Samsung Research

Samsung AI Center Cambridge

Fisher SAM

Information Geometry & Sharpness Aware Optimisation

Minyoung Kim

Coauthors: **Da Li, Shell Xu Hu, Timothy Hospedales**

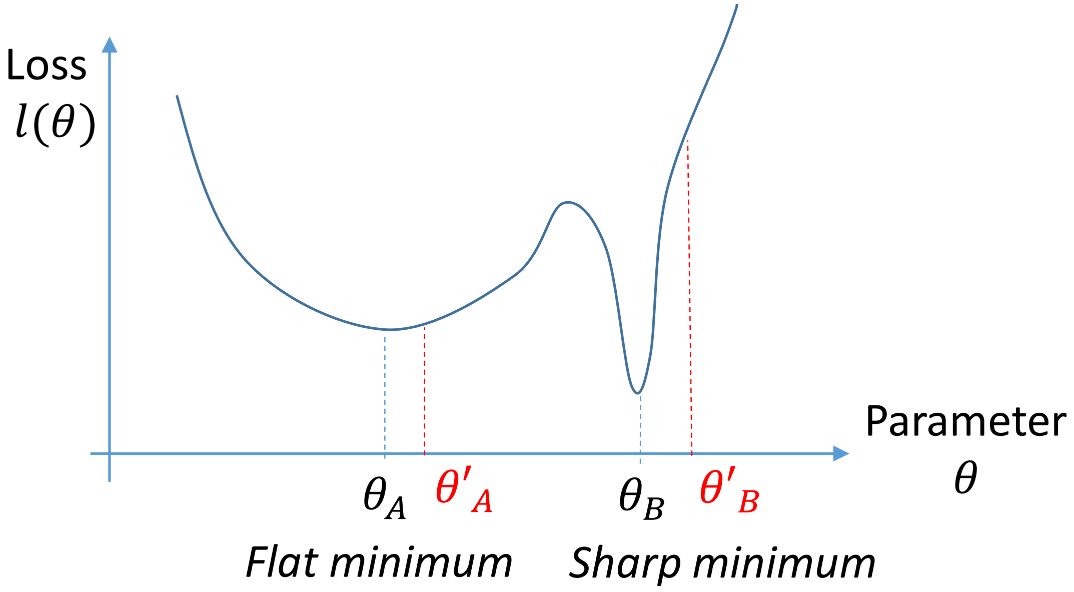
ICML 2022 Short Talk

Flat Minima in Deep Learning

In many cases,

DL → Minimising a loss function $l(\theta)$

Highly non-convex (many local minima)



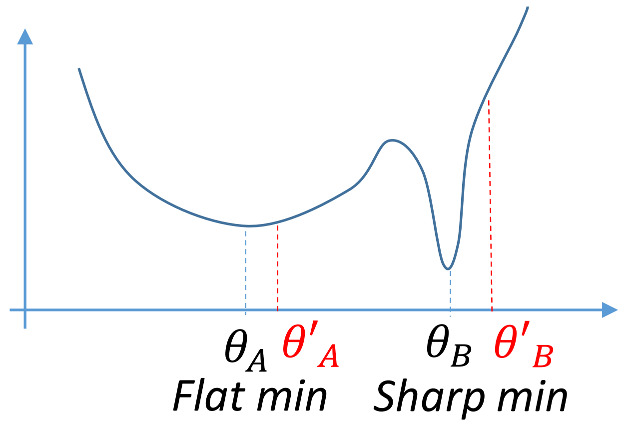
Q) Which is better, θ_A or θ_B?

A) We prefer θ_A to θ_B even though $l(\theta_A) > l(\theta_B)$

Why? Because θ_A is more robust.

Imagine some perturbation: $\theta_A \rightarrow \theta'_A,\ \theta_B \rightarrow \theta'_B \Rightarrow l(\theta'_A) \ll l(\theta'_B)$

Let's seek for a Flat Minimum



Flat minima = Robust models
= Resilient to data noise or model corruption
(often encountered in AI applications)

But, how?

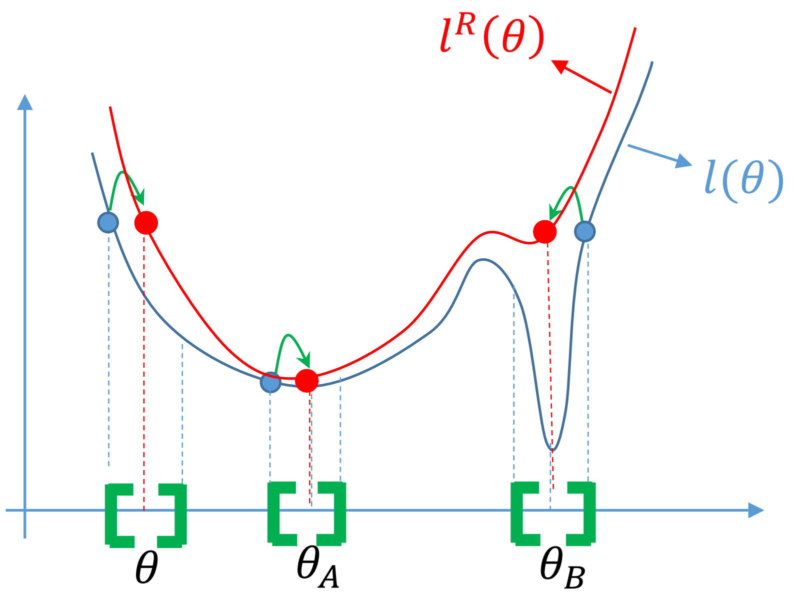
Sharpness-Aware Minimization (SAM) (Foret et al, 2021)

Idea of SAM:

Define a robust loss $l^R(\theta)$ as worst-case loss within a neighborhood of θ

$$l^R(\theta) = \max_{\epsilon \in N_\theta} l(\theta + \epsilon)$$

Neighborhood around θ



Sharpness-aware minimization for efficiently improving generalization, Foret et al., ICLR 2021

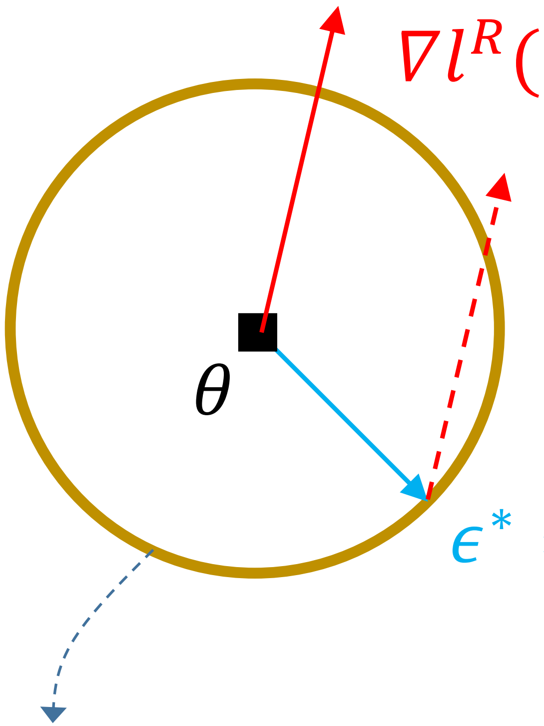
SAM(Foret et al, 2021) is Efficient

Computing $\nabla l^R(\theta)$ only amounts to evaluating two gradients!

$$l^R(\theta) = \max_{\epsilon \in N_\theta} l(\theta + \epsilon)$$

Neighborhood around θ

$$\nabla l^R(\theta) \approx \nabla l(\theta + \epsilon^*)$$



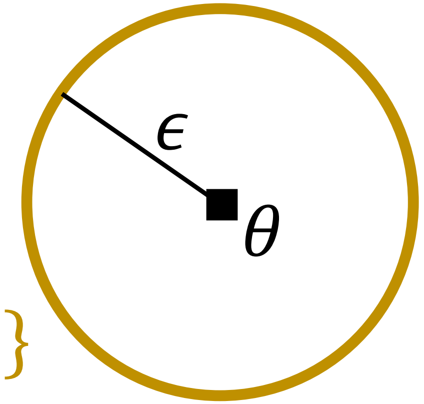
$$\epsilon^* \approx \gamma \frac{\nabla l(\theta)}{||\nabla l(\theta)||}$$

$N_\theta = \{\epsilon\colon ||\epsilon|| \leq \gamma\}$ (Euclidean ball)

But, SAM has an Issue

It's about the **Euclidean neighborhood** in SAM:

$$l^R(\theta) = \max_{\epsilon \in N_\theta} l(\theta + \epsilon) \qquad N_\theta = \{\epsilon : ||\epsilon|| \leq \gamma\}$$



But the parameter space is usually **not Euclidean**!

- $l(\theta)$ depends on θ through $p(y|x,\theta)$, eg, $l(\theta) = \mathbf{E}_{x,y}[-\log p(y|x,\theta)]$
- The distance measure $d(\theta, \theta')$ is **Fisher information metric**:

$d(\theta, \theta') \propto$ ~~$||\theta - \theta'||$~~

(for $\theta \approx \theta'$) $(\theta - \theta')^\top F(\theta)(\theta - \theta')$ ✓

$\boldsymbol{F(\theta)} = \mathbf{E}_{x,\theta}[\nabla \log p(y|x,\theta)\, \nabla \log p(y|x,\theta)^\top]$

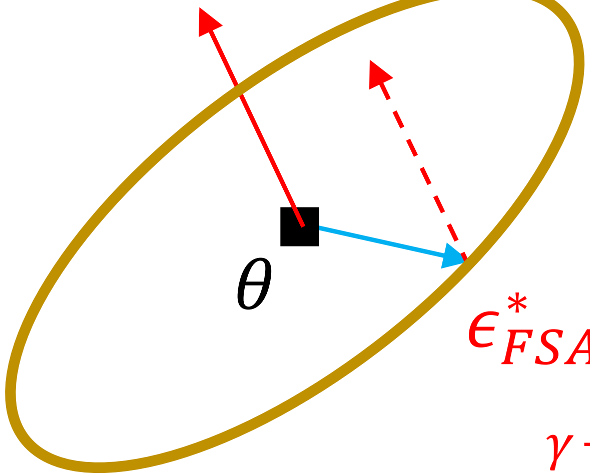
(Approximated by **Diagonal Empirical Gradient-Magnitude**)

(Our Approach) Fisher SAM

Idea: **Use Fisher-driven neighborhood** instead of Euclidean

$$l_{FSAM}(\theta) = \max_{\epsilon^\top F(\theta)\epsilon \le \gamma^2} l(\theta + \epsilon)$$

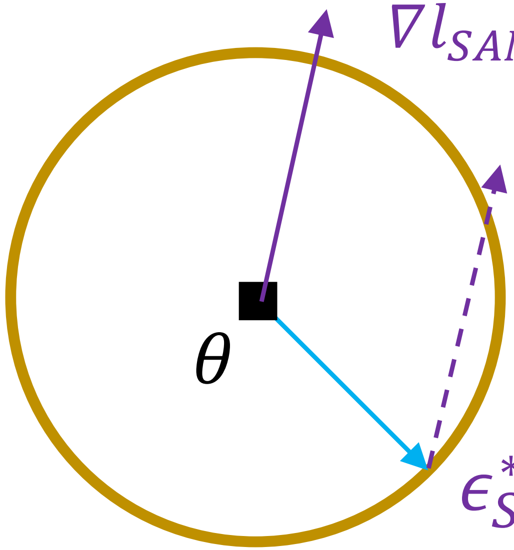
$$\nabla l_{FSAM}(\theta) \approx \nabla l(\theta + \epsilon^*)$$



$$\epsilon^*_{FSAM} \approx \gamma \frac{F(\theta)^{-1}\nabla l(\theta)}{\sqrt{\nabla l(\theta) F(\theta)^{-1} \nabla l(\theta)}}$$

$$l_{SAM}(\theta) = \max_{||\epsilon||^2 \le \gamma^2} l(\theta + \epsilon)$$

$$\nabla l_{SAM}(\theta) \approx \nabla l(\theta + \epsilon^*)$$

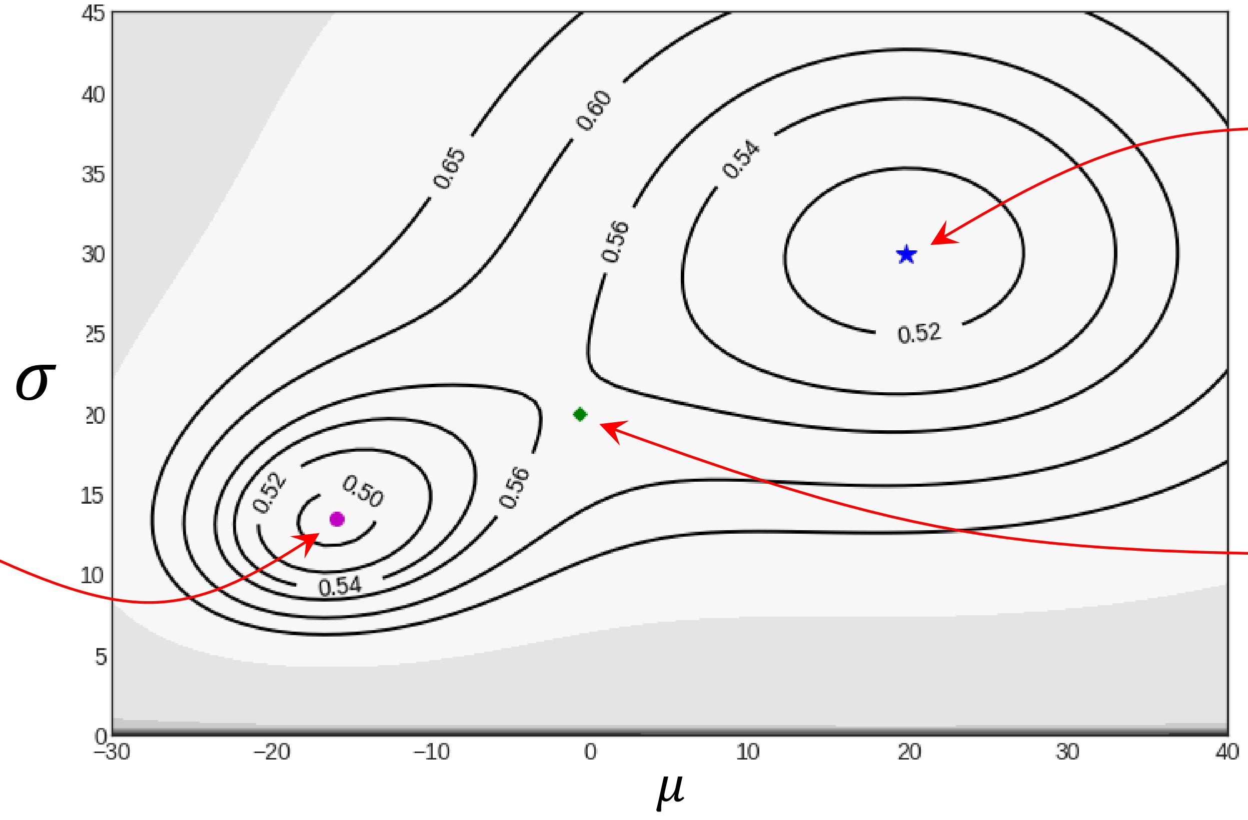


$$\epsilon^*_{SAM} \approx \gamma \frac{\nabla l(\theta)}{||\nabla l(\theta)||}$$

Illustration: 2D Toy Example

$$l(\theta) = -\log\left(\alpha_1 e^{-E_1(\theta)/\beta_1^2} + \alpha_2 e^{-E_2(\theta)/\beta_2^2}\right),$$ where

$$E_i(\theta) = \mathrm{KL}\left(p(x;\theta) || N(x; m_i, s_i^2)\right), \; i = 1, 2. \quad p(x;\theta) = N(x;\mu,\sigma^2)$$



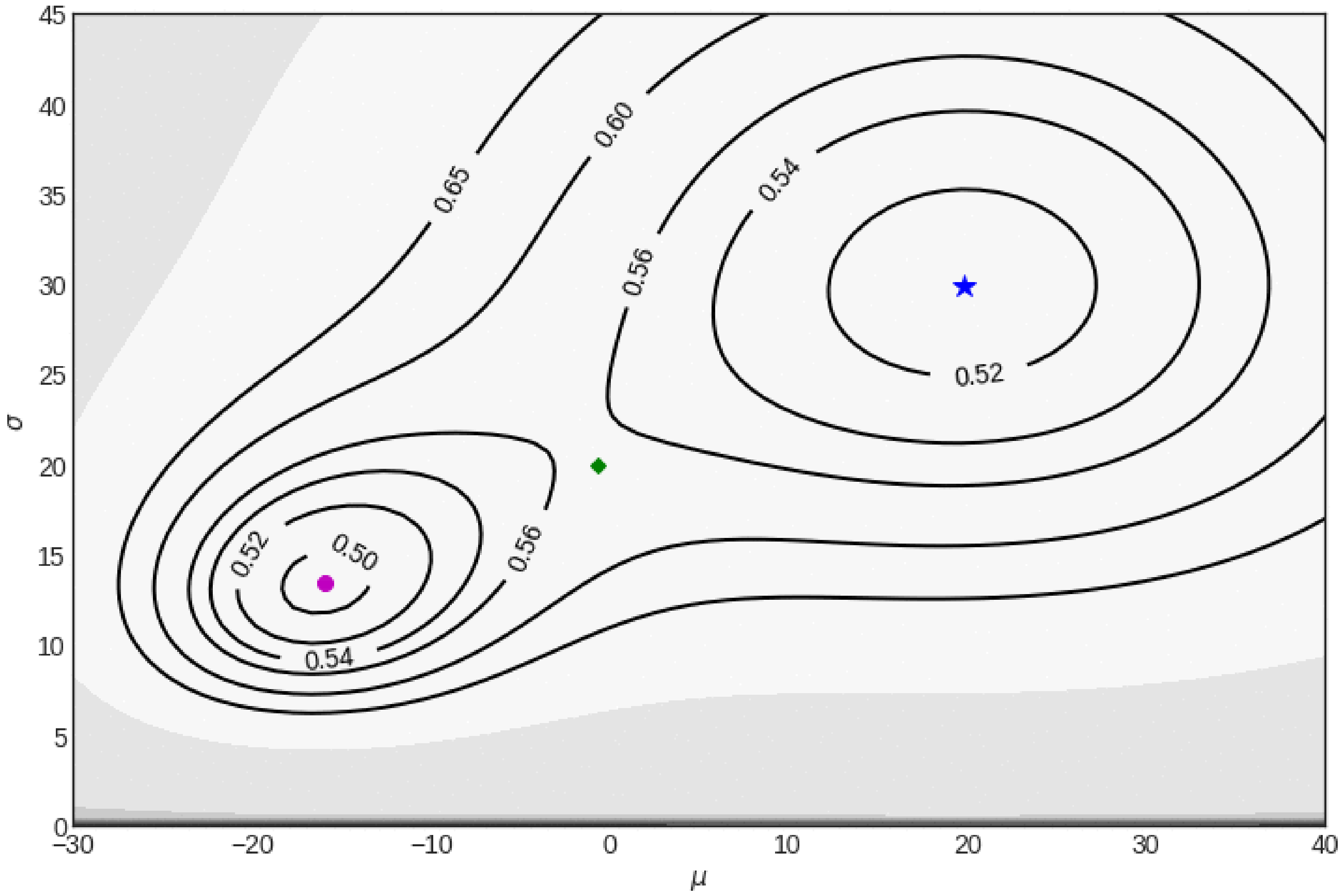
Flat minimum θ^{flat} ($l(\theta^{flat}) = 0.51$)

We want to land here!

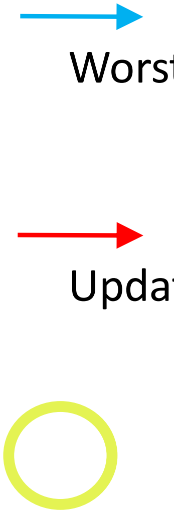
Sharp minimum θ^{sharp} ($l(\theta^{sharp}) = 0.49$)

Initial iterate

(Our) Fisher SAM



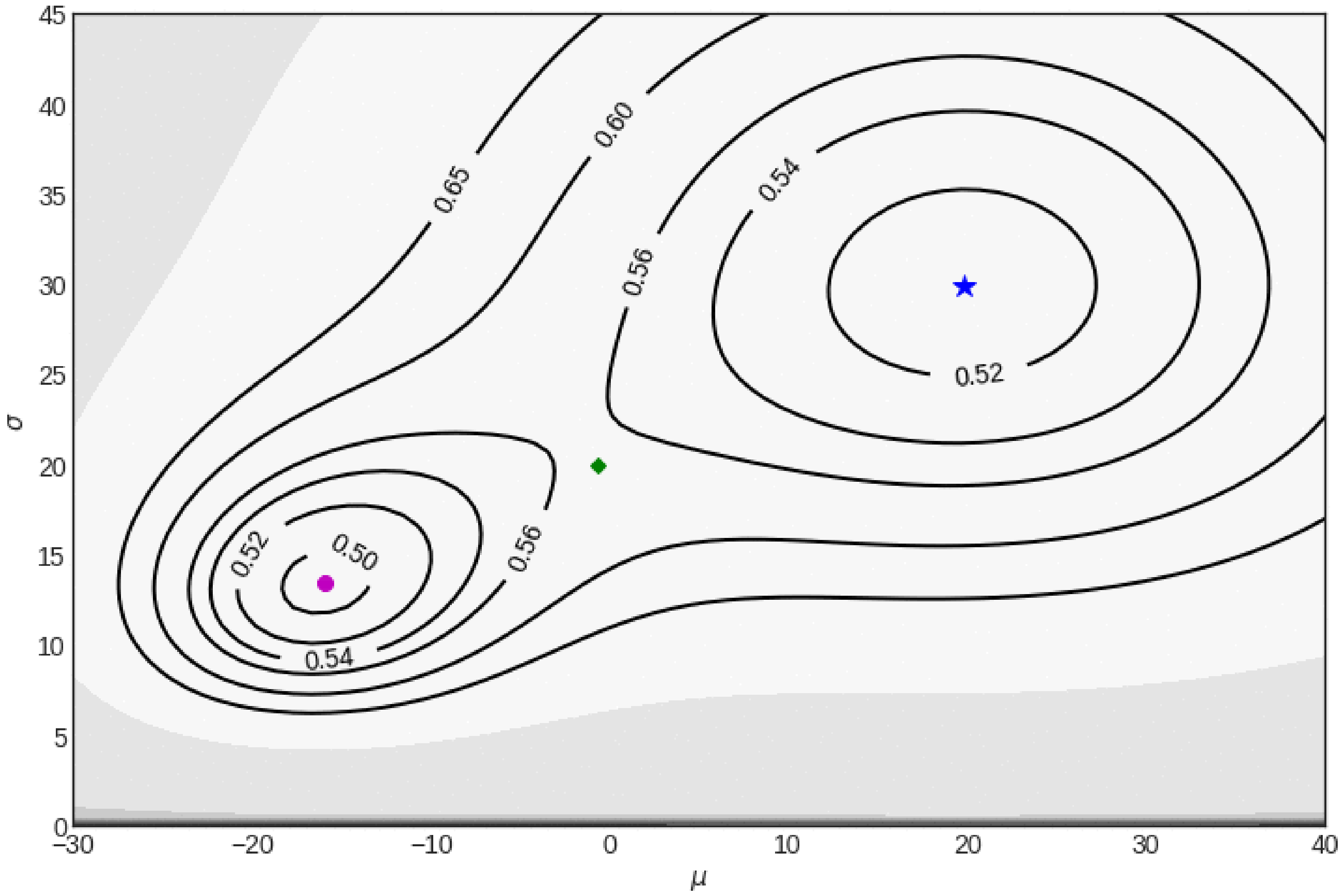
(Competitor) SAM(Foret et al, 2021)



Worst-case ϵ

Update direction

Neighborhood



Results on Image Classification

- Compare generalisation performance of:
  - SGD = vanilla (non-robust) optimization
  - SAM (Foret et al. 2021) = robust optim w/ Euclidean-ball neighborhood
  - ASAM (Kwon et al. 2021) = robust optim w/ parameter-scaled neighborhood
  - FSAM = proposed Fisher SAM (Fisher info neighborhood)

(Datasets = CIFAR-10/100 / 8 different neural networks)

Table 1. Test accuracies on CIFAR-10 and CIFAR-100.

	CIFAR-10				CIFAR-100			
	SGD	SAM	ASAM	FSAM	SGD	SAM	ASAM	FSAM
DenseNet-121	$91.83^{\pm 0.13}$	$92.44^{\pm 0.28}$	$92.70^{\pm 0.30}$	$\mathbf{92.81}^{\pm 0.17}$	$71.26^{\pm 0.15}$	$72.83^{\pm 0.01}$	$73.10^{\pm 0.23}$	$\mathbf{73.15}^{\pm 0.33}$
ResNet-20	$92.91^{\pm 0.13}$	$92.99^{\pm 0.16}$	$92.92^{\pm 0.15}$	$\mathbf{93.18}^{\pm 0.11}$	$68.24^{\pm 0.34}$	$68.61^{\pm 0.26}$	$68.68^{\pm 0.11}$	$\mathbf{69.04}^{\pm 0.30}$
ResNet-56	$95.37^{\pm 0.06}$	$95.59^{\pm 0.14}$	$95.63^{\pm 0.07}$	$\mathbf{95.71}^{\pm 0.08}$	$75.52^{\pm 0.27}$	$76.44^{\pm 0.26}$	$76.32^{\pm 0.14}$	$\mathbf{76.86}^{\pm 0.16}$
VGG-19-BN	$95.70^{\pm 0.09}$	$96.11^{\pm 0.09}$	$95.97^{\pm 0.10}$	$\mathbf{96.17}^{\pm 0.07}$	$73.45^{\pm 0.32}$	$77.25^{\pm 0.24}$	$74.36^{\pm 0.19}$	$\mathbf{77.86}^{\pm 0.22}$
ResNeXt-29-32x4d					$79.36^{\pm 0.19}$	$82.63^{\pm 0.16}$	$82.41^{\pm 0.31}$	$\mathbf{82.92}^{\pm 0.15}$
WRN-28-2	$95.56^{\pm 0.22}$	$96.28^{\pm 0.14}$	$96.25^{\pm 0.07}$	$\mathbf{96.51}^{\pm 0.08}$	$78.85^{\pm 0.25}$	$79.87^{\pm 0.13}$	$80.17^{\pm 0.14}$	$\mathbf{80.22}^{\pm 0.26}$
WRN-28-10	$97.12^{\pm 0.10}$	$97.56^{\pm 0.06}$	$97.63^{\pm 0.04}$	$\mathbf{97.89}^{\pm 0.07}$	$83.47^{\pm 0.21}$	$\mathbf{85.60}^{\pm 0.05}$	$85.20^{\pm 0.18}$	$\mathbf{85.60}^{\pm 0.11}$
PyramidNet-272	$97.73^{\pm 0.04}$	$97.91^{\pm 0.02}$	$97.91^{\pm 0.01}$	$\mathbf{97.93}^{\pm 0.04}$	$83.46^{\pm 0.02}$	$85.19^{\pm 0.04}$	$85.05^{\pm 0.11}$	$\mathbf{86.93}^{\pm 0.14}$

Transfer Learning

- Setup
  - From the vision transformer model (ViT-base) pretrained on ImageNet,
  - We finetune the model on CIFAR-10 with different losses (SGD/SAM/FSAM)

- Results (test accuracy %)

SGD	SAM (Foret et al)	ASAM (Kwon et al)	FSAM (Ours)
87.97 ± 0.12	87.99 ± 0.09	87.97 ± 0.08	**88.39 ± 0.13**

Robustness to Data Noise

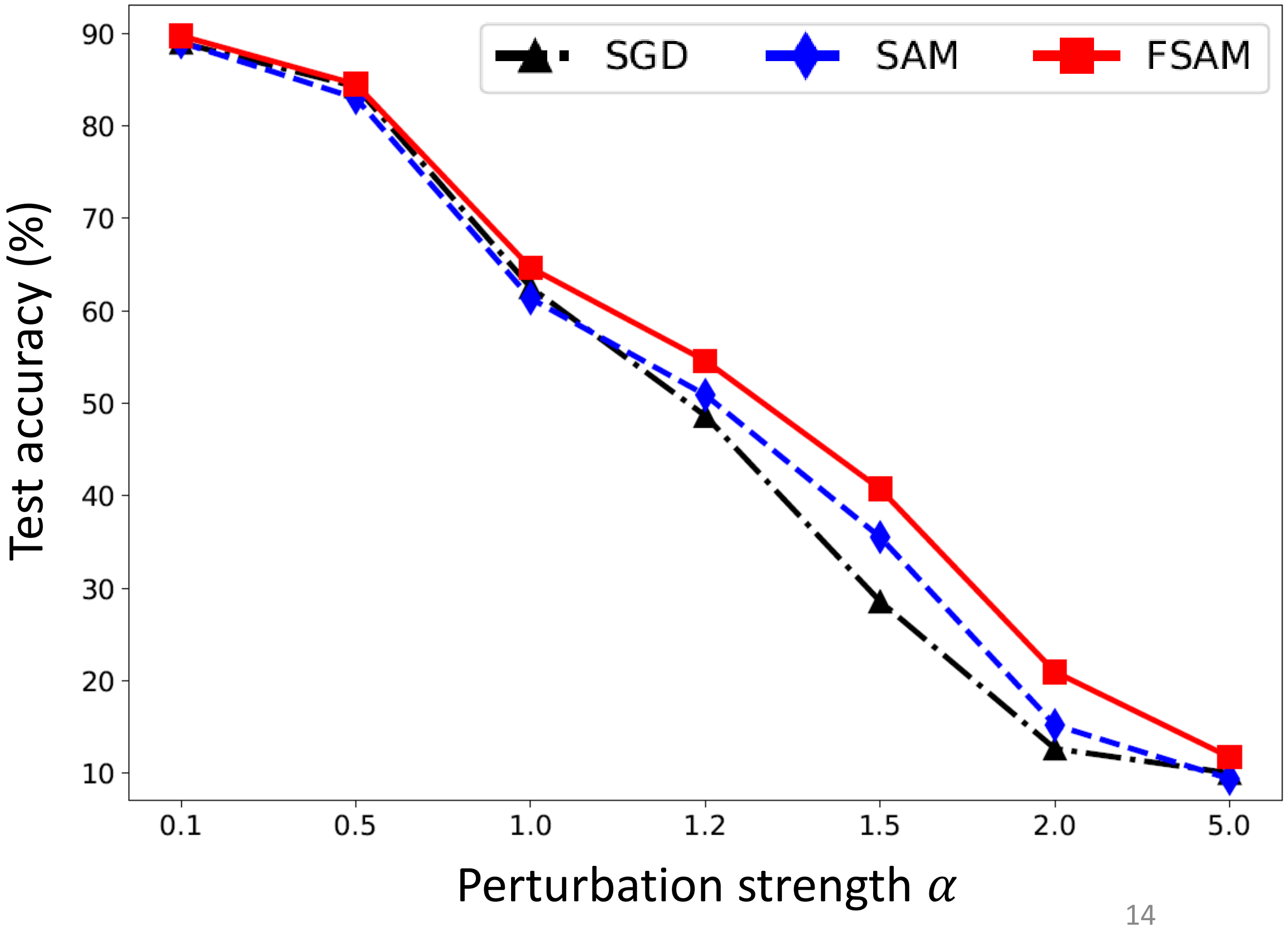
- Injecting label noise (perturbing the loss)
  - We inject label noise by randomly flipping class labels in training data
  - Different noise levels: 20/40/60/80%
- Check which of SGD, SAM, ASAM, and FSAM is the most robust
- Backbones: ResNet-32 Datasets: CIFAR-10

Table 2. Test accuracies on CIFAR-10 with label noise.

Noise rate	SGD	SAM	ASAM	FSAM
0.2	$87.97^{\pm 0.04}$	$\mathbf{93.12}^{\pm 0.24}$	$92.26^{\pm 0.33}$	$93.03^{\pm 0.11}$
0.4	$83.60^{\pm 0.59}$	$90.54^{\pm 0.19}$	$88.47^{\pm 0.06}$	$\mathbf{90.95}^{\pm 0.17}$
0.6	$76.97^{\pm 0.31}$	$85.39^{\pm 0.52}$	$82.32^{\pm 0.55}$	$\mathbf{85.76}^{\pm 0.21}$
0.8	$66.32^{\pm 0.27}$	$74.31^{\pm 1.02}$	$70.56^{\pm 0.27}$	$\mathbf{74.66}^{\pm 0.67}$

Robustness to Parameter Perturbation

- Setup
  - After training models with SGD/SAM/FSAM, we adversarially perturb the learned model parameters to see how test accuracy drop.
  - Perturbation magnitude varies (from weak to strong).
  - Backbone =ResNet34, Data = CIFAR-10



Theoretical Justification

The following holds for any θ w.p. at least $1 - \delta$ over S

$$\mathbb{E}_\epsilon[l_D(\theta + \epsilon)] \leq l^\gamma_{FSAM}(\theta; S) + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n - 1}}$$

Geometry-aware perturbation, $\epsilon \sim N(0, \rho^2 F^{-1}(\theta))$

Generalisation error

Training data

$\dim(\theta)$

Size of S

Conclusion

- A novel sharpness-aware loss that respects the underlying (Fisher) geometry of the parameter manifold
- Empirical evidence + theoretical bound on generalization error
- Possible future works
  - Combined with natural gradient updates
  - Distributed gradient update (related to Federated Learning)

Thank you!

Q&A