Towards Understanding Sharpness-Aware Minimization

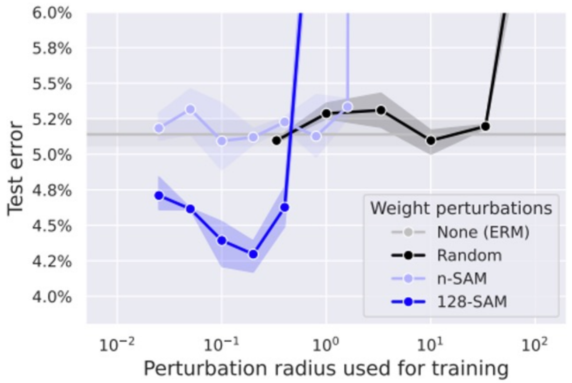
Maksym Andriushchenko (EPFL), Nicolas Flammarion (EPFL)

ICML | 2022

1. ***m*-sharpness** matters in m-SAM

$$\boldsymbol{m}\textbf{-SAM}: \min_{w \in \mathbb{R}^{|w|}} \sum_{\substack{\mathcal{S} \subset \mathcal{S}_{train}, \\ |\mathcal{S}|=m}} \max_{\|\delta\|_2 \leq \rho} \sum_{i \in \mathcal{S}} \ell_i(w+\delta)$$

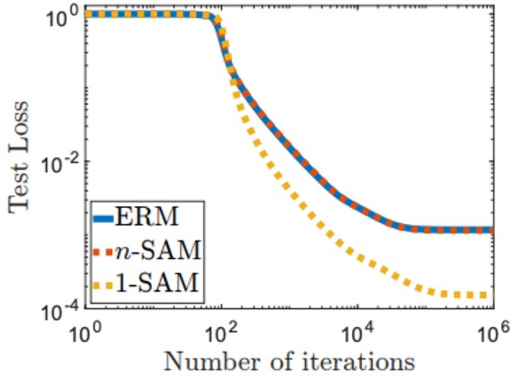
ResNet-18 on CIFAR-10



⚠ The PAC-Bayes generalization bound doesn't explain this

2. The **implicit bias** of 1-SAM vs. n-SAM and ERM can be well understood for diagonal linear networks

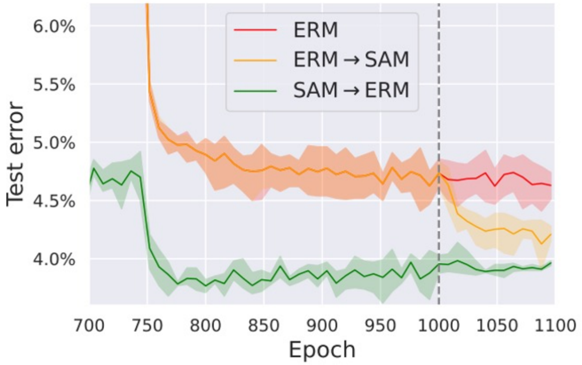
Diagonal linear network



💡 Simple models can be surprisingly predictive

3. m-SAM has some interesting effects: running ERM → SAM **gradually improves generalization**

ResNet-18 on CIFAR-10



❗ The same also happens for diagonal linear networks

joint work with
Prof. Nicolas Flammarion

28 October 2022

ELLIS Mathematics of Deep Learning reading group

Background: Sharpness-Aware Minimization

- Sharpness-Aware Minimization (SAM) [Foret et al., ICLR'21]:

$$w_{t+1} = w_t - \frac{\gamma_t}{|I_t|} \sum_{i \in I_t} \nabla \ell_i \Big(w_t + \frac{\rho_t}{|I_t|} \sum_{j \in I_t} \nabla \ell_j (w_t) \Big)$$

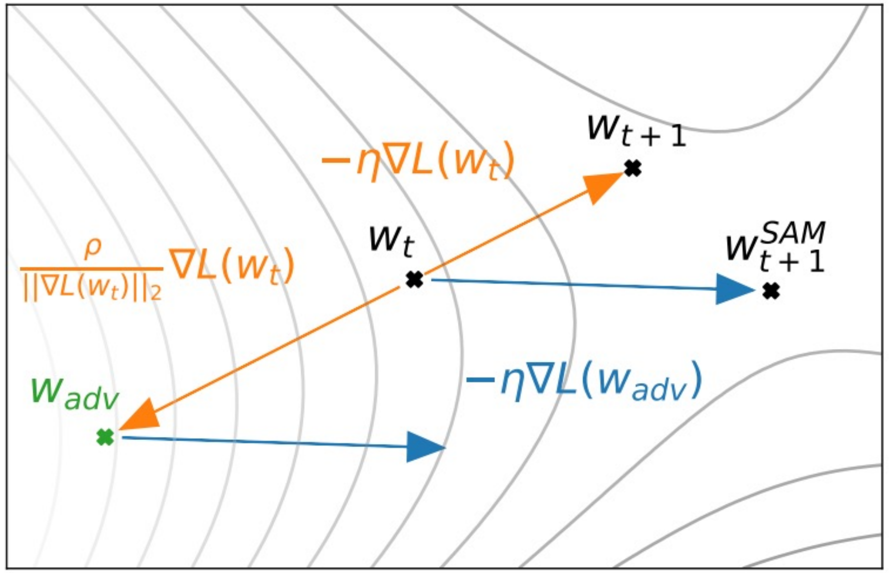
where ρ_t can optionally include $1/||\nabla||_2$

- Foret et al., ICLR'21 motivate SAM by minimization of **sharpness:**

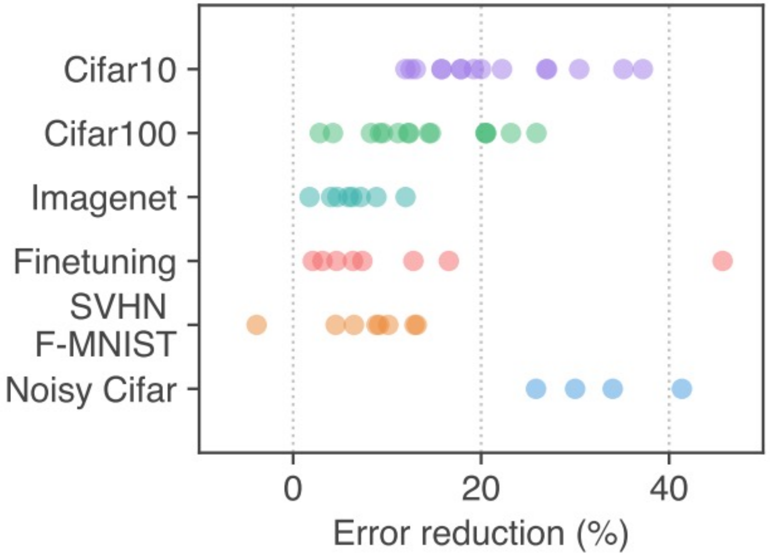
$$\min_{w \in \mathbb{R}^{|w|}} \max_{\|\delta\|_2 \le \rho} \frac{1}{n} \sum_{i=1}^{n} \ell_i (w + \delta)$$

- SAM consistently **improves generalization** in the state-of-the-art settings (!) and has only 2x computational overhead

Visual description of the SAM algorithm



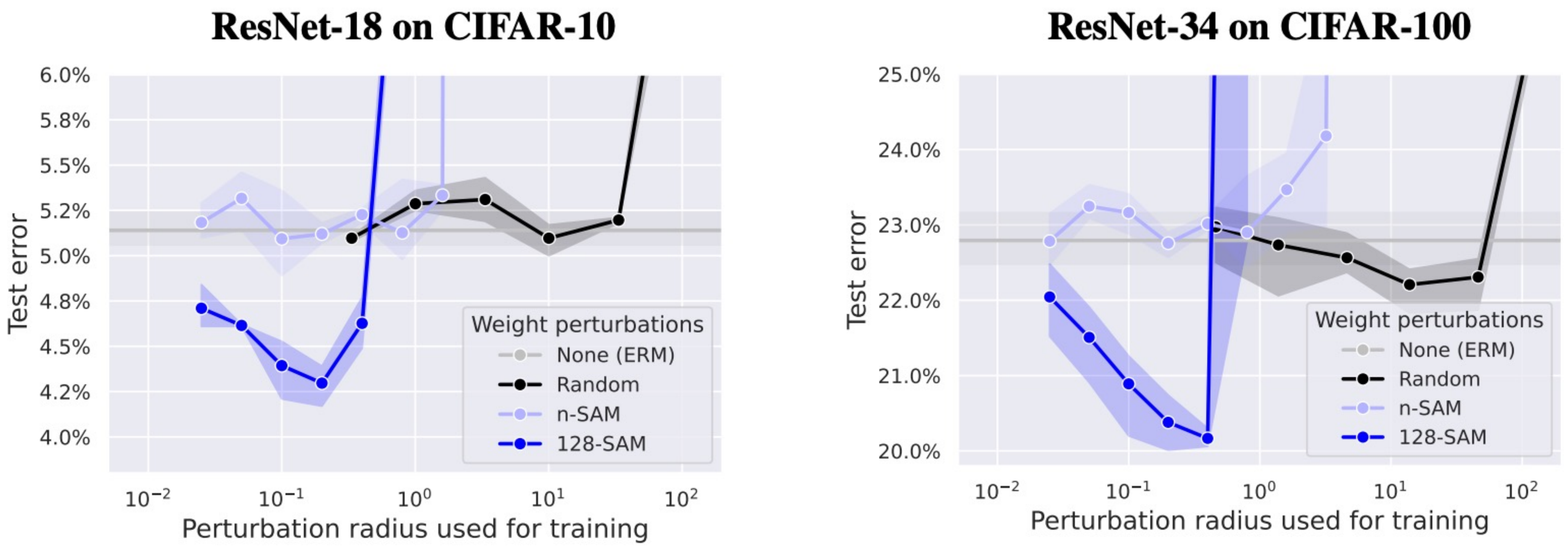
Source: Foret et al, ICLR'21



Which components of SAM are crucial?

$$\textbf{\textit{n}-SAM:} \min_{w \in \mathbb{R}^{|w|}} \max_{\|\delta\|_2 \le \rho} \sum_{i=1}^{n} \ell_i(w+\delta) \quad \rightarrow \quad \textbf{\textit{m}-SAM:} \min_{w \in \mathbb{R}^{|w|}} \sum_{\substack{\mathcal{S} \subset \mathcal{S}_{train}, \\ |\mathcal{S}|=m}} \max_{\|\delta\|_2 \le \rho} \sum_{i \in \mathcal{S}} \ell_i(w+\delta)$$

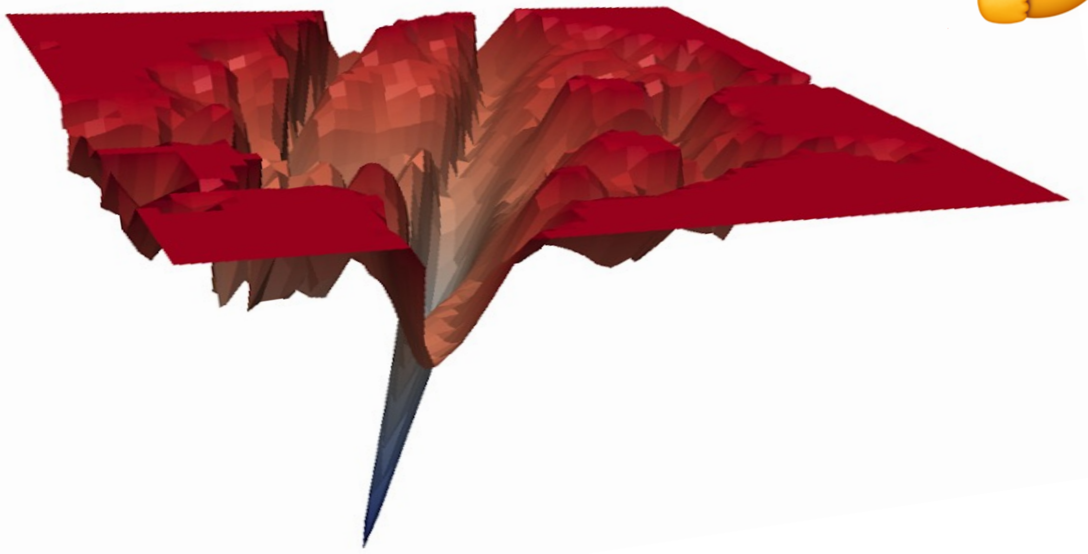
Worst-case weight perturbations, with a small m (aka $\boldsymbol{m}$**-sharpness**) are key!



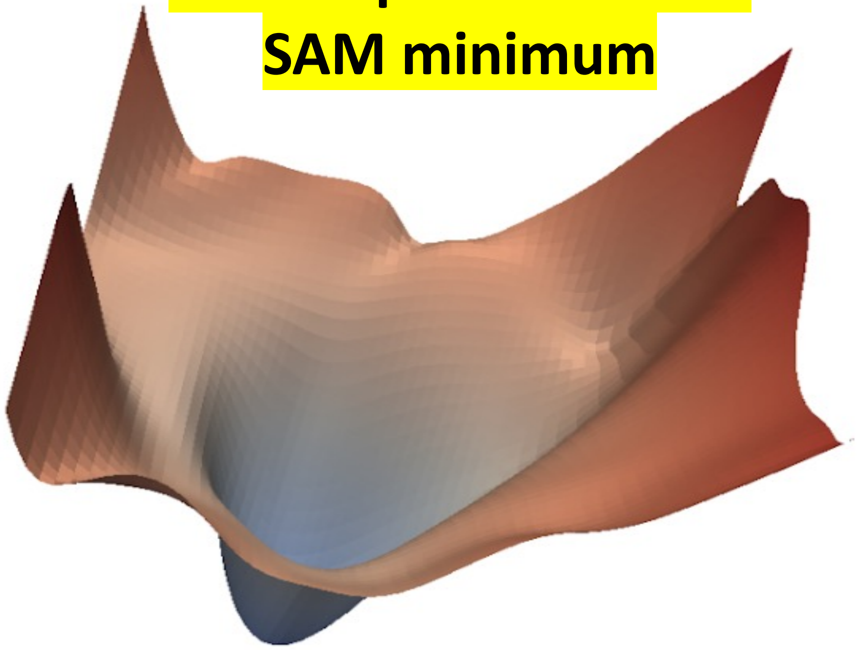
Note: state-of-the-art setting with weight decay, BatchNorm, and data augmentation

Sharp minima vs. flat minima?

2D subspace around an ERM minimum



2D subspace around a SAM minimum



Source of the loss surfaces: Sharpness-Aware Minimization for Efficiently Improving Generalization, ICLR'21

Importance of m-sharpness $\Rightarrow$ the common intuition about the benefits of converging to flat minima <u>of the training loss landscape</u> is unlikely to explain SAM!

PAC-Bayesian generalization bound and SAM?

A.1 PAC Bayesian Generalization Bound

Below, we state a generalization bound based on sharpness.

Theorem 2. *For any $\rho > 0$ and any distribution $\mathscr{D}$, with probability $1-\delta$ over the choice of the training set $\mathcal{S} \sim \mathscr{D}$,*

$$L_{\mathscr{D}}(\boldsymbol{w}) \leq \max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\epsilon}) + \sqrt{\frac{k \log\left(1+\frac{\|\boldsymbol{w}\|_2^2}{\rho^2}\left(1+\sqrt{\frac{\log(n)}{k}}\right)^2\right) + 4\log\frac{n}{\delta} + \tilde{O}(1)}{n-1}} \tag{4}$$

where $n = |\mathcal{S}|$, k is the number of parameters and we assumed $L_{\mathscr{D}}(\boldsymbol{w}) \leq \mathbb{E}_{\epsilon_i \sim \mathcal{N}(0,\rho)}[L_{\mathscr{D}}(\boldsymbol{w}+\boldsymbol{\epsilon})]$.

Source: Sharpness-Aware Minimization for Efficiently Improving Generalization, ICLR'21

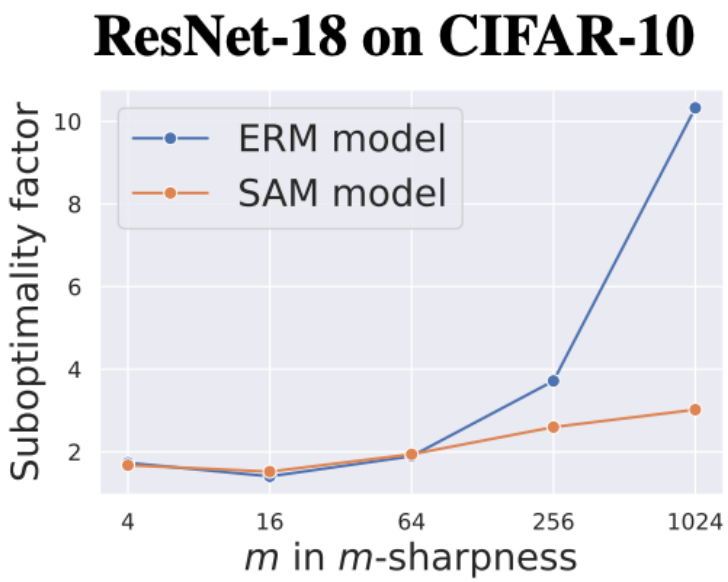
Importance of m-sharpness $\Rightarrow$ PAC-Bayes generalization is derived for random perturbations and can't explain the success of m-SAM

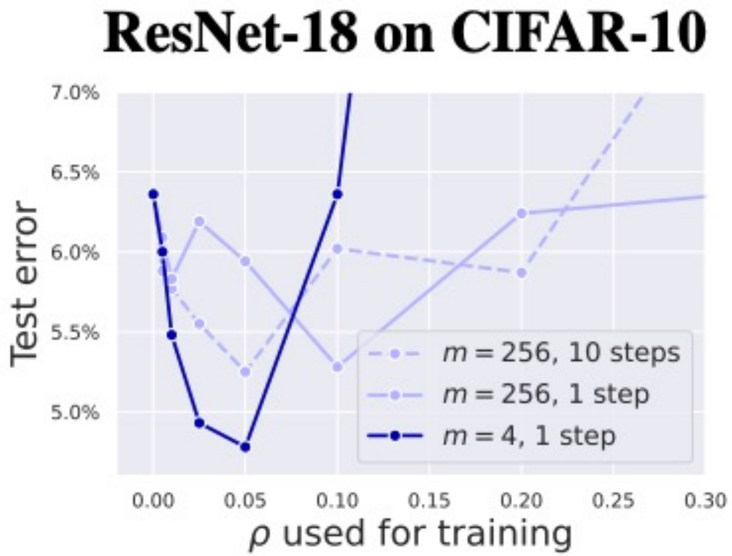
So why can m-sharpness be helpful in m-SAM?

Maybe some straightforward explanations?

- **Hypothesis 1**: *with a lower m, the ascent step of SAM more accurately maximizes the inner max*.
  → Some evidence towards this hypothesis, but using >1 step for the inner max **doesn't help**.

- **Hypothesis 2**: *the regularization effect of BatchNorm used with smaller batches (aka Ghost BatchNorm)*
  → Also **no**, we can see the generalization improvements from m-SAM also with other normalization schemes

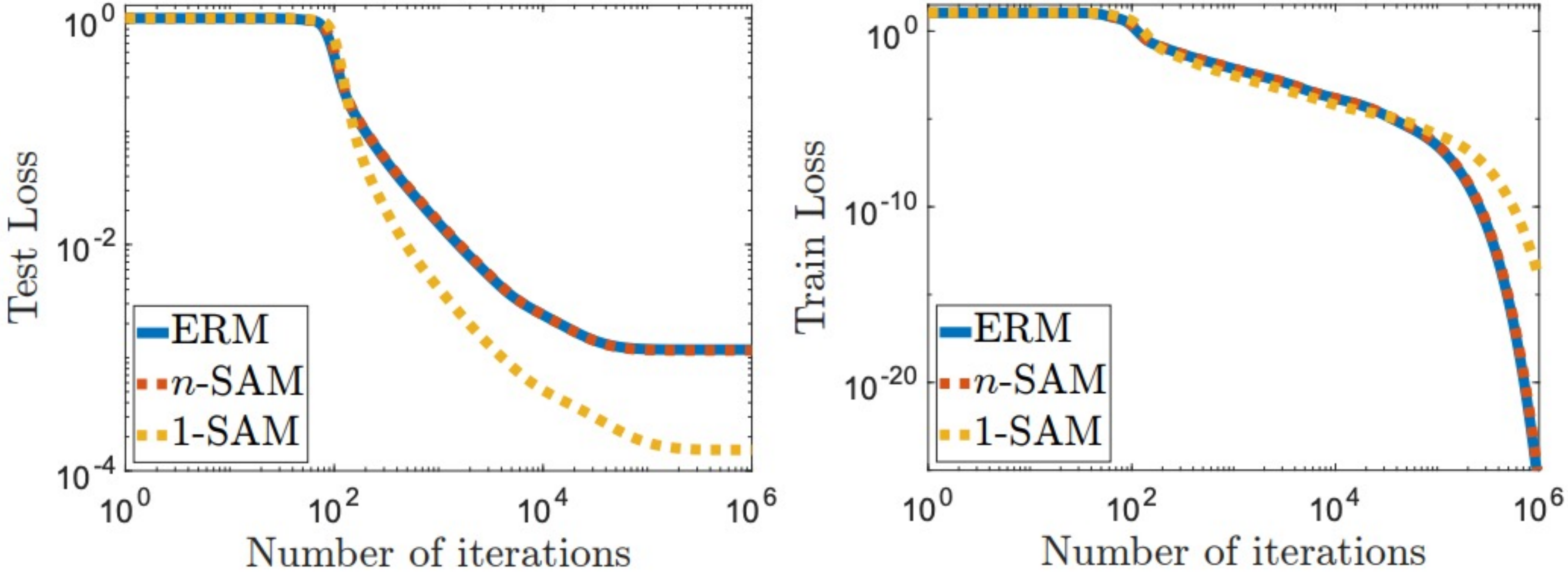




Our approach: understanding m-SAM on simple models

We will use **diagonal linear networks** $\boldsymbol{f(x) = \langle x, u \odot v \rangle}$ for sparse regression that shows different generalization depending on the initialization scale and SGD noise

1-SAM for $f(x)$ generalizes significantly better than ERM and n-SAM!



We are also able to capture it **theoretically**: 1-SAM promotes **sparsity** in terms of the linear predictor $u \odot v$ (and much more than n-SAM) ⬇️

A detour: implicit bias in machine learning

- Understanding deep learning requires rethinking generalization (ICLR'17): the key regularization effect for overparametrized networks must come from the opt. algorithm

- So what do we mean by the implicit bias? Say, $L^\star$ is an optimal predictor on the training set, then algorithm A induces an implicit bias $\phi(\beta)$ if

$$\beta_A = \underset{\beta \ s.t.\ L(\beta) = L^\star}{\arg\min}\, \phi(\beta)$$

- For example, for gradient descent on linear models: $\phi(\beta) = ||\beta - \beta_0||_2$

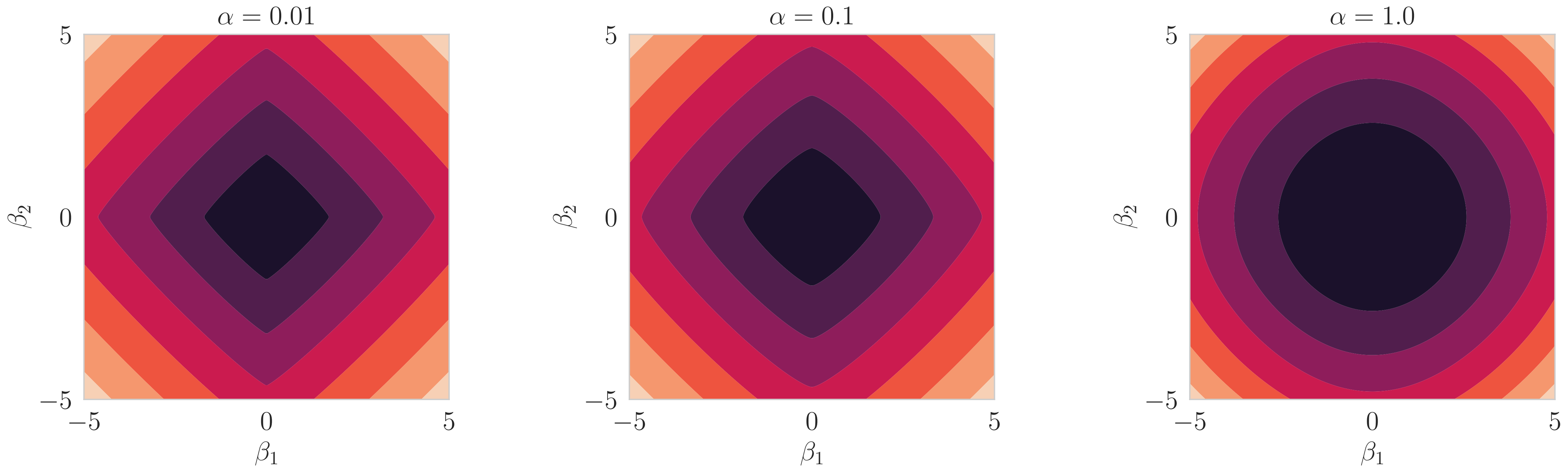
- [Woodworth et al., 2020]: for diagonal linear neural networks (overparam. regression with squared loss) solved with gradient flow, the initialization scale α matters

$$f_{u,v}(x) = \langle x, u \odot v \rangle \qquad (\mathrm{u}_\alpha^\infty, \mathrm{v}_\alpha^\infty) = \underset{u,v \in \mathbb{R}^d \ s.t.\ X(u \odot v) = y}{\arg\min}\, \phi_\alpha(u \odot v)$$

Diagonal linear networks: role of the initialization scale

- The role of α in the hyperbolic entropy: interpolation **between ℓ_1 and ℓ_2 norms**

$$\phi_\alpha(\beta) = \alpha \sum_{i=1}^{d} q\left(\frac{\beta_i}{\alpha^2}\right) \quad \text{where} \quad \mathrm{q(z)} = 2 - \sqrt{4 + z^2} + z \cdot \operatorname{arcsinh}(z/2)$$



close to ℓ_1 | a mix of ℓ_1 and ℓ_2 | close to ℓ_2

Diagonal linear networks: effect of SAM

- Our result for 1-SAM and n-SAM: **both decrease the effective parameter α in the hyperbolic entropy $\phi_\alpha(\beta)$ but n-SAM reduces it significantly more**

Theorem 1 (Informal). *Assuming global convergence, the solutions selected by the full-batch versions of the* 1*-SAM and n-SAM algorithms taken with infinitesimally small step sizes and initialized at $w_+ = w_- = \alpha \in \mathbb{R}^d_{>0}$, solve the optimization problem (6) with effective parameters:*

$$\alpha_{\text{1-SAM}} = \alpha \odot e^{-\rho \Delta_{\text{1-SAM}} + O(\rho^2)}, \quad \alpha_{n\text{-SAM}} = \alpha \odot e^{-\rho \Delta_{n\text{-SAM}} + O(\rho^2)},$$

where $\Delta_{\text{1-SAM}}, \Delta_{n\text{-SAM}} \in \mathbb{R}^d_+$ for which typically:

$$\|\Delta_{\text{1-SAM}}\|_1 \approx d \int_0^\infty L(w(s))ds \quad \text{and}$$
$$\|\Delta_{n\text{-SAM}}\|_1 \approx \frac{d}{n} \int_0^\infty L(w(s))ds.$$

- So 1-SAM promotes **sparsity** of the linear predictor $u \odot v$ (and much more than n-SAM)
- **This implicit bias of SAM can explain its generalization benefits for this problem**

Optimization theory: general convergence results for SAM

- We analyze the convergence of the stochastic version of m-SAM ($m = |I_t|$):

$$w_{t+1} = w_t - \frac{\gamma_t}{|I_t|} \sum_{i \in I_t} \nabla \ell_i \big(w_t + \frac{\rho_t}{|I_t|} \sum_{j \in I_t} \nabla \ell_j (w_t) \big)$$

- Note: the same batch I_t is used for the inner and outer updates (as in SAM)

- However, we don't consider ℓ_2 gradient normalization (i.e., $||\nabla L||_2$) but we show empirically that it's not important for generalization

- **Why interesting**:
  - we need to sufficiently minimize the loss
  - the implicit bias result **requires** convergence to a global min
  - and in practice **we converge to nearly zero training loss even with SAM** (!)

General convergence results for SAM

Assumptions

(A1) (Bounded variance). *There exists* $\sigma \geq 0$ *s.t.* $\mathbb{E}[\|\nabla \ell_i(w) - \nabla L(w)\|^2] \leq \sigma^2$ *for all* $i \sim \mathcal{U}([\![1, n]\!])$ *and* $w \in \mathbb{R}^d$.

(A2) (Individual β-smoothness). *There exists* $\beta \geq 0$ *s.t.* $\|\nabla \ell_i(w) - \nabla \ell_i(v)\| \leq \beta \|w - v\|$ *for all* $w, v \in \mathbb{R}^d$ *and* $i \in [\![1, n]\!]$.

(A3) (Polyak-Lojasiewicz). *There exists* $\mu > 0$ *s.t.* $\frac{1}{2}\|\nabla L(w)\|^2 \geq \mu(L(w) - L_*)$ *for all* $w, v \in \mathbb{R}^d$.

Convergence theorem

Theorem 2. *Assume* **(A1)** *and* **(A2)** *for the iterates (4). Then for any number of iterations* $T \geq 0$, *batch size* b, *and step sizes* $\gamma_t = \frac{1}{\sqrt{T}\beta}$ *and* $\rho_t = \frac{1}{T^{1/4}\beta}$, *we have:*

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1}\|\nabla L(w_t)\|^2\right] \leq \frac{4\beta}{\sqrt{T}}(L(w_0) - L_*) + \frac{8\sigma^2}{b\sqrt{T}},$$

In addition, under **(A3)**, *with step sizes* $\gamma_t = \min\{\frac{8t+4}{3\mu(t+1)^2}, \frac{1}{2\beta}\}$ *and* $\rho_t = \sqrt{\gamma_t/\beta}$:

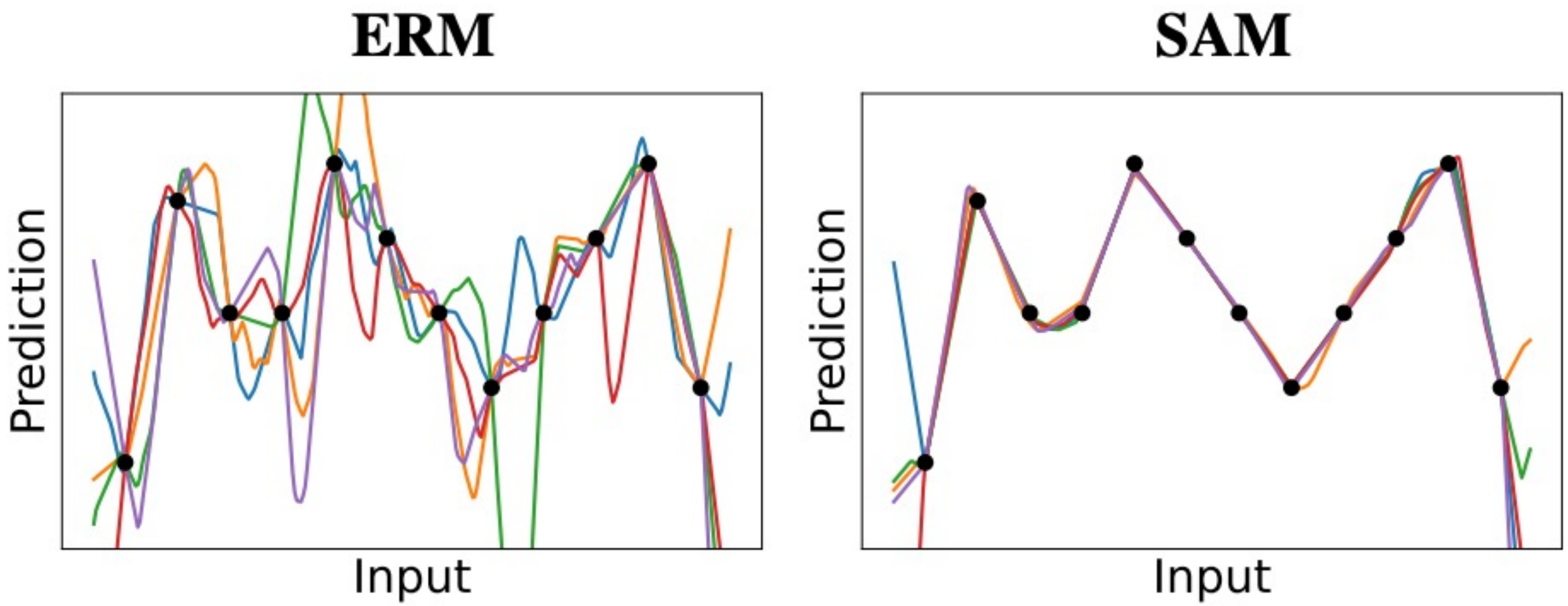
$$\mathbb{E}\left[L(w_T)\right] - L_* \leq \frac{3\beta^2(L(w_0) - L_*)}{\mu^2 T^2} + \frac{22\beta\sigma^2}{\mu^2 bT}.$$

Some people had the intuition that SAM helps generalization
because it prevents convergence → not true

Now let's switch gears and explore the effect of SAM empirically

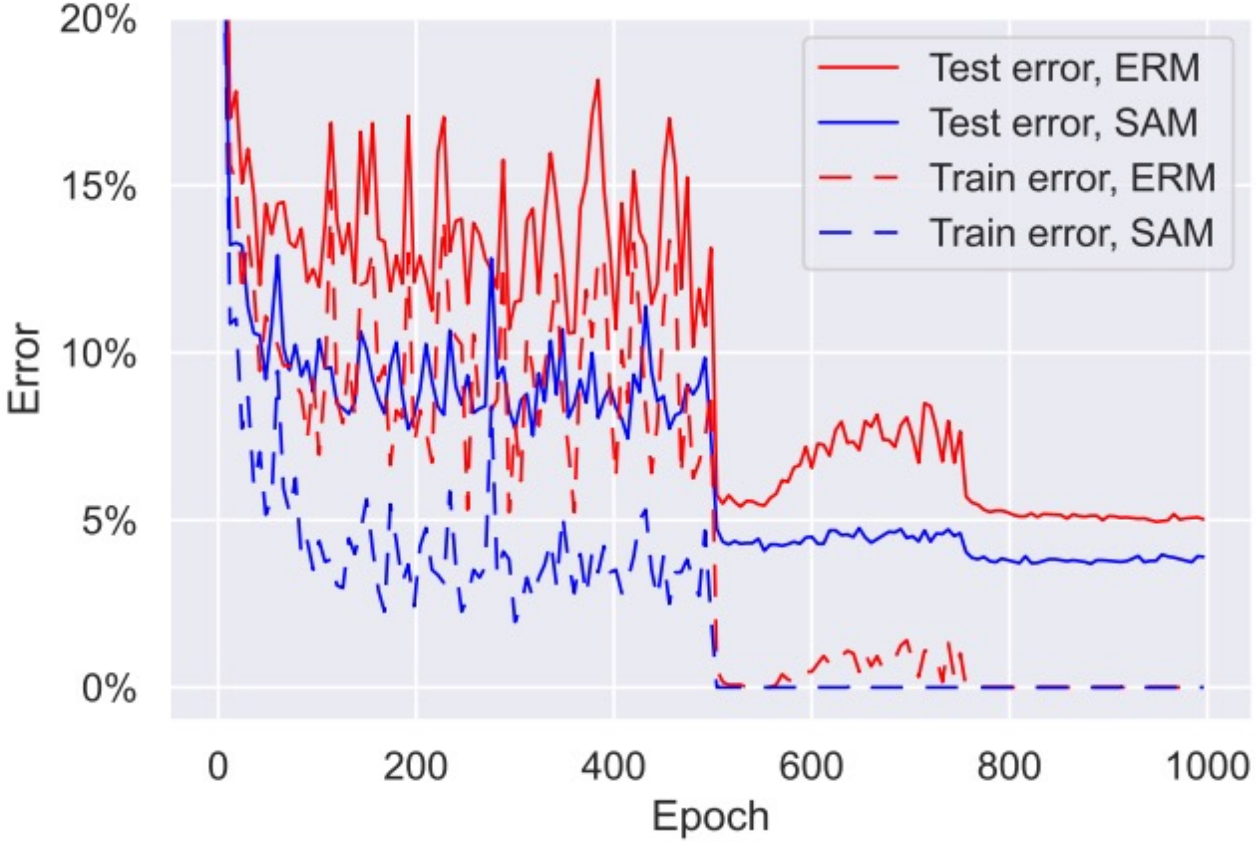
m-SAM for 2-layer ReLU networks: sparsity bias

For **non-linear** networks, we can observe some interesting properties empirically



Using SAM for 2-layer ReLU networks on simple 1D regression also leads to a **sparsifying effect** but in terms of the **ReLU kinks**

What happens for deep networks: convergence and generalization

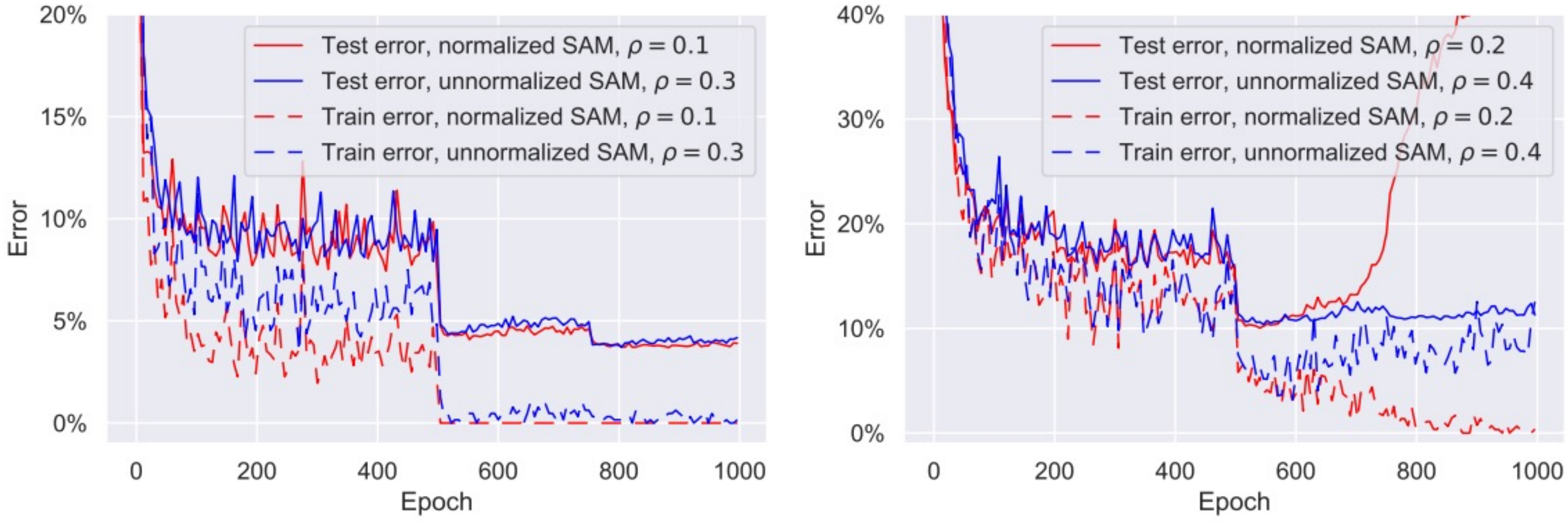


Setting:
ResNet-18 on CIFAR-10 with data augmentation

- Both ERM and SAM converge to nearly zero training loss: 0.0012 for ERM vs 0.0009 for SAM => **our convergence result is relevant**
- However, the SAM model has **much better generalization performance**: 3.76% vs 5.03% test error

What happens for deep networks: normalization in SAM

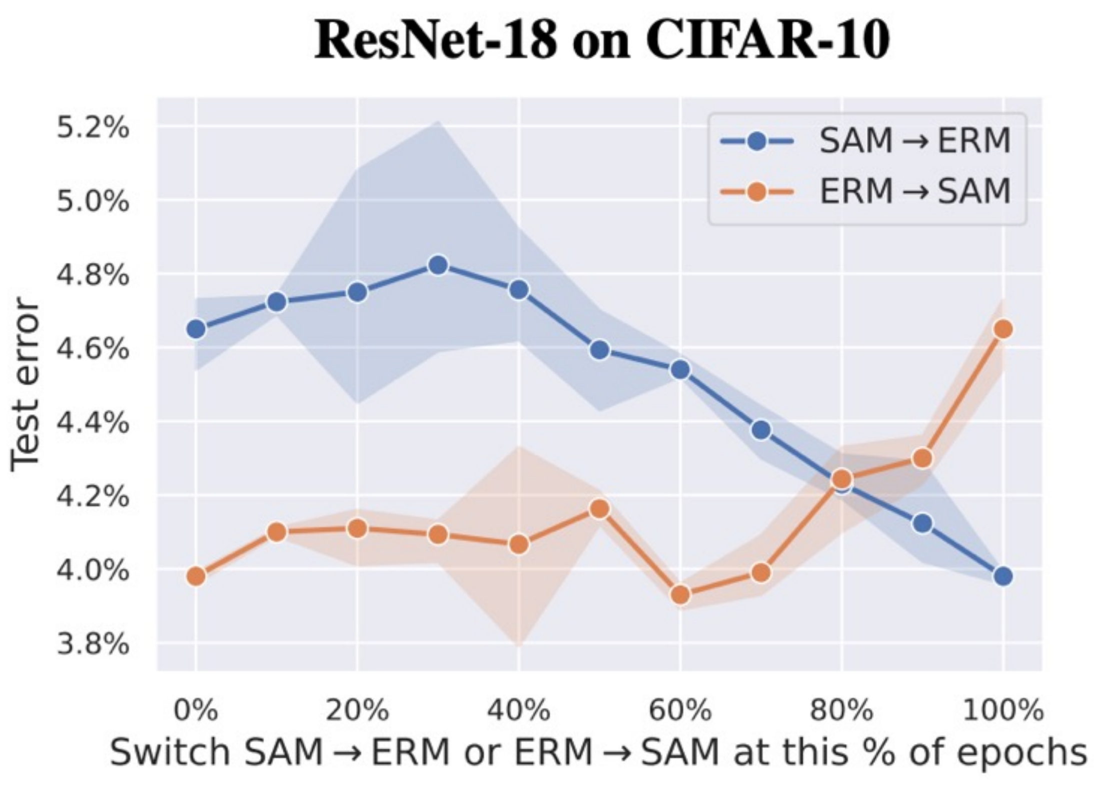
- Our convergence result holds for **unnormalized SAM**, i.e. we assumed no scaling of the SAM updates by $||\nabla L||_2$ (as this may prevent convergence in some cases)
- But empirically normalization isn't important for improving generalization

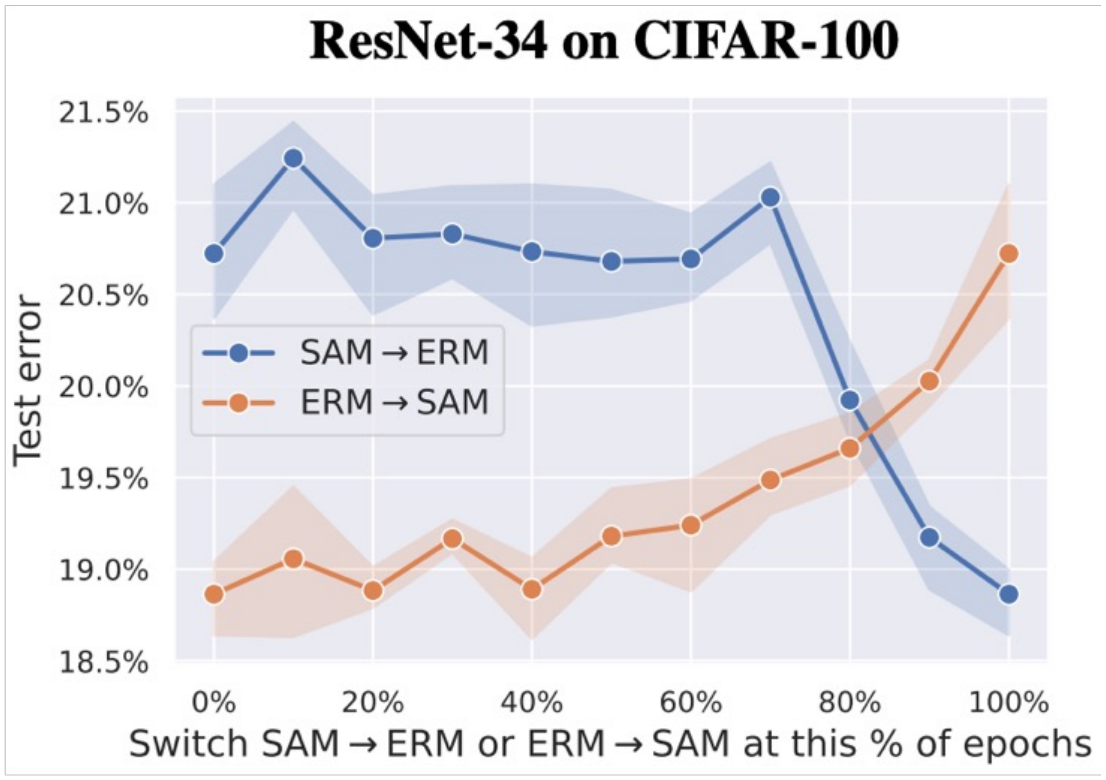


(a) Without label noise **(b)** With 60% label noise

At which stage of training the effect of SAM is important? (part I)

Here we switch **from SAM to ERM** and **from ERM to SAM** at different stages of training



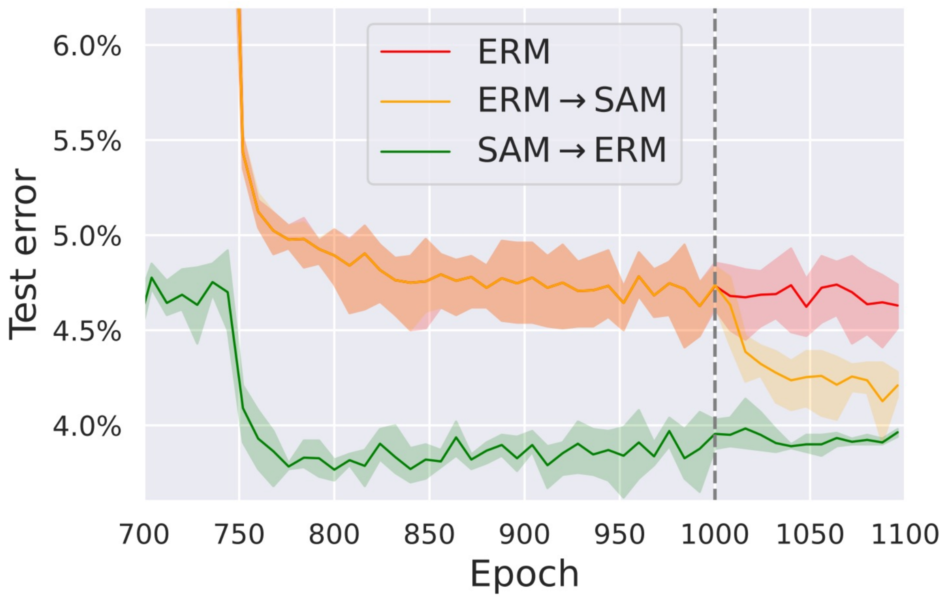


→ SAM has the most important effect in the second half of the training

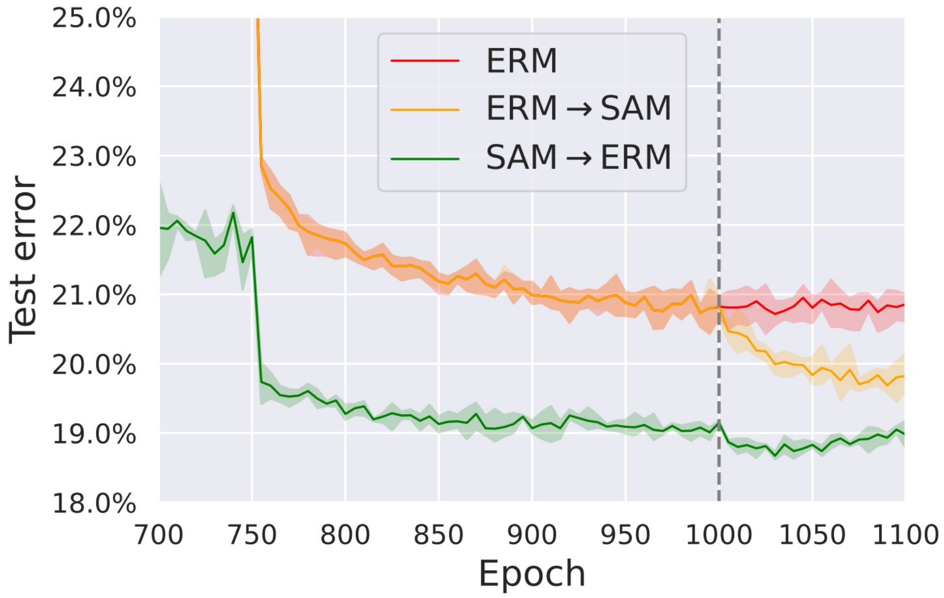
At which stage of training the effect of SAM is important? (part II)

A curious property of SAM: if we finetune an ERM model with SAM
<u>on the same dataset</u>, we get a **significant generalization improvement**

ResNet-18 on CIFAR-10



ResNet-34 on CIFAR-100



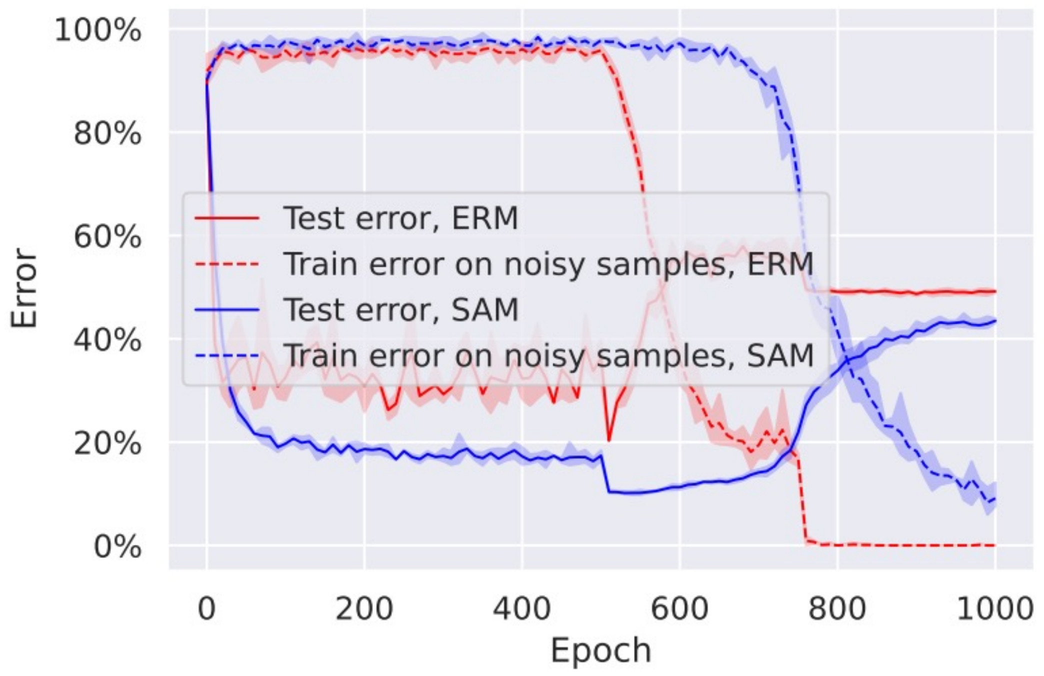
+ minima of ERM and ERM→SAM are linearly connected

And it's not so mysterious: **exactly the same** phenomena are observed also for **diagonal linear networks** where we can explain the dynamics quite well!

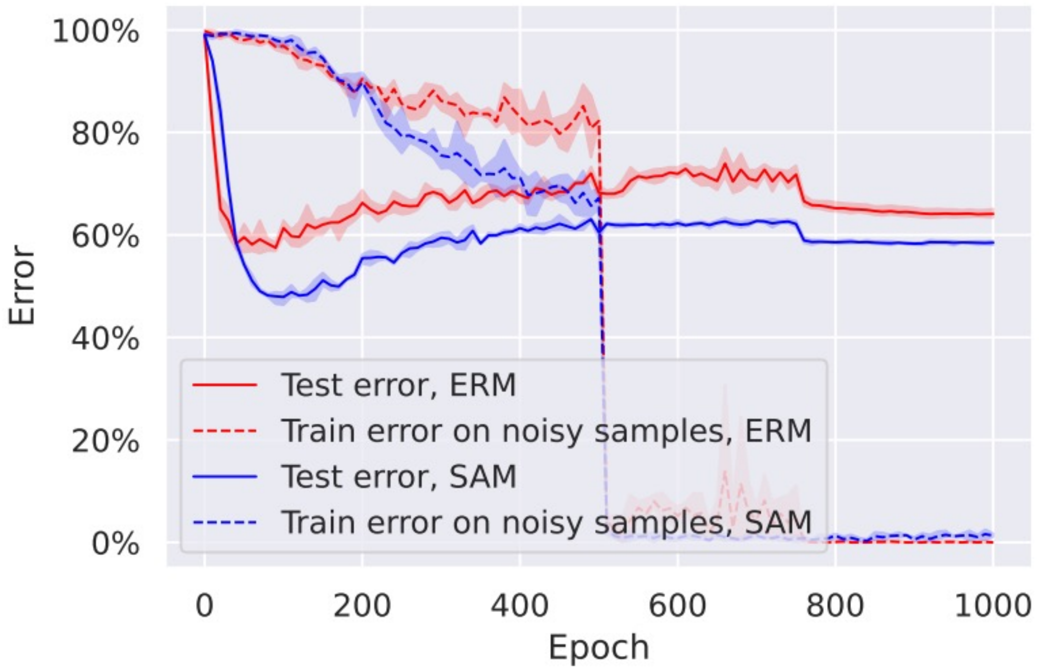
What happens with SAM on mislabelled data?

Convergence of SAM to global minima can also have a **negative impact**
→ e.g., SAM overfits similarly to ERM when trained on mislabelled data

ResNet-18 on CIFAR-10



ResNet-34 on CIFAR-100



This also suggests that the beneficial effect of SAM is observed not only close to a minimum but also **along the whole optimization trajectory**

Future directions

1. What is the implicit bias of SAM for non-linear neural networks in terms of the learned function?

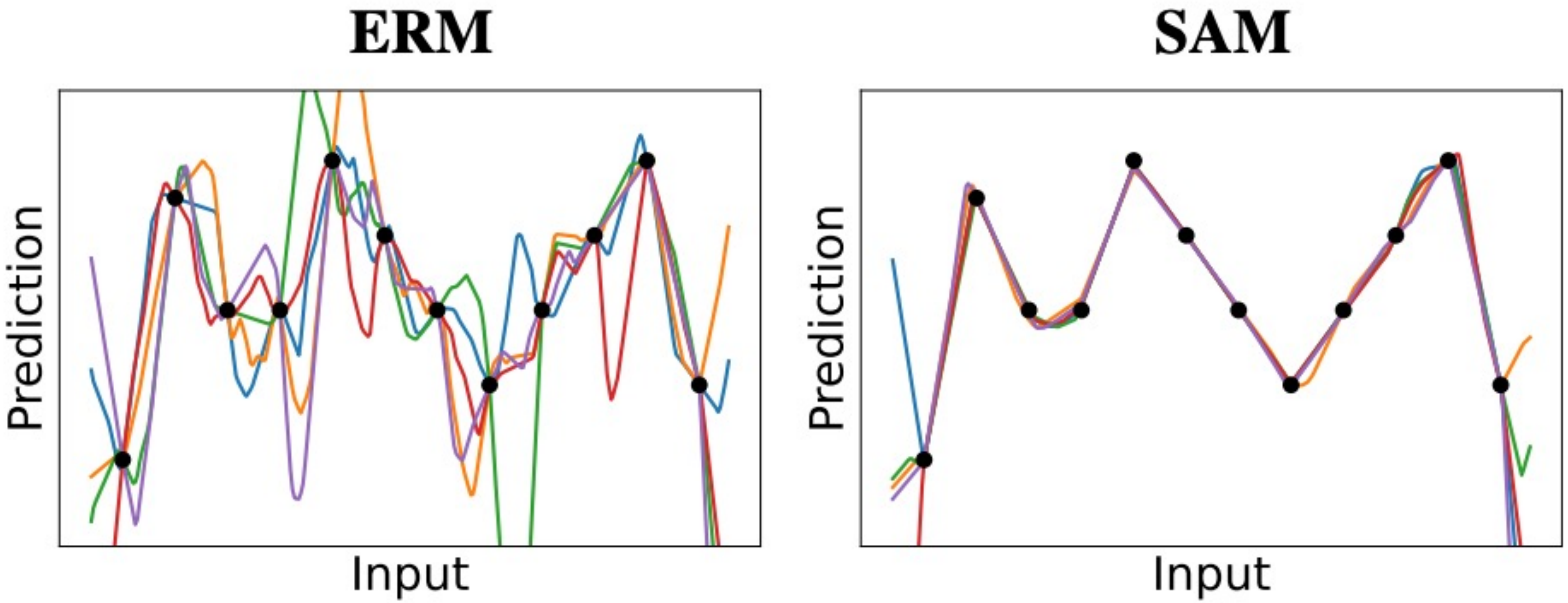
2. Why does sharpness still makes sense despite its obvious flaws (Sharp Minima Can Generalize For Deep Nets (ICML'17))?

3. Why is SAM so beneficial for vision transformers: When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations (ICLR'22)?

4. More in-depth exploration of SAM in the noisy label setting: why does it work?

Before we conclude, a few more words about 1.

A follow-up on the sparsity observation

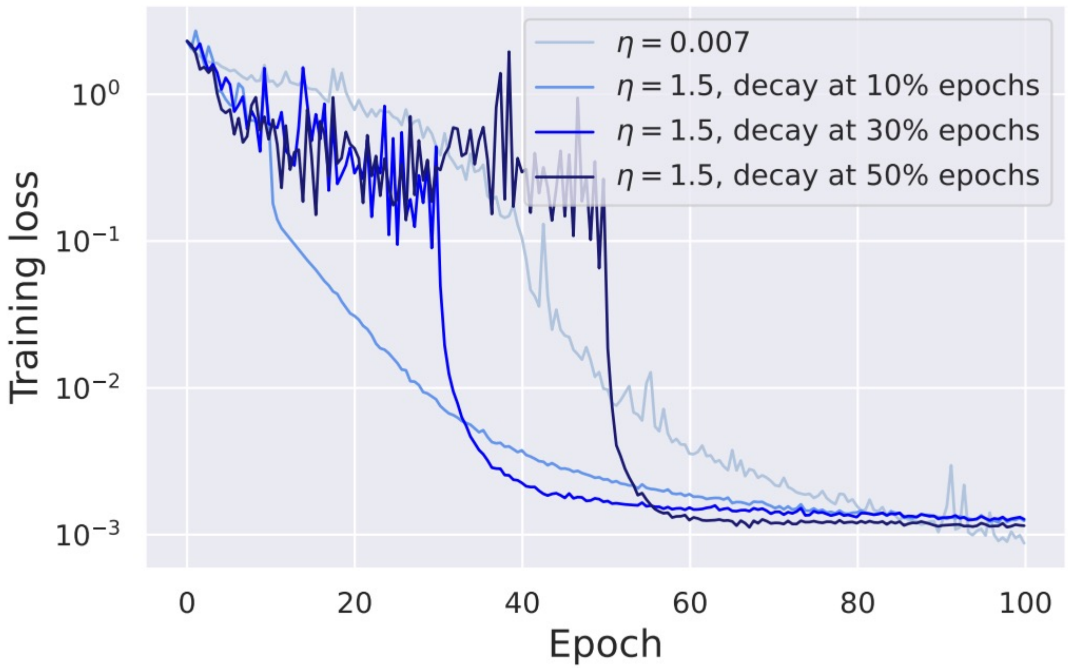


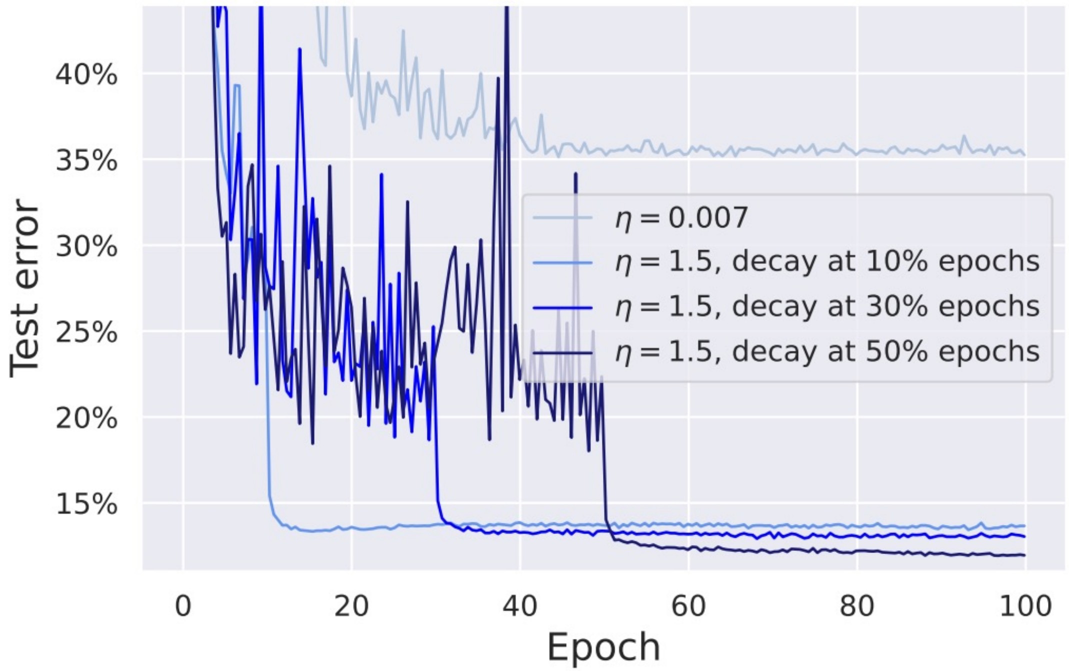
This observation is quite curious. Can we understand it better?

Can the same effect be achieved with standard SGD?

New paper: SGD with large step sizes learns sparse features

A typical training dynamics for a ResNet-18 on CIFAR-10

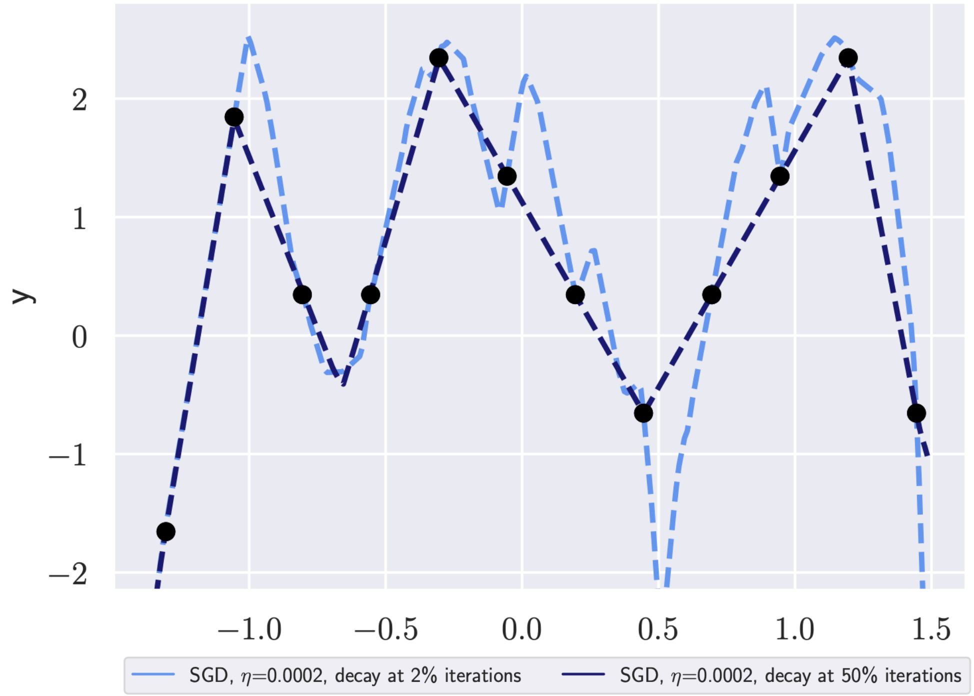


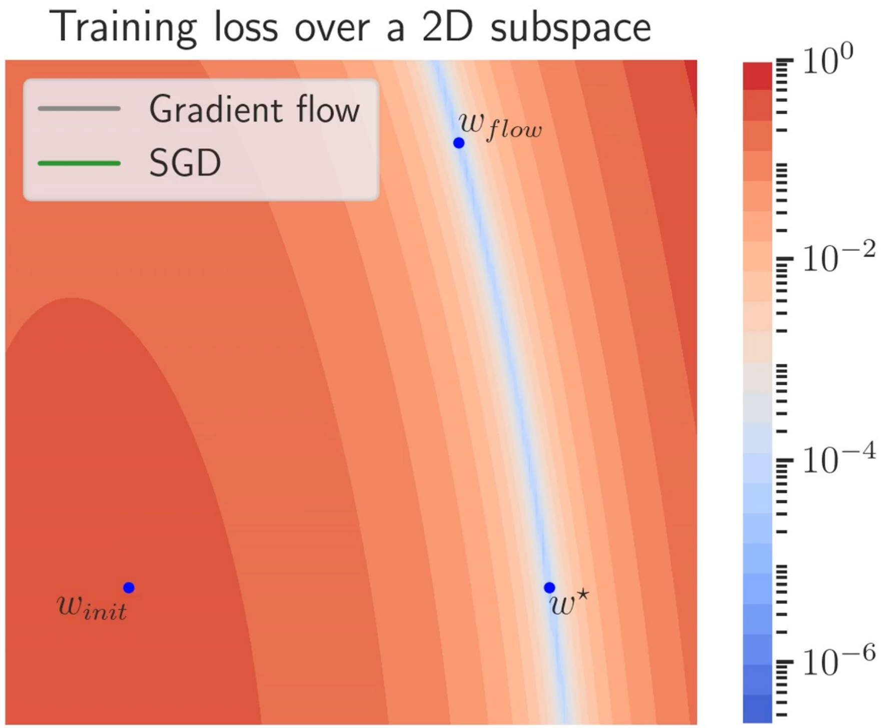


Setting: no momentum, no data augmentation.

1. Why does the training loss stabilizes?
2. What kind of hidden dynamics is happening in this phase?
3. Is it related to sparsity of the predictor?

New paper: SGD with large step sizes learns sparse features





- **Our picture**: SGD noise drives the iterates to a sparse solution which we observe on many models (from **diagonal linear networks** to **ResNets on CIFAR-100**)
- It's important that **we don't converge too early and keep benefitting from the noise**
- **Relation to sharpness**: the slow noisy dynamics can be seen as minimization of *some* sharpness-related criterion (but unclear which exactly; rank of the NTK feature matrix seems to be a good proxy)

Thanks for your attention!

Happy to answer your questions and chat more :)

Paper: https://arxiv.org/abs/2206.06232
Code: https://github.com/tml-epfl/understanding-sam