ICML 2022

PoF: Post-Training of Feature Extractor for Improving Generalization

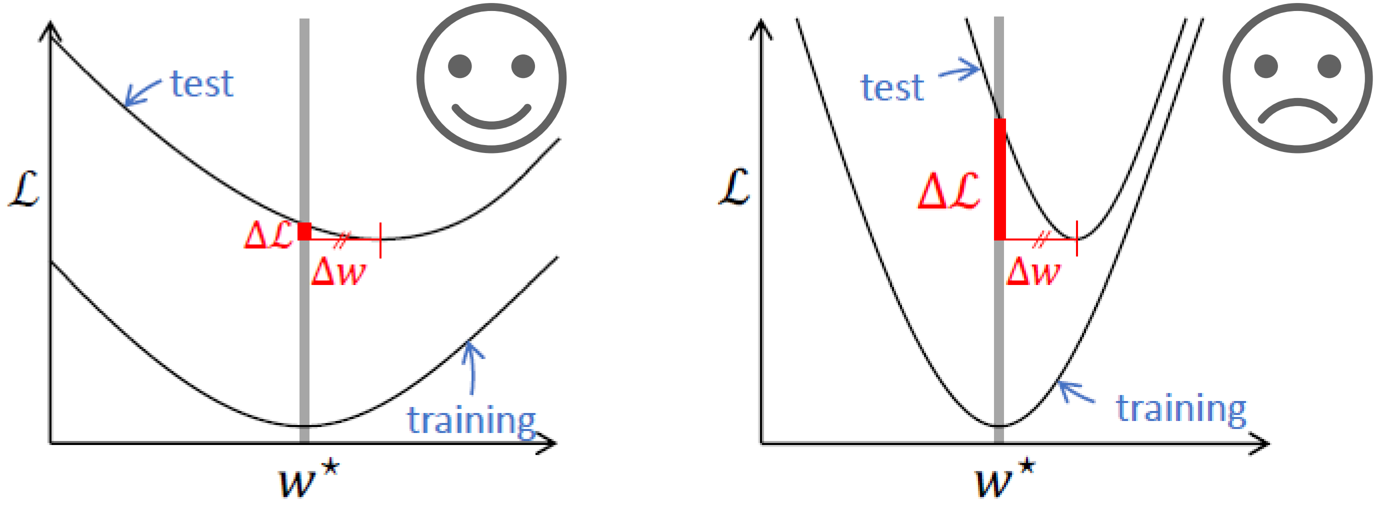
Ikuro Sato[*12], Ryota Yamada[*1],
Masayuki Tanaka[1], Nakamasa Inoue[1], Rei Kawakami[12]

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

Background

Loss landscape and generalization ability

Previous studies showed that flatter minima tend to generalize better.



(a) Flatter minima (b) Sharper minima

$\Delta\mathcal{L}$ Curvature-based loss increment

Hochreiter & Schmidhuber, 1997; Keskar et al., 2017; Dziugaite & Roy, 2017; Jiang et al., 2020; Dinh et al., 2017.

Related work

SAM: Sharpness Aware Minimization (P. Foret et al., 2021)

- Can find a flatter minimum within a “ball” of fixed radius.
- High performance gains

“ball” of fixed radius

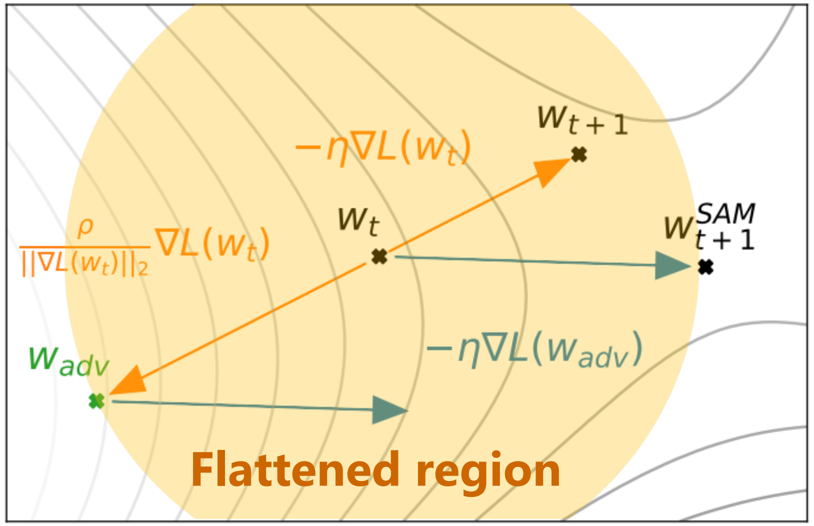


Table: Top-1 test error rates.

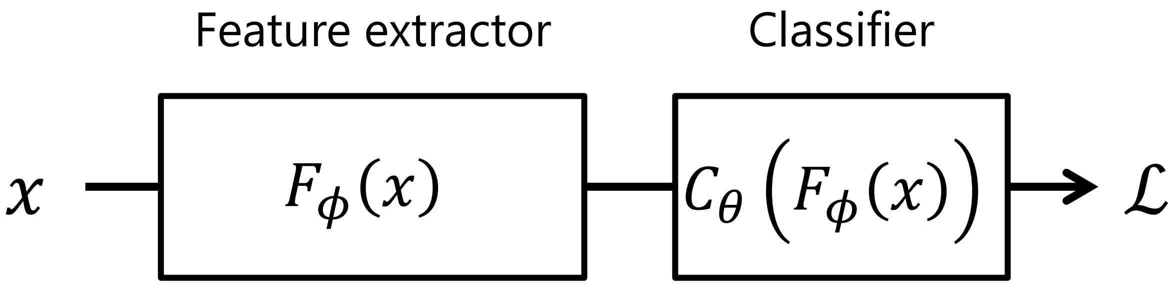
Dataset	EffNet-b7 + SAM	EffNet-b7	Prev. SOTA (ImageNet only)
FGVC_Aircraft	$6.80_{\pm 0.06}$	$8.15_{\pm 0.08}$	**5.3** (TBMSL-Net)
Flowers	$\mathbf{0.63}_{\pm 0.02}$	$1.16_{\pm 0.05}$	0.7 (BiT-M)
Oxford_IIIT_Pets	$\mathbf{3.97}_{\pm 0.04}$	$4.24_{\pm 0.09}$	4.1 (Gpipe)
Stanford_Cars	$5.18_{\pm 0.02}$	$5.94_{\pm 0.06}$	**5.0** (TBMSL-Net)
CIFAR-10	$\mathbf{0.88}_{\pm 0.02}$	$0.95_{\pm 0.03}$	1 (Gpipe)
CIFAR-100	$\mathbf{7.44}_{\pm 0.06}$	$7.68_{\pm 0.06}$	7.83 (BiT-M)
Birdsnap	$\mathbf{13.64}_{\pm 0.15}$	$14.30_{\pm 0.18}$	15.7 (EffNet)
Food101	$7.02_{\pm 0.02}$	$7.17_{\pm 0.03}$	7.0 (Gpipe)
ImageNet	$15.14_{\pm 0.03}$	15.3	**14.2** (KDforAA)

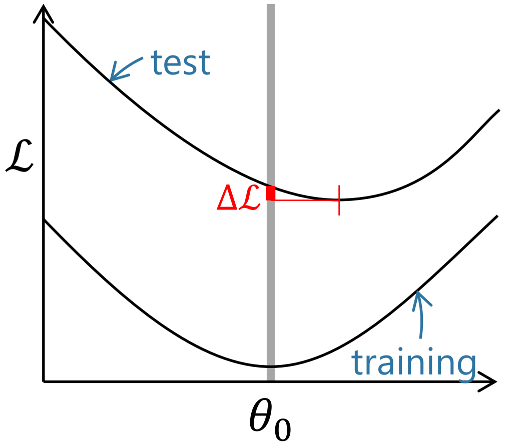
Method

Aim of this work

To develop an optimization method that optimizes feature extractor s.t. loss landscape in the classifier parameter space becomes flatter.

Assumption: Sharp landscape tends to be more harmful in higher-layer parameter space.

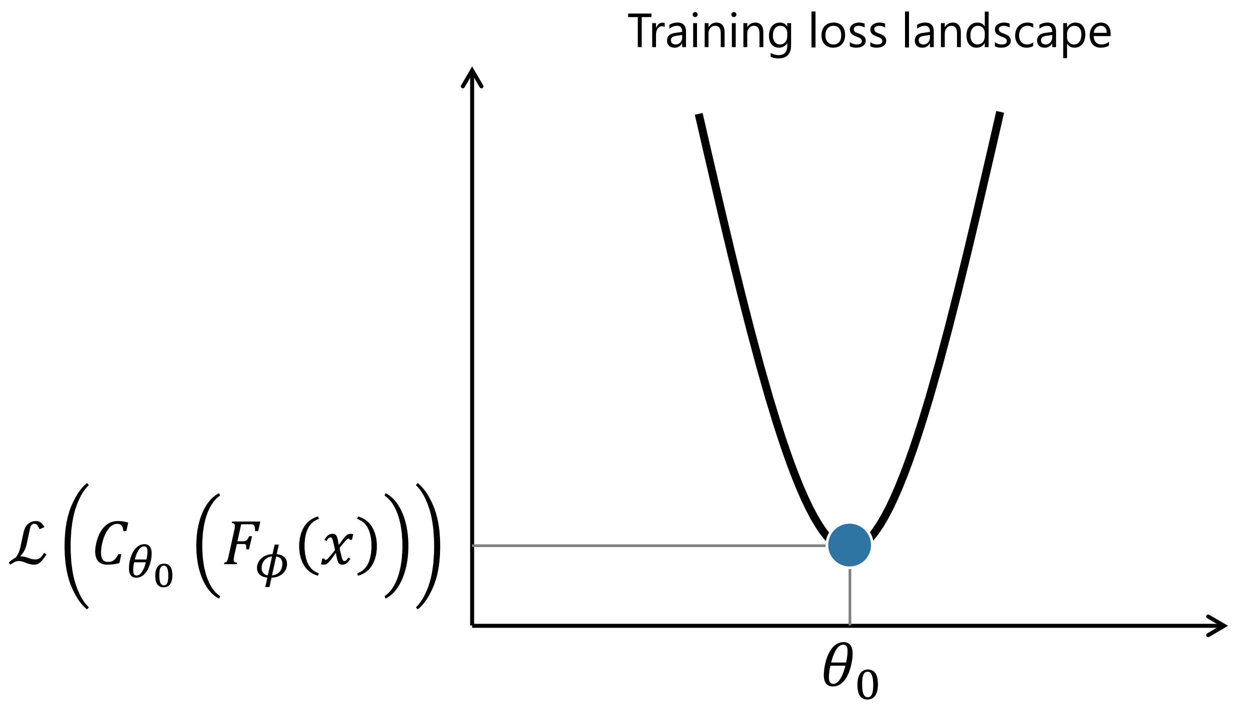




Classifier parameter space

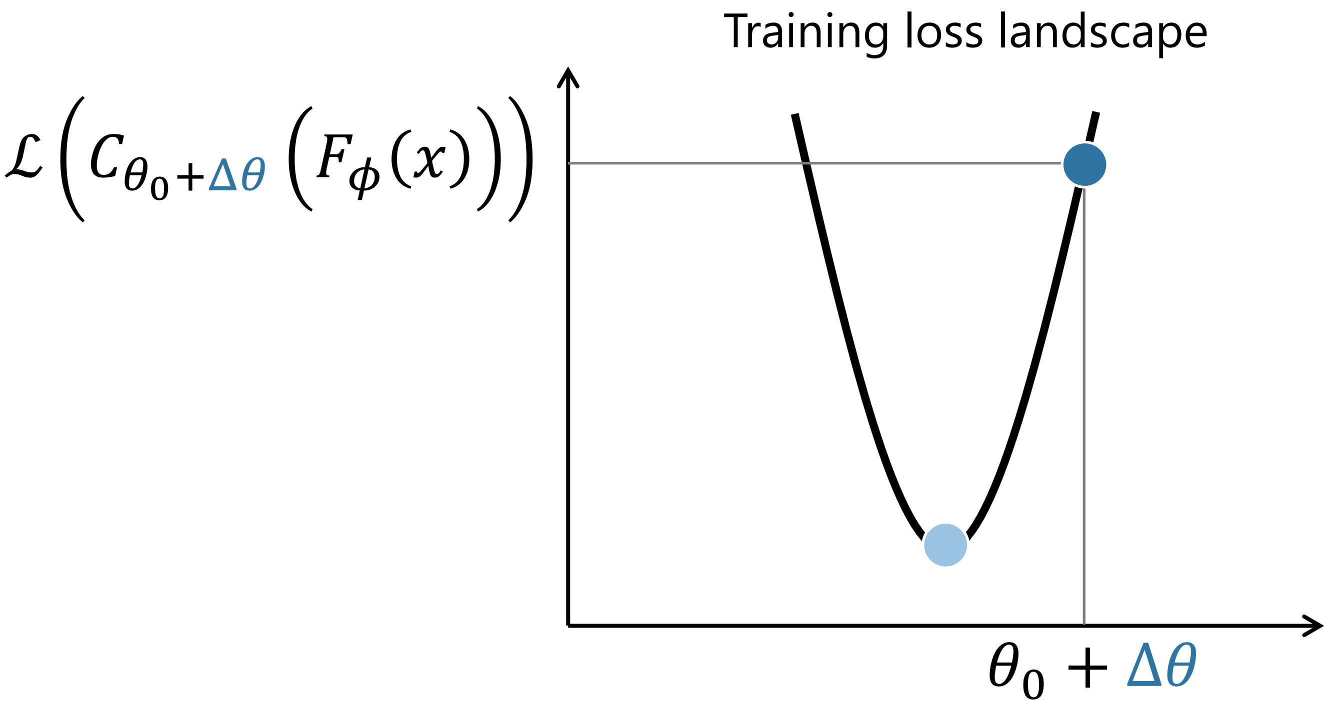
PoF: Post-training of Feature-extractor

Post-trains the feature-extractor part of DNN with parameter-perturbed classifiers.



PoF: Post-training of Feature-extractor

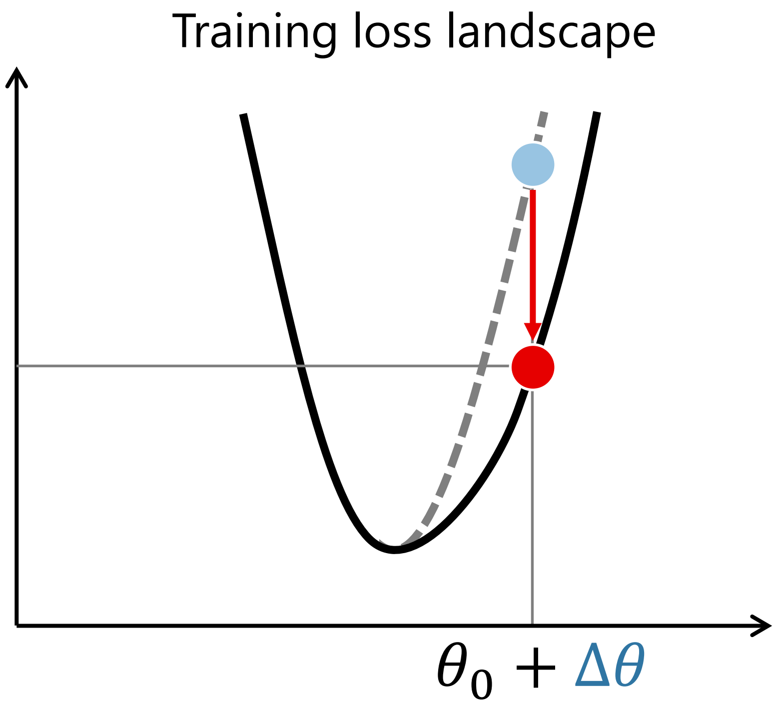
Post-trains the feature-extractor part of DNN with parameter-perturbed classifiers.



PoF: Post-training of Feature-extractor

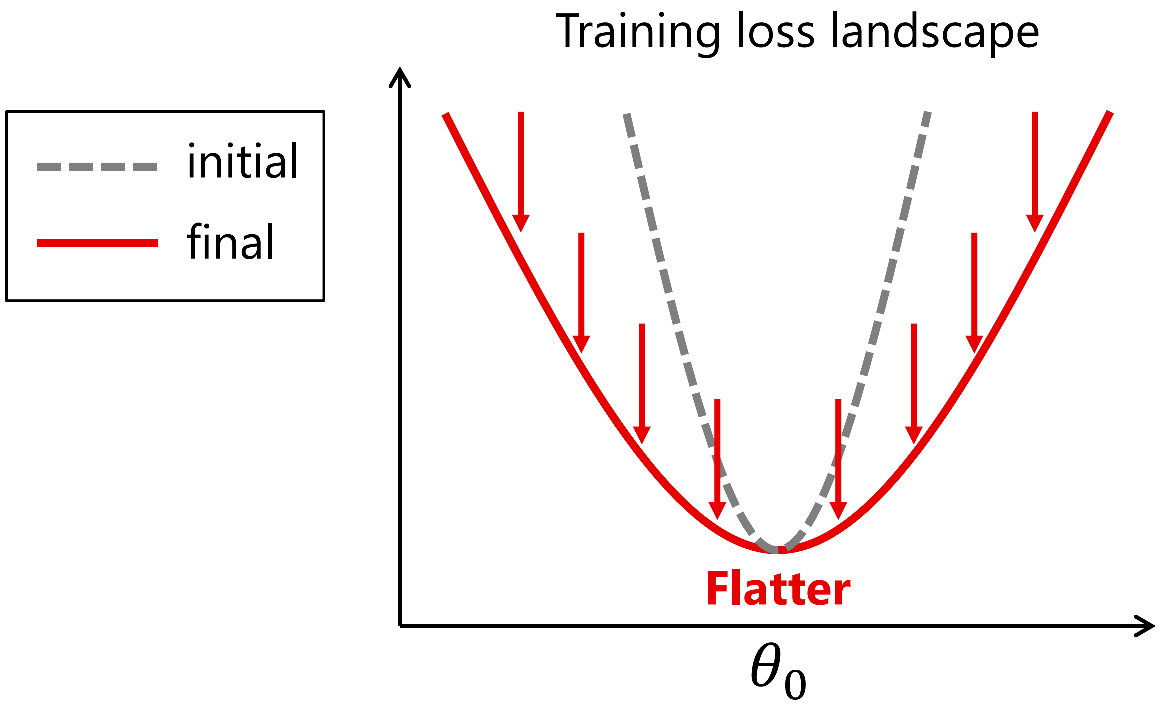
Post-trains the feature-extractor part of DNN with parameter-perturbed classifiers.

$$\mathcal{L}\left(C_{\theta_0+\Delta\theta}\left(F_{\phi+\Delta\phi}(x)\right)\right)$$



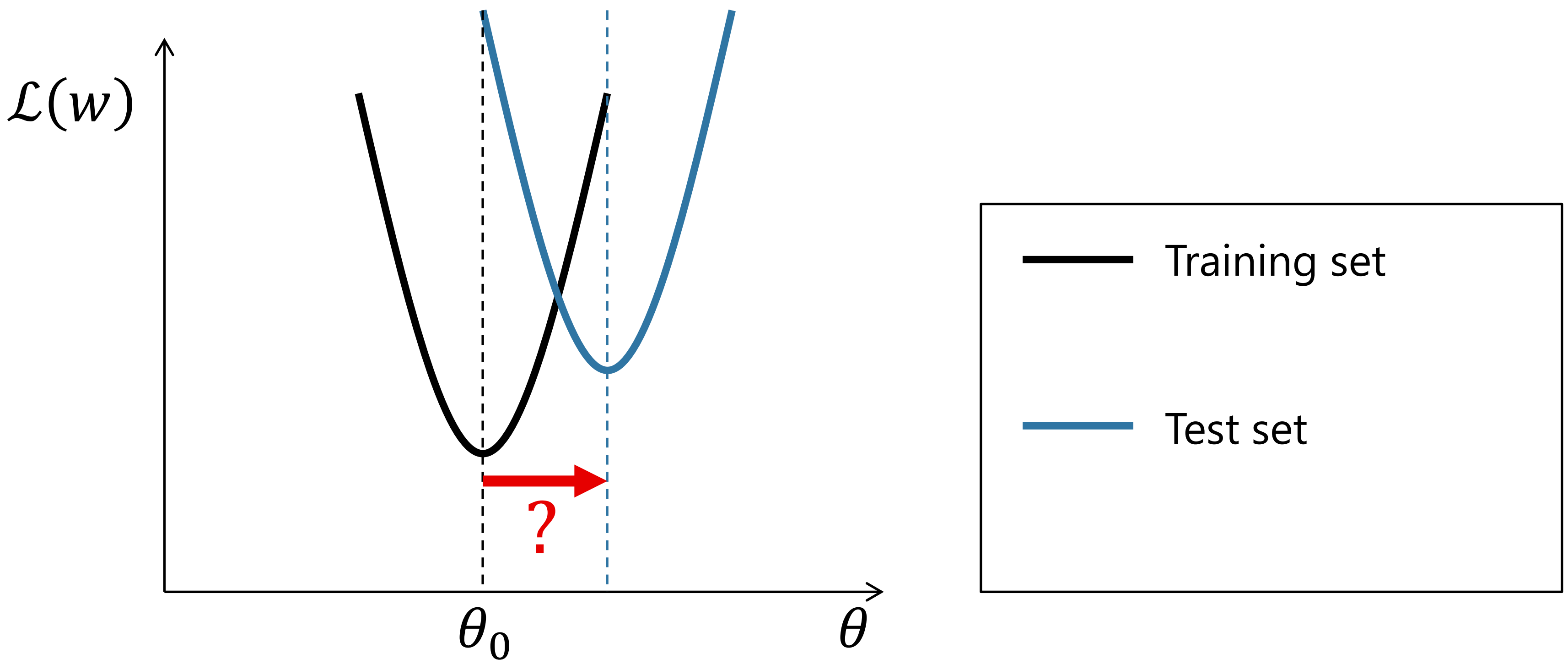
PoF: Post-training of Feature-extractor

Post-trains the feature-extractor part of DNN with parameter-perturbed classifiers.



Range of parameter perturbation

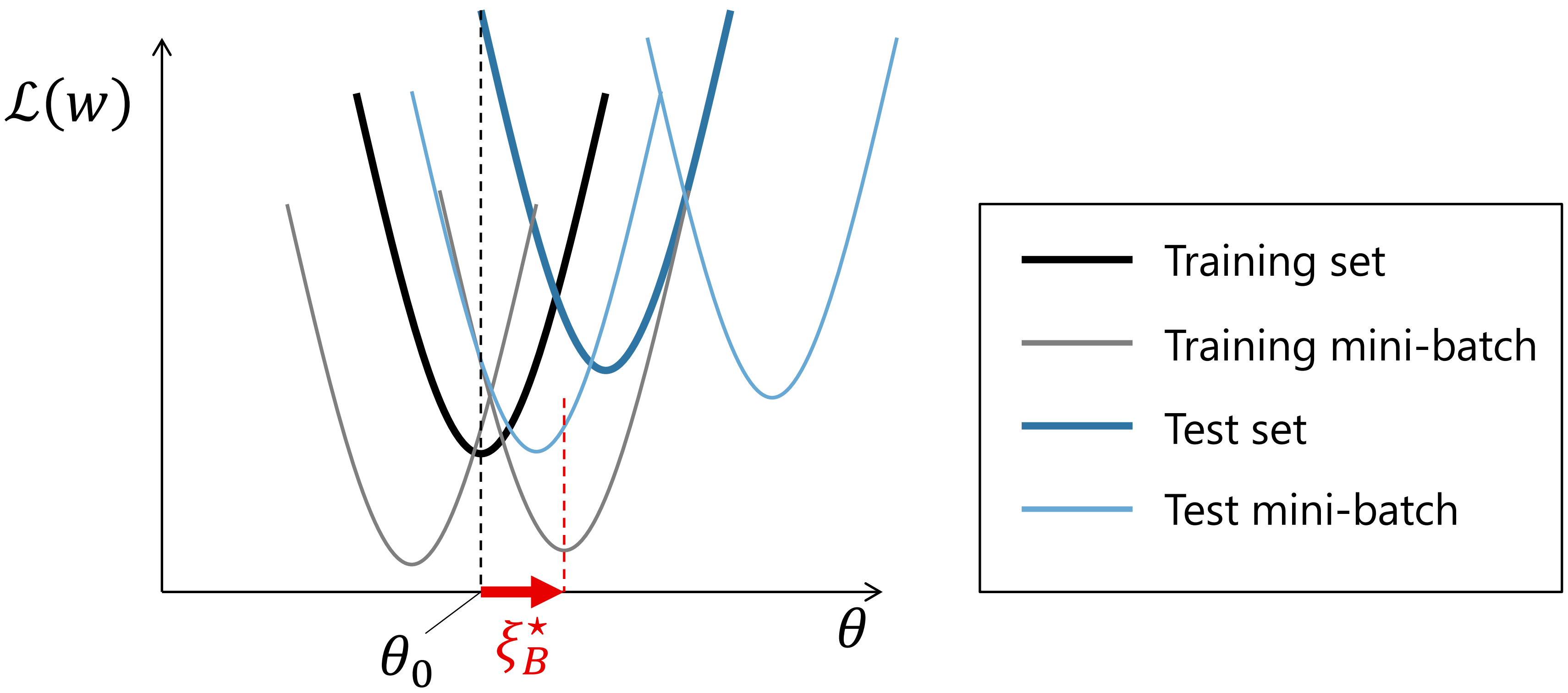
How can one estimate good perturbation range?



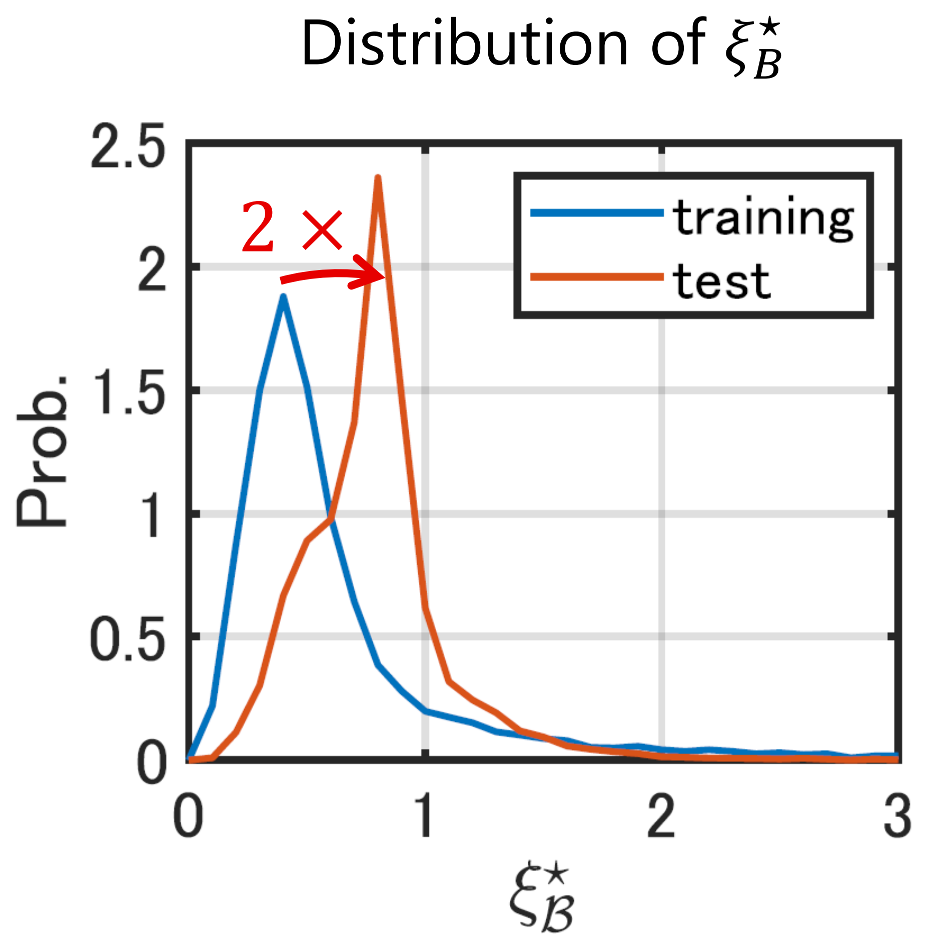
Range of parameter perturbation

How can one estimate good perturbation range?

→ Use *mini-batch statistics!*



Range of parameter perturbation



from toy experiment

→ The peak for test set is roughly **2x** as that for training set.

Algorithm

Normalized mini-batch gradient

Classifier-parameter perturbation

$$\xi_{\mathcal{B}}^{\star} = \arg\min_{\xi \geq 0} \mathcal{L}_{\mathcal{B}}\left(\phi^{(t)}, \theta_0 - \xi \hat{\mathcal{L}}_{B}'\right)$$

$\times\, n$

Feature-extractor update

$$\phi^{(t+1)} = \phi^{(t)} - \eta \frac{\partial \mathcal{L}_{\tilde{B}}\left(\phi, \theta_0 - 2\xi_{\mathcal{B}}^{\star} \hat{\mathcal{L}}_{B}'\right)}{\partial \phi}, \eta > 0$$

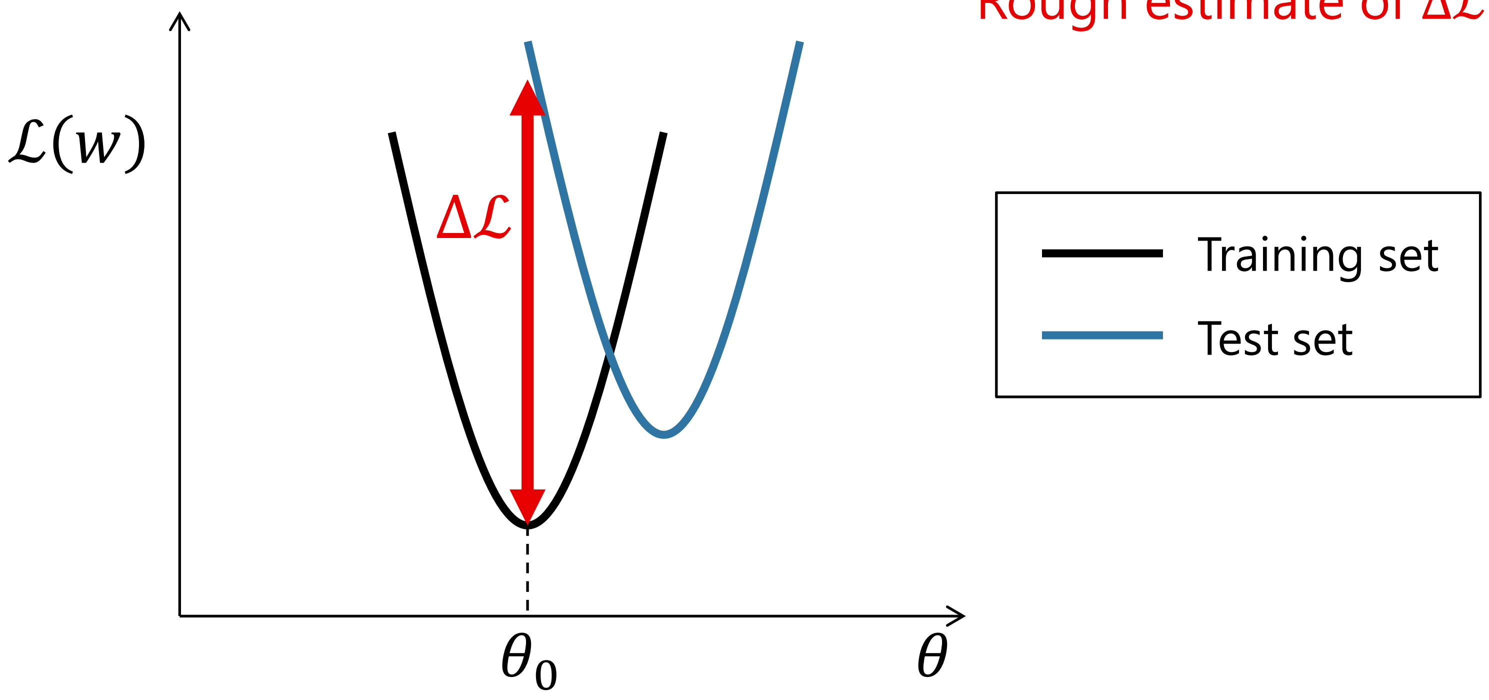
Theory

With certain assumption, we can derive **effective loss**,

Hessian

$$\mathcal{L}_{\mathcal{D}}\left(\phi, \theta_0 - 2\xi_B^{\star}\hat{\mathcal{L}}'_B\right) \simeq \mathcal{L}_{\mathcal{D}}(\phi, \theta_0) + \boxed{2\xi_B^{\star}\hat{\mathcal{L}}_B'^{\top}\mathcal{H}_{\mathcal{D}}(\phi, \theta_0)\hat{\mathcal{L}}'_B}$$

Rough estimate of $\Delta\mathcal{L}$



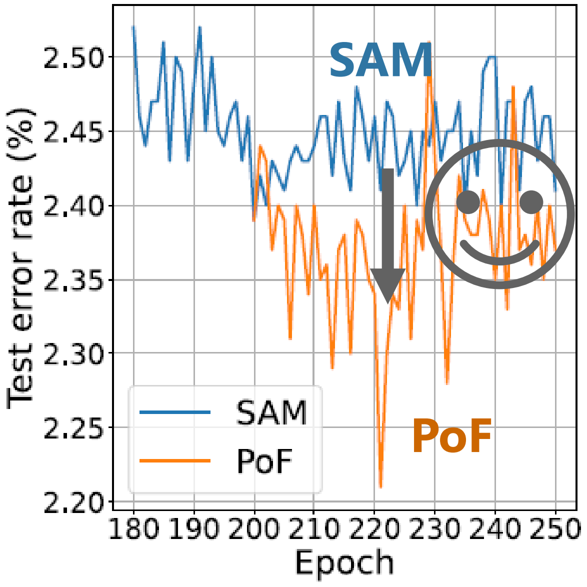
Evaluation

Generalization

PoF can improve generalization of network trained by SAM on 3/4 datasets.

Table: Test error rates

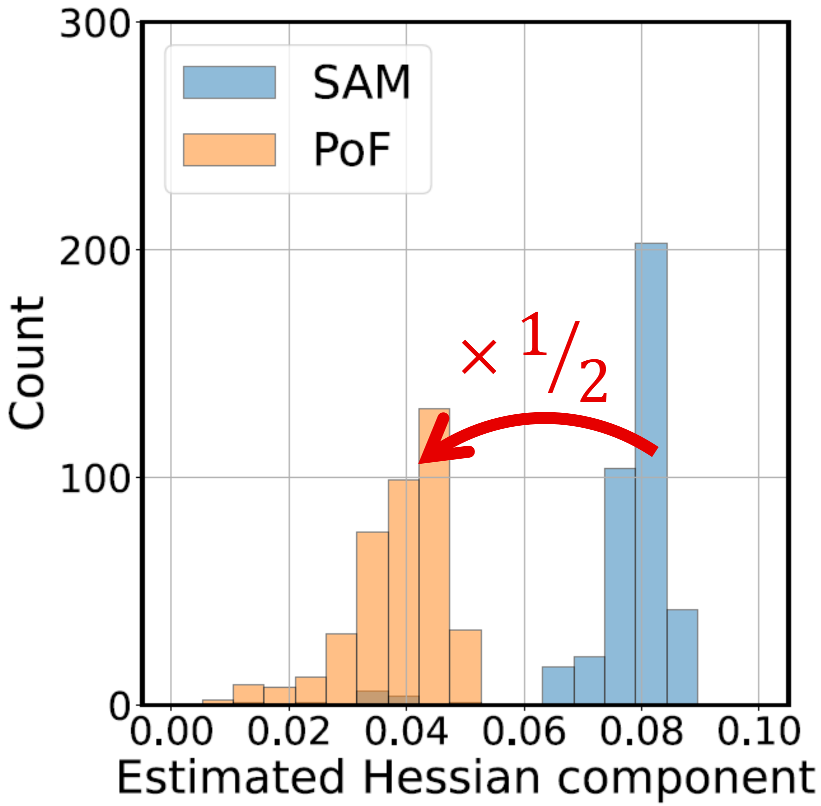
Method	Dataset			
	CIFAR-10	CIFAR-100	SVHN	Fashion
SGD (200 epochs)	3.22±0.14	18.23±0.35	1.67±0.03	4.60±0.11
SGD (250 epochs)	3.14±0.13	18.40±0.35	1.67±0.03	4.63±0.14
SAM (200 epochs)	2.50±0.07	16.27±0.09	1.64±0.04	4.14±0.09
SAM (250 epochs)	2.53±0.08	16.32±0.20	1.63±0.03	**4.12**±0.05
PoF (210 epochs)	**2.41**±0.02	**16.07**±0.15	1.60±0.04	4.25±0.05
PoF (250 epochs)	**2.41**±0.06	16.60±0.05	**1.55**±0.02	4.35±0.07



(b) Zoomed view

Loss curvatures

The largest Hessian component gets halved by PoF.



(b) Test dataset

Summary

- Proposed PoF: Post-training of Feature extractor
- The flattening range is controlled in a data-driven manner so that the algorithm rough reduces $\Delta\mathcal{L}$.
- Confirmed generalization improvement over SAM on 3/4 datasets.