Electronic Imaging 2021, Image Processing: Algorithms and Systems

Does end-to-end trained deep model always perform better than non-end-to-end counterpart?

Ikuro Sato[1,2]
Guoqing Liu[1]
Kohta Ishikawa[1]
Teppei Suzuki[1]
Masayuki Tanaka[2]

[1]Denso IT Laboratory, Inc., Japan
[2]Tokyo Institute of Technology, Japan

Outline

- Introduction
- Overview: FOCA

FOCA: Feature-extractor Optimization through Classifier Anonymization
I. Sato, et al., ICML2019.

- Experiment
  - Improvement over Sato et al.
  - Comparison with end-to-end training methods
  - Effect of network fine-tuning after FOCA
- Summary

Outline

- **Introduction**
- Overview: FOCA

FOCA: Feature-extractor Optimization through Classifier Anonymization
I. Sato, et al., ICML2019.

- Experiment
  - Improvement over Sato et al.
  - Comparison with end-to-end training methods
  - Effect of network fine-tuning after FOCA
- Summary

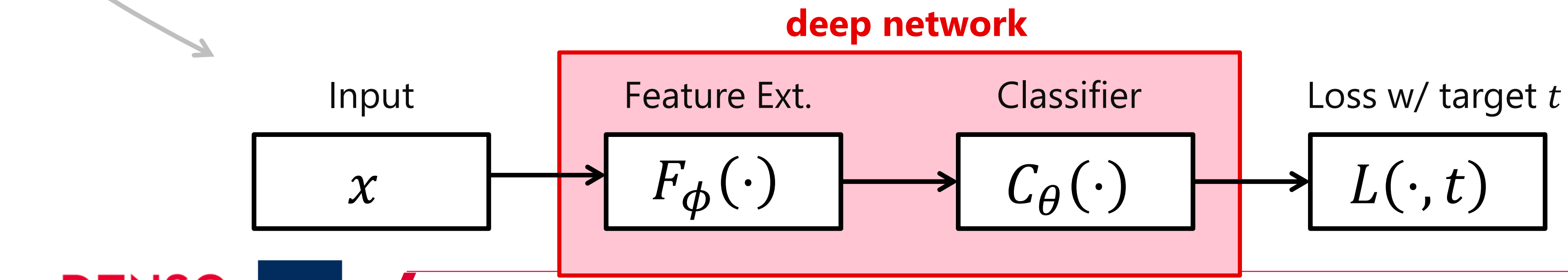
Flourishing E2E network optimization

Successful by replacing intermediate tasks with learnable layers

eg)

- CNN ← feature extractor (eg. SIFT) + pooling (eg. Fisher Vector) + classif.
- Spatial Transformer ← coordinate preprocessing + classif.
- Faster RCNN ← region proposal + classif.
- Monocular depth ← optical flow + epipolar geometry estimation
- PointNet ← voxelization + classif.

D. Lowe, 2004. F. Perronnin+, 2007.
M. Jaderberg+, 2015.
S. Ren+, 2015.
C. Godard+, 2016.
C. Qi+, 2017.



Is E2E optimization always good?

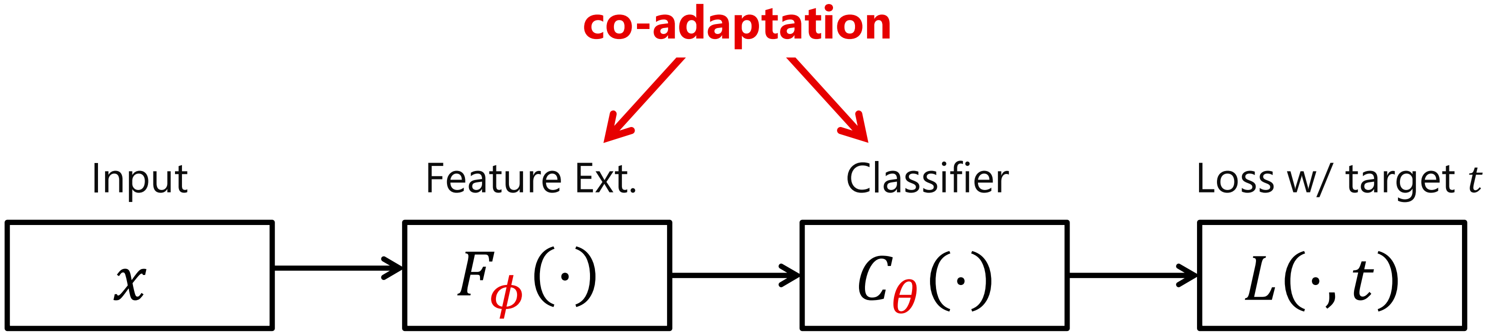
Co-adaptation between feature extractor and classifier can occur.

G. Hinton+, 2012.

- Feature distribution is only good at a particular decision boundary.
- Vice versa.

E2E optimization

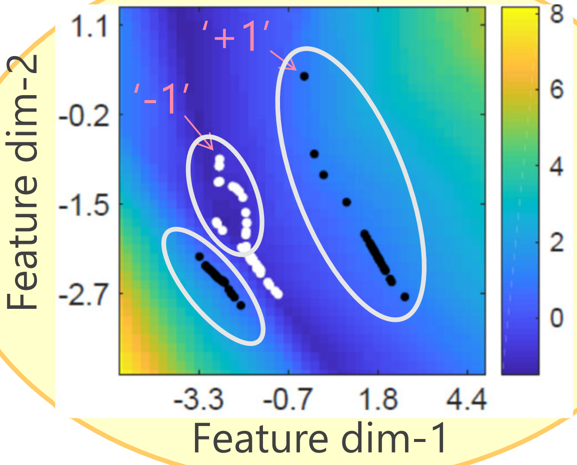
$$(\phi^\star, \theta^\star) = \arg\min_{\phi,\theta} \frac{1}{\|\mathcal{D}\|_0} \sum_{(x,t)\in\mathcal{D}} L\left(C_\theta\left(F_\phi(x)\right), t\right)$$



Is E2E optimization always good?

Worst cases: excessively complex feature distribution

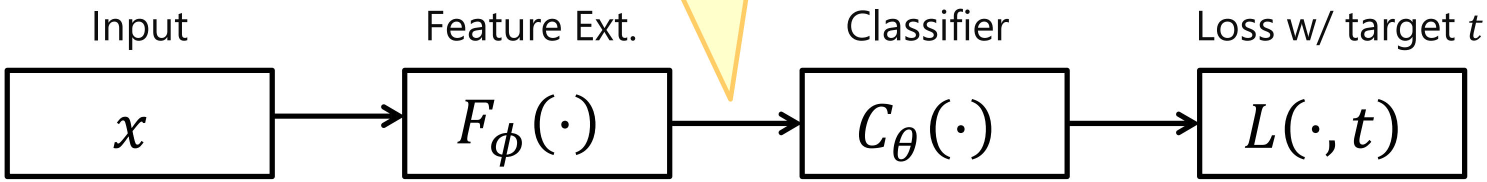
Toy ex.)
2-class regression



color: C_θ value

- Disjointed
- Split

→ Vulnerable to a small change in the feature distribution, *i.e.*, bad transferability. J. Yosinski+, 2014.



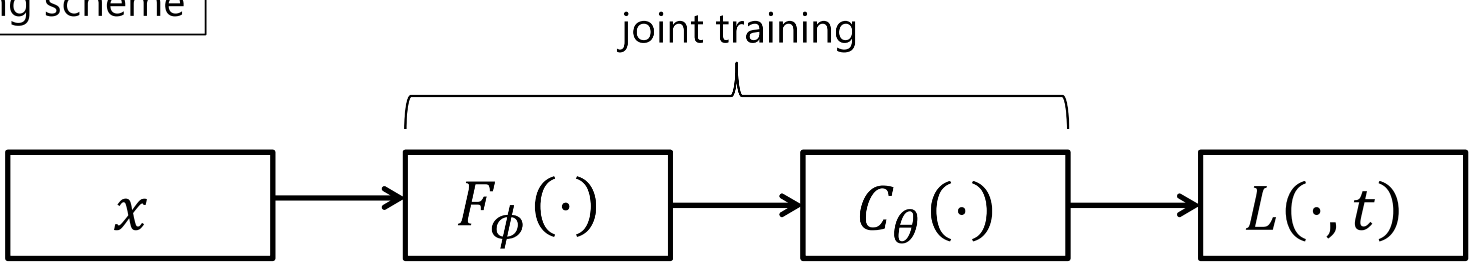
Question we try to answer

Q. Does end-to-end (E2E) trained deep model always perform better than non-end-to-end counterpart?

A. **Not always.** We show empirical evidences where a non-E2E training method known as FOCA outperforms strong E2E counterparts in image classification tasks.

FOCA: Feature-extractor Optimization through Classifier Anonymization

E2E training scheme



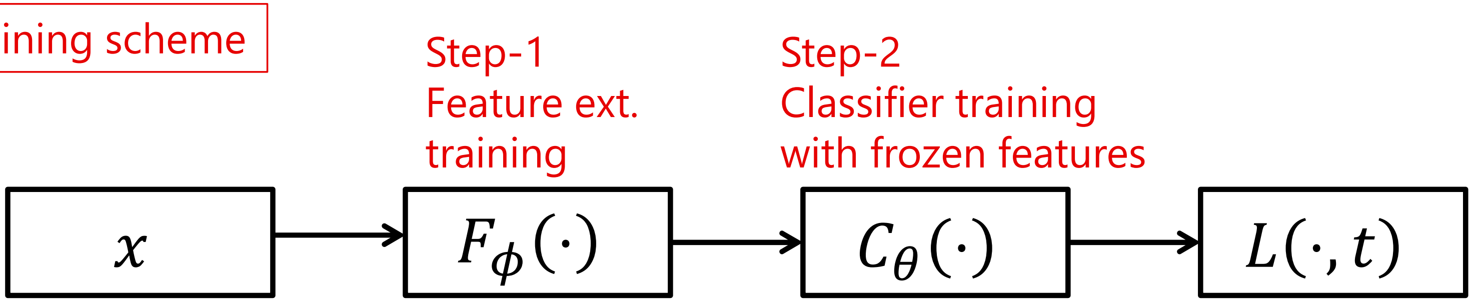
Our answer

Q. Does end-to-end (E2E) trained deep model always perform better than non-end-to-end counterpart?

A. **Not always.** We show empirical evidences where a non-E2E training method known as FOCA outperforms strong E2E counterparts in image classification tasks.

FOCA: Feature-extractor Optimization through Classifier Anonymization

FOCA's training scheme



Step-1
Feature ext.
training

Step-2
Classifier training
with frozen features

$$x \rightarrow F_{\phi}(\cdot) \rightarrow C_{\theta}(\cdot) \rightarrow L(\cdot, t)$$

Outline

- Introduction
- **Overview: FOCA**

  FOCA: Feature-extractor Optimization through Classifier Anonymization
  I. Sato, et al., ICML2019.
- Experiment
  - Improvement over Sato et al.
  - Comparison with end-to-end training methods
  - Effect of network fine-tuning after FOCA
- Summary

FOCA: Feature-extractor Optimization through Classifier Anonymization

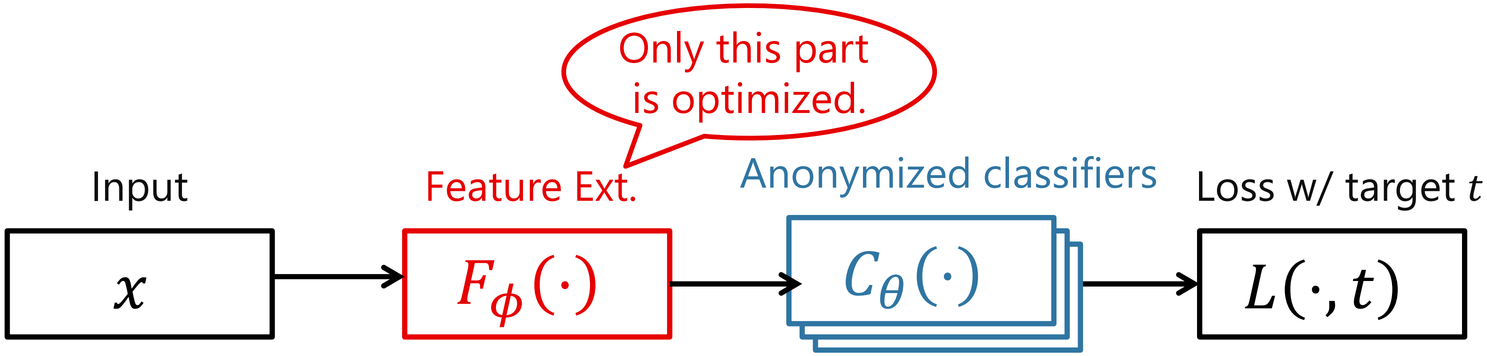
FOCA

$$\phi^\star = \arg\min_{\phi} \frac{1}{\|\mathcal{D}\|_0} \sum_{(x,t)\in\mathcal{D}} \mathbb{E}_{\theta\sim\Theta_\phi} L\left(C_\theta\left(F_\phi(x)\right), t\right)$$

Not $\arg\min_{\phi,\,\theta} \cdots$

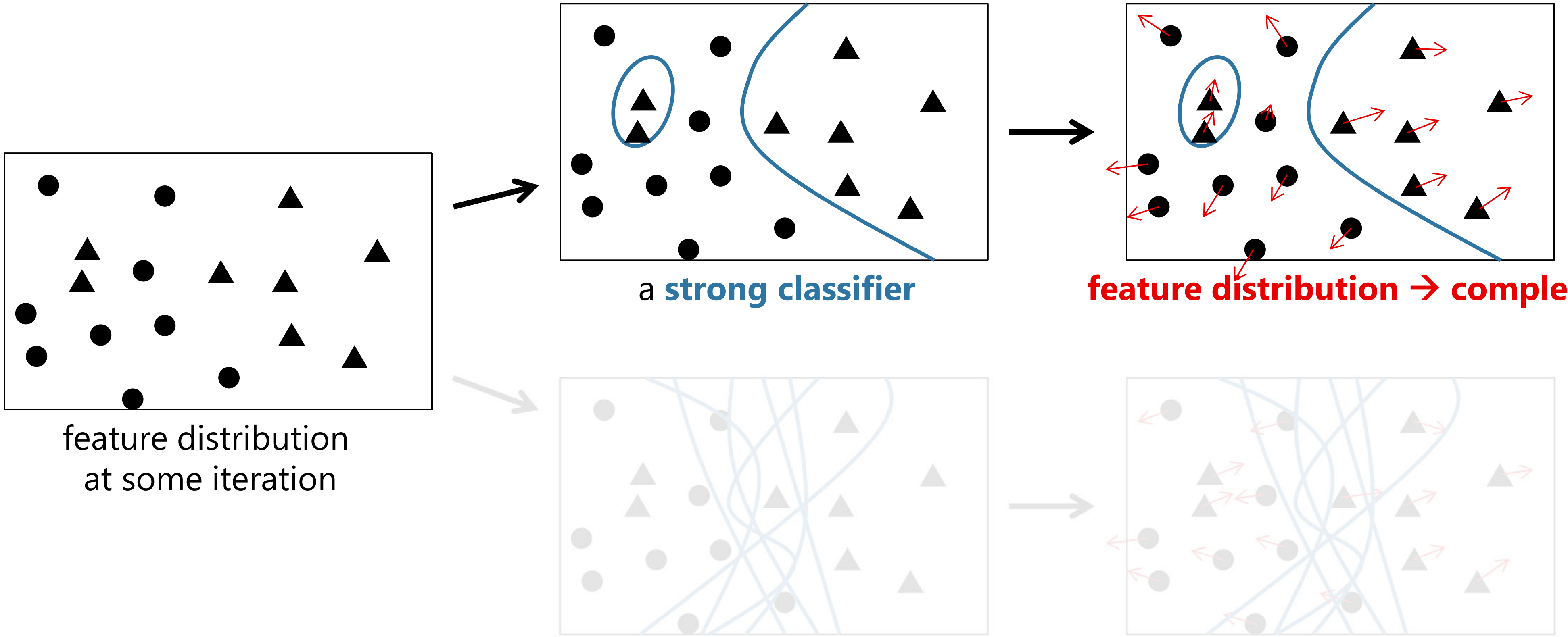
Random weak classifier: $\theta\sim\Theta_\phi$

→ Feature extractor is optimized wrt an ensemble of weak classifiers, not a particular strong classifier.



Why *weak* ??

strong classifier → Features do adapt…



feature distribution
at some iteration

a **strong classifier**

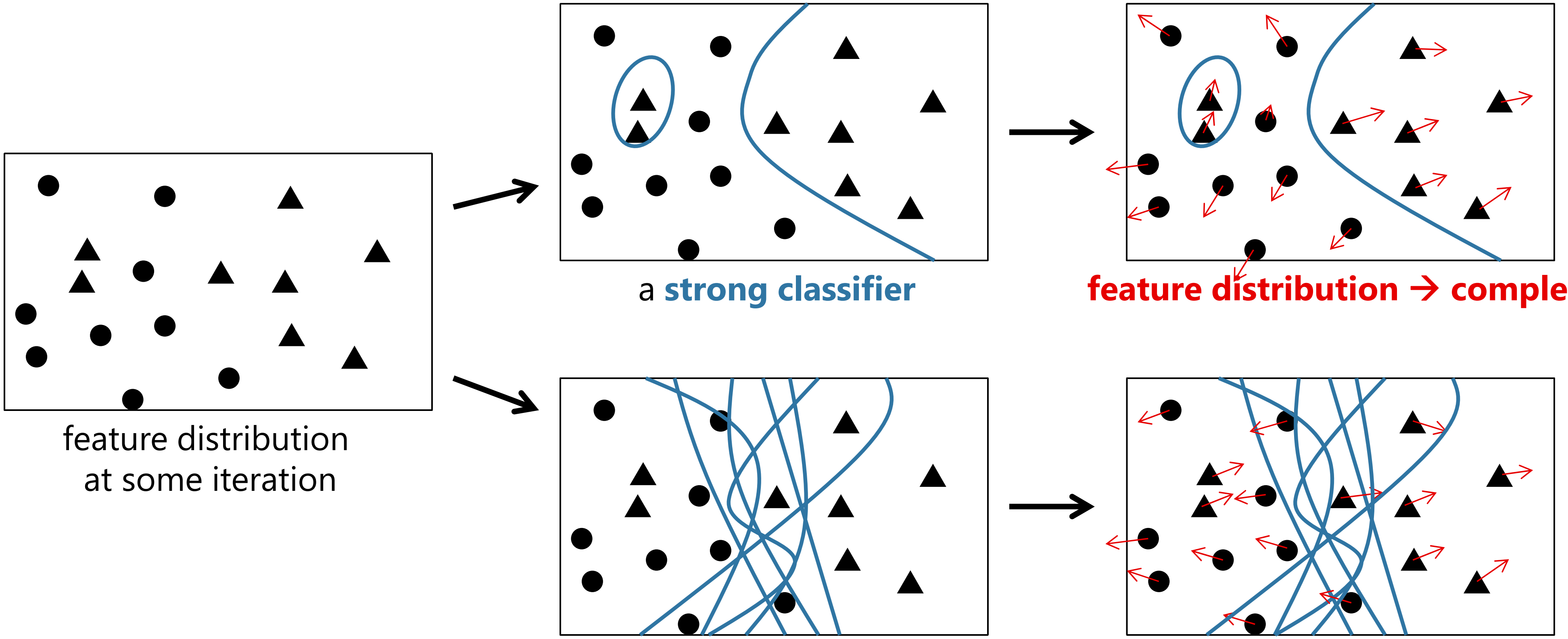
feature distribution → complex

a set of **weak classifiers**

feature distribution → simple

Why *weak* ??

many random weak classifiers → Features do not adapt to a particular one.



feature distribution at some iteration

a **strong classifier**

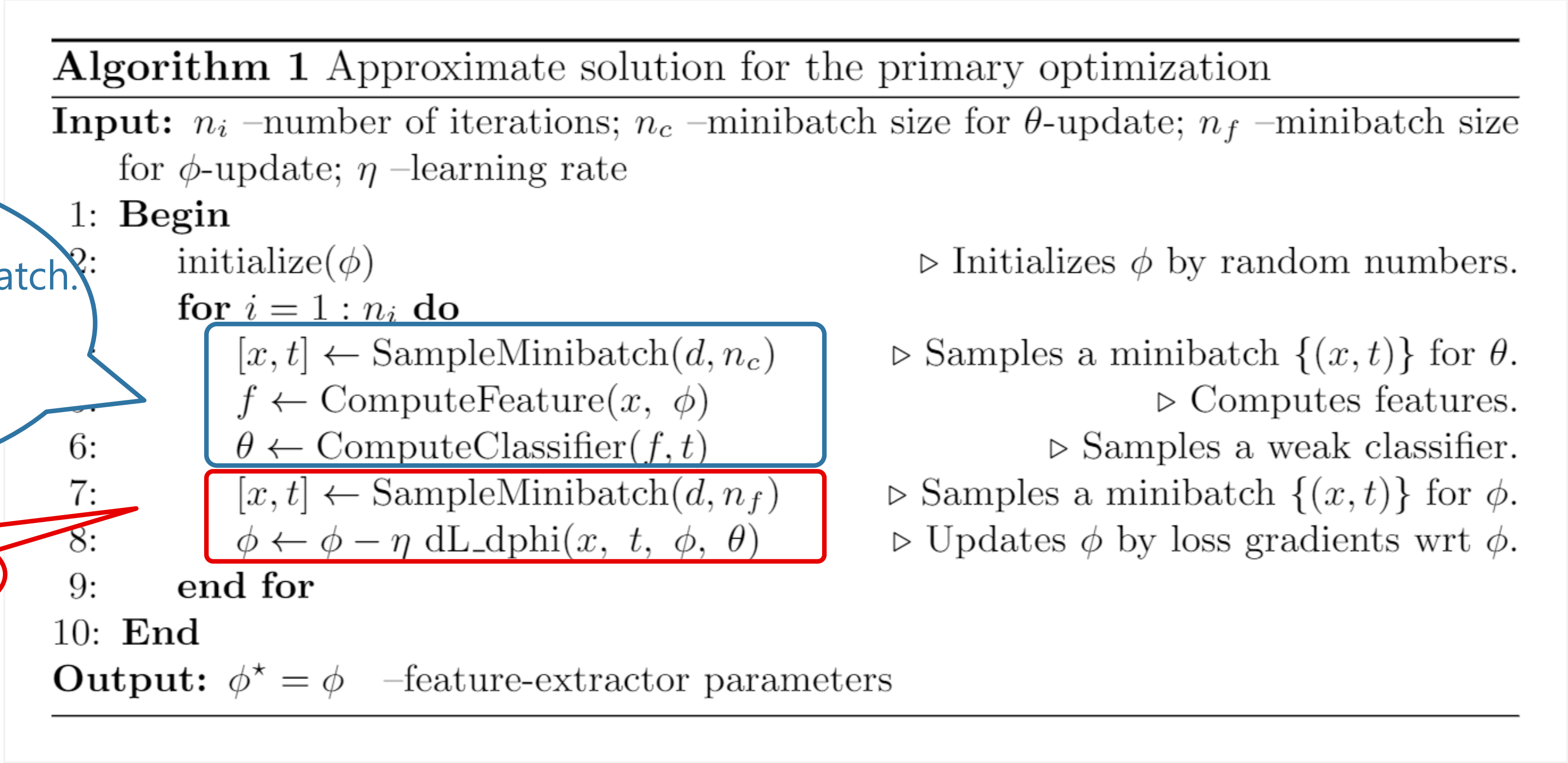
feature distribution → complex

a set of **weak classifiers**

feature distribution → simple

Pseudocode

source code: https://github.com/DensoITLab/FOCA-v1



Algorithm 1 Approximate solution for the primary optimization

Input: n_i –number of iterations; n_c –minibatch size for θ-update; n_f –minibatch size for ϕ-update; η –learning rate

```
Begin
    initialize($\phi$)                                  ▷ Initializes $\phi$ by random numbers.
    for $i = 1 : n_i$ do
        $[x, t] \leftarrow$ SampleMinibatch($d, n_c$)    ▷ Samples a minibatch $\{(x, t)\}$ for $\theta$.
        $f \leftarrow$ ComputeFeature($x, \phi$)         ▷ Computes features.
        $\theta \leftarrow$ ComputeClassifier($f, t$)    ▷ Samples a weak classifier.
        $[x, t] \leftarrow$ SampleMinibatch($d, n_f$)    ▷ Samples a minibatch $\{(x, t)\}$ for $\phi$.
        $\phi \leftarrow \phi - \eta$ dL_dphi($x, t, \phi, \theta$)   ▷ Updates $\phi$ by loss gradients wrt $\phi$.
    end for
End
```

Output: $\phi^\star = \phi$ –feature-extractor parameters

Optimizes θ with a small batch. Works weakly to the entire dataset.

Updates ϕ with θ.

* Weak classifier θ is discarded after a single use.

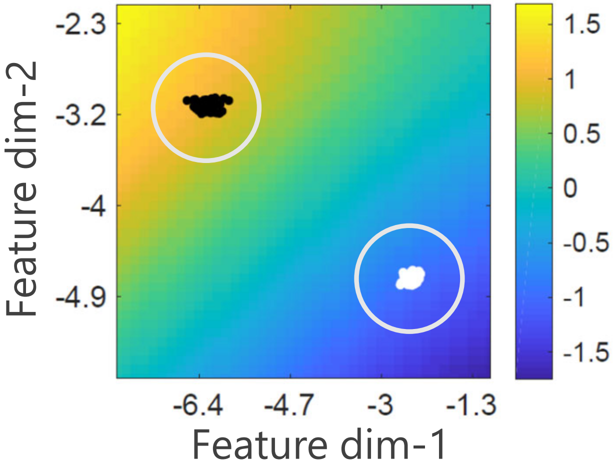
Property: simple feature distribution

In words [I. Sato, et al., ICML2019],

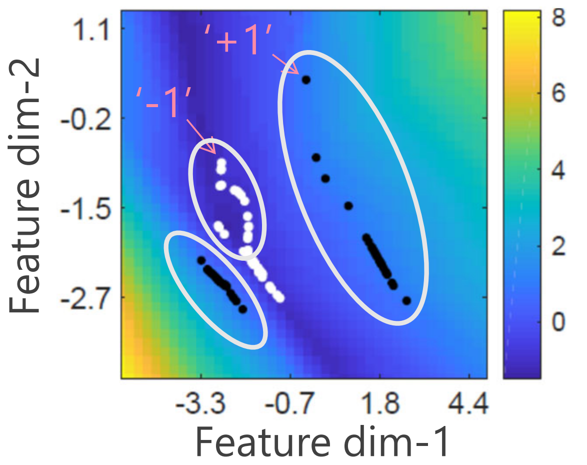
If feature extractor has an enough representation ability, all input data of the same class are projected to a single point in the feature space in a class-separable way under certain conditions.

Features form simple point-like distribution per class (under some conditions).

FOCA



E2E



Outline

- Introduction
- Overview: FOCA

FOCA: Feature-extractor Optimization through Classifier Anonymization
I. Sato, et al., ICML2019.

- **Experiment**
  - **Improvement over Sato et al.**
  - **Comparison with end-to-end training methods**
  - **Effect of network fine-tuning after FOCA**
- Summary

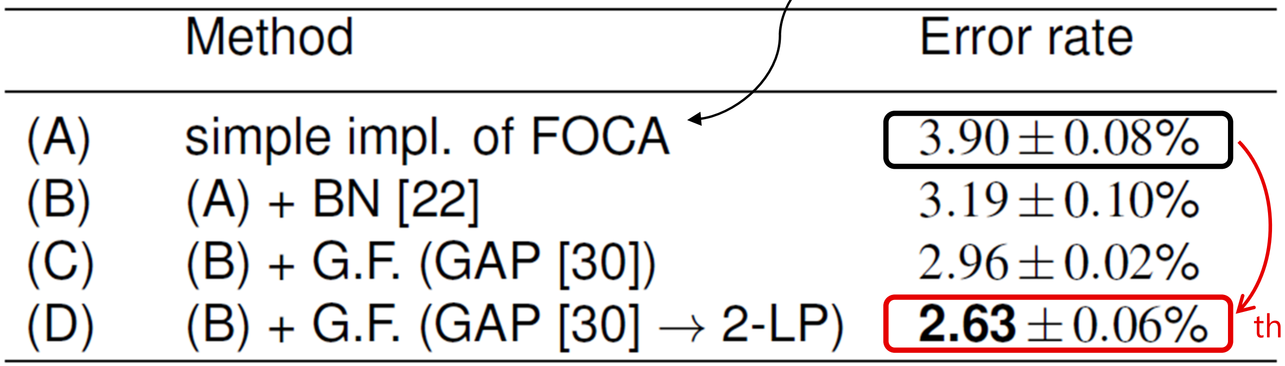
Improvement over Sato et al., ICML2019

Careful hyperparameter tuning with following techniques greatly improved FOCA’s generalization.

- *global features* (GF) with
  - Global Average Pooling (GAP) after convolution part
  - 2-layer perceptron (2-LP) after GAP
- Batch Normalization

Table 1. improvement over Sato et al., 2019. Wide ResNet (28-10) base network used in the feature extractor. CIFAR-10 dataset used.

Sato et al., ICML2019.



	Method	Error rate
(A)	simple impl. of FOCA	$\boxed{3.90 \pm 0.08\%}$
(B)	(A) + BN [22]	$3.19 \pm 0.10\%$
(C)	(B) + G.F. (GAP [30])	$2.96 \pm 0.02\%$
(D)	(B) + G.F. (GAP [30] → 2-LP)	$\boxed{\mathbf{2.63} \pm 0.06\%}$

this work

Comparison with E2E training methods

The non-E2E training method (FOCA) outperformed strong baselines that use E2E training under fair settings.

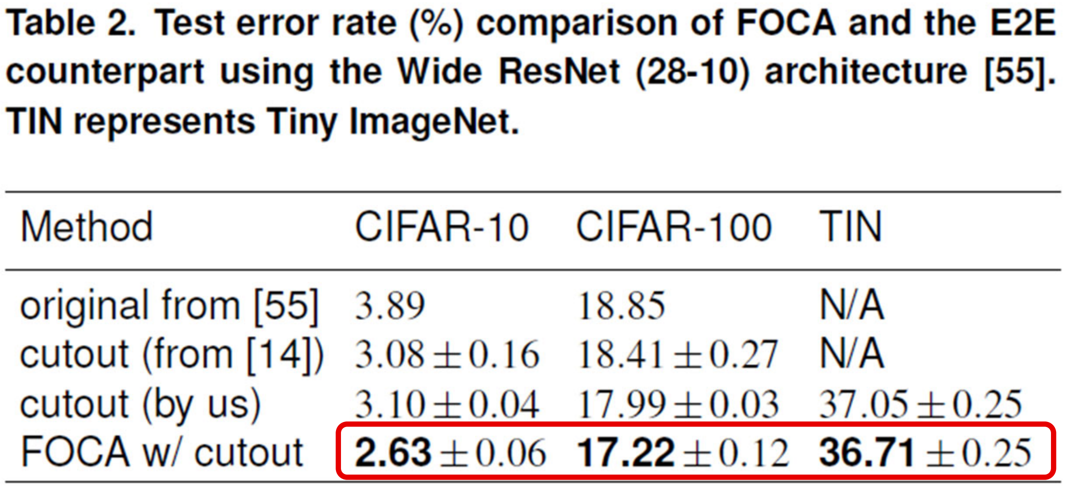
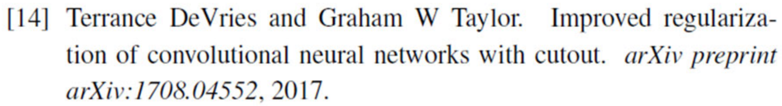


Table 2. Test error rate (%) comparison of FOCA and the E2E counterpart using the Wide ResNet (28-10) architecture [55]. TIN represents Tiny ImageNet.

Method	CIFAR-10	CIFAR-100	TIN
original from [55]	3.89	18.85	N/A
cutout (from [14])	3.08 ± 0.16	18.41 ± 0.27	N/A
cutout (by us)	3.10 ± 0.04	17.99 ± 0.03	37.05 ± 0.25
FOCA w/ cutout	**2.63** ± 0.06	**17.22** ± 0.12	**36.71** ± 0.25

this work



[14] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

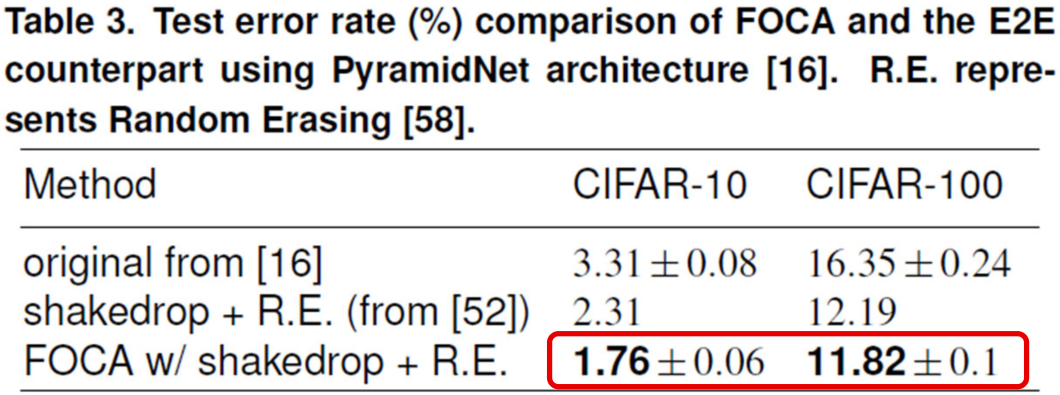
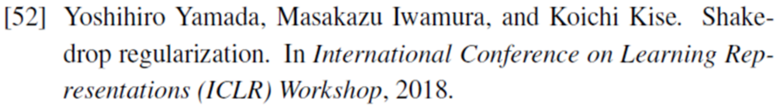


Table 3. Test error rate (%) comparison of FOCA and the E2E counterpart using PyramidNet architecture [16]. R.E. represents Random Erasing [58].

Method	CIFAR-10	CIFAR-100
original from [16]	3.31 ± 0.08	16.35 ± 0.24
shakedrop + R.E. (from [52])	2.31	12.19
FOCA w/ shakedrop + R.E.	**1.76** ± 0.06	**11.82** ± 0.1

this work



[52] Yoshihiro Yamada, Masakazu Iwamura, and Koichi Kise. Shakedrop regularization. In *International Conference on Learning Representations (ICLR) Workshop*, 2018.

Comparison with E2E training methods

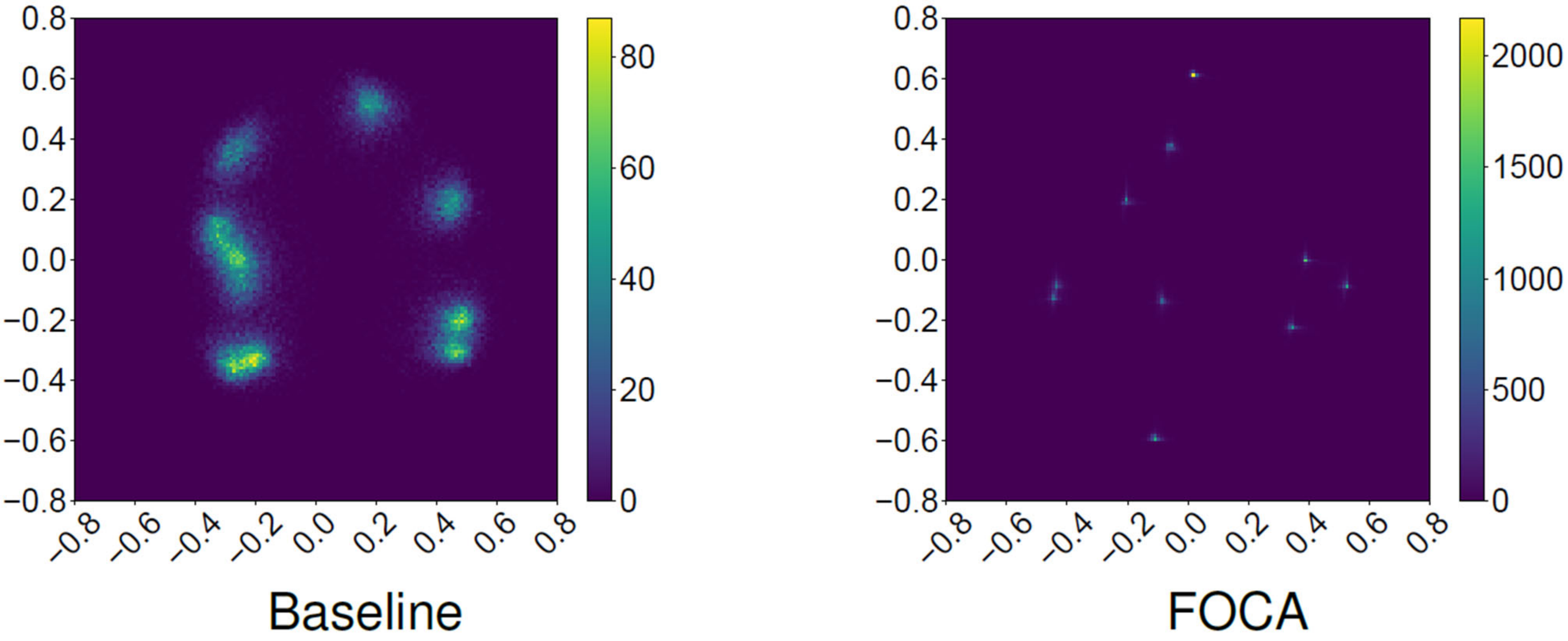
The nc ttings.

2D histograms of normalized CIFAR-10 features projected by PCA.
FOCA exhibits well-separated, point-like distribution.

Table 2. Tes d the E2E counterpart R.E. repre- TIN represe

Method	R-100
original fro	5 ±0.24
cutout (fro	9
cutout (by	2 ±0.1
FOCA w/ c	

[14] e- n-



Effect of network fine-tuning after FOCA

Aim To see if E2E network fine-tuning improve performance after FOCA.

so far

how about?

opt. step	process
1	feature extractor optimization
2	classifier optimization with frozen features
3	E2E network fine-tuning

Result E2E network fine-tuning yields no improvement or slightly worse performance.

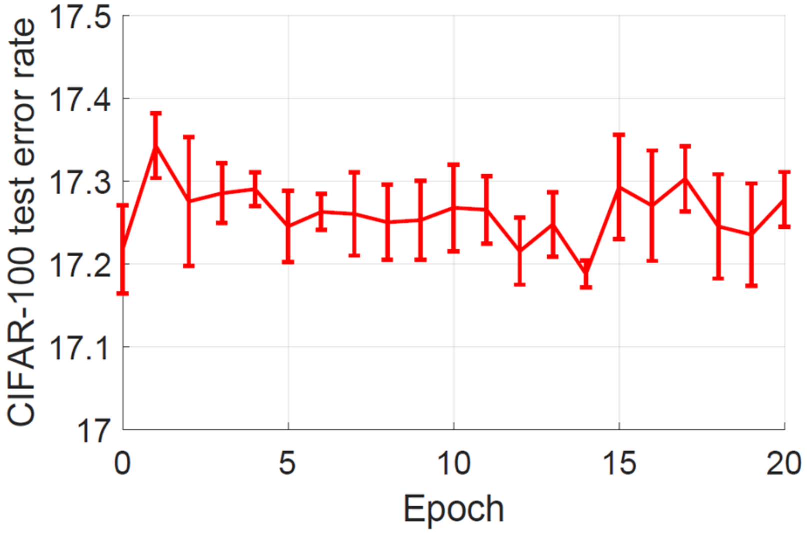


Fig. 1 CIFAR-100 test error rate curve.
Epoch 0 means the start of fine-tuning.
Similar results obtained for CIFAR-10 and Tiny ImageNet.

Outline

- Introduction
- Overview: FOCA

FOCA: Feature-extractor Optimization through Classifier Anonymization
I. Sato, et al., ICML2019.

- Experiment
  - Improvement over Sato et al.
  - Comparison with end-to-end training methods
  - Effect of network fine-tuning after FOCA
- **Summary**

Summary

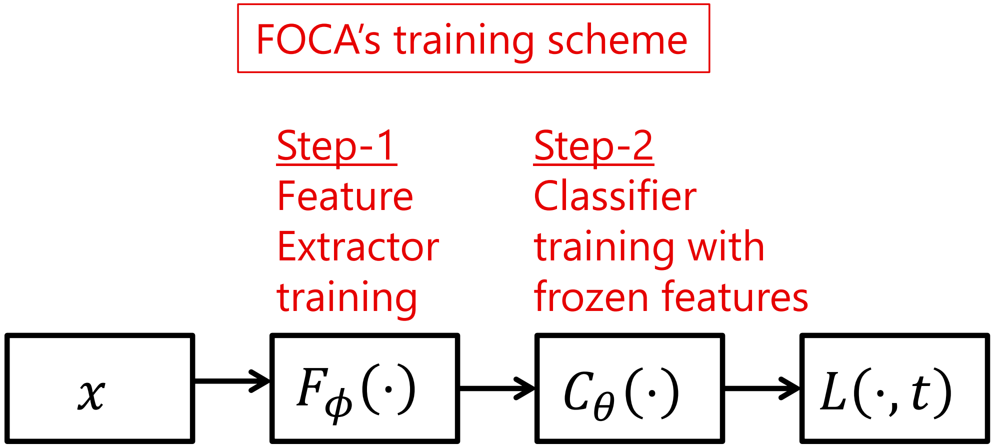
Question we try to answer

Does end-to-end trained deep model always perform better than non-end-to-end counterpart?

Our answer

Not always, with supportive evidences:

- We found evidences in which a non-E2E training method, FOCA, outperforms strong E2E training counterparts on CIFAR-10, 100, and Tiny ImageNet.
- E2E network fine-tuning after FOCA yields no improvement or slightly worse performance.



FOCA's training scheme

Step-1
Feature Extractor training

Step-2
Classifier training with frozen features

$x \rightarrow F_\phi(\cdot) \rightarrow C_\theta(\cdot) \rightarrow L(\cdot, t)$

DENSO
IT LAB