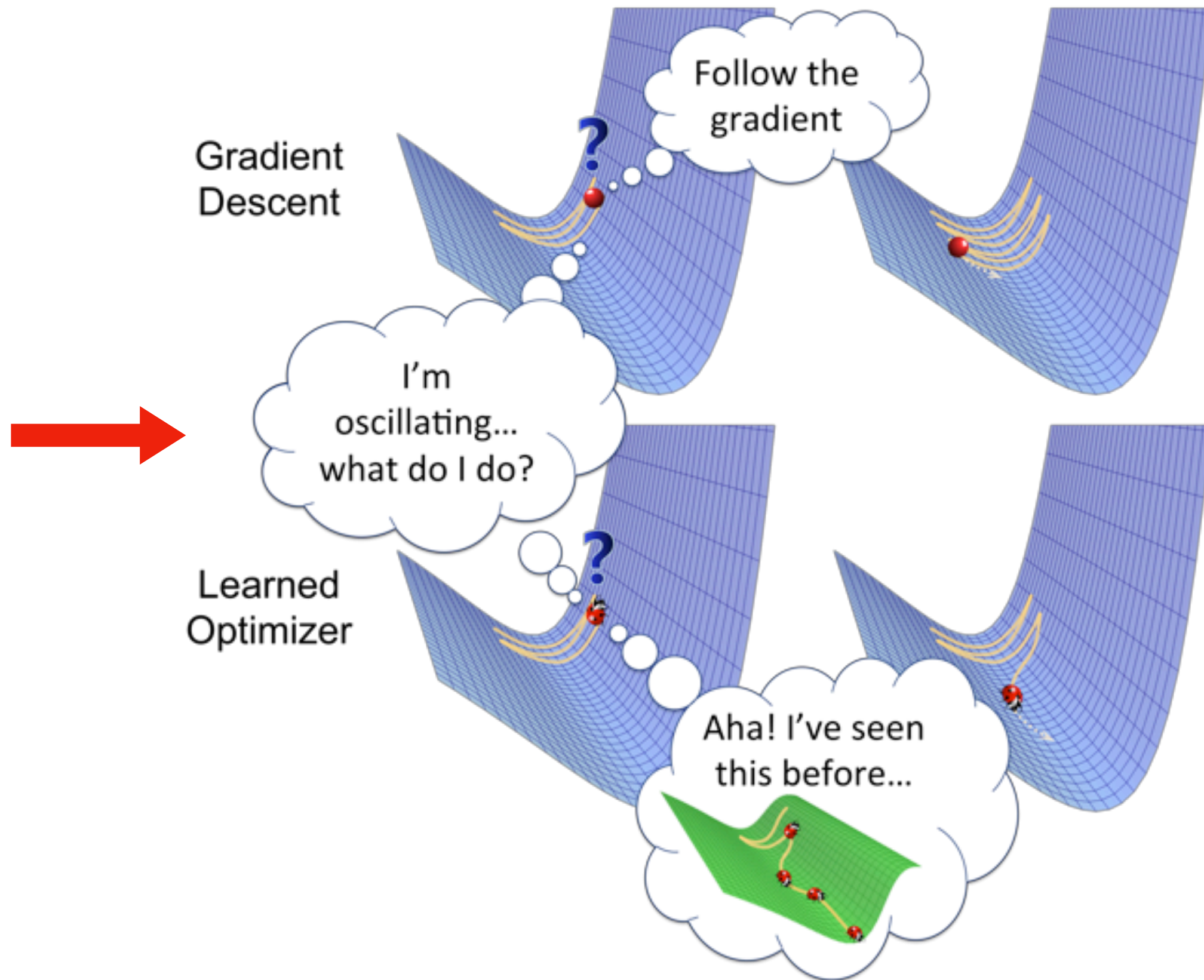- Meta-learning = learning to learn

What is Meta-learning?

- Meta-learning = learning to learn

- If you've learned 100 tasks already, can you figure out how to learn more efficiently?
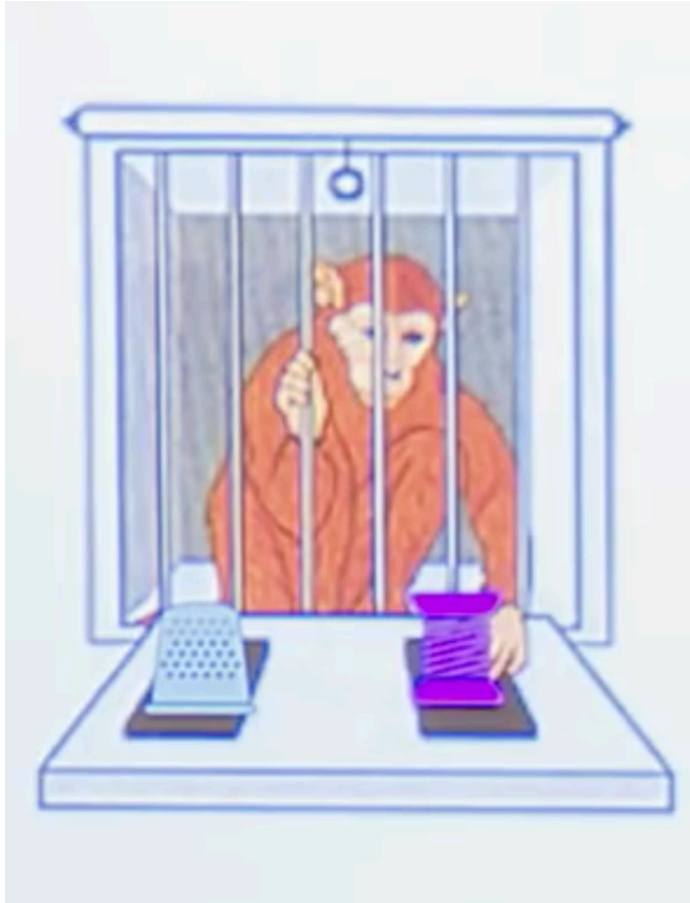
What is Meta-learning?

- Meta-learning = learning to learn

- If you've learned 100 tasks already, can you figure out how to learn more efficiently?

- Prior experience on "learning" helps to learn new tasks much faster!

What is Meta-learning?

- Meta-learning = learning to learn

- If you've learned 100 tasks already, can you figure out how to learn more efficiently?

- Prior experience on "learning" helps to learn new tasks much faster!

- Very close to multi-task learning and transfer-learning

What is Meta-learning?

- Meta-learning = learning to learn

- If you've learned 100 tasks already, can you figure out how to learn more efficiently?

- Prior experience on "learning" helps to learn new tasks much faster!

- Very close to multi-task learning and transfer-learning
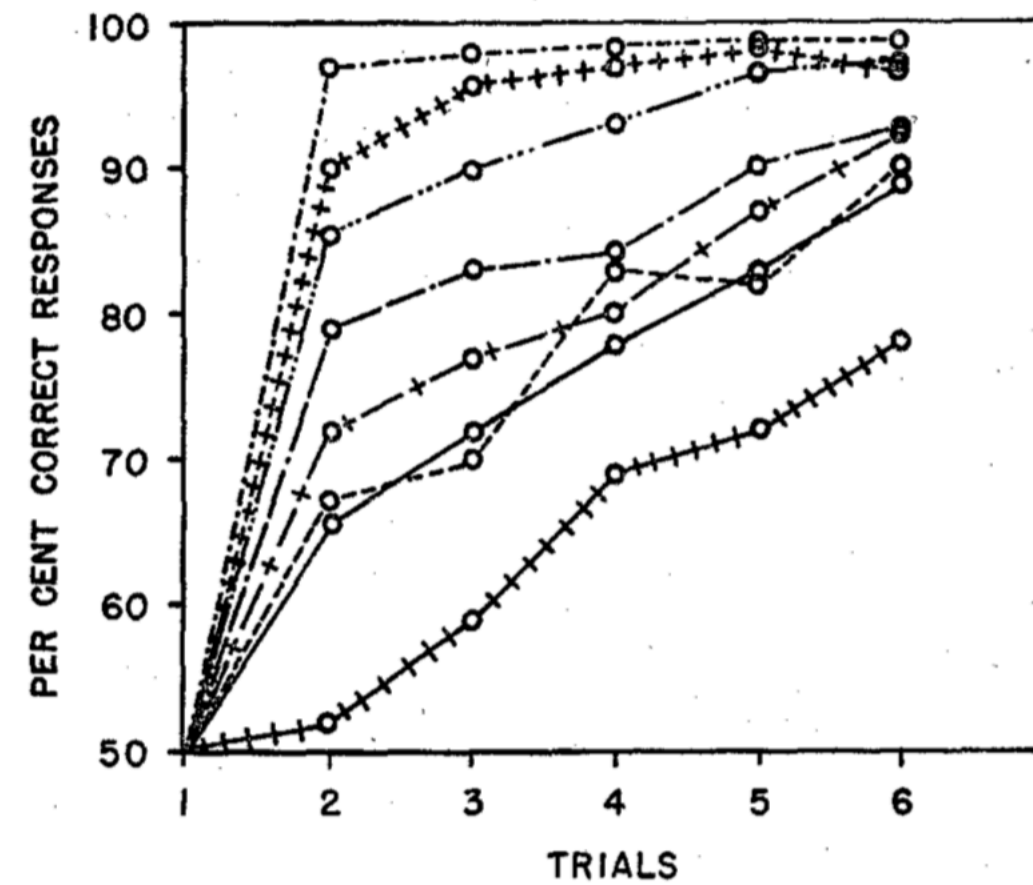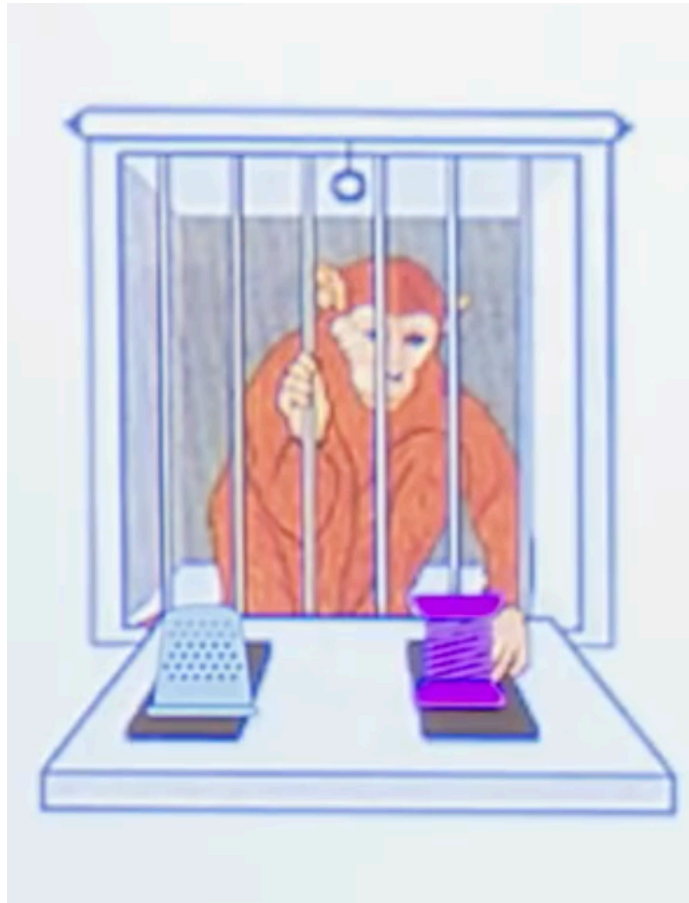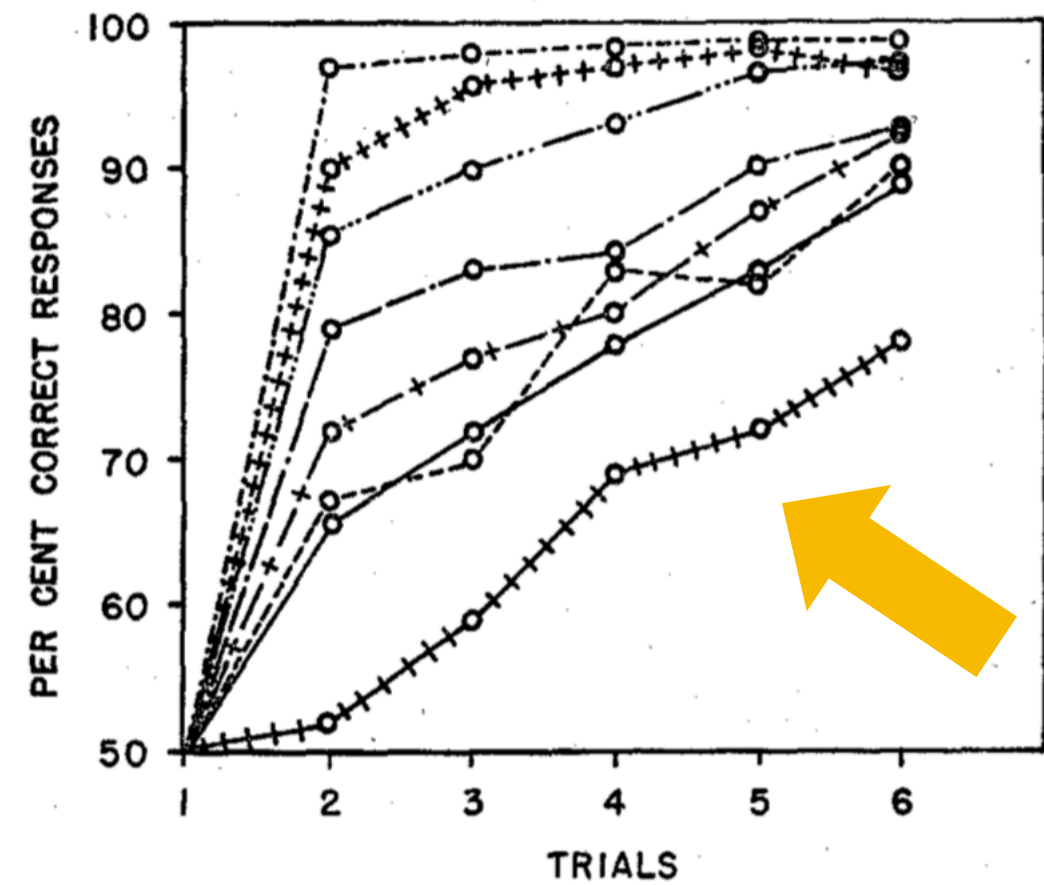
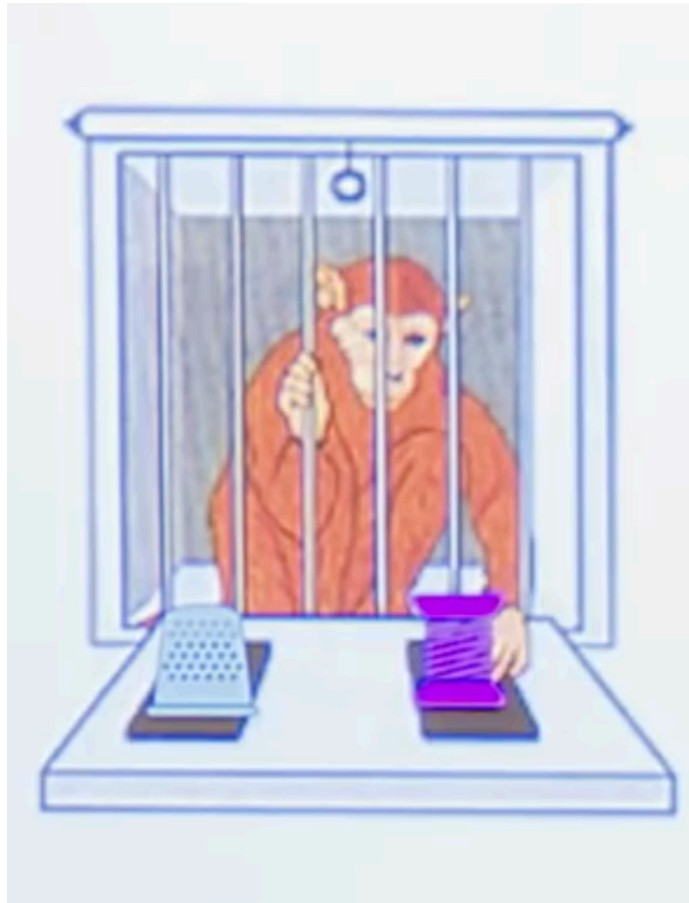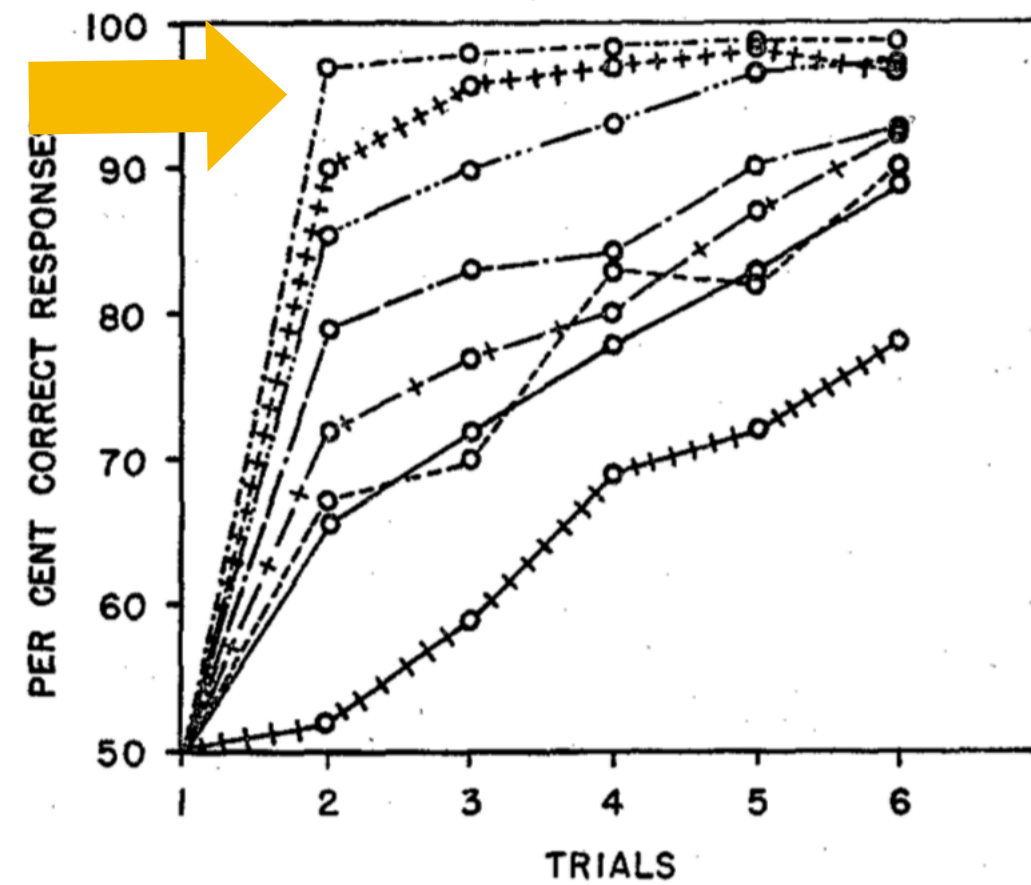- Many Formulations

Harlow Experiment
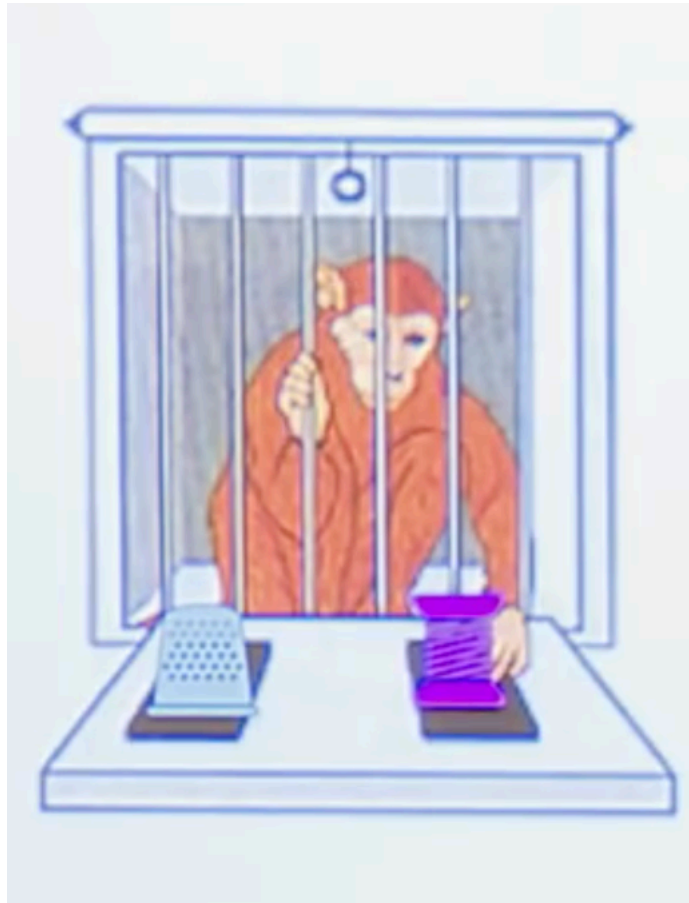
Harlow Experiment

Harlow Experiment

Harlow Experiment

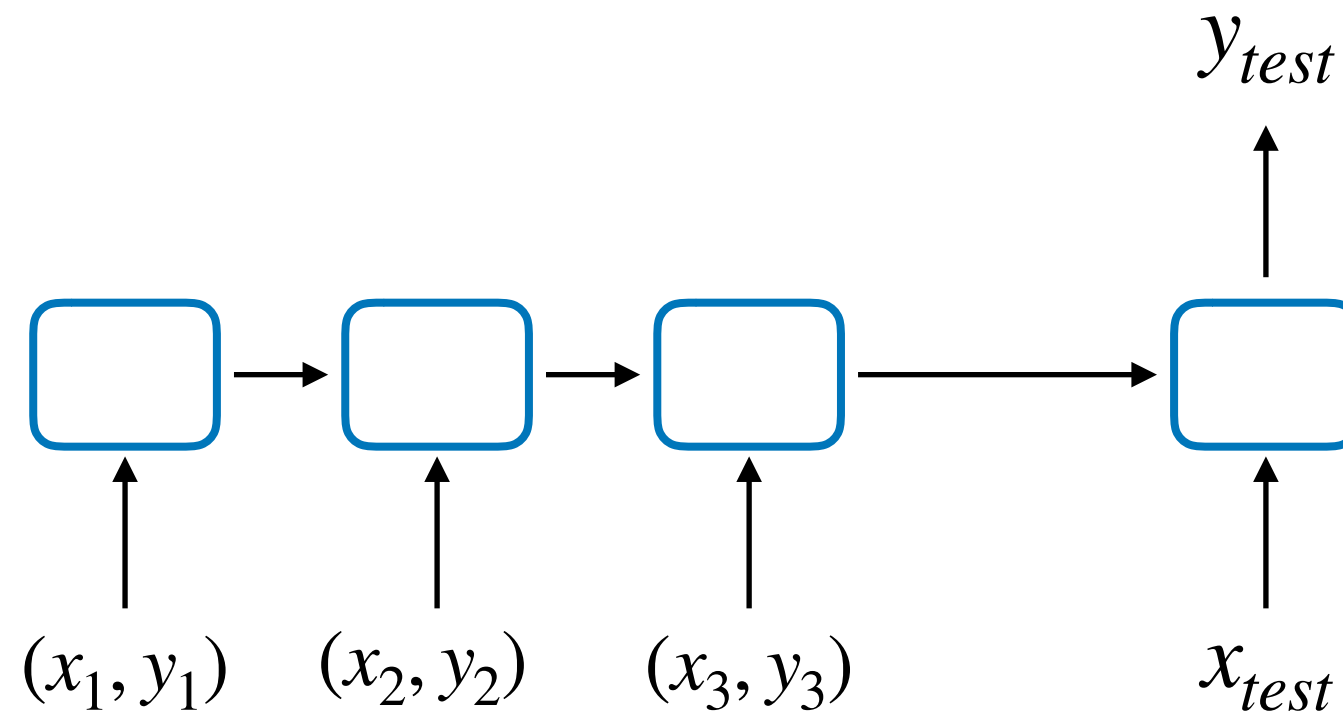# Harlow Experiment

RNN-based Meta-learning
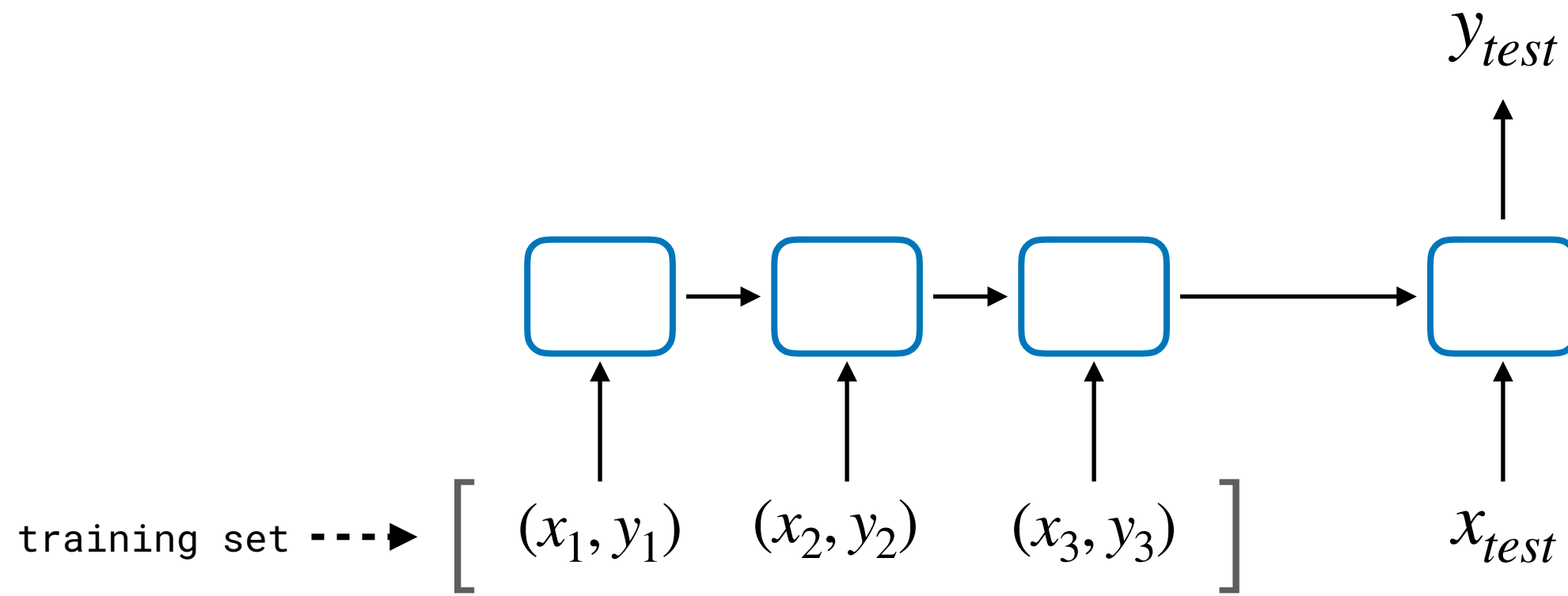
RNN-based Meta-learning

RNN-based Meta-learning

$$y_{test}$$

training set $\dashrightarrow$ $\left[\quad (x_1, y_1) \quad (x_2, y_2) \quad (x_3, y_3) \quad \right]$ $\qquad x_{test}$

RNN-based Meta-learning

y_{test} ◄--- test label

training set ---► $\left[\ (x_1, y_1)\quad (x_2, y_2)\quad (x_3, y_3)\ \right]$ x_{test} ◄--- test input

RNN-based Meta-learning

y_{test} ◀--- test label

training set ---▶ $\Big[\quad (x_1, y_1) \quad (x_2, y_2) \quad (x_3, y_3) \quad \Big]$ x_{test} ◀--- test input

supervised learning: $f(x) \to y$

input (e.g., image)   output (e.g., label)

28

RNN-based Meta-learning

y_{test} ◄--- test label

training set --►$\left[\quad (x_1, y_1) \quad (x_2, y_2) \quad (x_3, y_3) \quad \right]$ x_{test} ◄--- test input
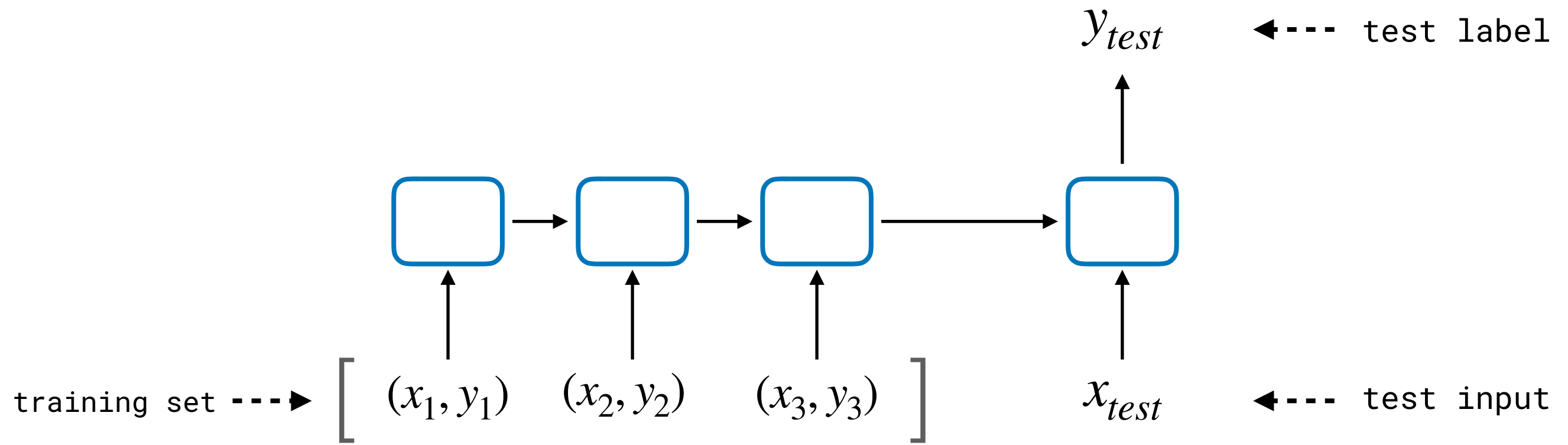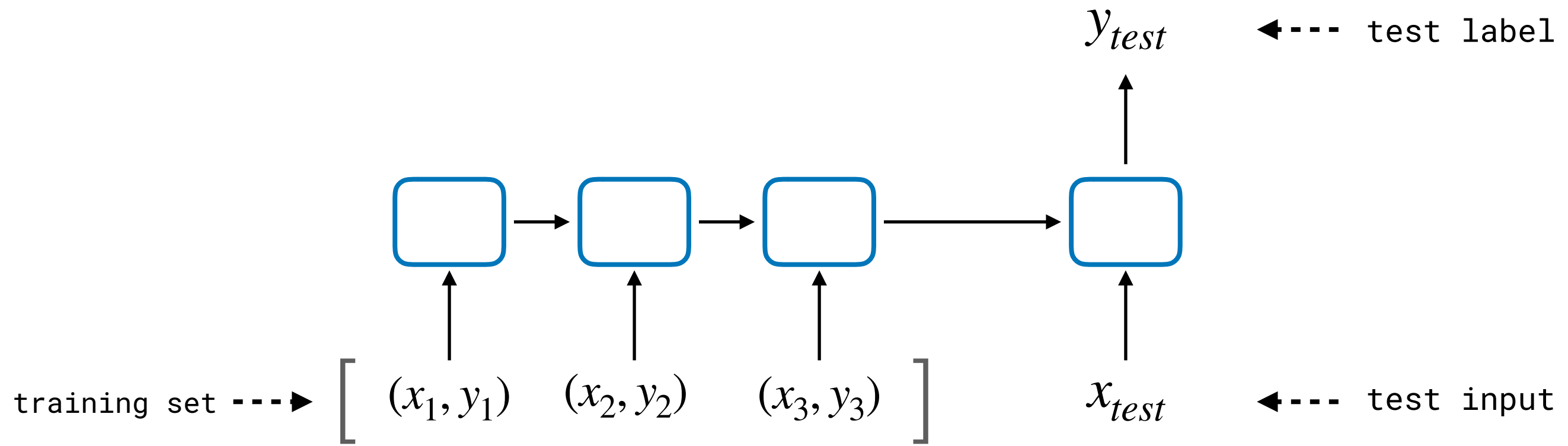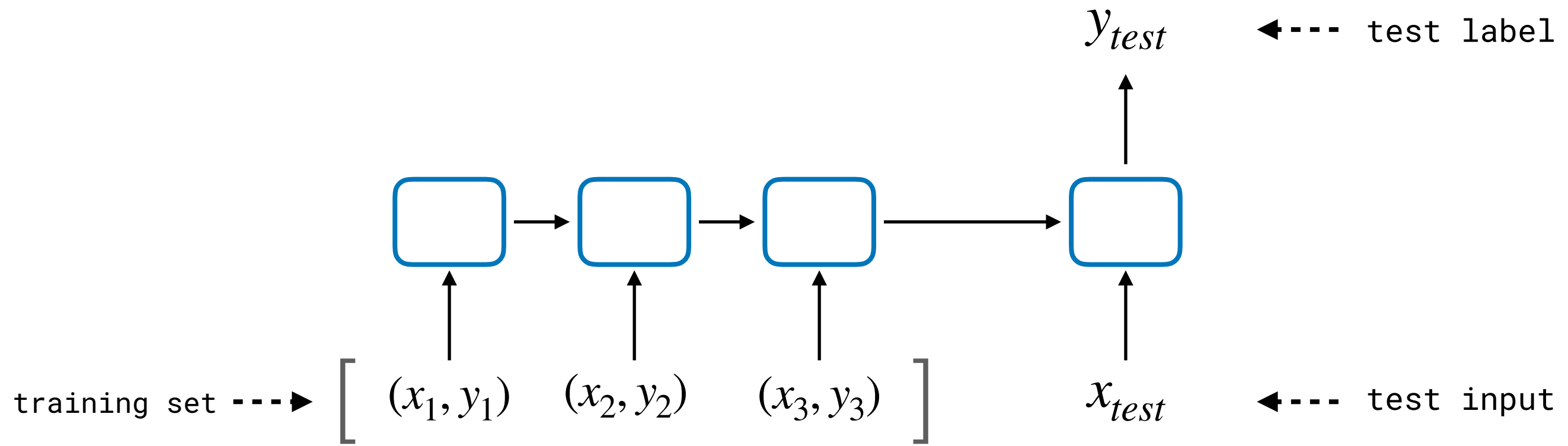
supervised learning: $f(x) \to y$

input (e.g., image)   output (e.g., label)

supervised meta-learning: $f(\mathcal{D}_{\text{train}}, x) \to y$

training set

Model Agnostic Meta-learning (MAML)

Model Agnostic Meta-learning (MAML)

Two main Component:

Model Agnostic Meta-learning (MAML)

Two main Component:

- Meta-learnee, Classifier **(C), f(x, θ) -> y**

Model Agnostic Meta-learning (MAML)

Two main Component:

- Meta-learnee, Classifier **(C), f(x, θ) -> y**

- Meta-learner, Teacher, Learning Procedure **(T)**

Model Agnostic Meta-learning (MAML)

Model Agnostic Meta-learning (MAML)

Model Agnostic Meta-learning (MAML)

Meta-learner $\longrightarrow$ θ

Model Agnostic Meta-learning (MAML)

Model Agnostic Meta-learning (MAML)

classifier → Gradient Decent

$$T_1 = (\mathscr{D}_1^{train}, \mathscr{D}_1^{test})$$

Meta-learner → θ

classifier → Gradient Decent

$$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$$

classifier → Gradient Decent

$$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$$

Model Agnostic Meta-learning (MAML)

$$T_1 = (\mathscr{D}_1^{train}, \mathscr{D}_1^{test})$$

$$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$$

$$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$$

Meta-learner

θ

classifier → Gradient Decent → θ_1^*

classifier → Gradient Decent → θ_2^*

classifier → Gradient Decent → θ_3^*

Model Agnostic Meta-learning (MAML)

Meta-learner → θ

classifier → Gradient Decent → θ_1^* → Calculate loss

$T_1 = (\mathcal{D}_1^{train}, \mathcal{D}_1^{test})$

$\mathcal{D}_1^{test}$

classifier → Gradient Decent → θ_2^* → Calculate loss

$T_2 = (\mathcal{D}_2^{train}, \mathcal{D}_2^{test})$

$\mathcal{D}_2^{test}$

classifier → Gradient Decent → θ_3^* → Calculate loss

$T_2 = (\mathcal{D}_2^{train}, \mathcal{D}_2^{test})$

$\mathcal{D}_3^{test}$

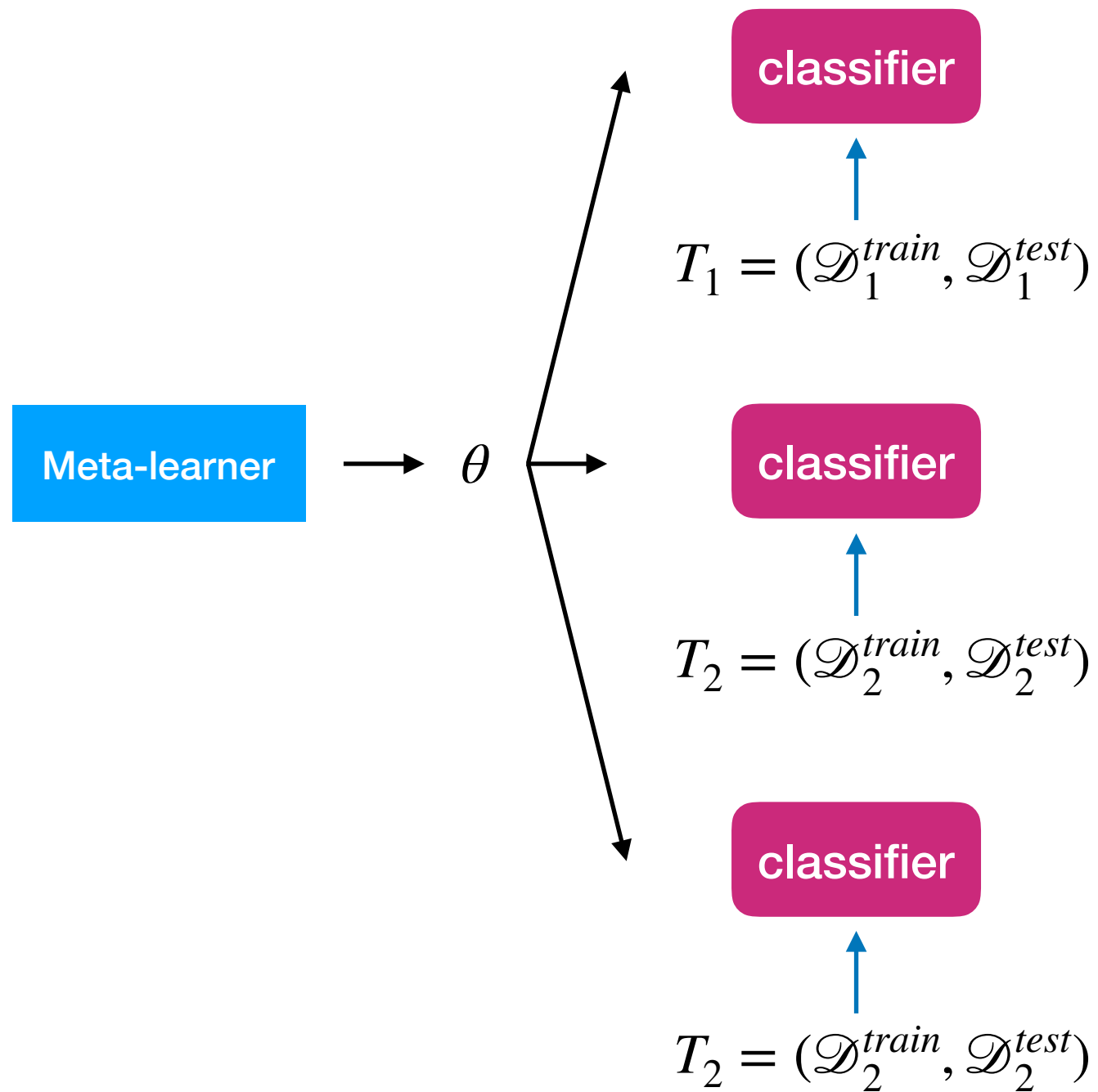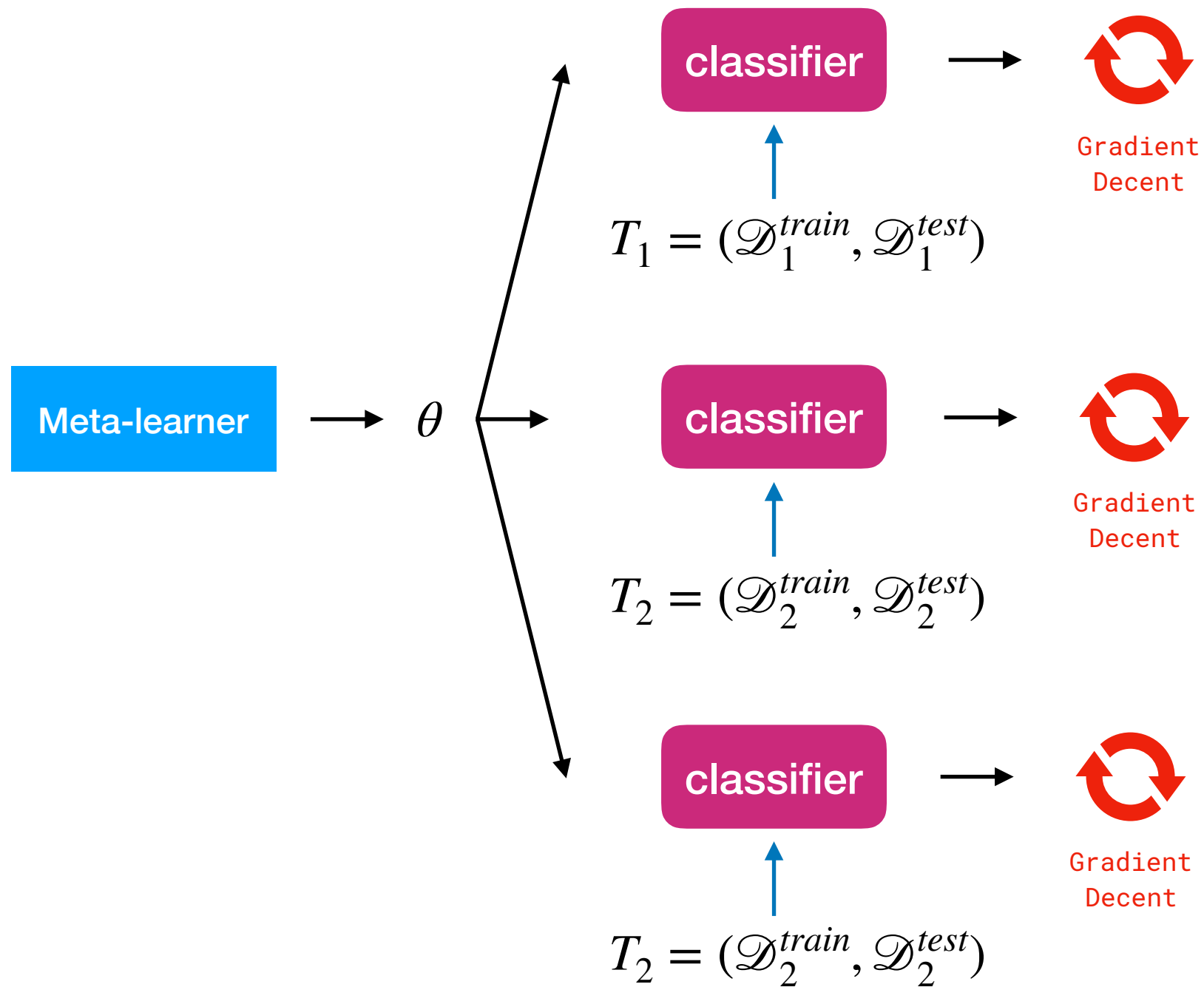# Model Agnostic Meta-learning (MAML)

Model Agnostic Meta-learning (MAML)

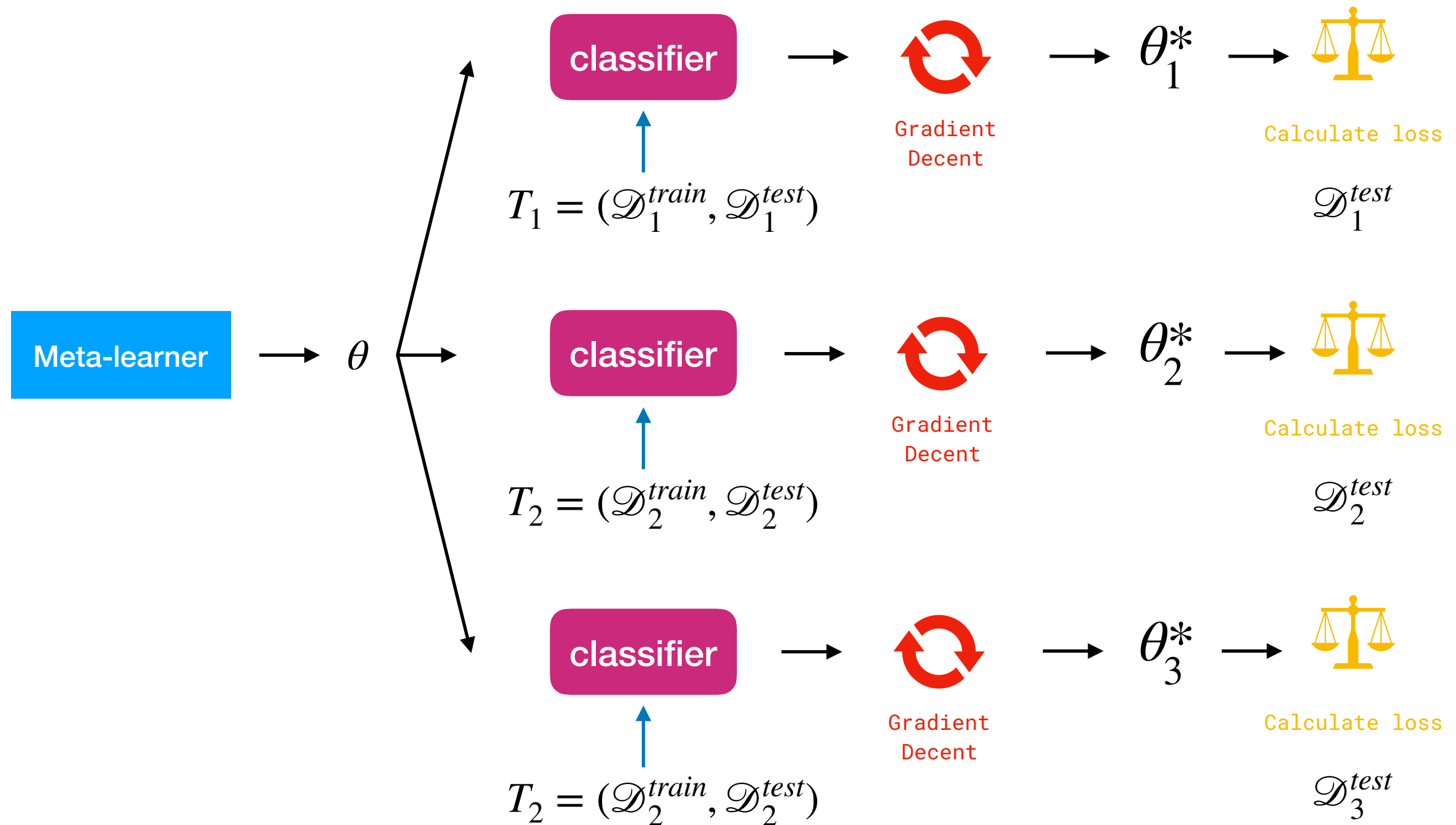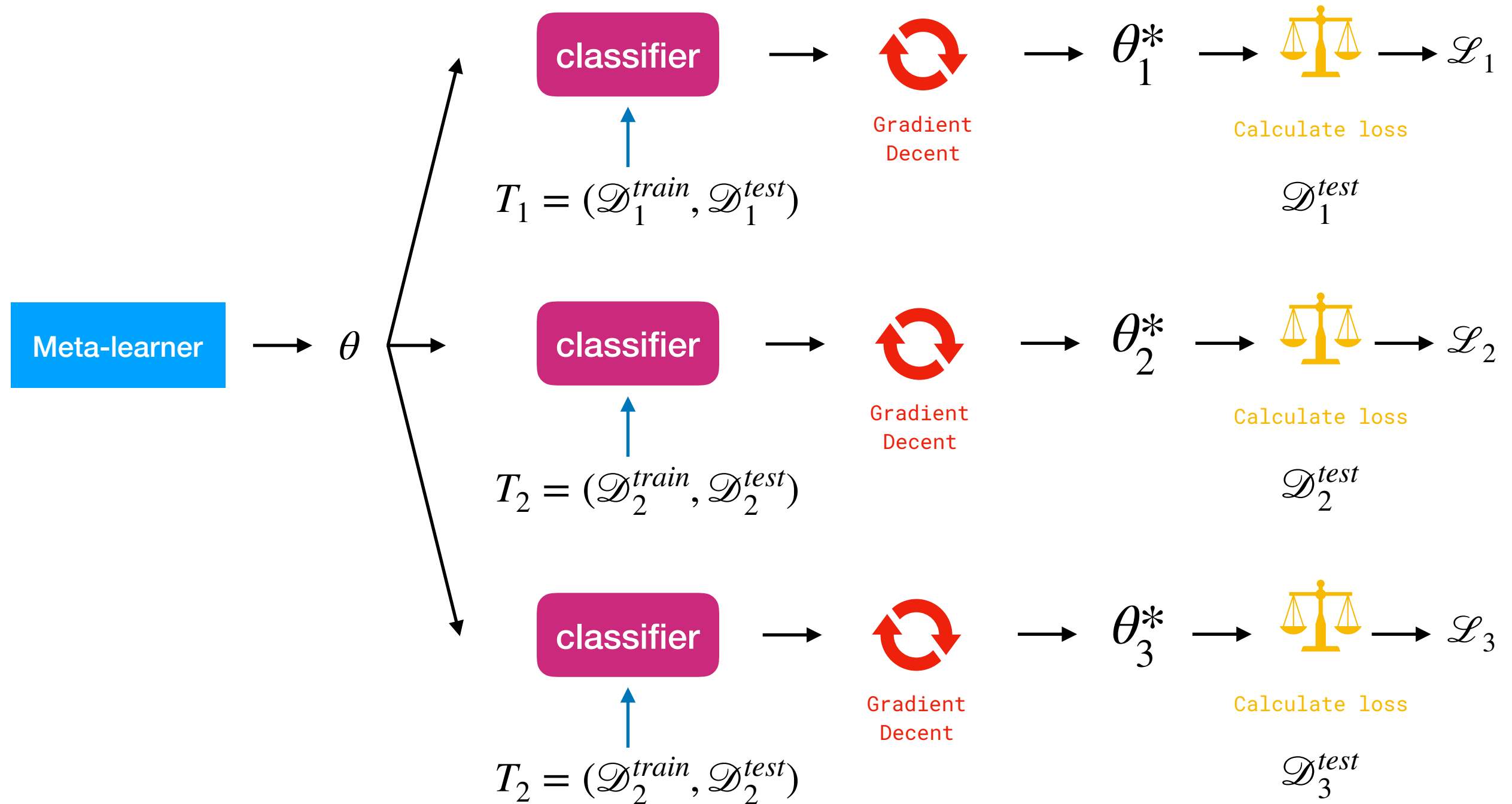classifier → Gradient Decent → θ_1^* → Calculate loss → $\mathscr{L}_1$

$T_1 = (\mathscr{D}_1^{train}, \mathscr{D}_1^{test})$

$\mathscr{D}_1^{test}$

Meta-learner → θ

classifier → Gradient Decent → θ_2^* → Calculate loss → $\mathscr{L}_2$

$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$

$\mathscr{D}_2^{test}$

classifier → Gradient Decent → θ_3^* → Calculate loss → $\mathscr{L}_3$

$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$

$\mathscr{D}_3^{test}$

Model Agnostic Meta-learning (MAML)

Meta-learner → θ

classifier

$T_1 = (\mathscr{D}_1^{train}, \mathscr{D}_1^{test})$

Gradient Decent

θ_1^*

Calculate loss

$\mathscr{D}_1^{test}$

$\mathscr{L}_1$

classifier

$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$

Gradient Decent

θ_2^*

Calculate loss

$\mathscr{D}_2^{test}$

$\mathscr{L}_2$

classifier

$T_2 = (\mathscr{D}_2^{train}, \mathscr{D}_2^{test})$

Gradient Decent

θ_3^*

Calculate loss

$\mathscr{D}_3^{test}$

$\mathscr{L}_3$

43

Model Agnostic Meta-learning (MAML)

f_θ : *Generic model parametrized by θ* (*aka meta* − *learner*)
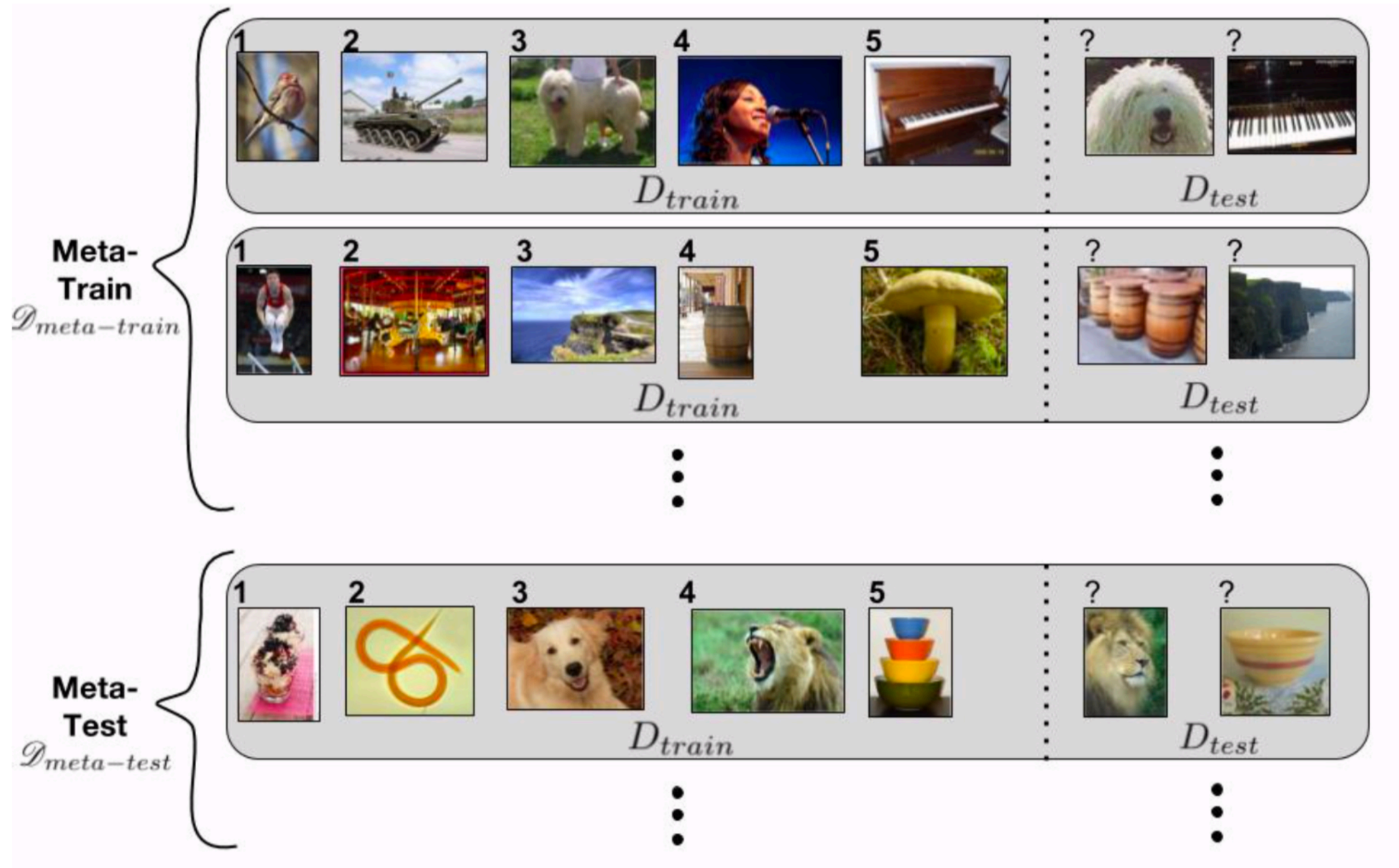
θ : *original paramters*

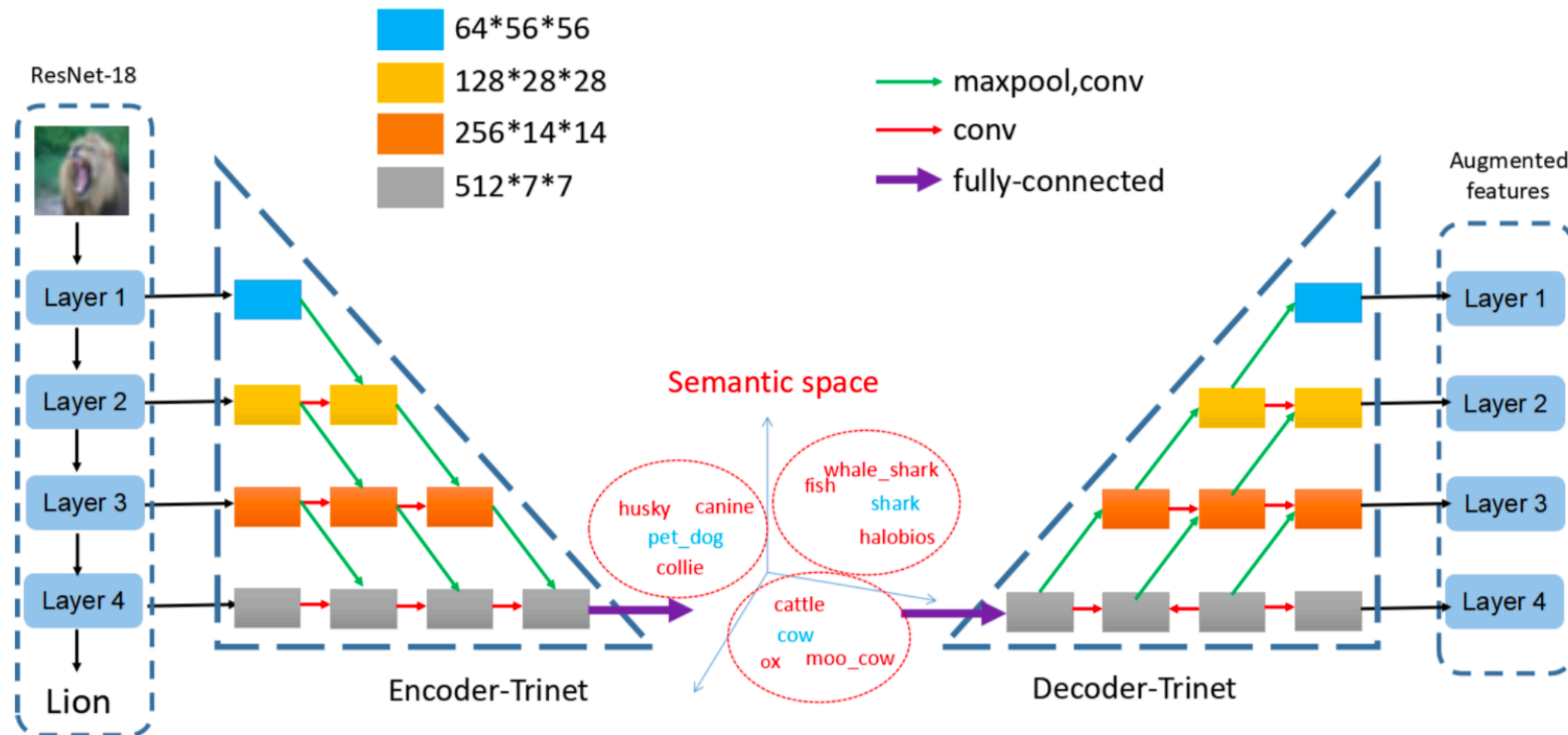*θ′* : *task* − *specific paramters*

$T_i = (\mathscr{D}^{train}, \mathscr{D}^{test})$

1: randomly initialize θ
2: **while** not done **do**
3:     Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4:     **for all** $\mathcal{T}_i$ **do**
5:         Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6:         Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7:     **end for**
8:     Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$
9: **end while**

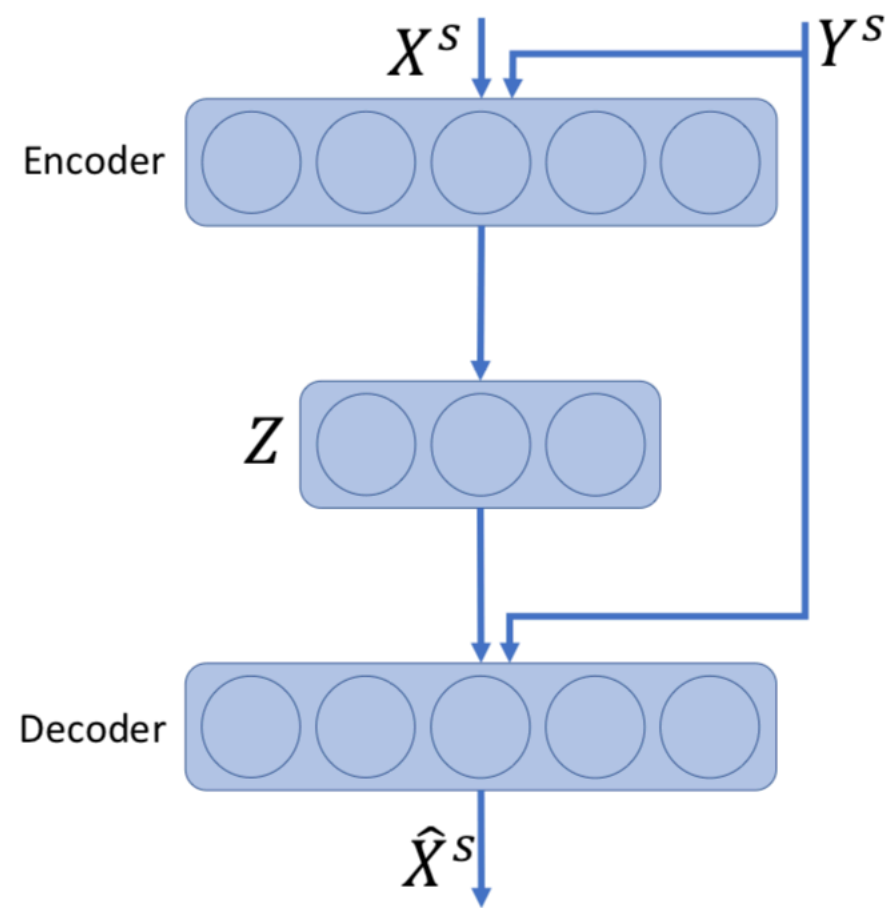# Model Agnostic Meta-learning (MAML)
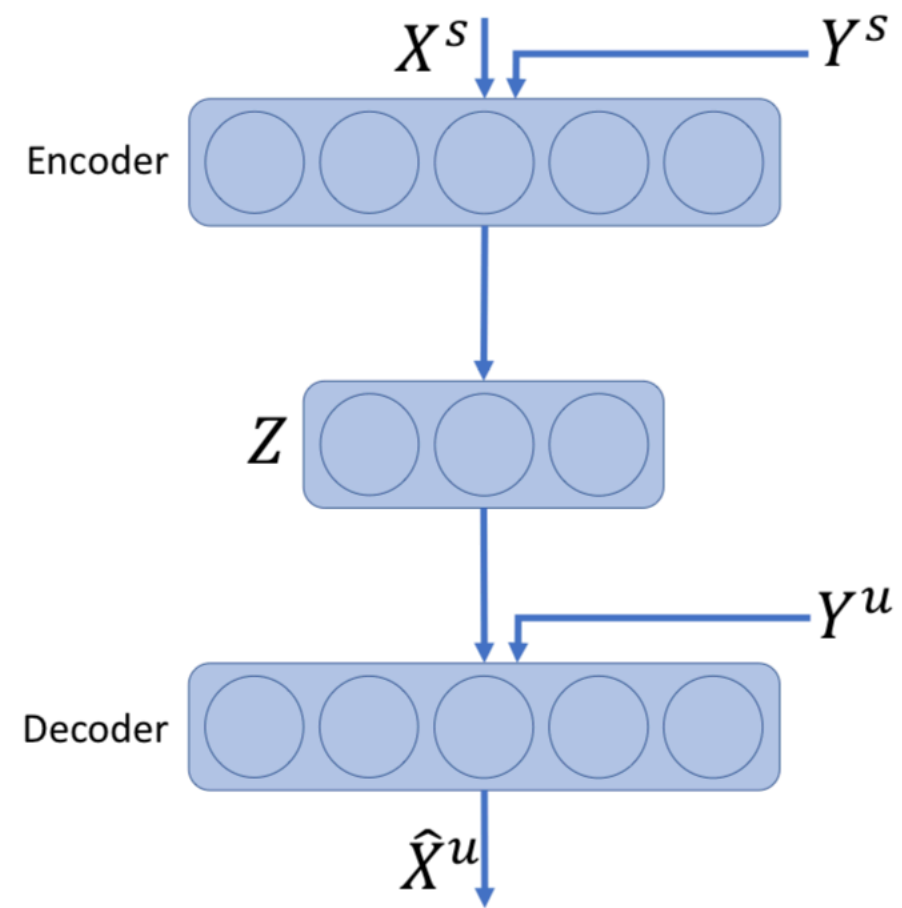
Data Augmentation

Data Augmentation

(a) Training phase:

(b) Sample synthesis phase:

Future Works

Fewshot learning in NLP

Future Works

Fewshot learning in NLP

- Data Augmentation

- MAML

- Pre-trained Language Models (e.x. ULMFiT, Google Bert)

Thank you!