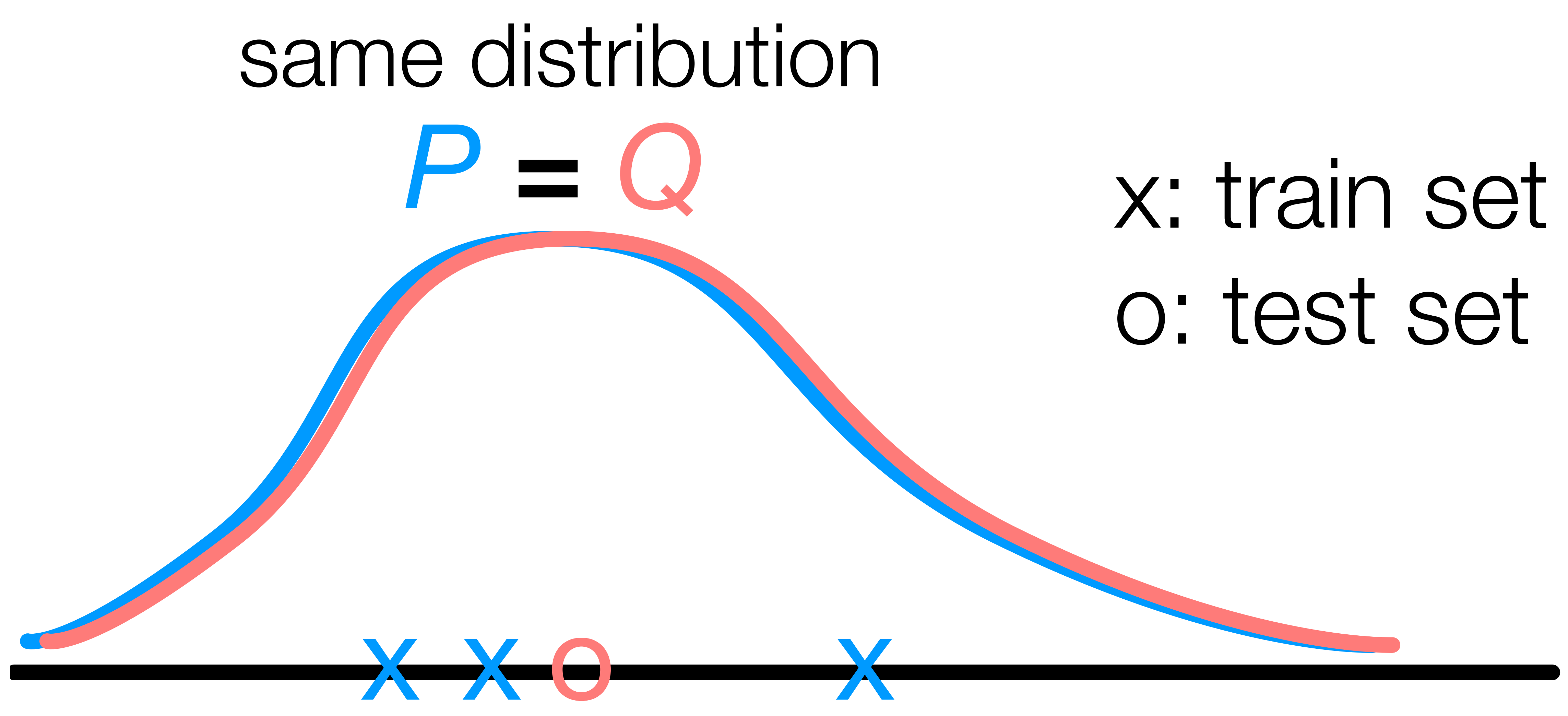
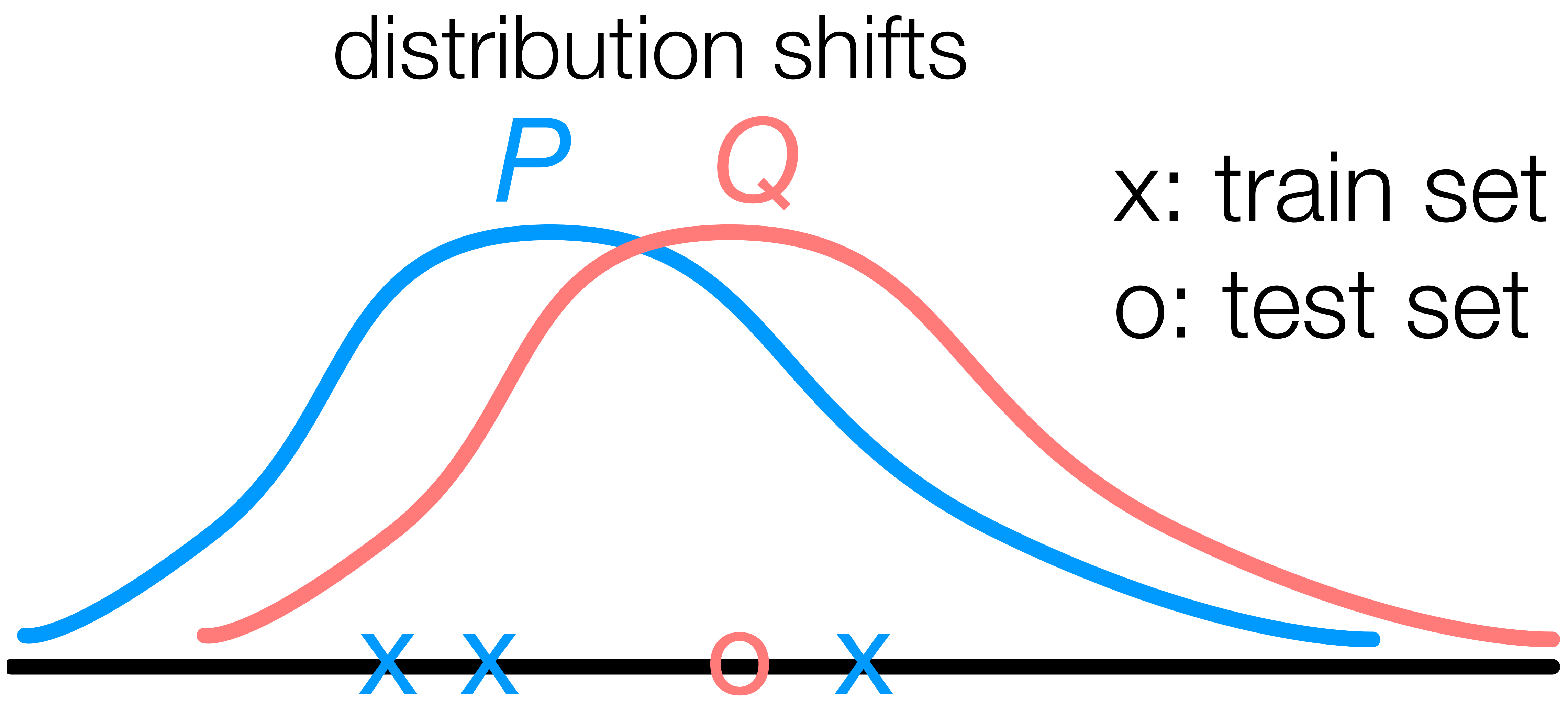
Test-Time Training with Self-Supervision for Generalization under Distribution Shifts

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, Moritz Hardt
UC Berkeley

ICML 2020

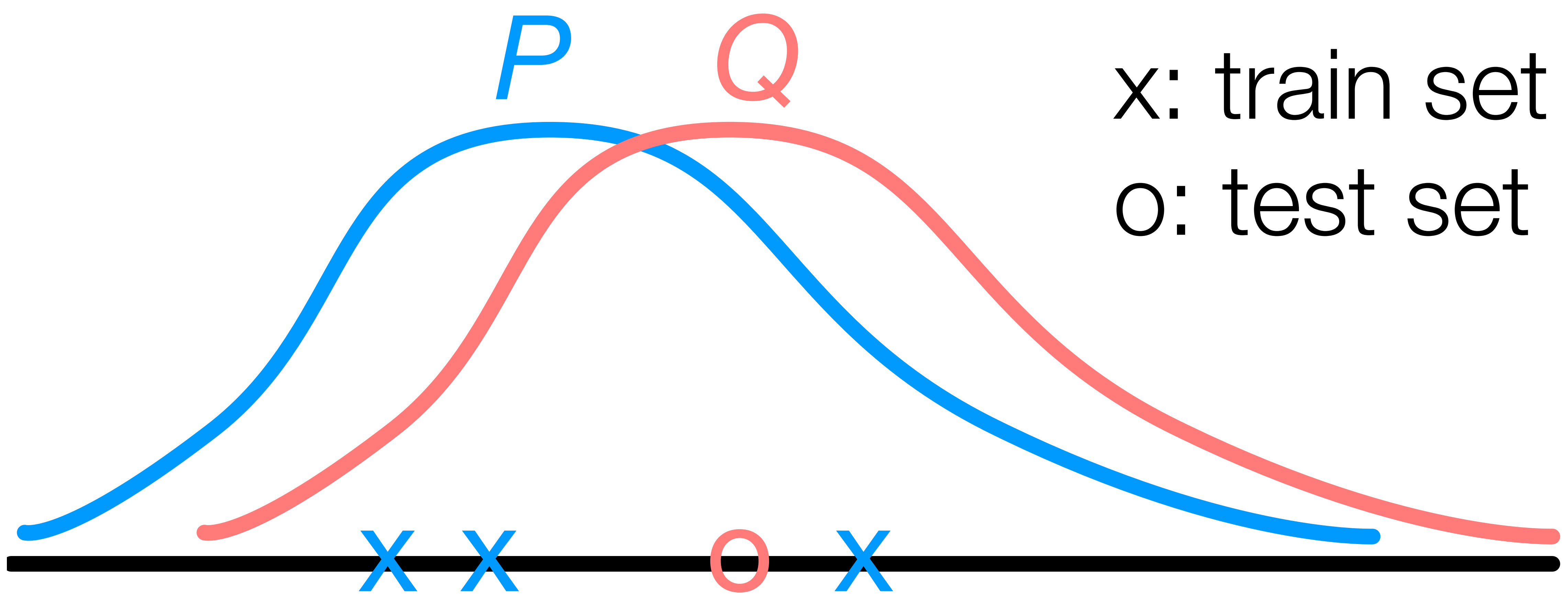


- **In theory**: same distribution for training and testing



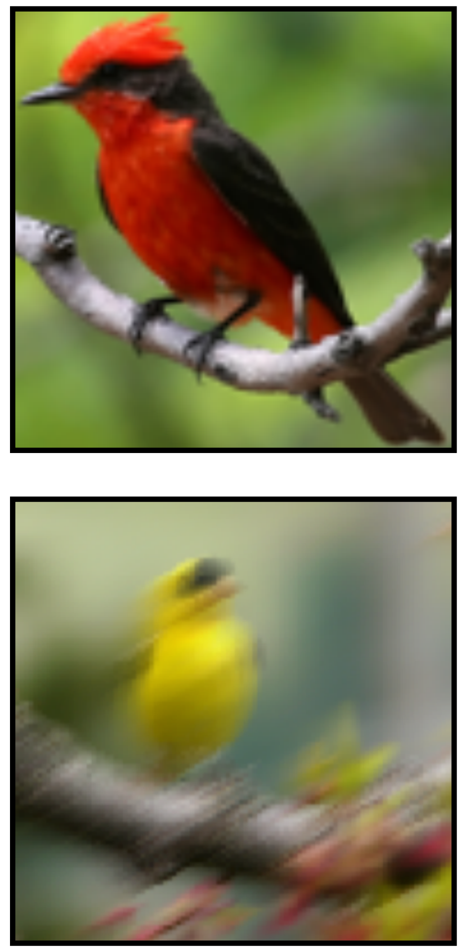
- **In theory**: same distribution for training and testing
- **In the real word**: distribution shifts are everywhere

distribution shifts



x: train set
o: test set

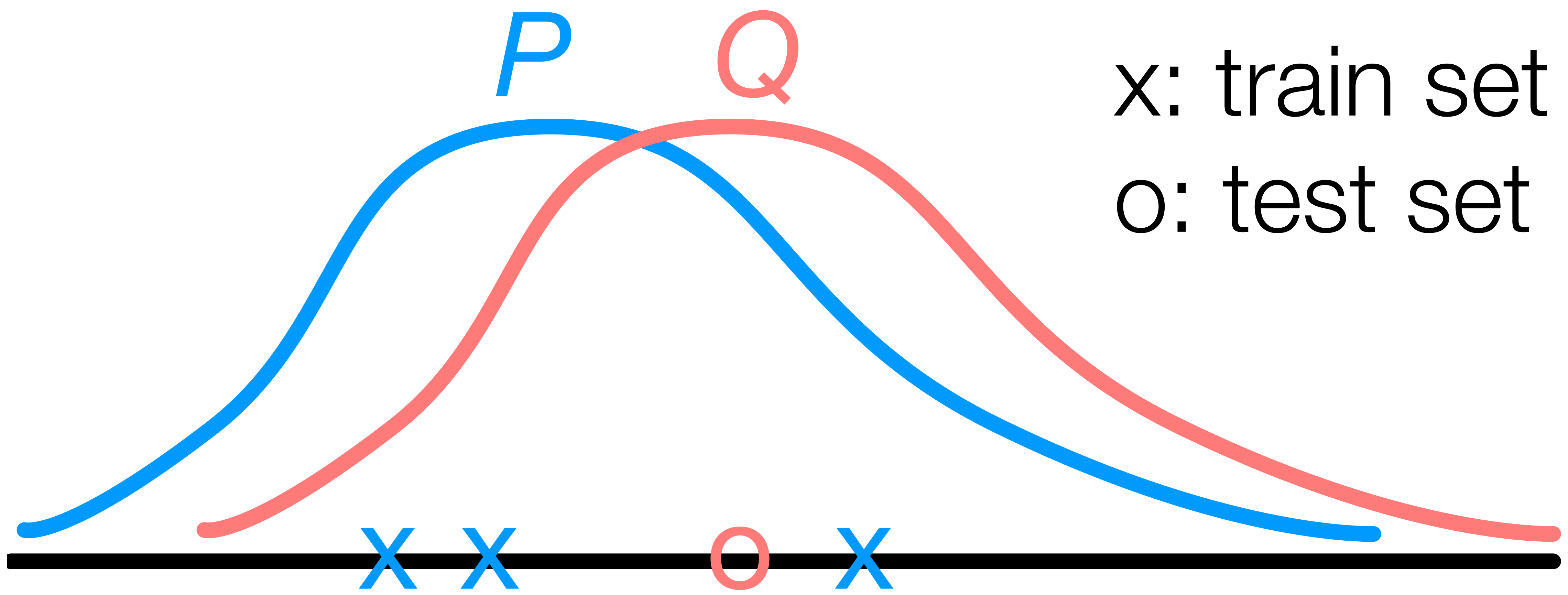
- **In theory**: same distribution for training and testing
- **In the real word**: distribution shifts are everywhere



Hendrycks and Dietterich, 2018

Recht, Roelofs, Schmidt and Shankar, 2019

Existing paradigms



x: train set
o: test set

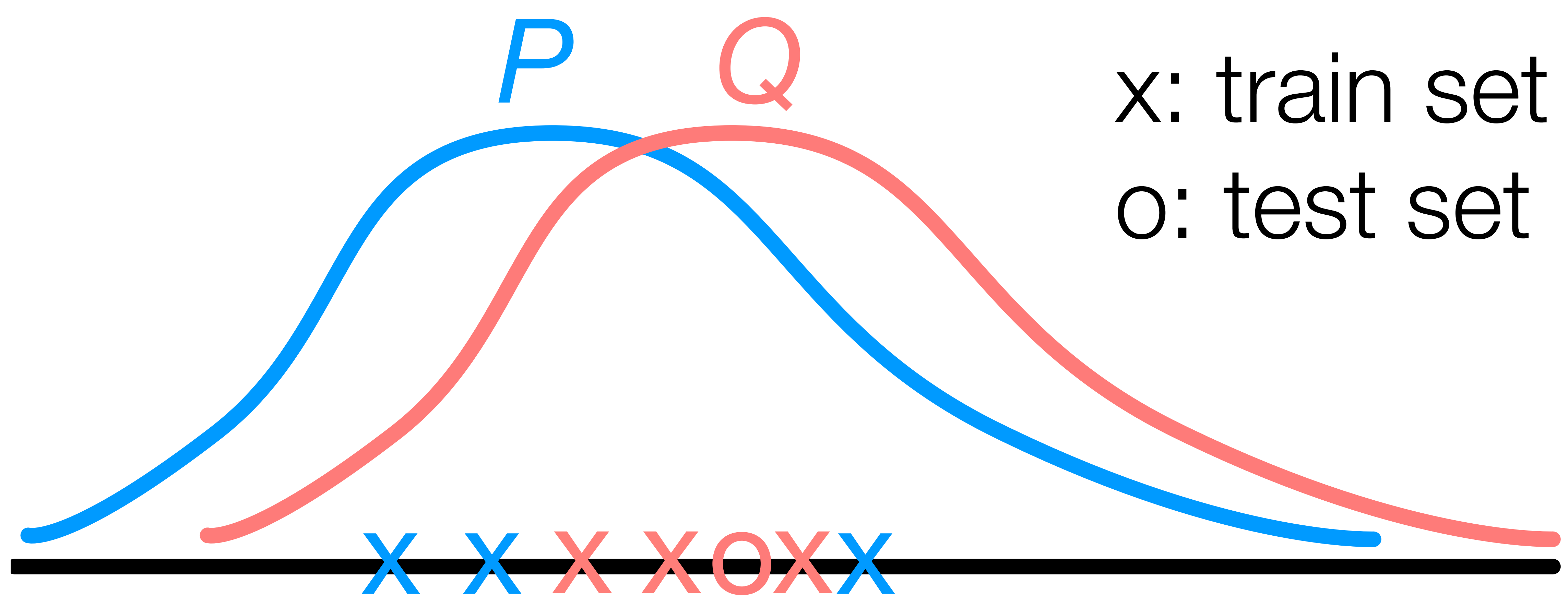
A Theory of Learning from Different Domains
Ben-David, Blitzer, Crammer, Kulesza, Pereira and Vaughan, 2009

Adversarial Discriminative Domain Adaptation
Tzeng, Hoffman, Saenko and Darrell, 2017

Unsupervised Domain Adaptation through Self-Supervision
Sun, Tzeng, Darrell and Efros, 2019

Existing paradigms

- **Domain adaptation**
  - Data from the test distribution



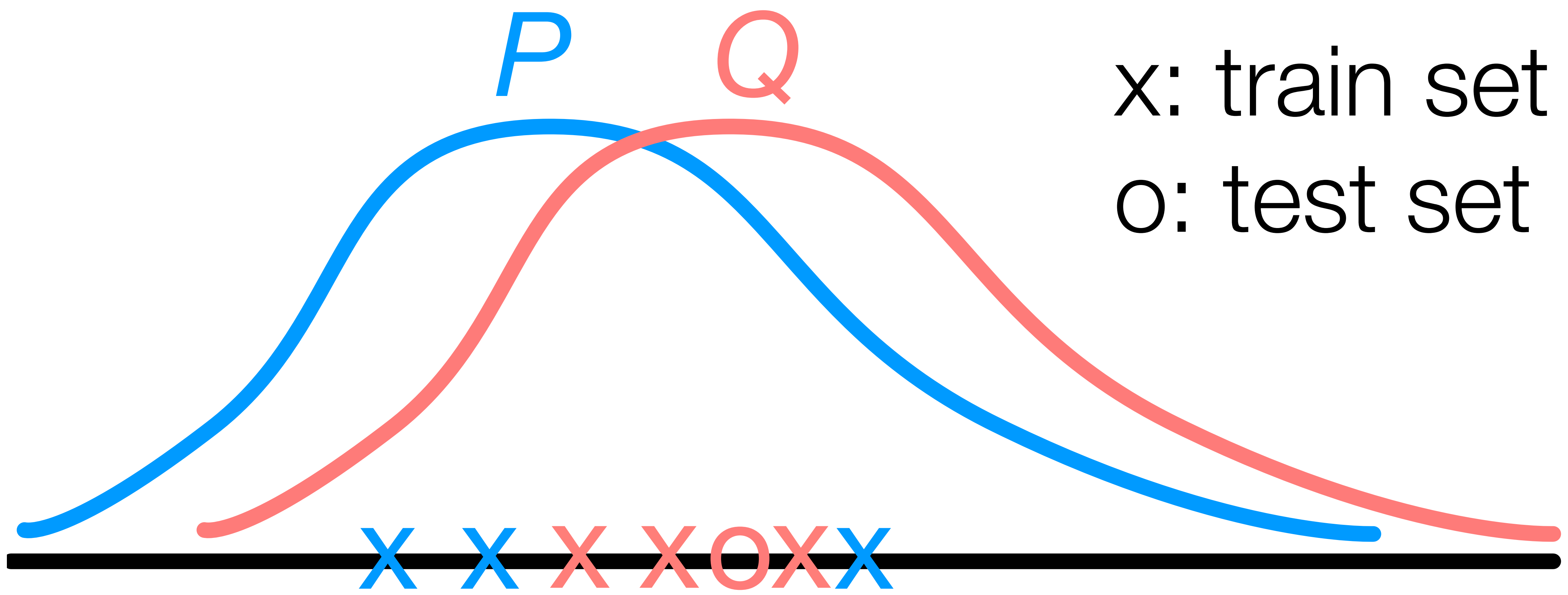
A Theory of Learning from Different Domains
Ben-David, Blitzer, Crammer, Kulesza, Pereira and Vaughan, 2009

Adversarial Discriminative Domain Adaptation
Tzeng, Hoffman, Saenko and Darrell, 2017

Unsupervised Domain Adaptation through Self-Supervision
Sun, Tzeng, Darrell and Efros, 2019

Existing paradigms

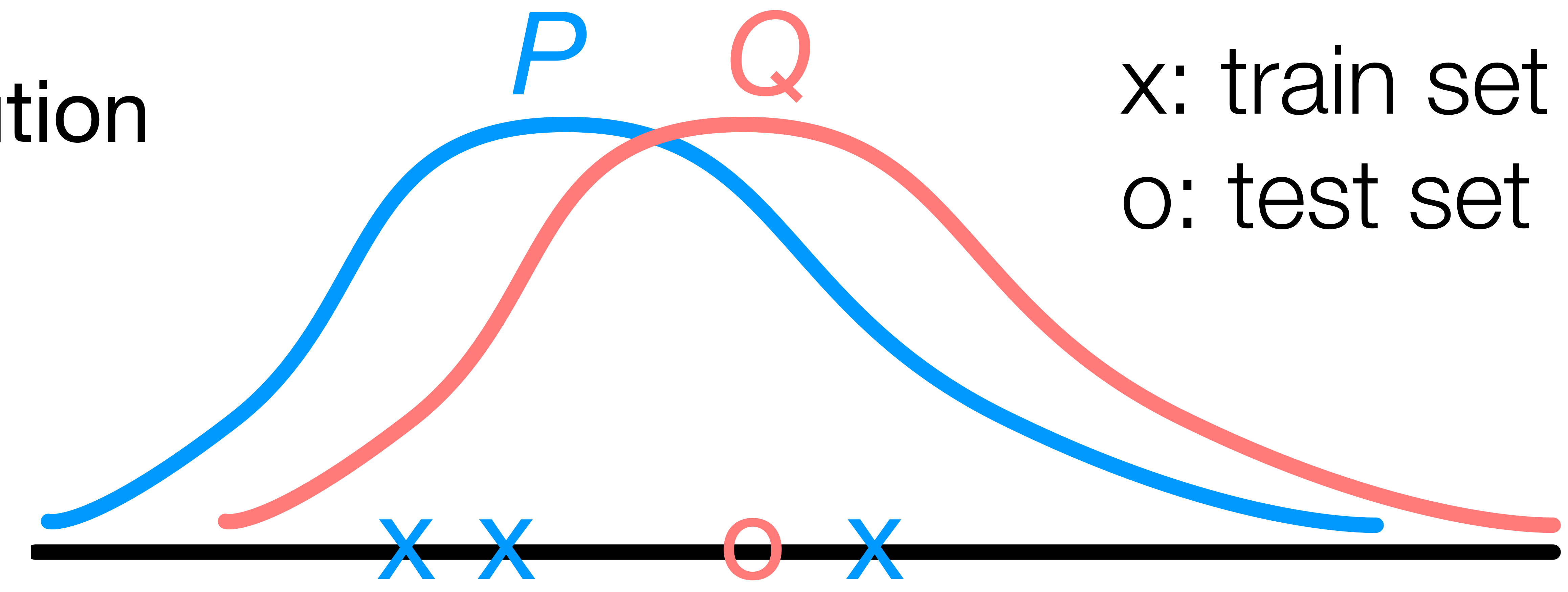
- **Domain adaptation**
  - Data from the test distribution (maybe unlabeled)
  - Hard to know the test distribution



x: train set
o: test set

Existing paradigms

- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution



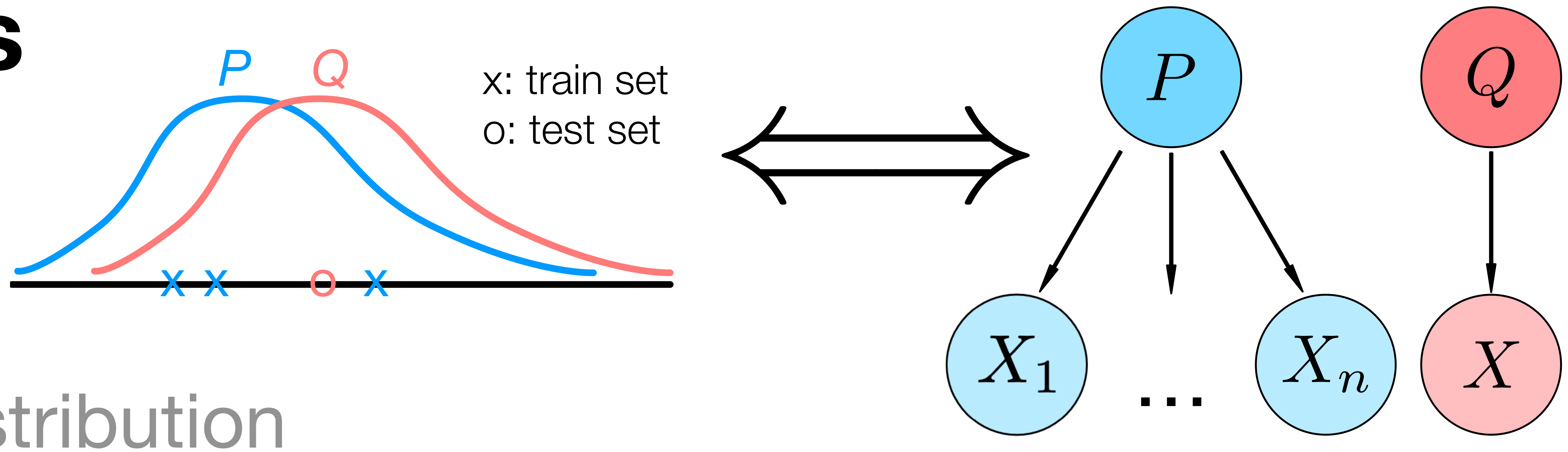
Domain generalization via invariant feature representation
Muandet, Balduzzi and Scholkopf, 2013

Domain generalization for object recognition with multi-task autoencoders
Ghifary, Bastiaan, Zhang and Balduzzi, 2015

Domain Generalization by Solving Jigsaw Puzzles
Carlucci, D'Innocente, Bucci, Caputo and Tommasi, 2019

Existing paradigms

distribution shifts



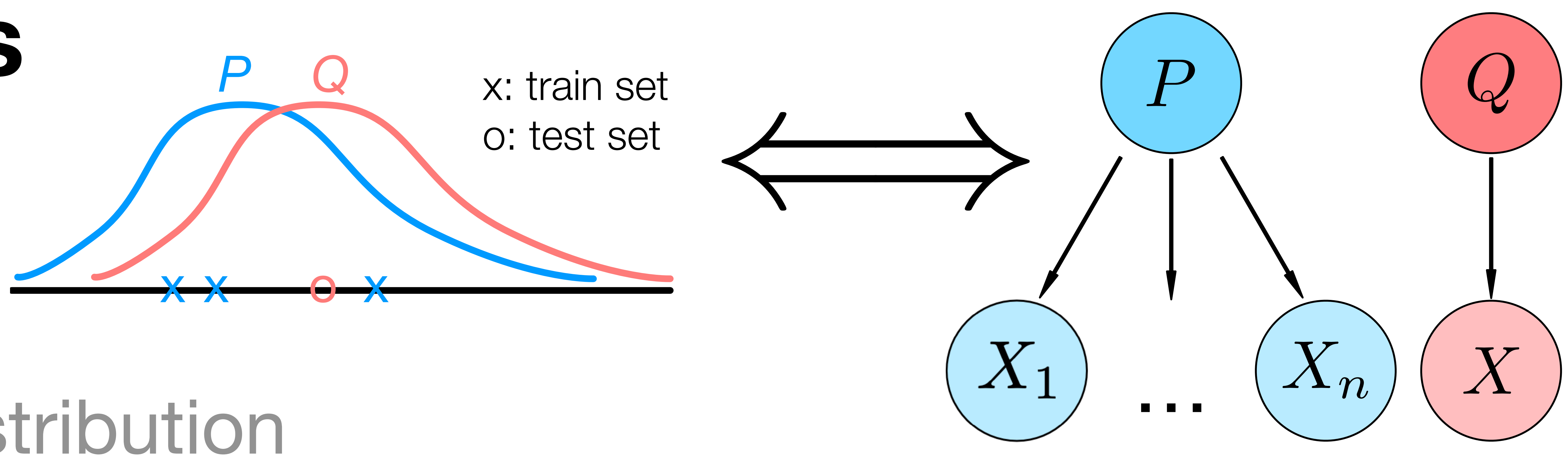
- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution

Domain generalization via invariant feature representation
Muandet, Balduzzi and Scholkopf, 2013

Domain generalization for object recognition with multi-task autoencoders
Ghifary, Bastiaan, Zhang and Balduzzi, 2015

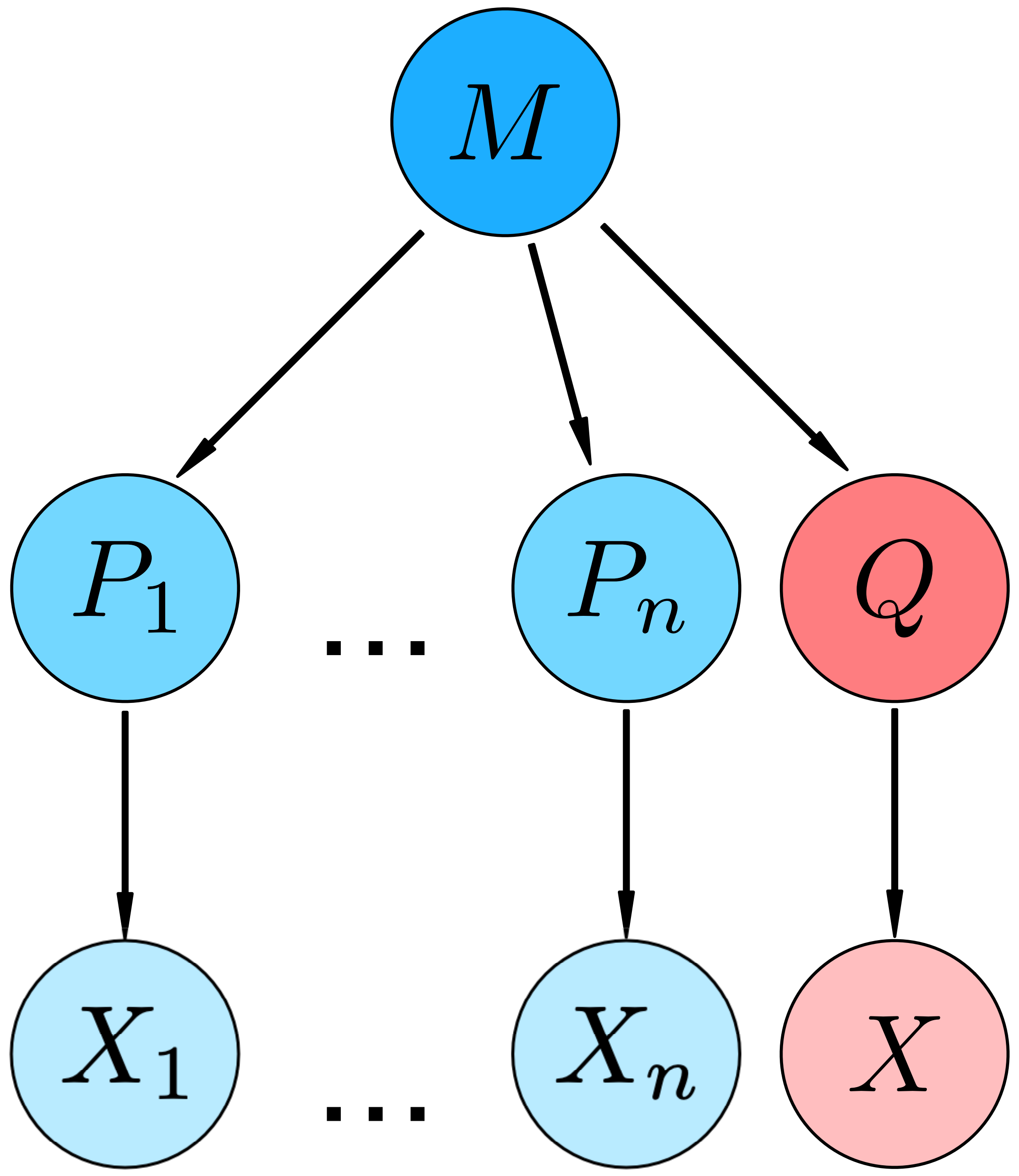
Domain Generalization by Solving Jigsaw Puzzles
Carlucci, D'Innocente, Bucci, Caputo and Tommasi, 2019

Existing paradigms



distribution shifts

- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution

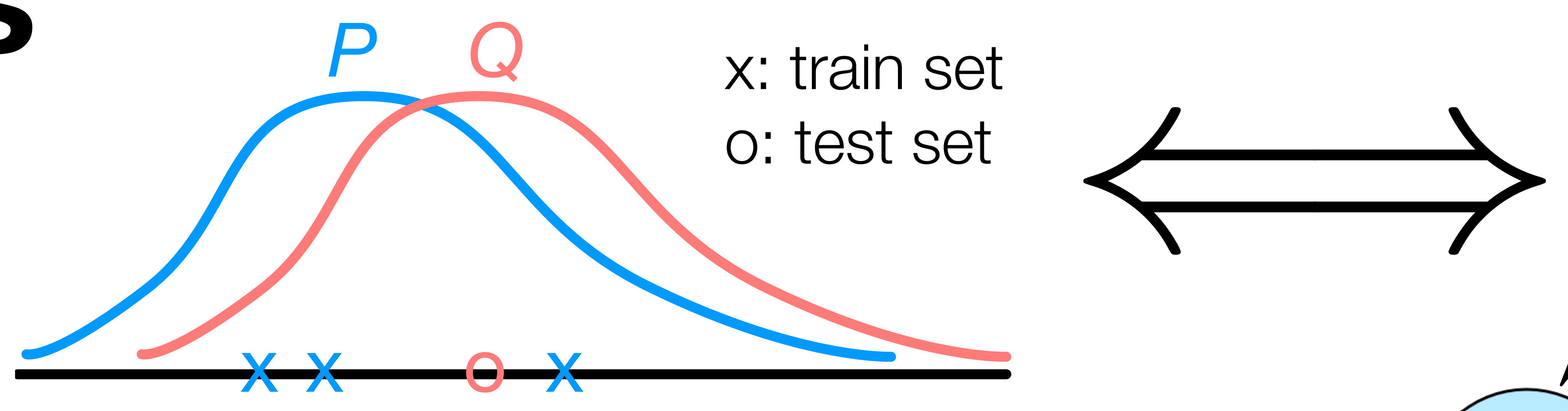


Domain generalization via invariant feature representation
Muandet, Balduzzi and Scholkopf, 2013

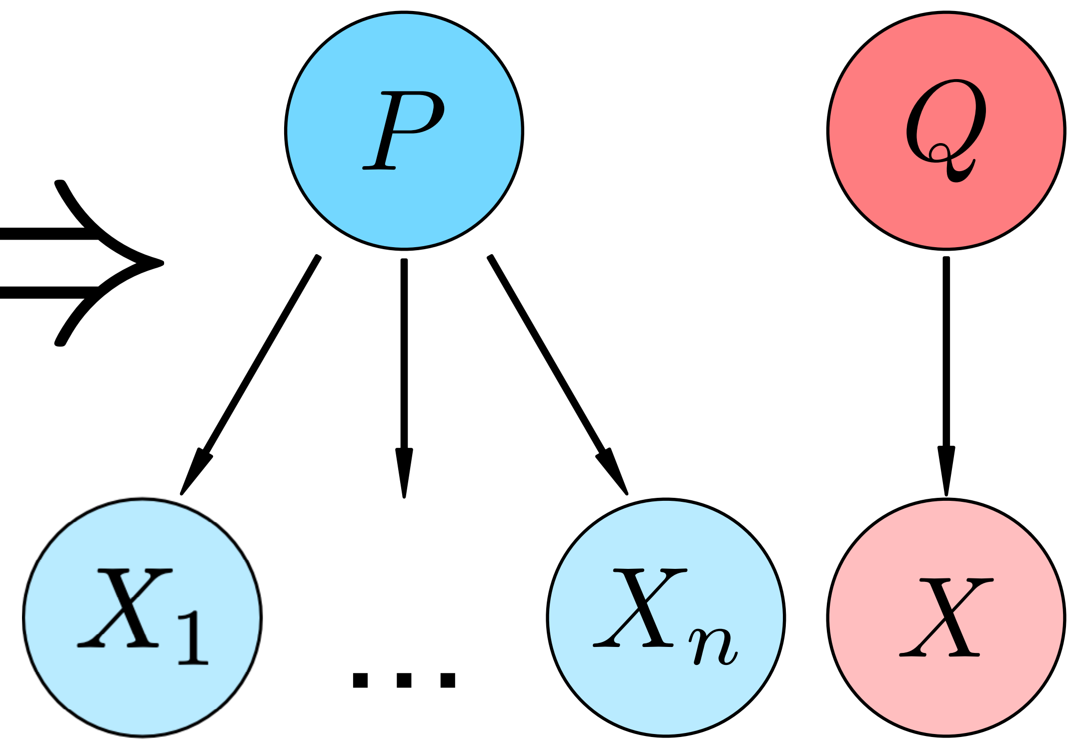
Domain generalization for object recognition with multi-task autoencoders
Ghifary, Bastiaan, Zhang and Balduzzi, 2015

Domain Generalization by Solving Jigsaw Puzzles
Carlucci, D'Innocente, Bucci, Caputo and Tommasi, 2019

Existing paradigms

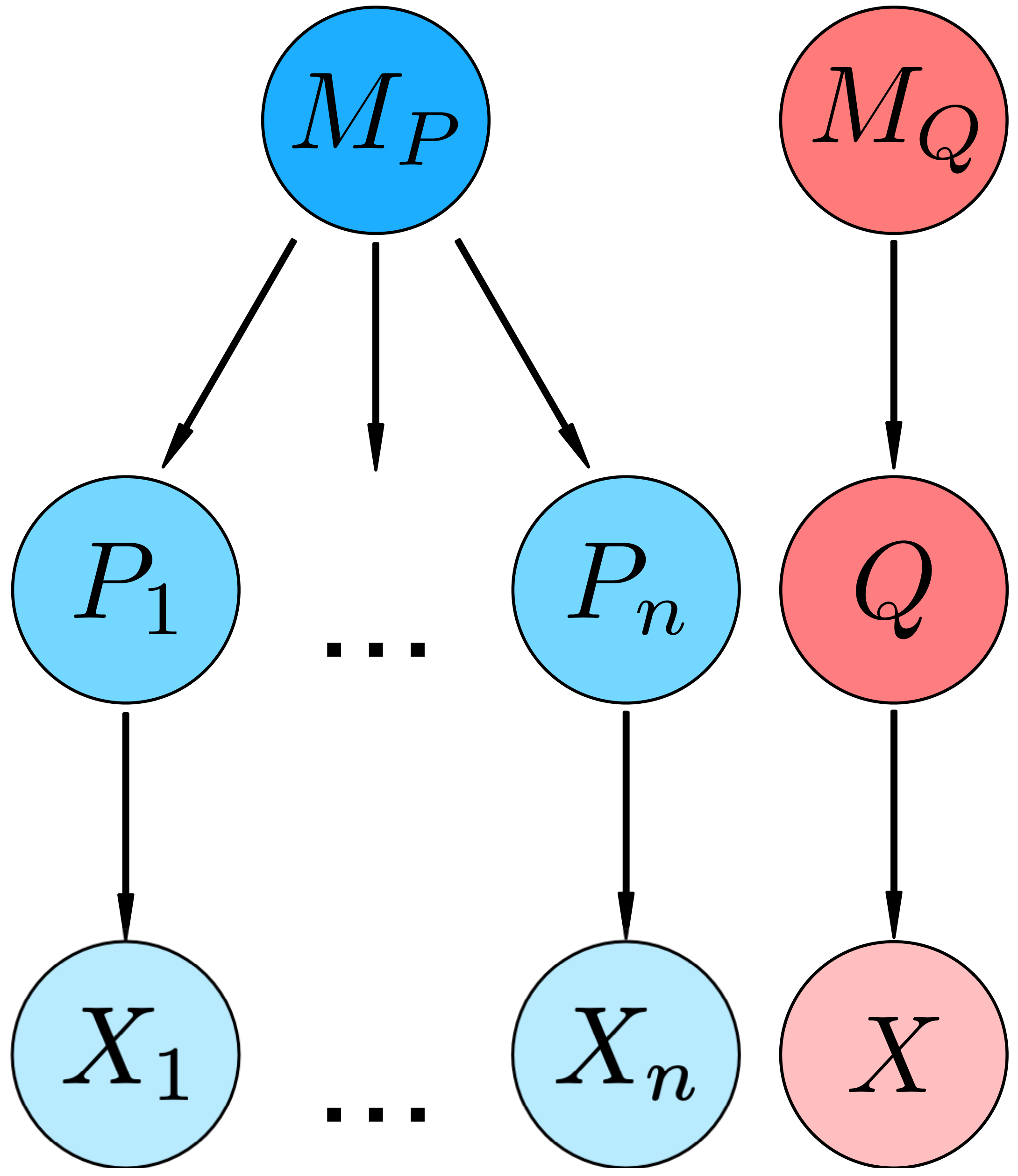


distribution shifts



- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution
  - Hard to know the meta distribution

meta distribution shifts



Domain generalization via invariant feature representation
Muandet, Balduzzi and Scholkopf, 2013

Domain generalization for object recognition with multi-task autoencoders
Ghifary, Bastiaan, Zhang and Balduzzi, 2015

Domain Generalization by Solving Jigsaw Puzzles
Carlucci, D'Innocente, Bucci, Caputo and Tommasi, 2019

Existing paradigms

Certifying some distributional robustness with principled adversarial training
Sinha, Namkoong and Duchi, 2017

Towards deep learning models resistant to adversarial attacks
Madry, Makelov, Schmidt, Tsipras and Vladu, 2017

Adversarially robust generalization requires more data
Schmidt, Santurkar, Tsipras, Talwar and Madry, 2018

- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution
  - Hard to know the meta distribution
- **Adversarial robustness**
  - Topological structure of the test distribution

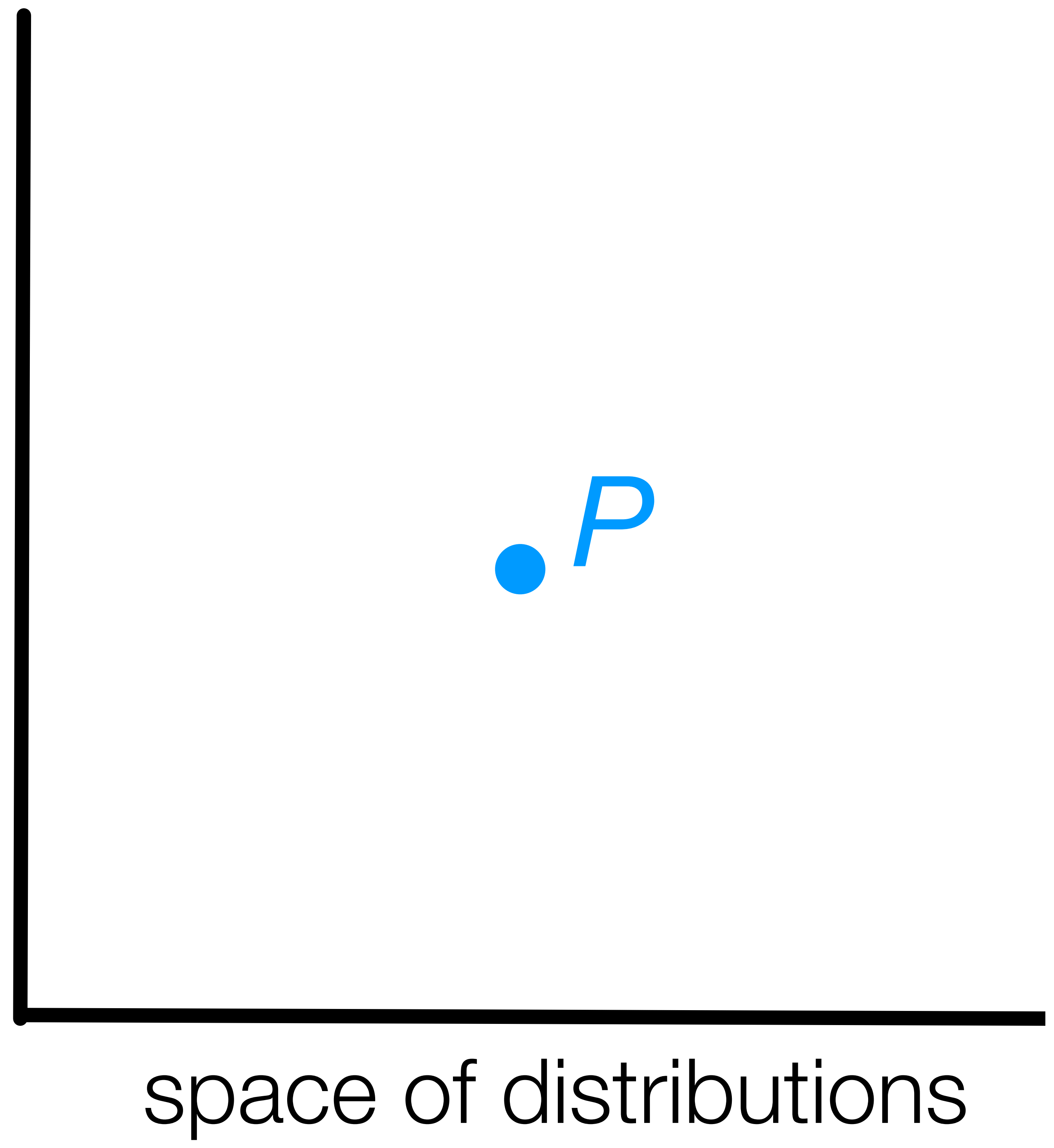
Certifying some distributional robustness with principled adversarial training
Sinha, Namkoong and Duchi, 2017

Towards deep learning models resistant to adversarial attacks
Madry, Makelov, Schmidt, Tsipras and Vladu, 2017

Adversarially robust generalization requires more data
Schmidt, Santurkar, Tsipras, Talwar and Madry, 2018

Existing paradigms

- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution
  - Hard to know the meta distribution
- **Adversarial robustness**
  - Topological structure of the test distribution



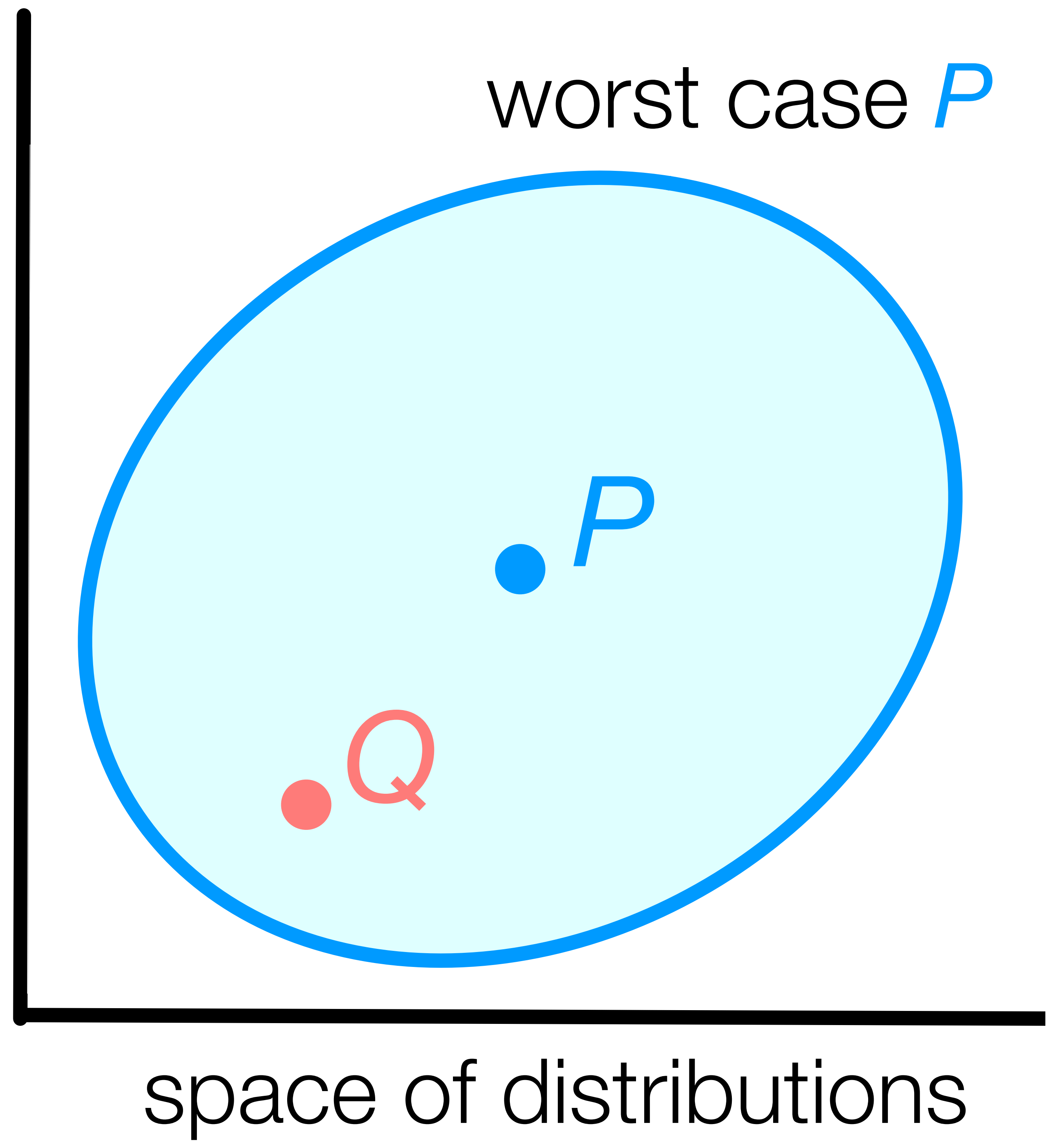
Certifying some distributional robustness with principled adversarial training
Sinha, Namkoong and Duchi, 2017

Towards deep learning models resistant to adversarial attacks
Madry, Makelov, Schmidt, Tsipras and Vladu, 2017

Adversarially robust generalization requires more data
Schmidt, Santurkar, Tsipras, Talwar and Madry, 2018

Existing paradigms

- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution
  - Hard to know the meta distribution
- **Adversarial robustness**
  - Topological structure of the test distribution



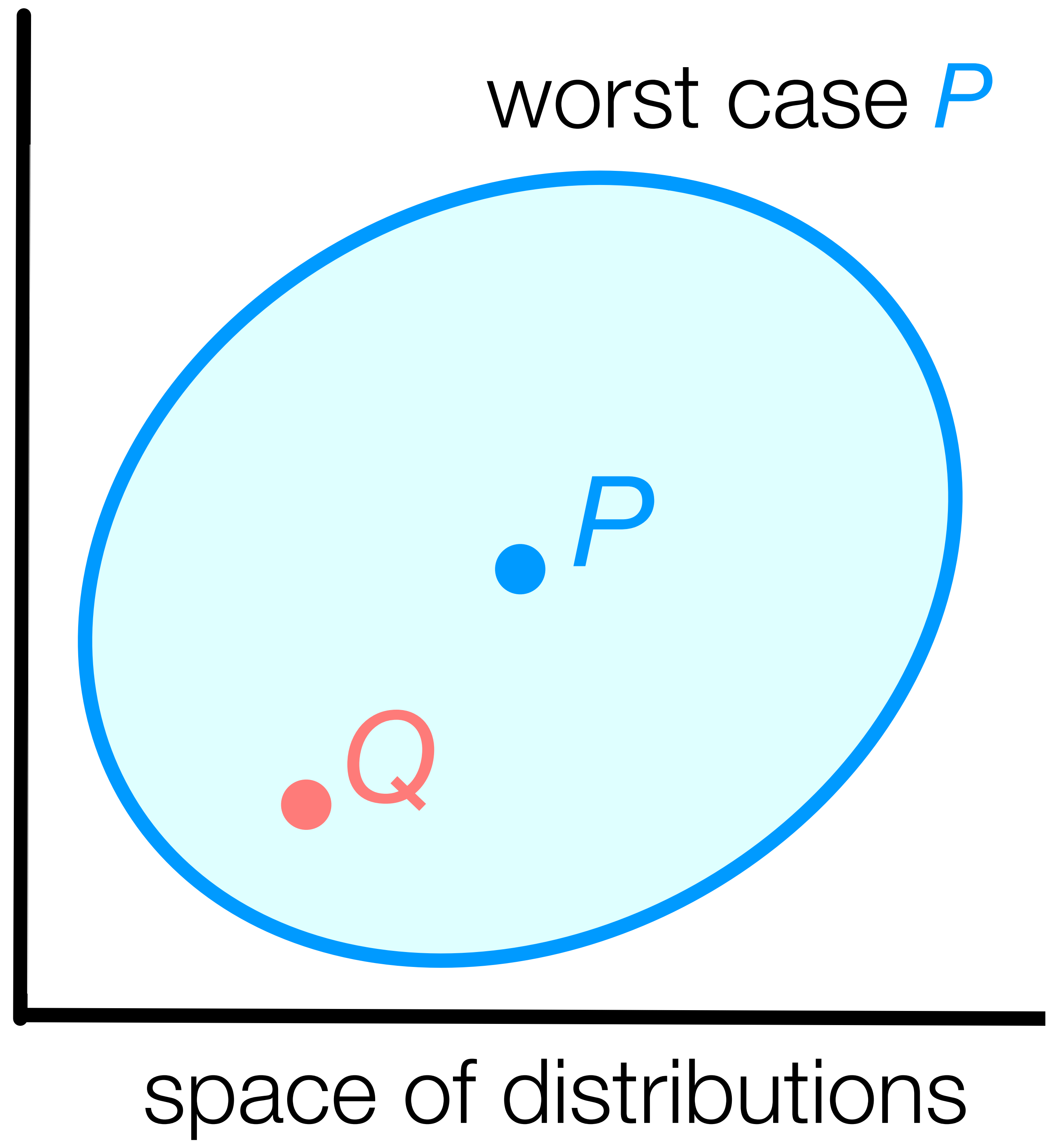
Existing paradigms

Certifying some distributional robustness with principled adversarial training
Sinha, Namkoong and Duchi, 2017

Towards deep learning models resistant to adversarial attacks
Madry, Makelov, Schmidt, Tsipras and Vladu, 2017

Adversarially robust generalization requires more data
Schmidt, Santurkar, Tsipras, Talwar and Madry, 2018

- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution
  - Hard to know the meta distribution
- **Adversarial robustness**
  - Topological structure of the test distribution
  - Hard to describe, especially in high dimension



Existing paradigms anticipate the distribution shifts

- **Domain adaptation**
  - Data from the test distribution
  - Hard to know the test distribution
- **Domain generalization**
  - Data from the meta distribution
  - Hard to know the meta distribution
- **Adversarial robustness**
  - Topological structure of the test distribution
  - Hard to describe, especially in high dimension

Test-Time Training (TTT)

- Does not anticipate the test distribution

Test-Time Training (TTT)

$$\text{standard test error} = \mathbb{E}_Q[\ell(x, y);\ \theta]$$

- Does not anticipate the test distribution
- The test sample x gives us a hint about Q

Test-Time Training (TTT)

$$\text{standard test error} = \mathbb{E}_Q[\ell(x, y);\ \theta]$$

$$\text{our test error} = \mathbb{E}_Q[\ell(x, y);\ \theta(x)]$$

- Does not anticipate the test distribution
- The test sample x gives us a hint about Q
- No fixed model, but adapt at test time

Test-Time Training (TTT)

$$\text{standard test error} = \mathbb{E}_Q[\ell(x, y);\ \theta]$$

$$\text{our test error} = \mathbb{E}_Q[\ell(x, y);\ \theta(x)]$$

- Does not anticipate the test distribution
- The test sample x gives us a hint about Q
- No fixed model, but adapt at test time
- One sample learning problem
- No label? Self-supervision!

Rotation prediction as self-supervision

(Gidaris et al. 2018)

x

- Create labels from unlabeled input

Unsupervised Representation Learning by Predicting Image Rotations
Gidaris, Singh and Komodakis, 2018

Rotation prediction as self-supervision

(Gidaris et al. 2018)

x y_{s}

0º

90º

180º

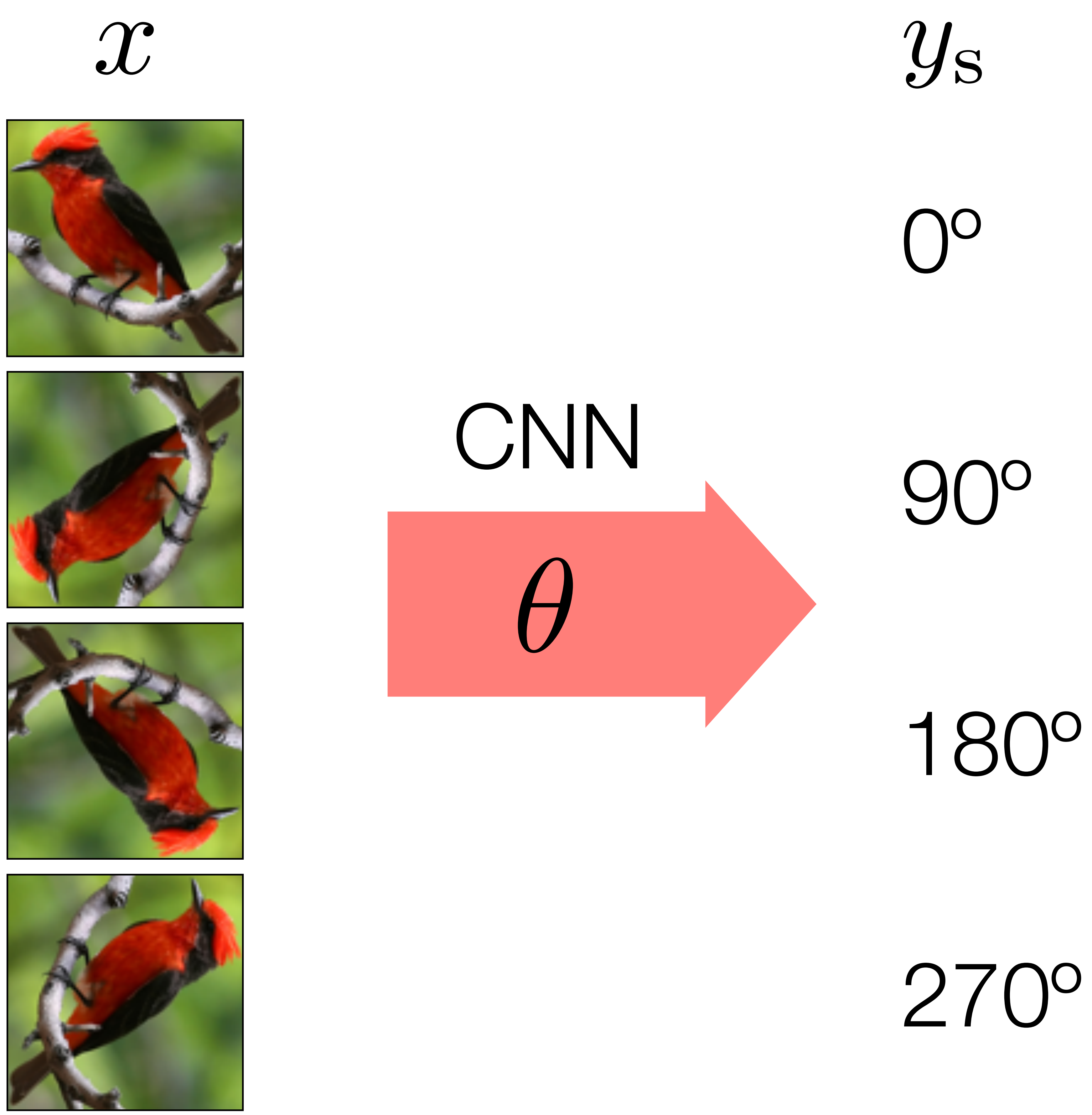
270º

- Create labels from unlabeled input
- Rotate input image by multiples of 90º

Unsupervised Representation Learning by Predicting Image Rotations
Gidaris, Singh and Komodakis, 2018

Rotation prediction as self-supervision

(Gidaris et al. 2018)

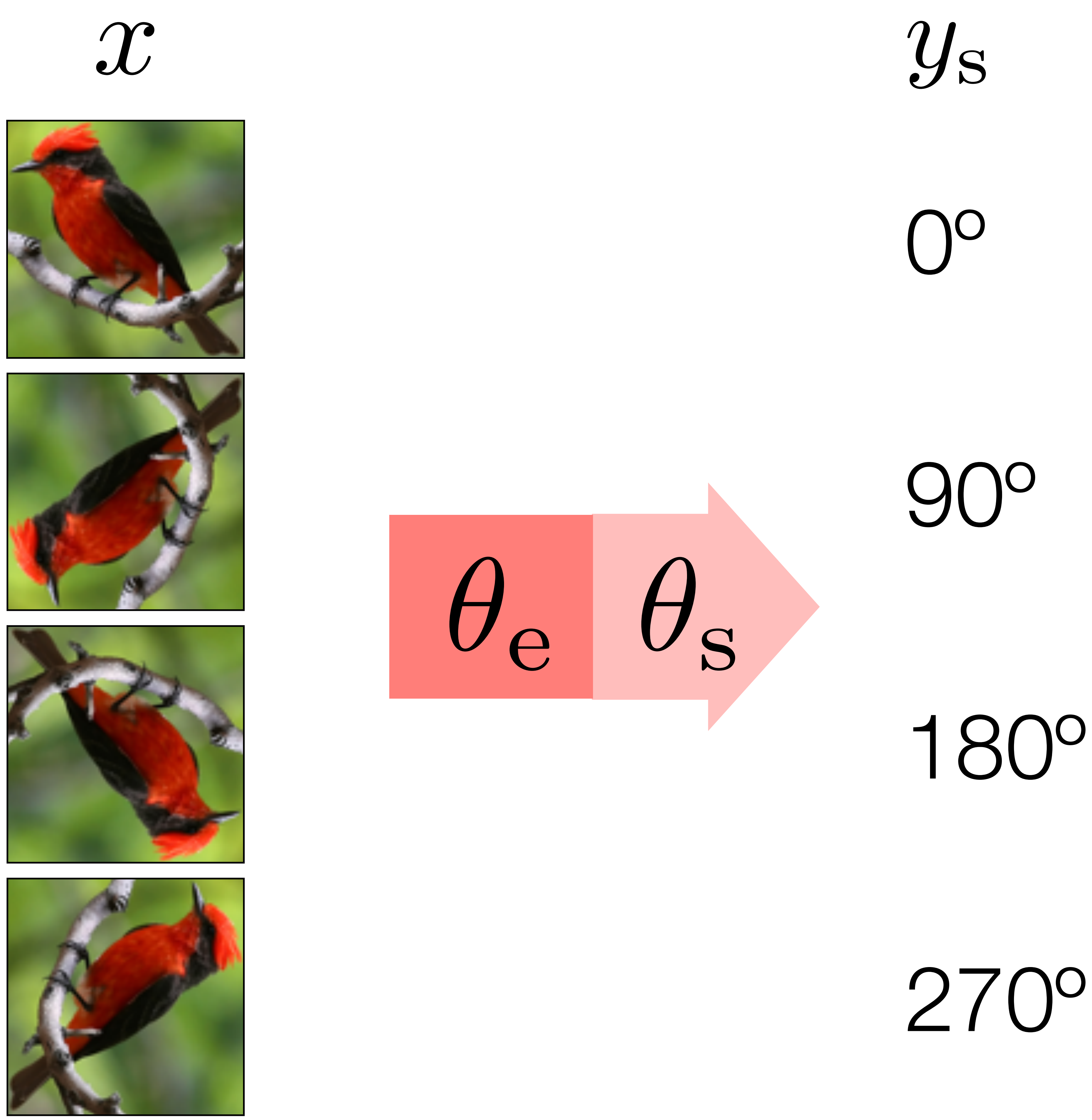


- Create labels from unlabeled input
- Rotate input image by multiples of 90°
- Produce a four-way classification problem

Unsupervised Representation Learning by Predicting Image Rotations
Gidaris, Singh and Komodakis, 2018

Rotation prediction as self-supervision

(Gidaris et al. 2018)



- Create labels from unlabeled input
- Rotate input image by multiples of 90°
- Produce a four-way classification problem
- Usually a pre-training step

Unsupervised Representation Learning by Predicting Image Rotations
Gidaris, Singh and Komodakis, 2018

Rotation prediction as self-supervision

(Gidaris et al. 2018)

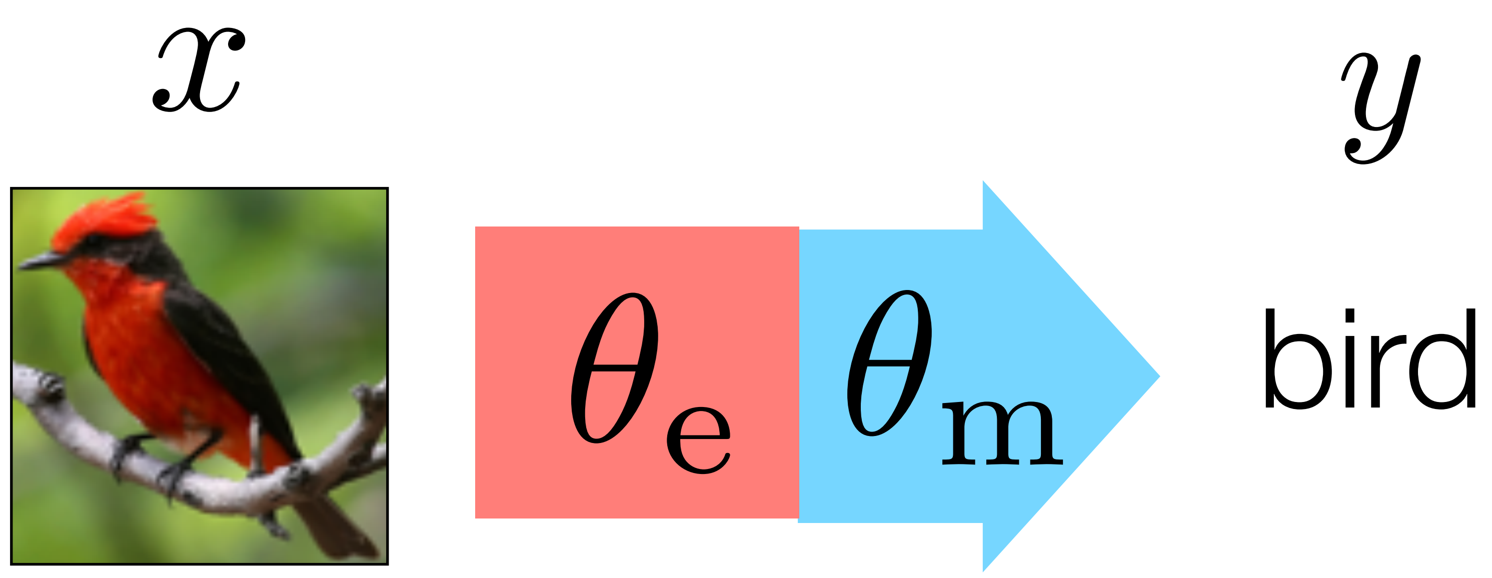
θ_{e}

- Create labels from unlabeled input
- Rotate input image by multiples of 90º
- Produce a four-way classification problem
- Usually a pre-training step
    - After training, take feature extractor

Unsupervised Representation Learning by Predicting Image Rotations
Gidaris, Singh and Komodakis, 2018

Rotation prediction as self-supervision

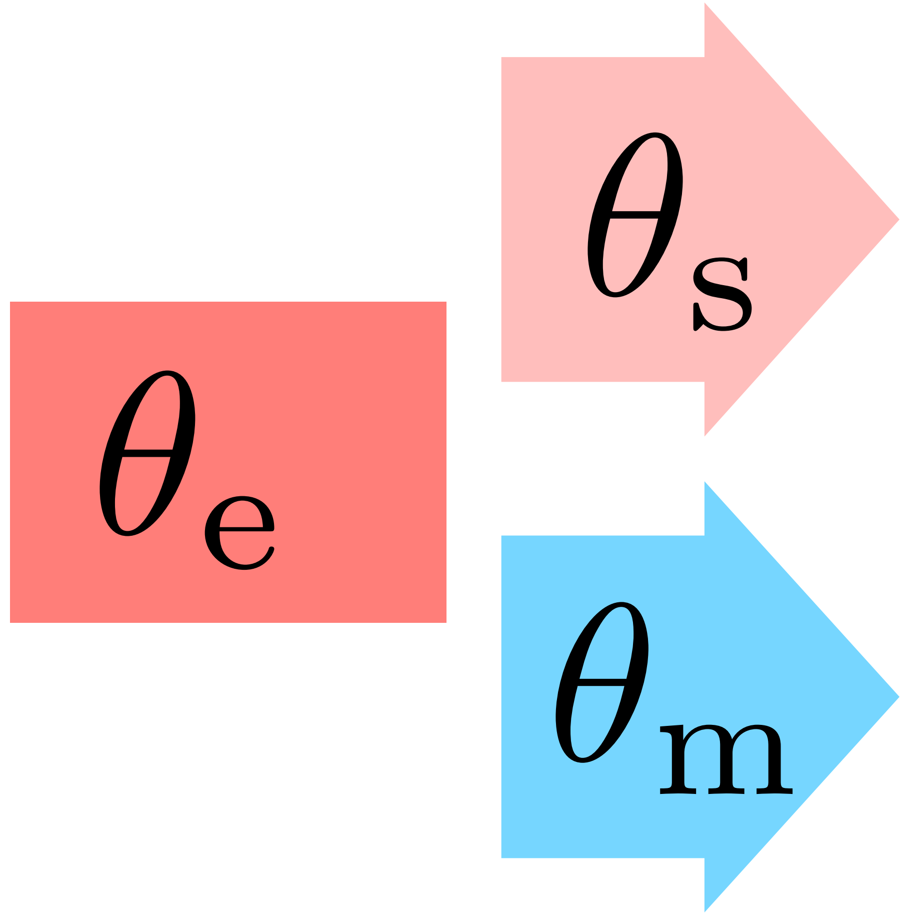
(Gidaris et al. 2018)



- Create labels from unlabeled input
- Rotate input image by multiples of 90°
- Produce a four-way classification problem
- Usually a pre-training step
  - After training, take feature extractor
  - Use it for a downstream main task

Unsupervised Representation Learning by Predicting Image Rotations
Gidaris, Singh and Komodakis, 2018

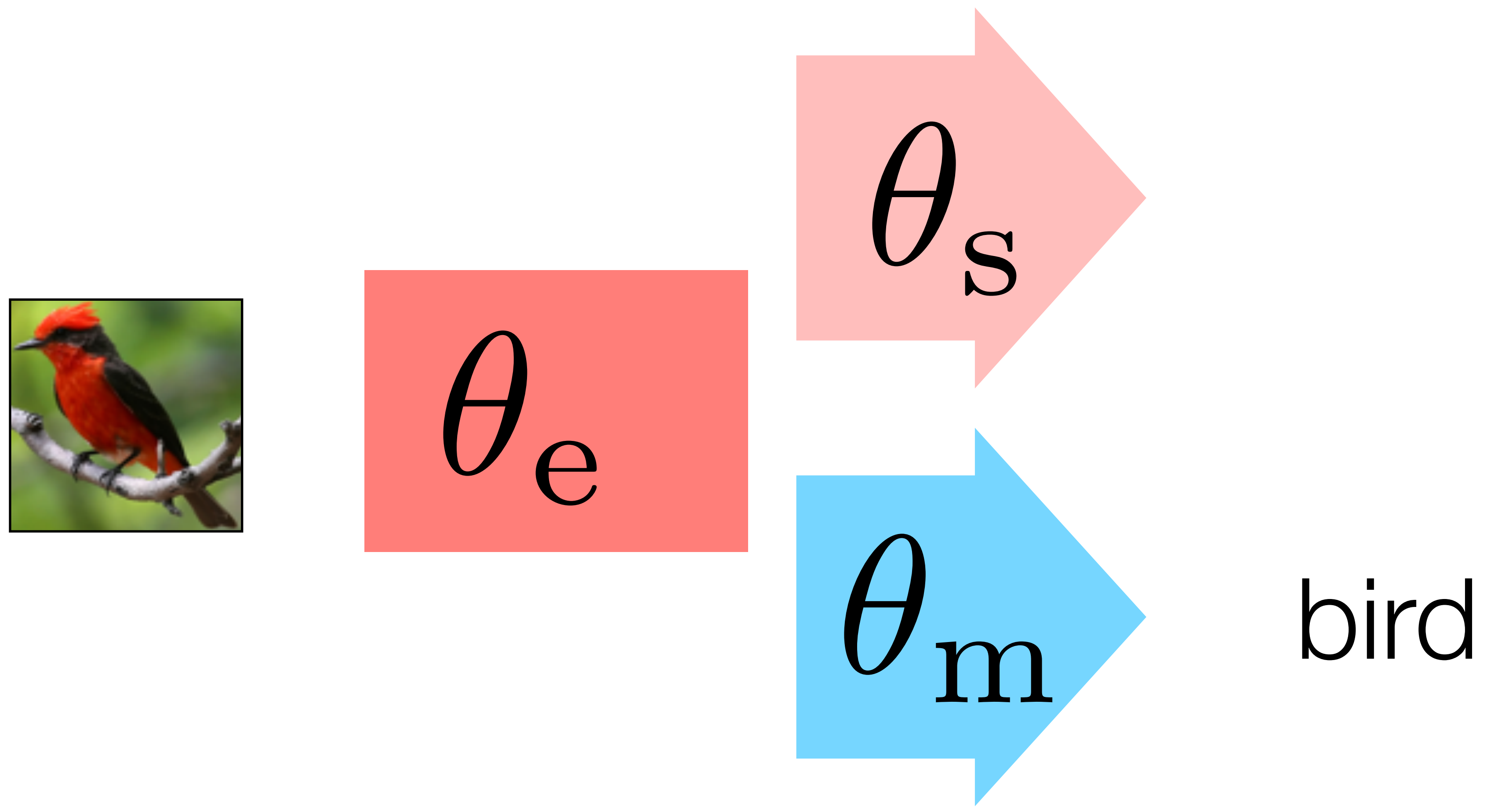
Algorithm for TTT



network
architecture

Algorithm for TTT

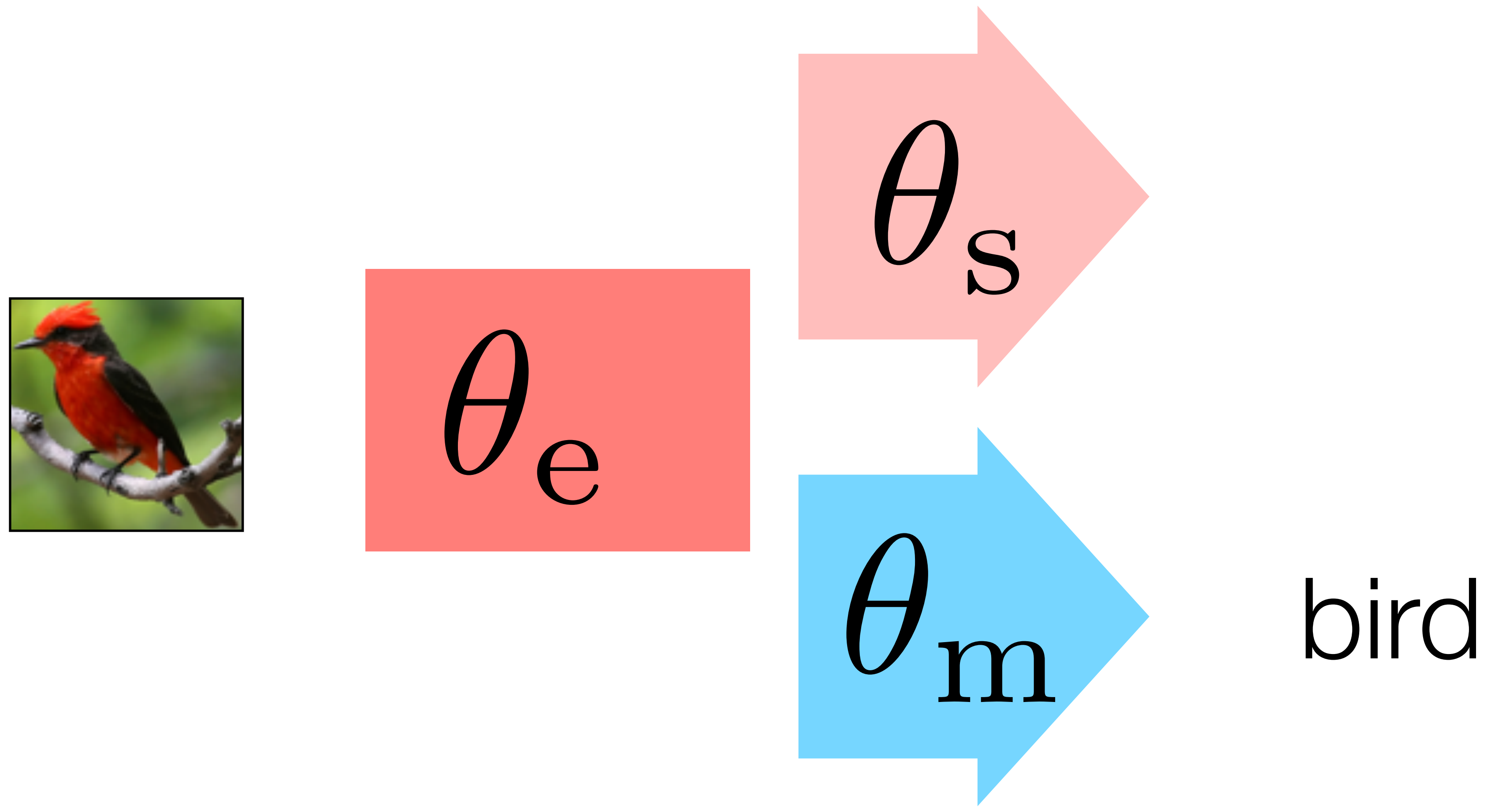
training



Algorithm for TTT

training

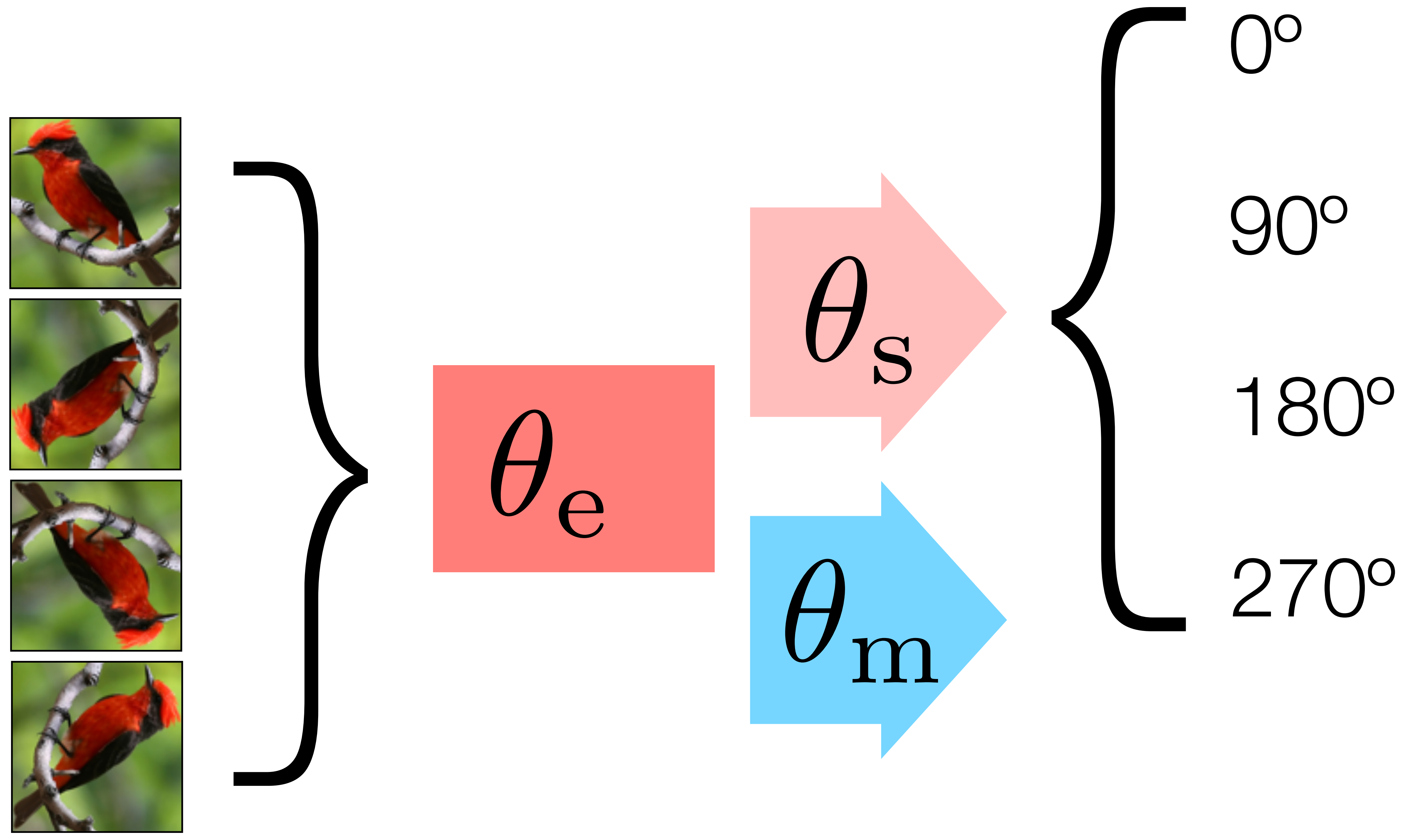
$$\ell_{\mathrm{m}}(x, y; \theta_{\mathrm{e}}, \theta_{\mathrm{m}})$$



Algorithm for TTT

training

$\ell_{\mathrm{m}}(x, y; \theta_{\mathrm{e}}, \theta_{\mathrm{m}})$

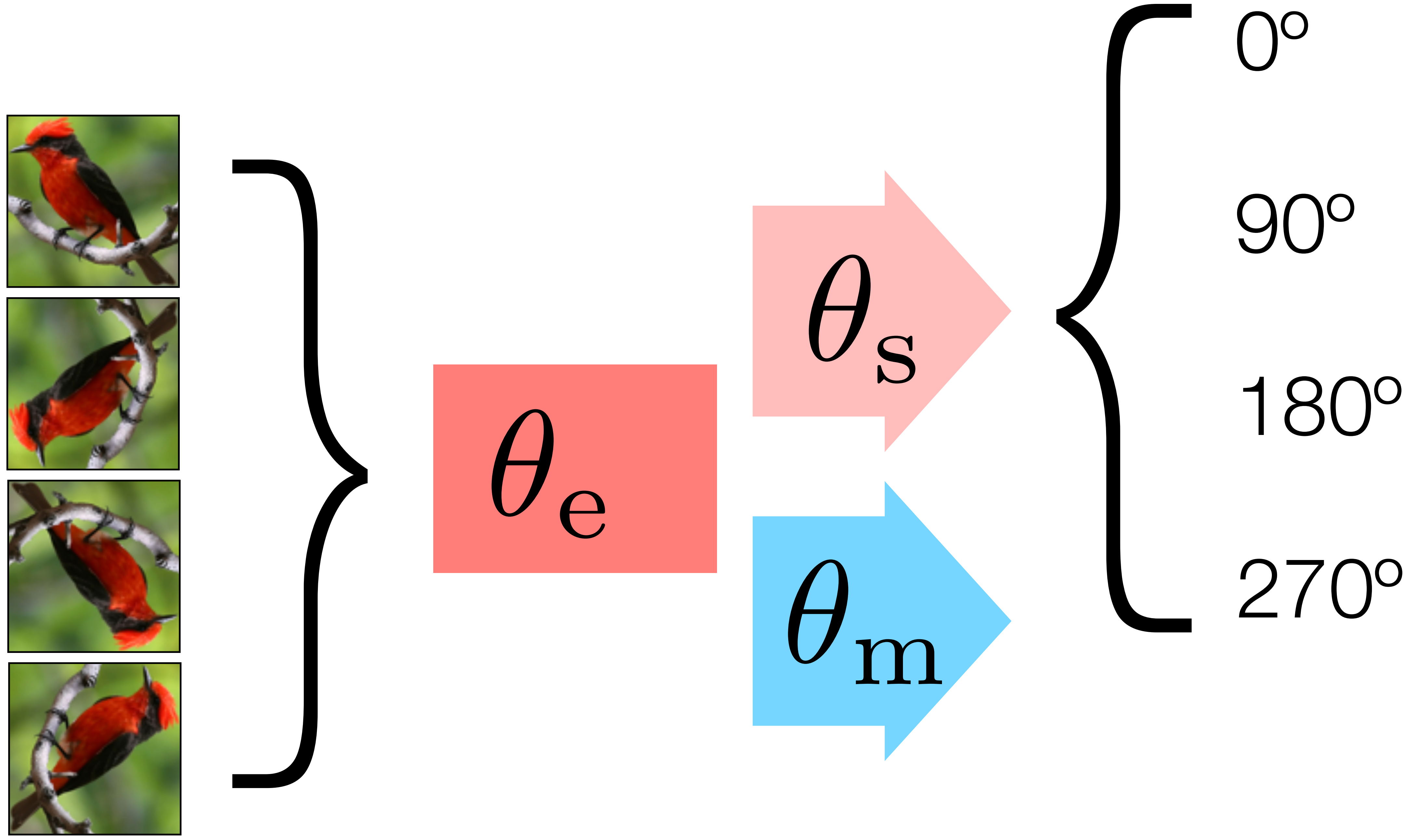


Algorithm for TTT

training

$$\ell_{\mathrm{m}}(x, y; \theta_{\mathrm{e}}, \theta_{\mathrm{m}})$$

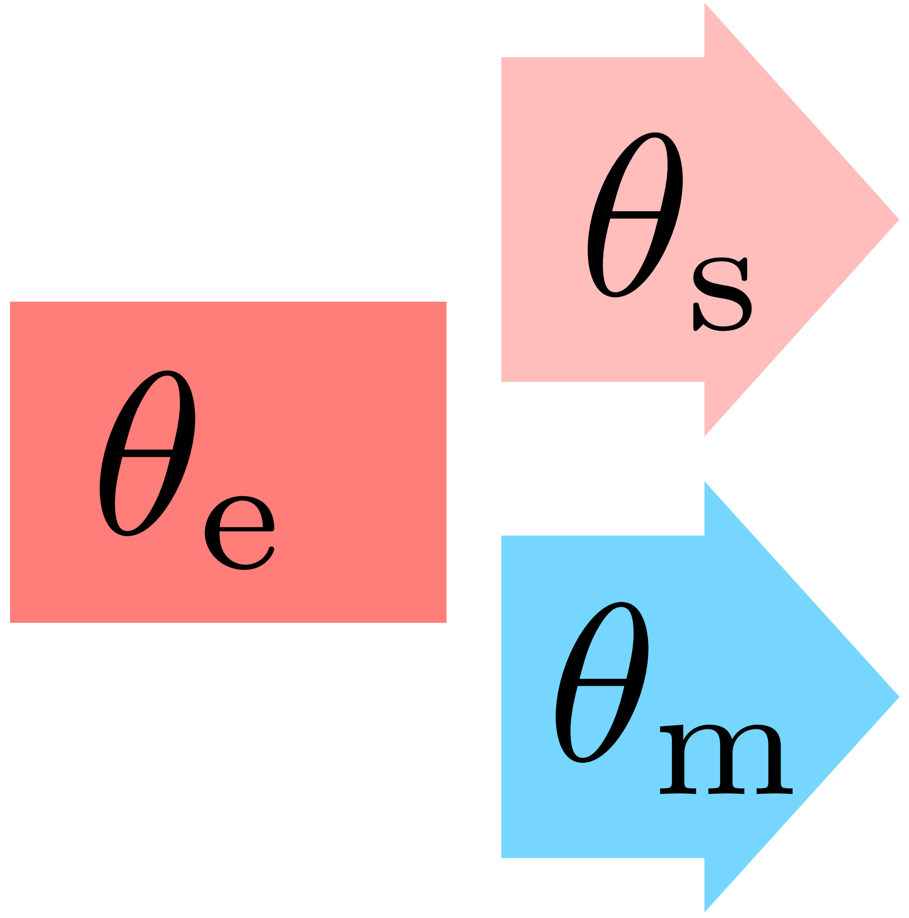
$$+\ell_s(x, y_{\mathrm{s}}; \theta_e, \theta_s)$$



Algorithm for TTT

training

$$\min_{\theta_\mathrm{e},\theta_\mathrm{s},\theta_\mathrm{m}} \mathbb{E}_P\left[\begin{array}{l}\ell_\mathrm{m}(x, y; \theta_\mathrm{e}, \theta_\mathrm{m}) \\ + \ell_s(x, y_\mathrm{s}; \theta_e, \theta_s)\end{array}\right]$$

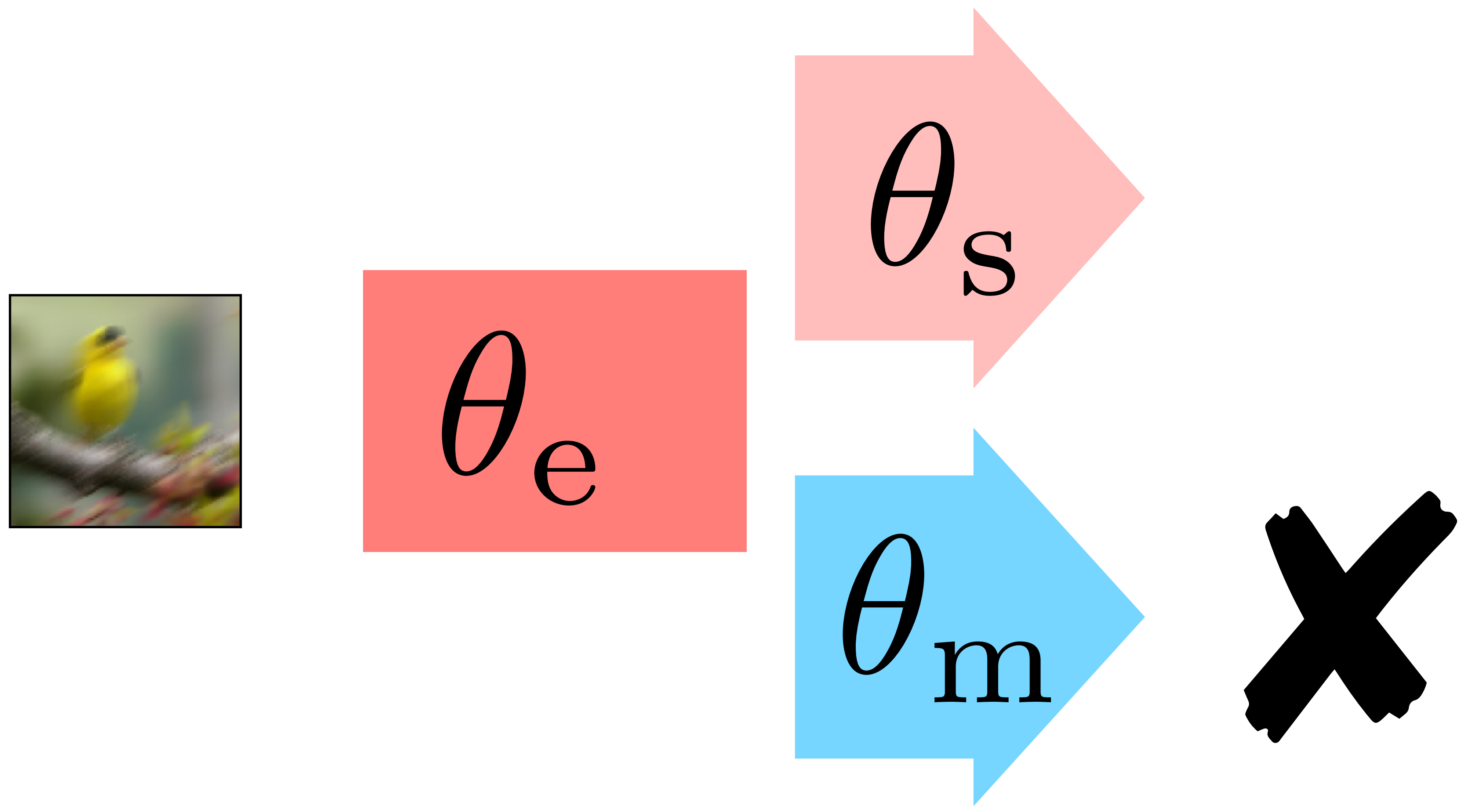


Algorithm for TTT

training

$$\min_{\theta_e,\theta_s,\theta_m} \mathbb{E}_P \left[\begin{array}{l} \ell_m(x, y; \theta_e, \theta_m) \\ + \ell_s(x, y_s; \theta_e, \theta_s) \end{array} \right]$$

testing

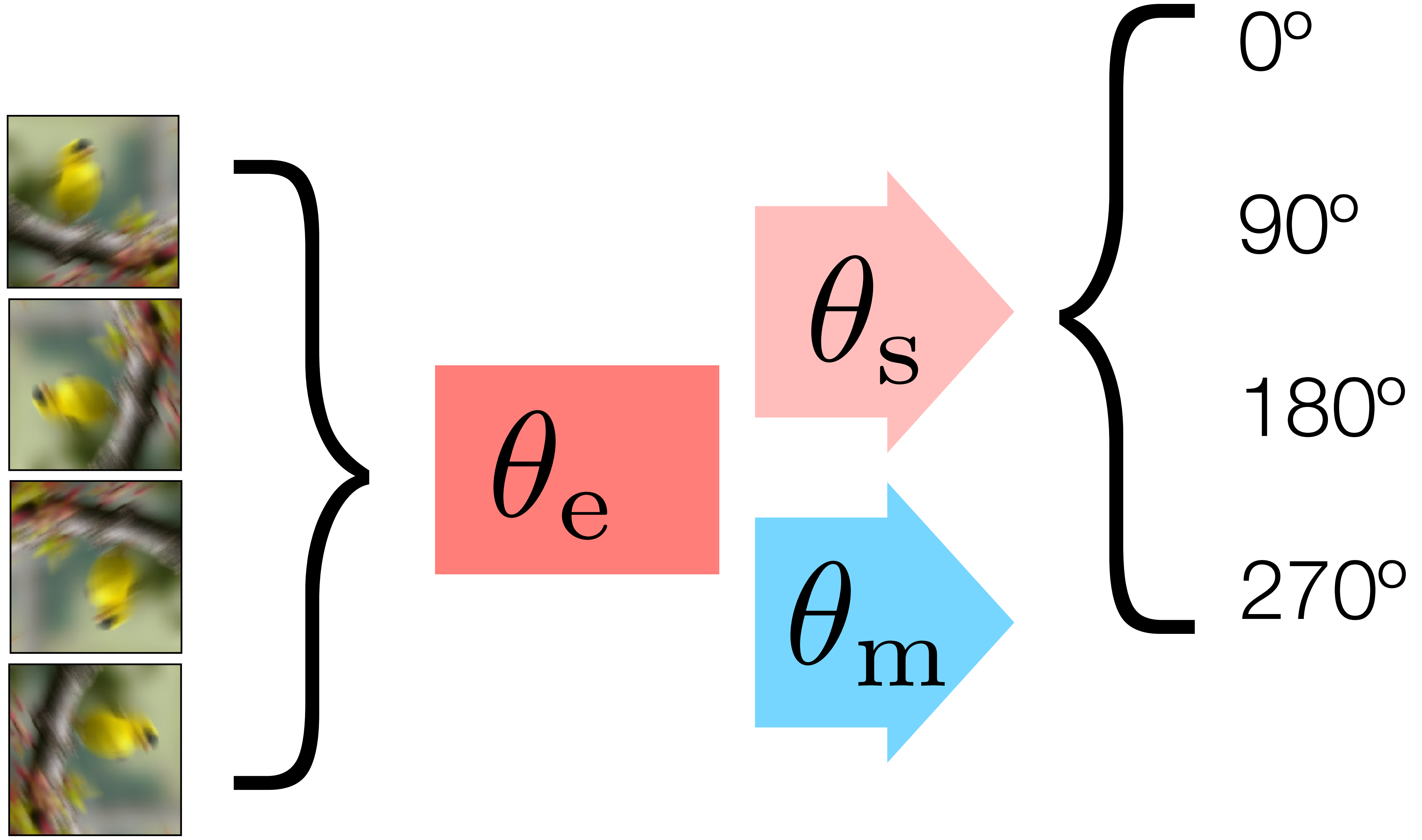


Algorithm for TTT

training

$$\min_{\theta_e, \theta_s, \theta_m} \mathbb{E}_P \left[\begin{aligned} &\ell_m(x, y; \theta_e, \theta_m) \\ &+ \ell_s(x, y_s; \theta_e, \theta_s) \end{aligned} \right]$$

testing



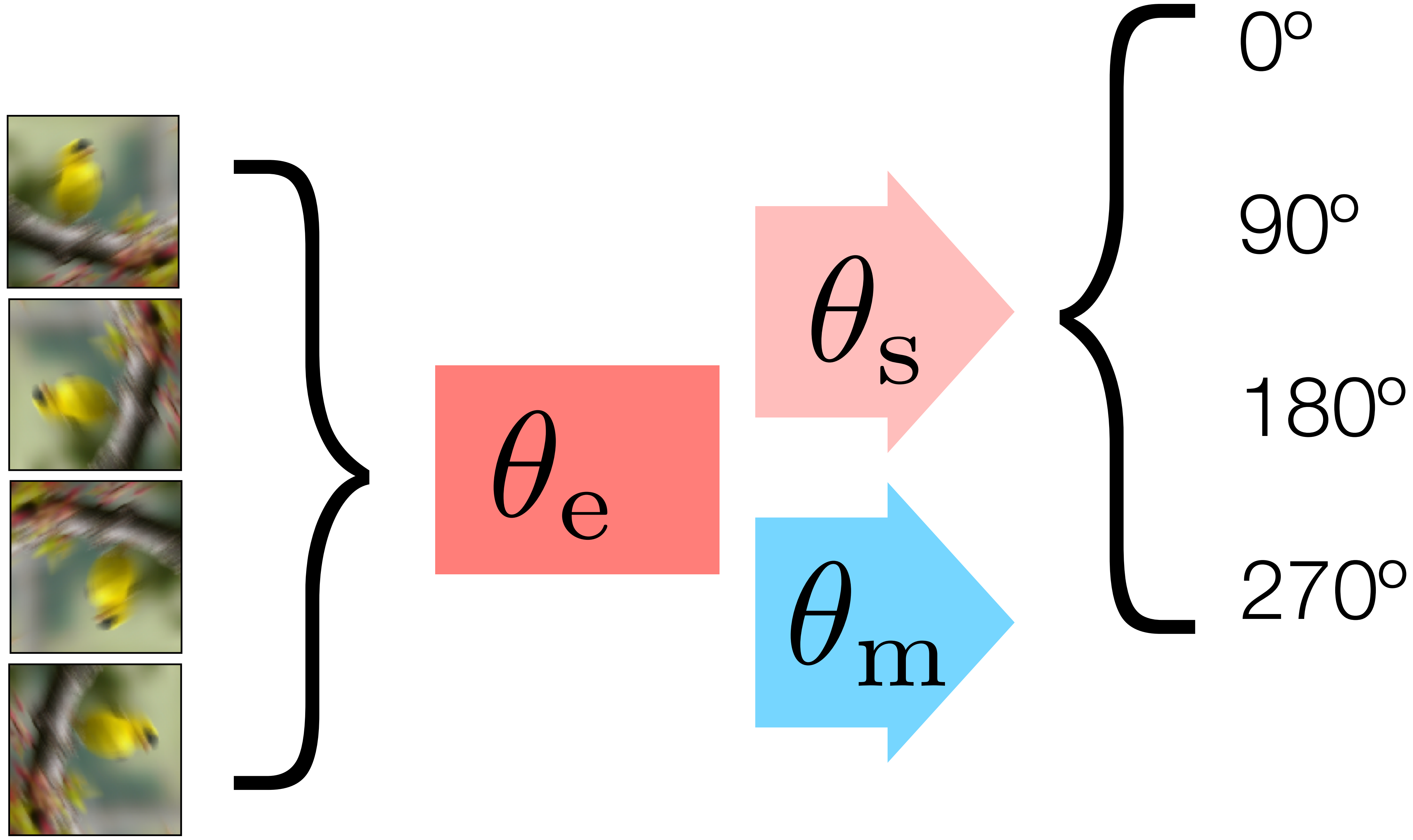
Algorithm for TTT

training

$$\min_{\theta_\mathrm{e},\theta_\mathrm{s},\theta_\mathrm{m}} \mathbb{E}_P \begin{bmatrix} \ell_\mathrm{m}(x, y; \theta_\mathrm{e}, \theta_\mathrm{m}) \\ + \ell_s(x, y_\mathrm{s}; \theta_e, \theta_s) \end{bmatrix}$$

testing

$$\min_{\theta_\mathrm{e},\theta_\mathrm{s}} \left[\ell_s(x, y_\mathrm{s}; \theta_e, \theta_s) \right]$$



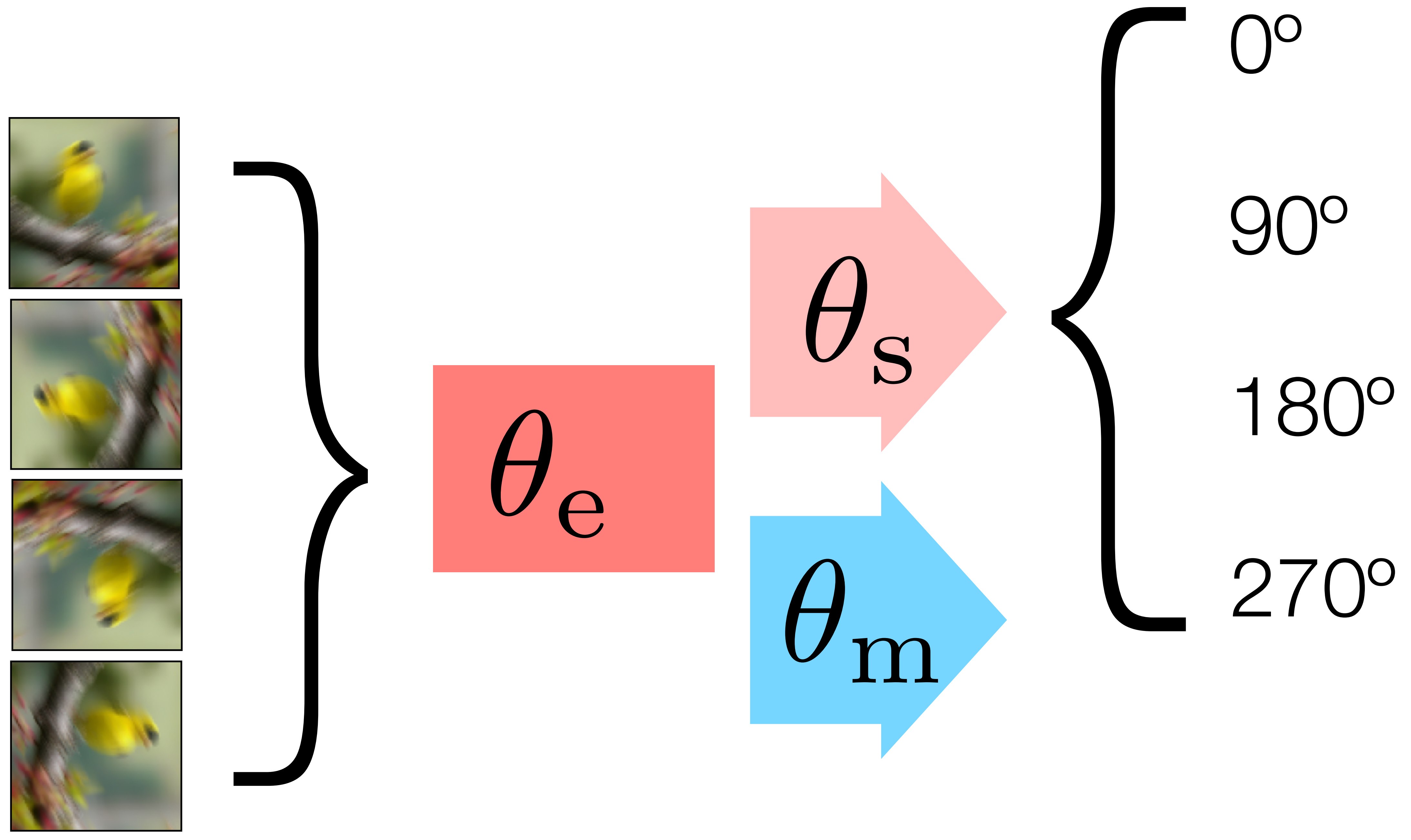
Algorithm for TTT

training

$$\min_{\theta_e,\theta_s,\theta_m} \mathbb{E}_P \left[\begin{array}{l} \ell_m(x, y; \theta_e, \theta_m) \\ + \ell_s(x, y_s; \theta_e, \theta_s) \end{array} \right]$$

testing

$$\min_{\theta_e,\theta_s} \mathbb{E}_Q \left[\ell_s(x, y_s; \theta_e, \theta_s) \right]$$



Algorithm for TTT

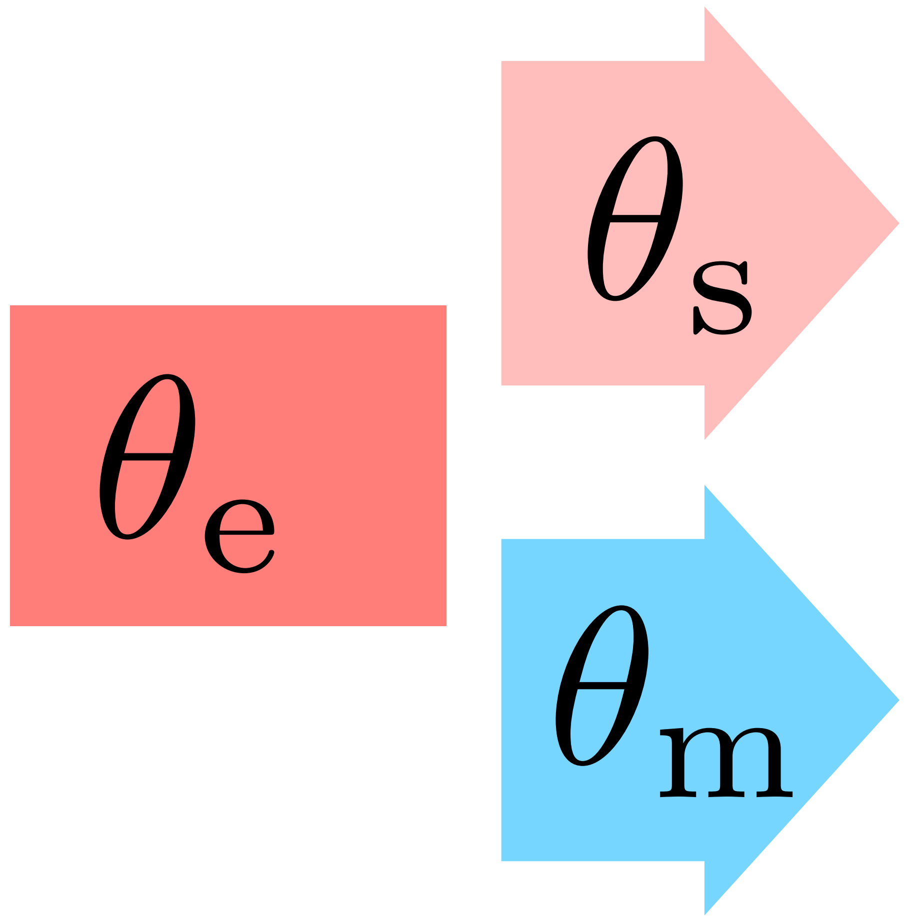
training

$$\min_{\theta_\mathrm{e},\theta_\mathrm{s},\theta_\mathrm{m}} \mathbb{E}_P \left[\begin{array}{l} \ell_\mathrm{m}(x, y; \theta_\mathrm{e}, \theta_\mathrm{m}) \\ + \ell_s(x, y_\mathrm{s}; \theta_e, \theta_s) \end{array} \right]$$

testing

$$\min_{\theta_\mathrm{e},\theta_\mathrm{s}} \mathbb{E}_Q \left[\ell_s(x, y_\mathrm{s}; \theta_e, \theta_s) \right]$$

$\rightarrow \theta(x)$: make prediction on x



Algorithm for TTT

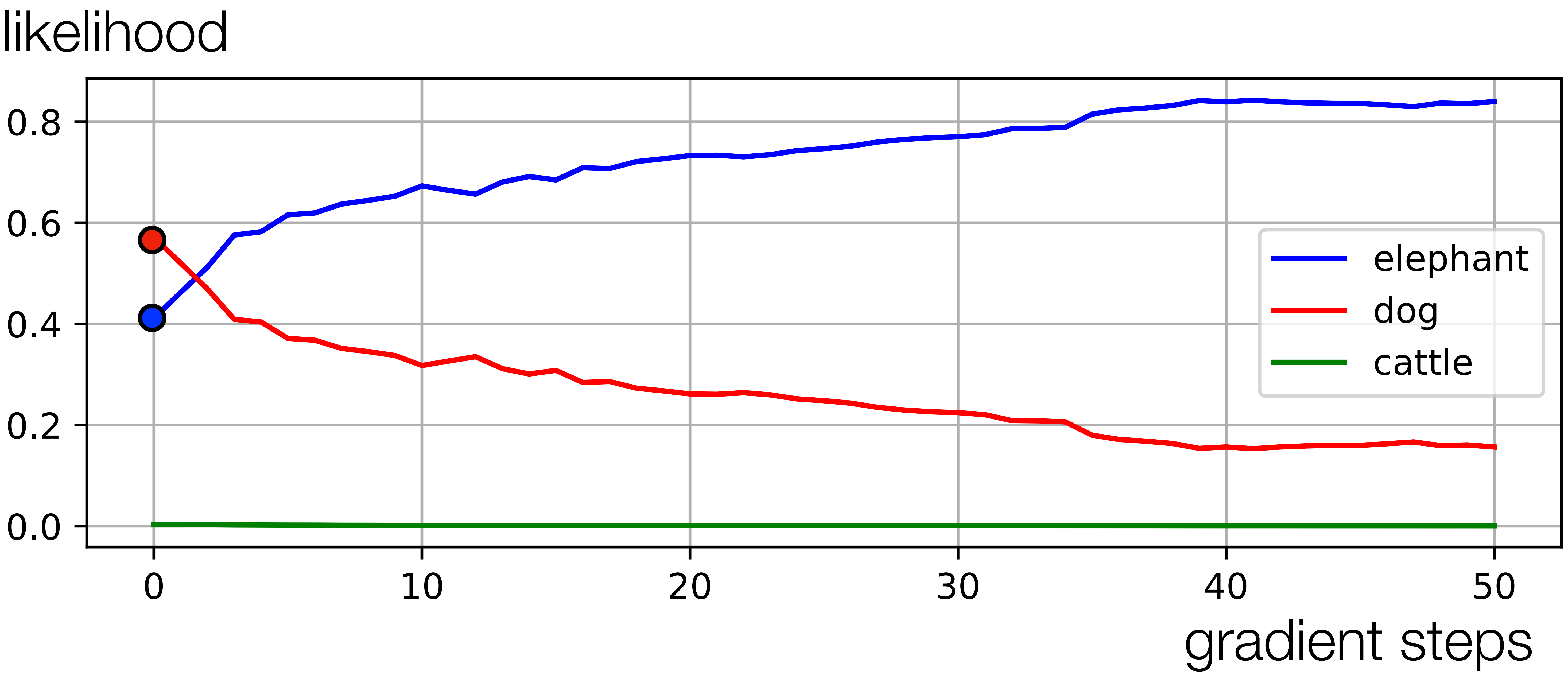
training

$$\min_{\theta_{\mathrm{e}},\theta_{\mathrm{s}},\theta_{\mathrm{m}}} \mathbb{E}_P \left[\begin{array}{l} \ell_{\mathrm{m}}(x, y; \theta_{\mathrm{e}}, \theta_{\mathrm{m}}) \\ + \ell_s(x, y_{\mathrm{s}}; \theta_e, \theta_s) \end{array} \right]$$

testing

$$\min_{\theta_{\mathrm{e}},\theta_{\mathrm{s}}} \mathbb{E}_Q \left[\ell_s(x, y_{\mathrm{s}}; \theta_e, \theta_s) \right]$$

$\rightarrow \theta(x)$: make prediction on x



Algorithm for TTT

training

$$\min_{\theta_e, \theta_s, \theta_m} \mathbb{E}_P \left[\begin{array}{l} \ell_m(x, y; \theta_e, \theta_m) \\ + \ell_s(x, y_s; \theta_e, \theta_s) \end{array} \right]$$

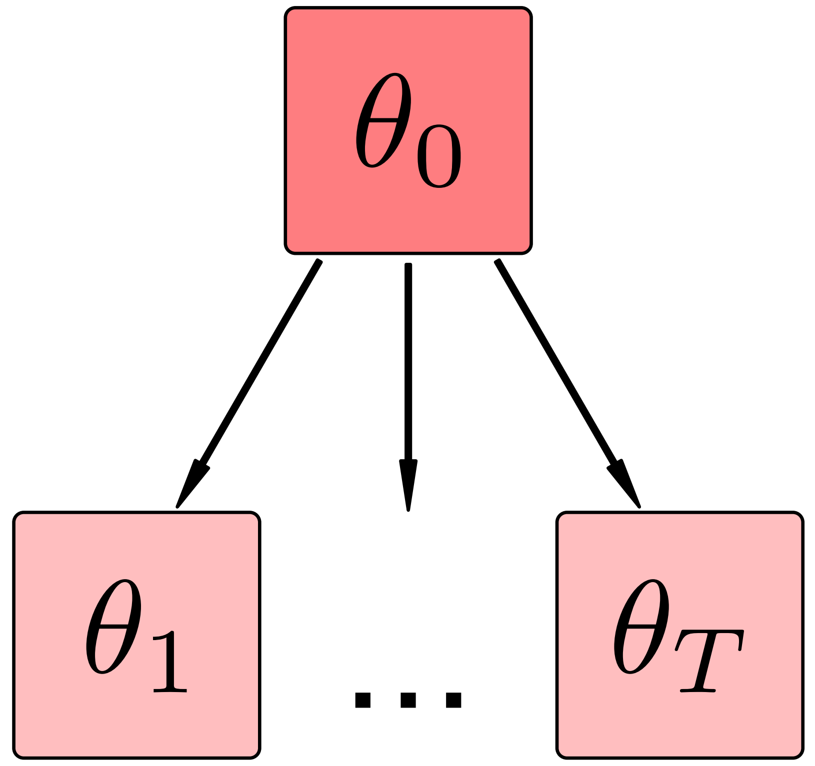
testing

$$\min_{\theta_e, \theta_s} \mathbb{E}_Q \left[\ell_s(x, y_s; \theta_e, \theta_s) \right]$$

$\rightarrow \theta(x)$: make prediction on x

multiple test samples $x_1, ..., x_T$

θ_0: parameters after joint training



Algorithm for TTT

training

$$\min_{\theta_e, \theta_s, \theta_m} \mathbb{E}_P \left[\begin{array}{l} \ell_m(x, y; \theta_e, \theta_m) \\ + \ell_s(x, y_s; \theta_e, \theta_s) \end{array} \right]$$

testing

$$\min_{\theta_e, \theta_s} \mathbb{E}_Q \left[\ell_s(x, y_s; \theta_e, \theta_s) \right]$$

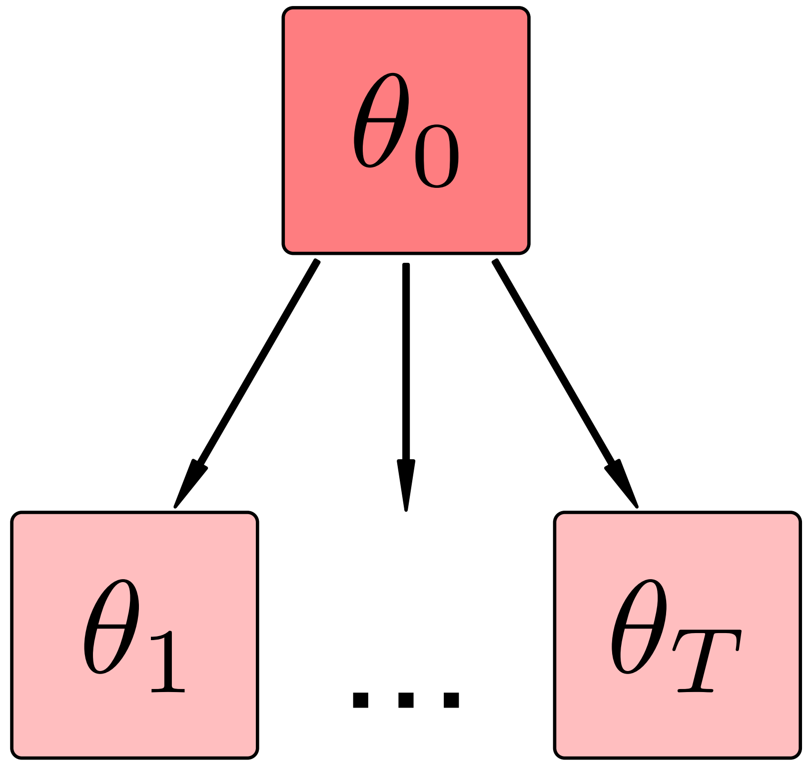
$\rightarrow \theta(x)$: make prediction on x

multiple test samples $x_1, \ldots, x_T$

θ_0: parameters after joint training

standard version

no assumption on the test samples



Algorithm for TTT

training

$$\min_{\theta_e, \theta_s, \theta_m} \mathbb{E}_P \left[\begin{array}{l} \ell_m(x, y; \theta_e, \theta_m) \\ + \ell_s(x, y_s; \theta_e, \theta_s) \end{array} \right]$$

testing

$$\min_{\theta_e, \theta_s} \mathbb{E}_Q \left[\ell_s(x, y_s; \theta_e, \theta_s) \right]$$

$\rightarrow \theta(x)$: make prediction on x

multiple test samples $x_1, \ldots, x_T$

θ_0: parameters after joint training

standard version

no assumption on the test samples

$$\begin{tikzcd} & \theta_0 \arrow[dl] \arrow[d] \arrow[dr] & \\ \theta_1 & \ldots & \theta_T \end{tikzcd}$$

online version

$x_1, \ldots, x_T$ come from the same Q

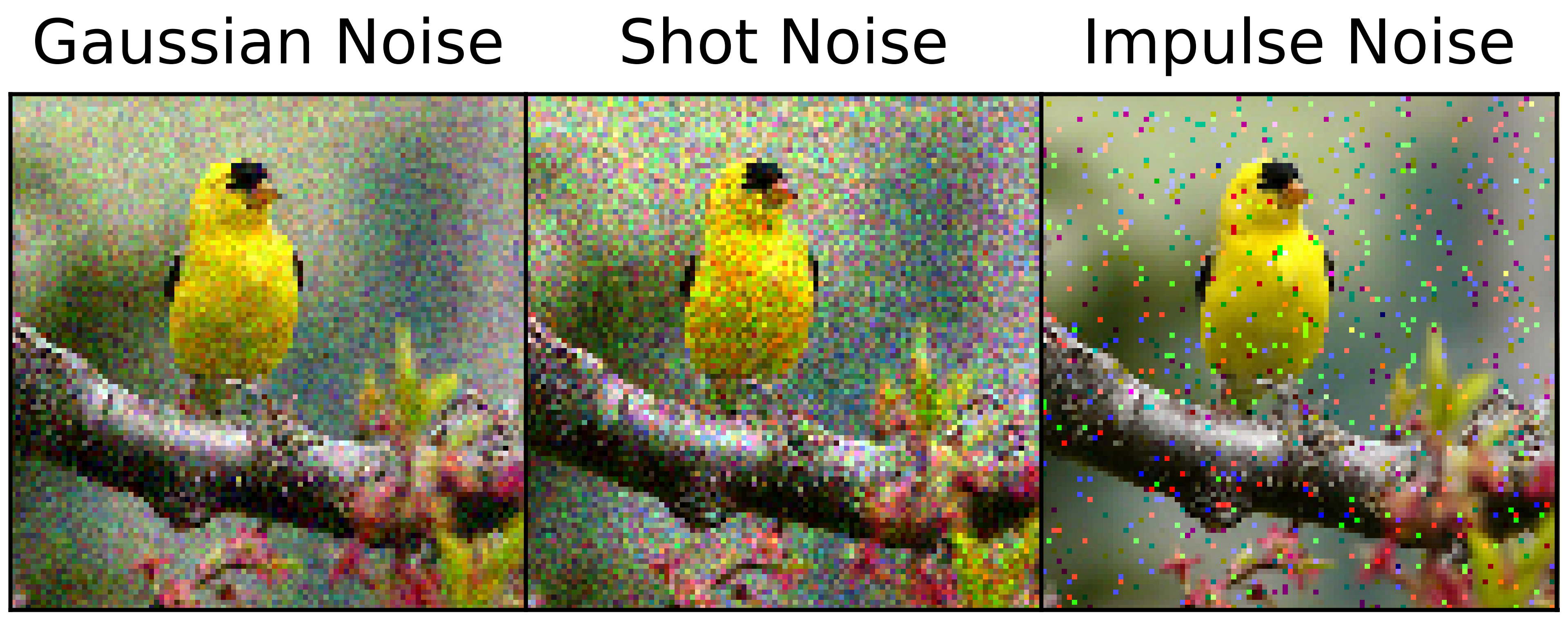
or smoothly changing $Q_1, \ldots, Q_T$

$$\theta_0 \rightarrow \theta_1 \rightarrow \ldots \rightarrow \theta_T$$

Results

Object recognition with corruptions

- 15 corruptions
- CIFAR-10: 10 classes
- ImageNet: 1000 classes
- No knowledge of the corruptions during training



Benchmarking Neural Network Robustness to Common Corruptions and Perturbations
Hendrycks and Dietterich, 2018

Results on CIFAR-10-C

Joint training reported here is our improved implementation of their method. Please see our paper for clarification, and their paper for their original results.

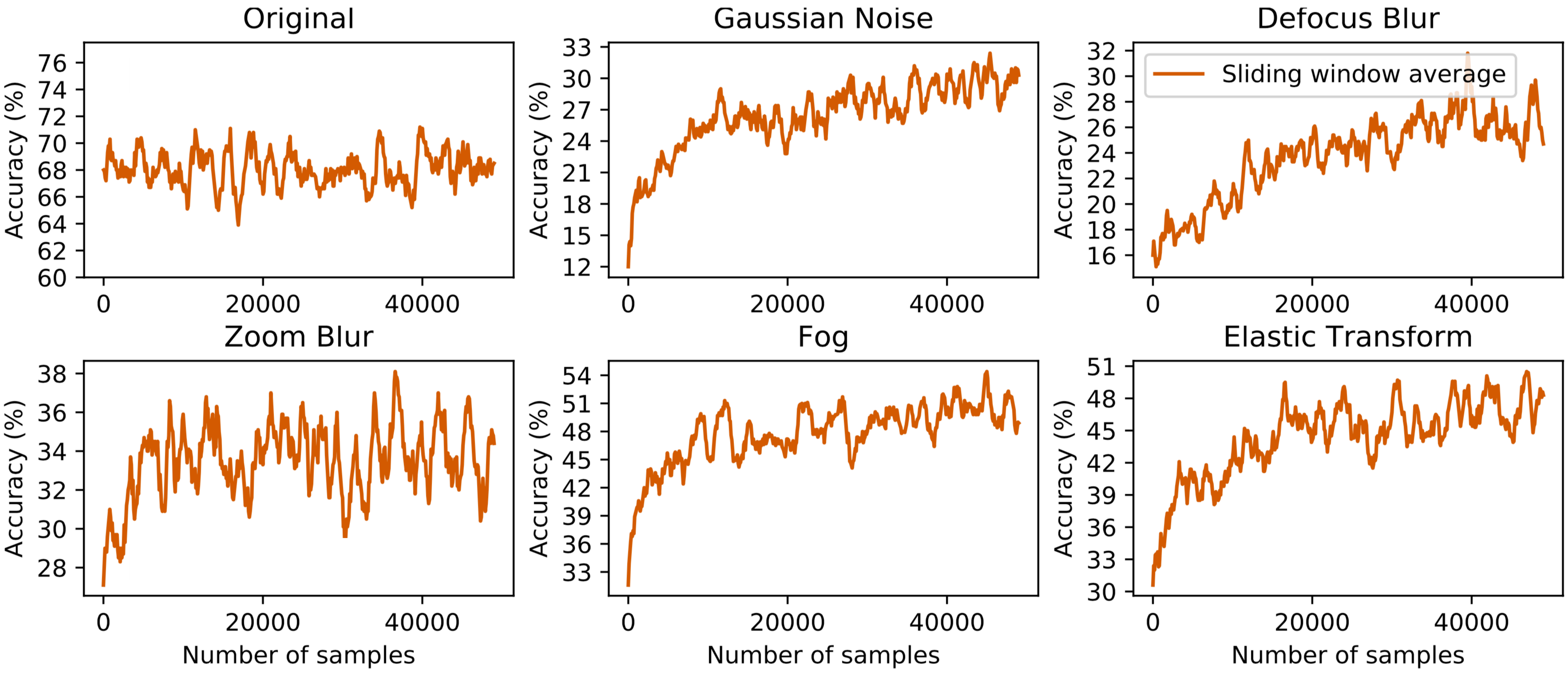
Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty
Hendrycks, Mazeika, Kadavath and Song, 2019

Results on ImageNet-C

Joint training reported here is our improved implementation of their method. Please see our paper for clarification, and their paper for their original results.

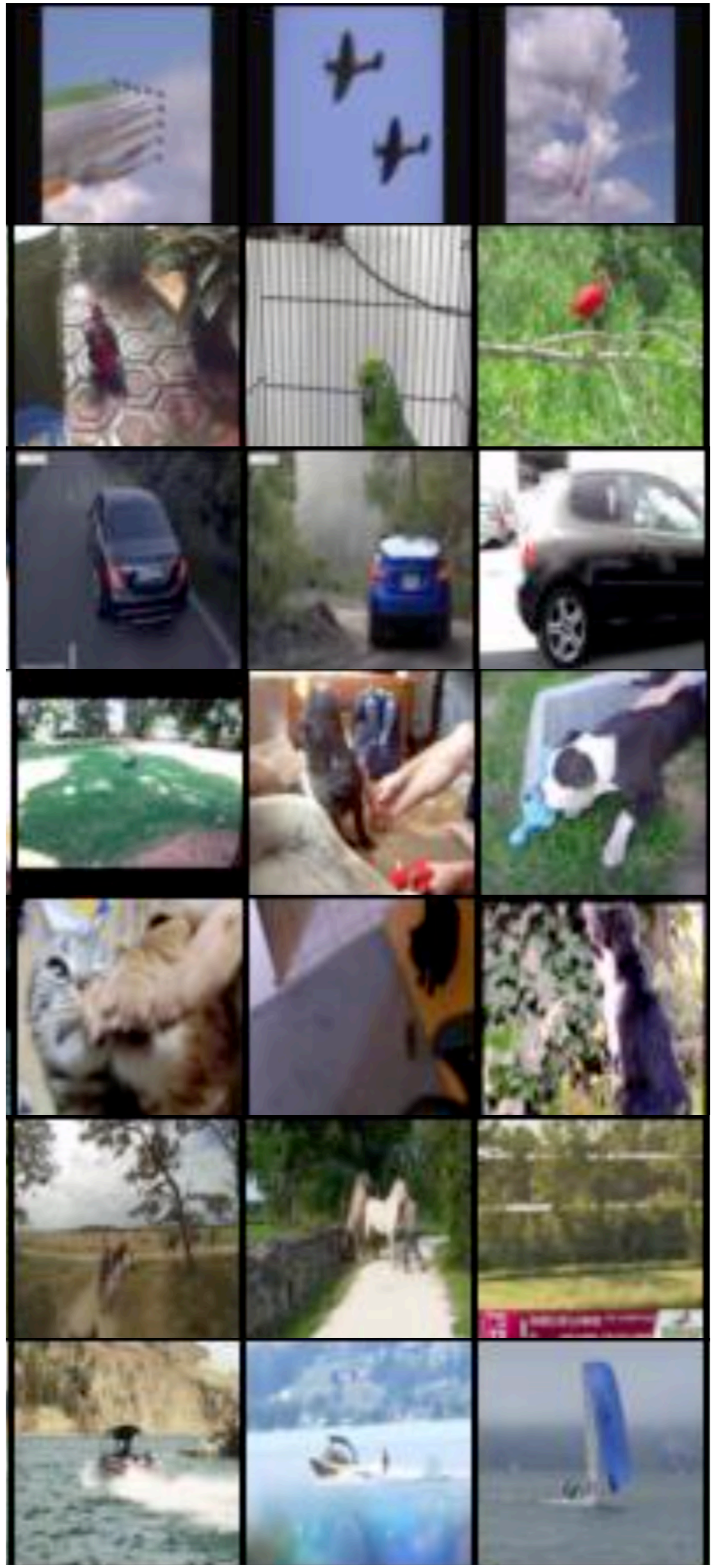
Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty
Hendrycks, Mazeika, Kadavath and Song, 2019

The online version on ImageNet-C



From still images to videos

- Videos of objects in motion
- 7 classes from CIFAR-10
- 30 classes from ImageNet
- Train on CIFAR-10 / ImageNet
- Test on video frames



airplane
bird
car
dog
cat
horse
ship

A systematic framework for natural perturbations from videos
Shankar, Dave, Roelofs, Ramanan, Recht and Schmidt, 2019

Results

Method	CIFAR-10 accuracy (%)	ImageNet accuracy (%)
Object recognition task only	41.4	62.7
Joint training (Hendrycks et al. 2019)	42.4	63.5
TTT standard	45.2	63.8
TTT online	45.4	64.3

Positive examples

Join training: dog

TTT: elephant

Join training: dog

TTT: cattle

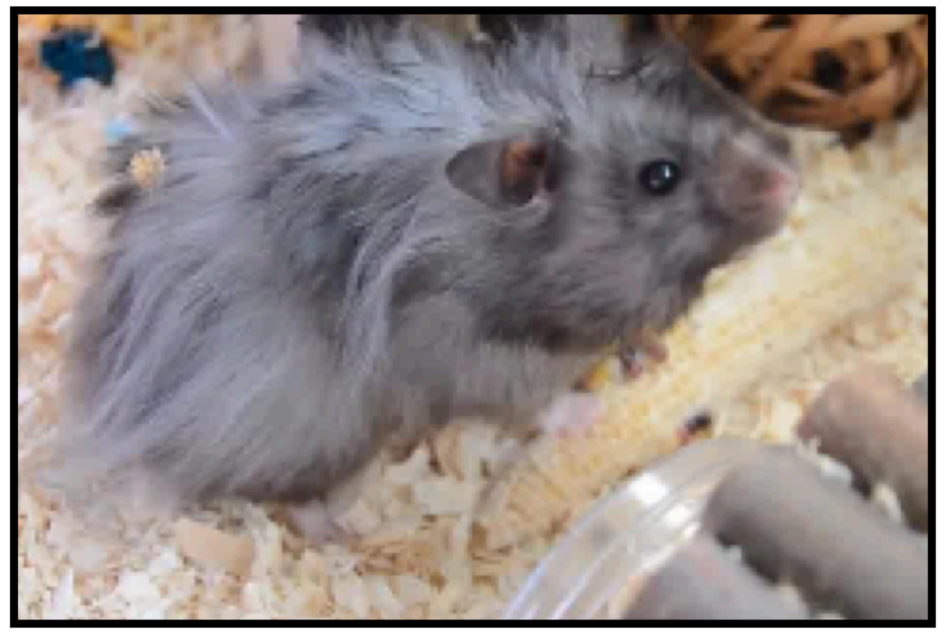
Join training: car

TTT: bus

Results

Method	CIFAR-10 accuracy (%)	ImageNet accuracy (%)
Object recognition task only	41.4	62.7
Joint training (Hendrycks et al. 2019)	42.4	63.5
TTT standard	45.2	63.8
TTT online	45.4	64.3

Negative examples



Join training: hamster

TTT: cat

Join training: snake

TTT: lizard

Join training: turtle

TTT: lizard

Results

Method	CIFAR-10 accuracy (%)	ImageNet accuracy (%)
Object recognition task only	41.4	62.7
Joint training (Hendrycks et al. 2019)	42.4	63.5
TTT standard	45.2	63.8
TTT online	45.4	64.3

Negative examples

Join training: airplane

TTT: bird

Join training: airplane

TTT: watercraft

Rotation prediction is quite limiting!

CIFAR-10.1

- New test set on CIFAR-10
- Cannot notice the distribution shifts
- Still an open problem

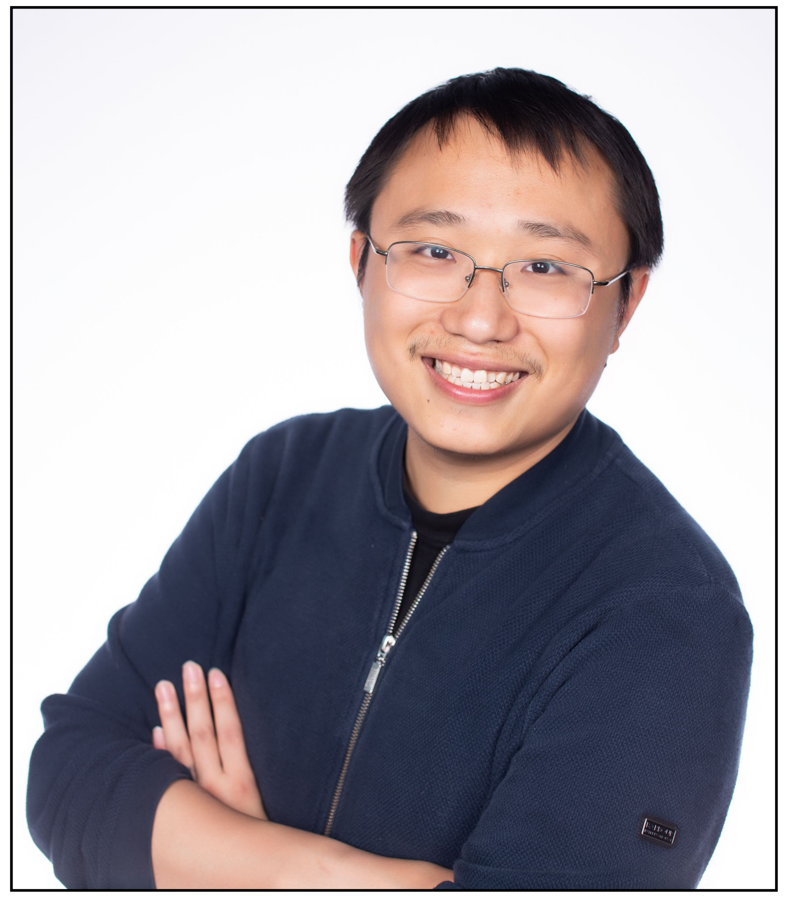
Do CIFAR-10 Classifiers Generalize to CIFAR-10?
Recht, Roelofs, Schmidt and Shankar, 2019

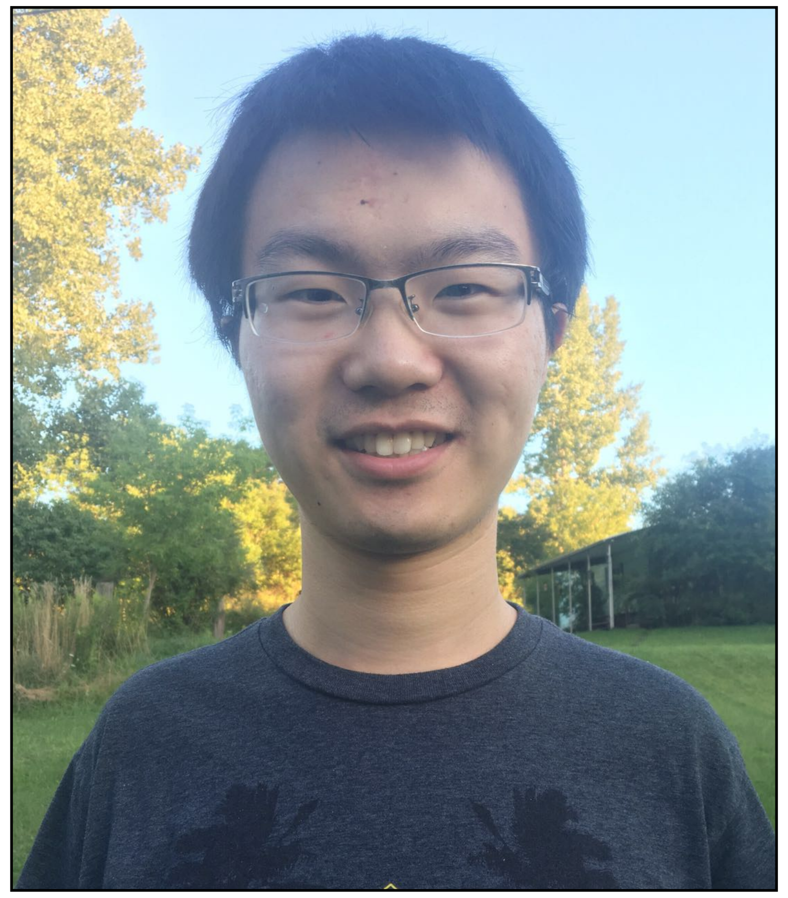
Results

Method	Error (%)
Object recognition task only	17.4
Joint training (Hendrycks et al. 2019)	16.7
TTT standard	15.9

Conclusion

- Boundary between labeled and unlabeled samples
  - Broken down by self-supervision
- Boundary between training and testing
  - We are trying to break this down


Xiaolong Wang


Zhuang Liu


John Miller


Alyosha Efros


Moritz Hardt