WHEN VISION TRANSFORMERS OUTPERFORM RESNETS WITHOUT PRE-TRAINING OR STRONG DATA AUGMENTATIONS

Xiangning Chen[1,2]* **Cho-Jui Hsieh**[2] **Boqing Gong**[1]

[1]Google Research [2]Department of Computer Science, UCLA

ABSTRACT

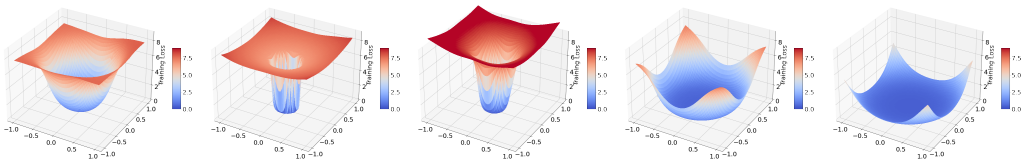
Vision Transformers (ViTs) and MLPs signal further efforts on replacing hand-wired features or inductive biases with general-purpose neural architectures. Existing works empower the models by massive data, such as large-scale pre-training and/or repeated strong data augmentations, and still report optimization-related problems (e.g., sensitivity to initialization and learning rates). Hence, this paper investigates ViTs and MLP-Mixers from the lens of loss geometry, intending to improve the models' data efficiency at training and generalization at inference. Visualization and Hessian reveal extremely sharp local minima of converged models. By promoting smoothness with a recently proposed sharpness-aware optimizer, we substantially improve the accuracy and robustness of ViTs and MLP-Mixers on various tasks spanning supervised, adversarial, contrastive, and transfer learning (e.g., +5.3% and +11.0% top-1 accuracy on ImageNet for ViT-B/16 and Mixer-B/16, respectively, with the simple Inception-style preprocessing). We show that the improved smoothness attributes to sparser active neurons in the first few layers. The resultant ViTs outperform ResNets of similar size and throughput when trained from scratch on ImageNet without large-scale pre-training or strong data augmentations. They also possess more perceptive attention maps. Our model checkpoints are released at `https://github.com/google-research/vision_transformer`.

1 INTRODUCTION

Transformers (Vaswani et al., 2017) have become the de-facto model of choice in natural language processing (NLP) (Devlin et al., 2018; Radford et al., 2018). In computer vision, there has recently been a surge of interest in end-to-end Transformers (Dosovitskiy et al., 2021; Touvron et al., 2021b; Liu et al., 2021b; Fan et al., 2021; Arnab et al., 2021; Bertasius et al., 2021; Akbari et al., 2021) and MLPs (Tolstikhin et al., 2021; Touvron et al., 2021a; Liu et al., 2021a; Melas-Kyriazi, 2021), prompting the efforts to replace hand-wired features or inductive biases with general-purpose neural architectures powered by data-driven training. We envision these efforts may lead to a unified knowledge base that produces versatile representations for different data modalities, simplifying the inference and deployment of deep learning models in various application scenarios.

Despite the appealing potential of moving toward general-purpose neural architectures, the lack of convolution-like inductive bias also challenges the training of vision Transformers (ViTs) and MLPs. When trained on ImageNet (Deng et al., 2009) with the conventional Inception-style data preprocessing (Szegedy et al., 2016), Transformers *"yield modest accuracies of a few percentage points below ResNets of comparable size"* (Dosovitskiy et al., 2021). To boost the performance, existing works resort to large-scale pre-training (Dosovitskiy et al., 2021; Arnab et al., 2021; Akbari et al., 2021) and repeated strong data augmentations (Touvron et al., 2021b), resulting in excessive demands of data, computing, and sophisticated tuning of many hyperparameters. For instance, Dosovitskiy et al. (Dosovitskiy et al., 2021) pre-train ViTs using 304M labeled images, and Touvron et al. (2021b) repeatedly stack four strong image augmentations.

*Work done as a student researcher at Google.



(a) ResNet (b) ViT (c) Mixer (d) ViT-SAM (e) Mixer-SAM

Figure 1: Cross-entropy loss landscapes of ResNet-152, ViT-B/16, and Mixer-B/16. ViT and MLP-Mixer converge to sharper regions than ResNet when trained on ImageNet with the basic Inception-style preprocessing. SAM, a sharpness-aware optimizer, significantly smooths the landscapes.

In this paper, we show ViTs can outperform ResNets (He et al., 2016) of even bigger sizes in both accuracy and various forms of robustness by using a principled optimizer, without the need for large-scale pre-training or strong data augmentations. MLP-Mixers (Tolstikhin et al., 2021) also become on par with ResNets.

We first study the architectures fully trained on ImageNet from the lens of loss landscapes and draw the following findings. First, visualization and Hessian matrices of the loss landscapes reveal that Transformers and MLP-Mixers converge at extremely sharp local minima, whose largest principal curvatures are almost an order of magnitude bigger than ResNets. Such effect accumulates when the gradients backpropagate from the last layer to the first, and the initial embedding layer suffers the largest eigenvalue of the corresponding sub-diagonal Hessian. Second, the networks all have very small training errors, and MLP-Mixers are more prone to overfitting than ViTs of more parameters (probably because of the difference in self-attention). Third, ViTs and MLP-Mixers have worse "trainabilities" than ResNets following the neural tangent kernel analyses (Xiao et al., 2020).

We conjecture that the convolution-induced translation equivariance and locality help ResNets escape from bad local minima when trained on visual data. However, we need improved learning algorithms to prevent them from happening to the convolution-free ViTs and and MLP-Mixers. The first-order optimizers (e.g., SGD and Adam (Kingma & Ba, 2015)) only seek the model parameters that minimize the training error. They dismiss the higher-order information such as flatness that correlates with the generalization (Keskar et al., 2017; Kleinberg et al., 2018; Jastrzębski et al., 2019; Smith & Le, 2018; Chaudhari et al., 2017).

The above study and reasoning lead us to the recently proposed sharpness-aware minimizer (SAM) (Foret et al., 2021) that explicitly smooths the loss geometry during model training. SAM strives to find a solution whose entire neighborhood has low losses rather than focus on any singleton point. We show that the resultant models exhibit smoother loss landscapes, and their generalization capabilities improve tremendously across different tasks including supervised, adversarial, contrastive, and transfer learning (e.g., +5.3% and +11.0% top-1 accuracy on ImageNet for ViT-B/16 and Mixer-B/16, respectively, with the simple Inception-style preprocessing). The enhanced ViTs achieve better accuracy and robustness than ResNets of similar and bigger sizes when trained from scratch on ImageNet, without large-scale pre-training or strong data augmentations.

By analyzing some intrinsic model properties, we find that the models after SAM reduce the Hessian eigenvalues by activating sparser neurons (on ImageNet), especially in the first few layers. The weight norms increase, implying the commonly used weight decay may not be an effective regularization alone. A side observation is that, unlike ResNets and MLP-Mixers, ViTs have extremely sparse active neurons, revealing the redundancy of input image patches and the capacity for network pruning. Another interesting finding is that ViTs' performance gain also translates to plausible attention maps containing more perspicuous information about semantic segmentation. Finally, we draw similarities between SAM and strong augmentations (e.g., mixup) in that they both smooth the average loss geometry and encourage the models to behave linearly between training images.

2 Background and Related Work

We briefly review ViTs, MLP-Mixers, and some related works in this section.

Dosovitskiy et al. (2021) show that a pure Transformer architecture (Vaswani et al., 2017) can achieve state-of-the-art accuracy on image classification by pre-training it on large datasets such

Table 1: Number of parameters, NTK condition number κ, Hessian dominate eigenvalue λ_{max}, accuracy on ImageNet, and accuracy/robustness on ImageNet-C. ViT and MLP-Mixer suffer divergent κ and converge to sharp regions of big λ_{max}; SAM rescues that and leads to better generalization.

	ResNet-152	ResNet-152-SAM	ViT-B/16	ViT-B/16-SAM	Mixer-B/16	Mixer-B/16-SAM
#Params	60M		87M		59M	
NTK κ †	2801.6		4205.3		14468.0	
Hessian λ_{max}	179.8	**42.0**	738.8	**20.9**	1644.4	**22.5**
ImageNet (%)	78.5	**79.3**	74.6	**79.9**	66.4	**77.4**
ImageNet-C (%)	50.0	**52.2**	46.6	**56.5**	33.8	**48.8**

† As it is prohibitive to compute the exact NTK, we approximate the value by averaging over its sub-diagonal blocks. Please see Appendix E for details.

as ImageNet-21k (Deng et al., 2009) and JFT-300M (Sun et al., 2017). Their vision Transformer (ViT) is a stack of residual blocks, each containing a multi-head self-attention, layer normalization (Ba et al., 2016), and a MLP layer. ViT first embeds an input image $x \in \mathbb{R}^{H\times W\times C}$ into a sequence of features $z \in \mathbb{R}^{N\times D}$ by applying a linear projection over N nonoverlapping image patches $x_p \in \mathbb{R}^{N\times(P^2\cdot C)}$, where D is the feature dimension, P is the patch resolution, and $N = HW/P^2$ is the sequence length. The self-attention layers in ViT are global and do not possess the locality and translation equivariance of convolutions. ViT is compatible with the popular architectures in NLP (Devlin et al., 2018; Radford et al., 2018) and, similar to its NLP counterparts, requires pre-training over massive datasets (Dosovitskiy et al., 2021; Akbari et al., 2021; Arnab et al., 2021) or strong data augmentations (Touvron et al., 2021b). Some works specialize the ViT architectures for the visual data (Liu et al., 2021b; Yuan et al., 2021; Fan et al., 2021; Bertasius et al., 2021).

More recent works find that the self-attention in ViT is not vital for performance, resulting in several architectures exclusively based on MLPs (Tolstikhin et al., 2021; Touvron et al., 2021a; Liu et al., 2021a; Melas-Kyriazi, 2021). Here we take MLP-Mixer (Tolstikhin et al., 2021) as an example. MLP-Mixer shares the same input layer as ViT; namely, it partitions an image into a sequence of nonoverlapping patches/tokens. It then alternates between token and channel MLPs, where the former allows feature fusion from different spatial locations.

We focus on ViTs and MLP-Mixers in this paper. We denote by "S" and "B" the small and base model sizes, respectively, and by an integer the image patch resolution. For instance, ViT-B/16 is the base ViT model taking as input a sequence of 16×16 patches. Appendices contain more details.

3 ViTs and MLP-Mixers Converge to Sharp Local Minima

The current training recipe of ViTs, MLP-Mixers, and related convolution-free architectures relies heavily on massive pre-training (Dosovitskiy et al., 2021; Arnab et al., 2021; Akbari et al., 2021) or a bag of strong data augmentations (Touvron et al., 2021b; Tolstikhin et al., 2021; Cubuk et al., 2019; 2020; Zhang et al., 2018; Yun et al., 2019). It highly demands data and computing, and leads to many hyperparameters to tune. Existing works report that ViTs yield inferior accuracy to the ConvNets of similar size and throughput when trained from scratch on ImageNet without the combination of those advanced data augmentations, despite using various regularization techniques (e.g., large weight decay, Dropout (Srivastava et al., 2014), etc.). For instance, ViT-B/16 (Dosovitskiy et al., 2021) gives rise to 74.6% top-1 accuracy on the ImageNet validation set (224 image resolution), compared with 78.5% of ResNet-152 (He et al., 2016). Mixer-B/16 (Tolstikhin et al., 2021) performs even worse (66.4%). There also exists a large gap between ViTs and ResNets in robustness tests (e.g., against 19 corruptions in ImageNet-C (Hendrycks & Dietterich, 2019)).

Moreover, Chen et al. (2021c) find that the gradients can spike and cause a sudden accuracy dip when training ViTs, and Touvron et al. (2021b) report the training is sensitive to initialization and hyperparameters. These all point to optimization problems. In this paper, we investigate the loss landscapes of ViTs and MLP-Mixers to understand them from the optimization perspective, intending to reduce their dependency on the large-scale pre-training or strong data augmentations.

ViTs and MLP-Mixers converge to extremely sharp local minima. It has been extensively studied that the convergence to a flat region whose curvature is small benefits the generalization of neural networks (Keskar et al., 2017; Kleinberg et al., 2018; Jastrzębski et al., 2019; Chen & Hsieh, 2020; Smith & Le, 2018; Zela et al., 2020; Chaudhari et al., 2017). Following Li et al. (2018), we plot the loss landscapes at convergence when ResNets, ViTs, and MLP-Mixers are trained from scratch

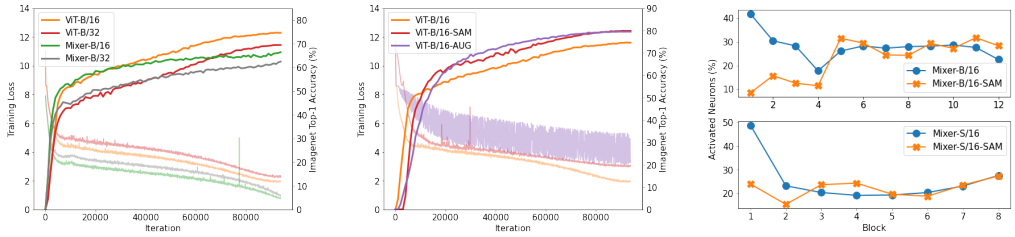


Figure 2: **Left** and **Middle**: ImageNet training error and validation accuracy vs. iteration for ViTs and MLP-Mixers. **Right**: Percentage of activated neurons at each block of MLP-Mixers.

on ImageNet with the basic Inception-style preprocessing (Szegedy et al., 2016) (see Appendices for details). As shown in Figures 1(a) to 1(c), ViTs and MLP-Mixers converge to much sharper regions than ResNets. In Table 1, we further validate the results by computing the dominate Hessian eigenvalue λ_{max}, which is a mathematical evaluation of the worst-case landscape curvature. The λ_{max} values of ViT and MLP-Mixer are orders of magnitude larger than that of ResNet, and MLP-Mixer suffers the largest curvature among the three species (see Section 4.4 for a detailed analysis).

Small training errors. This convergence to sharp regions coincides with the training dynamics shown in Figure 2 (left). Although Mixer-B/16 has fewer parameters than ViT-B/16 (59M vs. 87M), it has a smaller training error but much worse test accuracy, implying that using the cross-token MLP to learn the interplay across image patches is more prone to overfitting than ViTs' self-attention mechanism whose behavior is restricted by a softmax. Such a difference probably explains that it is easier for MLP-Mixers to get stuck in sharp local minima.

ViTs and MLP-Mixers have worse trainability. Furthermore, we discover that ViTs and MLP-Mixers suffer poor trainability, defined as the effectiveness of a network to be optimized by gradient descent (Xiao et al., 2020; Burkholz & Dubatovka, 2019; Shin & Karniadakis, 2020). Xiao et al. (2020) show that the trainability of a neural network can be characterized by the condition number of the associated neural tangent kernel (NTK), $\Theta(x, x') = J(x)J(x')^T$, where J is the Jacobian matrix. Denoting by $\lambda_1 \geq \cdots \geq \lambda_m$ the eigenvalues of NTK Θ_{train}, the smallest eigenvalue λ_m converges exponentially at a rate given by the condition number $\kappa = \lambda_1/\lambda_m$. If κ diverges then the network will become untrainable (Xiao et al., 2020; Chen et al., 2021a). As shown in Table 1, κ is pretty stable for ResNets, echoing previous results that ResNets enjoy superior trainability regardless of the depth (Yang & Schoenholz, 2017; Li et al., 2018). However, we observe that the condition number diverges when it comes to ViT and MLP-Mixer, confirming that the training of ViTs desires extra care (Chen et al., 2021c; Touvron et al., 2021b).

4 A Principled Optimizer for Convolution-Free Architectures

The commonly used first-order optimizers (e.g., SGD (Nesterov, 1983), Adam (Kingma & Ba, 2015)) only seek to minimize the training loss $L_{train}(w)$. They usually dismiss the higher-order information such as curvature that correlates with the generalization (Keskar et al., 2017; Chaudhari et al., 2017; Dziugaite & Roy, 2017). However, the objective L_{train} for deep neural networks are highly non-convex, making it easy to reach near-zero training error but high generalization error L_{test} during evaluation, let alone their robustness when the test sets have different distributions (Hendrycks & Dietterich, 2019; Hendrycks et al., 2020). ViTs and MLPs amplify such drawbacks of first-order optimizers due to the lack of inductive bias for visual data, resulting in excessively sharp loss landscapes and poor generalization, as shown in the previous section. We hypothesize that smoothing the loss landscapes at convergence can significantly improve the generalization ability of those convolution-free architectures, leading us to the recently proposed sharpness-aware minimizer (SAM) (Foret et al., 2021) that explicitly avoids sharp minima.

4.1 SAM: Overview

Intuitively, SAM (Foret et al., 2021) seeks to find the parameter w whose entire neighbours have low training loss L_{train} by formulating a minimax objective:

$$\min_{w} \max_{\|\epsilon\|_2 \leq \rho} L_{train}(w + \epsilon), \tag{1}$$

where ρ is the size of the neighbourhood ball. Without loss of generality, here we use l_2 norm for its strong empirical results (Foret et al., 2021) and omit the regularization term for simplicity. Since the exact solution of the inner maximization $\epsilon^\star = \arg\max_{\|\epsilon\|_2 \leq \rho} L_{train}(w + \epsilon)$ is hard to obtain, they employ an efficient first-order approximation:

$$\hat{\epsilon}(w) = \underset{\|\epsilon\|_2 \leq \rho}{\arg\max}\, L_{train}(w) + \epsilon^T \nabla_w L_{train}(w) = \rho \nabla_w L_{train}(w) / \|\nabla_w L_{train}(w)\|_2. \quad (2)$$

Under the l_2 norm, $\hat{\epsilon}(w)$ is simply a scaled gradient of the current weight w. After computing $\hat{\epsilon}$, SAM updates w based on the sharpness-aware gradient $\nabla_w L_{train}(w)|_{w+\hat{\epsilon}(w)}$.

4.2 Sharpness-aware optimization improves ViTs and MLP-Mixers

We train ViTs and MLP-Mixers with no large-scale pre-training or strong data augmentations. We directly apply SAM to the original ImageNet training pipeline of ViTs (Dosovitskiy et al., 2021) without changing any hyperparameters. The pipeline employs the basic Inception-style preprocessing (Szegedy et al., 2016). The original training setup of MLP-Mixers (Tolstikhin et al., 2021) includes a combination of strong data augmentations, and we replace it with the same Inception-style preprocessing for a fair comparison. Note that we perform grid search for the learning rate, weight decay, Dropout *before* applying SAM. Please see Appendices for details.

Smoother regions around the local minima. Thanks to SAM, both ViTs and MLP-Mixers converge at much smoother regions, as shown in Figures 1(d) and 1(e). The worst-case curvature, i.e., the largest eigenvalue λ_{max} of the Hessian matrix, also decreases to a small value (see Table 1).

Higher accuracy. What comes along is tremendously improved generalization performance. On the ImageNet validation set, SAM boosts the top-1 accuracy of ViT-B/16 from 74.6% to 79.9%, and Mixer-B/16 from 66.4% to 77.4%. For comparison, the improvement on a similarly sized ResNet-152 is 0.8%. Empirically, *the degree of improvement negatively correlates with the level of inductive biases built into the architecture.* ResNets with inherent translation equivalence and locality benefit less from landscape smoothing than the attention-based ViTs. MLP-Mixers gain the most from the smoothed loss geometry. Moreover, SAM brings larger improvements to the models of larger capacity (e.g., +4.1% for Mixer-S/16 vs. +11.0% for Mixer-B/16) and longer patch sequence (e.g., +2.1% for ViT-S/32 vs. +5.3% for ViT-S/8). Please see Table 2 for more results.

Better robustness. We also evaluate the models' robustness using ImageNet-R (Hendrycks et al., 2020) and ImageNet-C (Hendrycks & Dietterich, 2019) and find even bigger impacts of the smoothed loss landscapes. On ImageNet-C, which corrupts images by noise, bad weather, blur, etc., we report the average accuracy against 19 corruptions across five severity. As shown in Tables 1 and 2, the accuracies of ViT-B/16 and Mixer-B/16 increase by 9.9% and 15.0%, respectively, after SAM smooths their converged local regions.

4.3 ViTs outperform ResNets without pre-training or strong augmentations

The performance of a model architecture is often conflated with the training strategies (Bello et al., 2021), where data augmentations play a key role (Cubuk et al., 2019; 2020; Zhang et al., 2018; Xie et al., 2020; Chen et al., 2021b). However, the design of data augmentations requires substantial domain expertise and may not translate between images and videos, for instance. Thanks to the principled sharpness-aware optimizer, we can remove the advanced augmentations and focus on the architecture itself (with the basic Inception-style preprocessing).

When trained from scratch on ImageNet with SAM, *ViTs outperform ResNets of similar and greater sizes (also comparable throughput at inference)* regarding both clean accuracy (on ImageNet (Deng et al., 2009), ImageNet-ReaL (Beyer et al., 2020), and ImageNet V2 (Recht et al., 2019)) and robustness (on ImageNet-R (Hendrycks et al., 2020) and ImageNet-C (Hendrycks & Dietterich, 2019)). ViT-B/16 achieves 79.9%, 26.4%, and 56.6% top-1 accuracy on ImageNet, ImageNet-R, and ImageNet-C, while the counterpart numbers for ResNet-152 are 79.3%, 25.7%, and 52.2%, respectively (see Table 2). The gaps between ViTs and ResNets are even wider for small architectures. ViT-S/16 outperforms a similarly sized ResNet-50 by 1.4% on ImageNet, and 6.5% on ImageNet-C. SAM also significantly improves MLP-Mixers' results.

Table 2: Accuracy and robustness of ResNets, ViTs, and MLP-Mixers trained from scratch on ImageNet with SAM (improvement over the vanilla model is shown in the parentheses). We use the Inception-style preprocessing (with resolution 224) rather than a combination of strong data augmentations. ViTs achieve better accuracy and robustness than ResNets of similar size and throughput (calculated following Tolstikhin et al. (2021)), and MLP-Mixers become on par with ResNets.

Model	#params	Throughput (img/sec/core)	ImageNet	Real	V2	ImageNet-R	ImageNet-C
ResNet							
ResNet-50-SAM	25M	2161	76.7 (+0.7)	83.1 (+0.7)	64.6 (+1.0)	23.3 (+1.1)	46.5 (+1.9)
ResNet-101-SAM	44M	1334	78.6 (+0.8)	84.8 (+0.9)	66.7 (+1.4)	25.9 (+1.5)	51.3 (+2.8)
ResNet-152-SAM	60M	935	79.3 (+0.8)	84.9 (+0.7)	67.3 (+1.0)	25.7 (+0.4)	52.2 (+2.2)
ResNet-50x2-SAM	98M	891	79.6 (+1.5)	85.3 (+1.6)	67.5 (+1.7)	26.0 (+2.9)	50.7 (+3.9)
ResNet-101x2-SAM	173M	519	80.9 (+2.4)	86.4 (+2.4)	69.1 (+2.8)	27.8 (+3.2)	54.0 (+4.7)
ResNet-152x2-SAM	236M	356	81.1 (+1.8)	86.4 (+1.9)	69.6 (+2.3)	28.1 (+2.8)	55.0 (+4.2)
Vision Transformer							
ViT-S/32-SAM	23M	6888	70.5 (+2.1)	77.5 (+2.3)	56.9 (+2.6)	21.4 (+2.4)	46.2 (+2.9)
ViT-S/16-SAM	22M	2043	78.1 (+3.7)	84.1 (+3.7)	65.6 (+3.9)	24.7 (+4.7)	53.0 (+6.5)
ViT-S/14-SAM	22M	1234	78.8 (+4.0)	84.8 (+4.5)	67.2 (+5.2)	24.4 (+4.7)	54.2 (+7.0)
ViT-S/8-SAM	22M	333	81.3 (+5.3)	86.7 (+5.5)	70.4 (+6.2)	25.3 (+6.1)	55.6 (+8.5)
ViT-B/32-SAM	88M	2805	73.6 (+4.1)	80.3 (+5.1)	60.0 (+4.7)	24.0 (+4.1)	50.7 (+6.7)
ViT-B/16-SAM	87M	863	79.9 (+5.3)	85.2 (+5.4)	67.5 (+6.2)	26.4 (+6.3)	56.5 (+9.9)
MLP-Mixer							
Mixer-S/32-SAM	19M	11401	66.7 (+2.8)	73.8 (+3.5)	52.4 (+2.9)	18.6 (+2.7)	39.3 (+4.1)
Mixer-S/16-SAM	18M	4005	72.9 (+4.1)	79.8 (+4.7)	58.9 (+4.1)	20.1 (+4.2)	42.0 (+6.4)
Mixer-S/8-SAM	20M	1498	75.9 (+5.7)	82.5 (+6.3)	62.3 (+6.2)	20.5 (+5.1)	42.4 (+7.8)
Mixer-B/32-SAM	60M	4209	72.4 (+9.9)	79.0 (+10.9)	58.0 (+10.4)	22.8 (+8.2)	46.2 (12.4)
Mixer-B/16-SAM	59M	1390	77.4 (+11.0)	83.5 (+11.4)	63.9 (+13.1)	24.7 (+10.2)	48.8 (+15.0)
Mixer-B/8-SAM	64M	466	79.0 (+10.4)	84.4 (+10.1)	65.5 (+11.6)	23.5 (+9.2)	48.9 (+16.9)

Table 3: Dominant eigenvalue λ_{max} of the sub-diagonal Hessians for different network components, and norm of the model parameter w and the post-activation a_k of block k. Each ViT block consists of a MSA and a MLP, and MLP-Mixer alternates between a token MLP a channel MLP. Shallower layers have larger λ_{max}. SAM smooths every component.

Model	λ_{max} of diagonal blocks of Hessian							$\|w\|_2$	$\|a_1\|_2$	$\|a_6\|_2$	$\|a_{12}\|_2$
	Embedding	MSA/ Token MLP	MLP/ Channel MLP	Block1	Block6	Block12	Whole				
ViT-B/16	300.4	179.8	281.4	44.4	32.4	26.9	738.8	269.3	104.9	104.3	138.1
ViT-B/16-SAM	3.8	8.5	9.6	1.7	1.7	1.5	20.9	353.8	117.0	120.3	97.2
Mixer-B/16	1042.3	95.8	417.9	239.3	41.2	5.1	1644.4	197.6	96.7	135.1	74.9
Mixer-B/16-SAM	18.2	1.4	9.5	4.0	1.1	0.3	22.5	389.9	110.9	176.0	216.1

4.4 Intrinsic changes after SAM

We take a deeper look into the models to understand how they intrinsically change to reduce the Hessian' eigenvalue λ_{max} and what the changes imply in addition to the enhanced generalization.

Smoother loss landscapes for every network component. In Table 3, we break down the Hessian of the whole architecture into small diagonal blocks of Hessians concerning each set of parameters, attempting to analyze what specific components cause the blowing up of λ_{max} in the models trained without SAM. We observe that shallower layers have larger Hessian eigenvalues λ_{max}, and the first linear embedding layer incurs the sharpest geometry. This agrees with the finding in (Chen et al., 2021c) that spiking gradients happen early in the embedding layer. Additionally, the multi-head self-attention (MSA) in ViTs and the Token MLPs in MLP-Mixers, both of which mix information across spatial locations, have comparably lower λ_{max} than the other network components. SAM consistently reduces the λ_{max} of all network blocks.

We can gain insights into the above findings by the recursive formulation of Hessian matrices for MLPs (Botev et al., 2017). Let h_k and a_k be the pre-activation and post-activation values for layer k, respectively. They satisfy $h_k = W_k a_{k-1}$ and $a_k = f_k(h_k)$, where W_k is the weight matrix and f_k is the activation function (GELU (Hendrycks & Gimpel, 2020) in MLP-Mixers). Here we omit the bias term for simplicity. The diagonal block of Hessian matrix H_k with respect to W_k can be

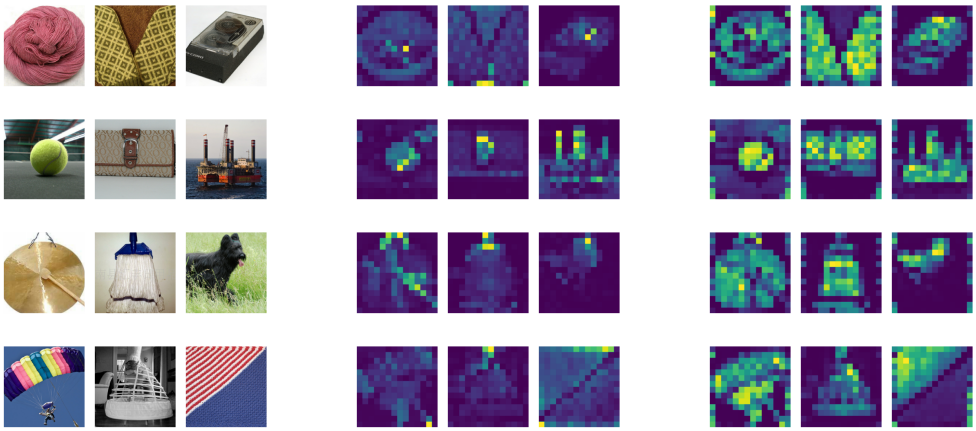


Figure 3: Raw images (**Left**) and attention maps of ViT-S/16 with (**Right**) and without (**Middle**) sharpness-aware optimization. ViT-S/16 with less sharp local optimum contains perceptive segmentation information in its attention maps.

recursively calculated as:

$$H_k = (a_{k-1}a_{k-1}^T) \otimes \mathcal{H}_k, \quad \mathcal{H}_k = B_k W_{k+1}^T \mathcal{H}_{k+1} W_{k+1} B_k + D_k, \tag{3}$$

$$B_k = \text{diag}(f_k'(h_k)), \qquad D_k = \text{diag}(f_k''(h_k)\frac{\partial L}{\partial a_k}), \tag{4}$$

where $\otimes$ is the Kronecker product, $\mathcal{H}_k$ is the pre-activation Hessian for layer k, and L is the objective function. Therefore, the Hessian norm accumulates as the recursive formulation backpropagates to shallow layers, explaining why the first block has much larger λ_{max} than the last block in Table 3.

Greater weight norms. After applying SAM, we find that the norm of the post-activation value a_{k-1} and the weight W_{k+1} become even bigger (see Table 3), indicating that the commonly used weight decay may not effectively regularize ViTs and MLP-Mixers.

Sparser active neurons in MLP-Mixers. Given the recursive formulation Equation (3), we identify another intrinsic measure of MLP-Mixers that contribute to the Hessian: the number of activated neurons. Indeed, B_k is determined by the activated neurons whose values are greater than zero, since the first-order derivative of GELU becomes much smaller when the input is negative. As a result, the number of active GELU neurons is directly connected to the Hessian norm. Figure 2 (right) shows the proportion of activated neurons for each block, counted using 10% of the ImageNet training set. We can see that SAM greatly reduces the proportion of activated neurons for the first few layers, pushing them to much sparser states. This result also suggests the potential redundancy of image patches.

ViTs' active neurons are highly sparse. Although Equations (3) and (4) only involve MLPs, we still observe a decrease of activated neurons in the first layer of ViTs (but not as significant as in MLP-Mixers). More interestingly, we find that the proportion of activated neurons in ViT is much smaller than another two architectures — less than 10% neurons have values greater than zero for most layers (In comparison, the fraction of activated neurons for ResNet is $> 50\%$). In other words, ViTs offer a huge potential for network pruning. This sparsity may also explain why one Transformer can handle multi-modality signals (vision, text, and audio) (Akbari et al., 2021).

More perceptive attention maps in ViTs. We visualize ViT-S/16's attention map of the classification token averaged over the last multi-head attentions in Figure 3 following Caron et al. (2021). Interestingly, the ViT model optimized with SAM can encode plausible segmentation information, giving rise to better interpretability than the one trained via the conventional SGD optimization.

4.5 SAM VS. STRONG AUGMENTATIONS

Previous sections show that SAM can improve the generalization (and robustness) of ViTs and MLP-Mixers. Meanwhile, another paradigm to train these models on ImageNet from scratch is to stack multiple strong augmentations (Touvron et al., 2021b;a; Tolstikhin et al., 2021). Hence, it is

Table 5: Data augmentations, SAM, and their combination applied to different model architectures trained on ImageNet and its subsets from scratch.

Dataset	#Images	ResNet-152				ViT-B/16				Mixer-B/16			
		Vanilla	SAM	AUG	SAM + AUG	Vanilla	SAM	AUG	SAM + AUG	Vanilla	SAM	AUG	SAM + AUG
ImageNet	1,281,167	78.5	79.3	78.8	78.9	74.6	79.9	79.6	81.5	66.4	77.4	76.5	78.1
i1k (1/2)	640,583	74.2	75.6	75.1	75.5	64.9	75.4	73.1	75.8	53.9	71.0	70.4	73.1
i1k (1/4)	320,291	68.0	70.3	70.2	70.6	52.4	66.8	63.2	65.6	37.2	62.8	61.0	65.8
i1k (1/10)	128,116	54.6	57.1	59.2	59.5	32.8	46.1	38.5	45.7	21.0	43.5	43.0	51.0

interesting to study the differences and similarities between the models trained by SAM and by using strong data augmentations. For the augmentation experiments, we follow Tolstikhin et al. (2021)'s pipeline that includes mixup (Zhang et al., 2018) and RandAugment (Cubuk et al., 2020).

Generalization. Table 5 shows the results of strong data augmentation, SAM, and their combination on ImageNet. Each row corresponds to a training set of a different fraction of ImageNet-1k. SAM benefits ViT-B/16 and Mixer-B/16 more than the strong data augmentations, especially when the training set is small. For instance, when the training set contains only 1/10 of ImageNet training images, ViT-B/16-SAM outperforms ViT-B/16-AUG by 7.6%. Apart from the improved validation accuracy, we also observe that both SAM and strong augmentations increase the training error (see Figure 2 (Middle) and Table 4), indicating their regularization effects. However, they have distinct training dynamics as the loss curve for ViT-B/16-AUG is much nosier than ViT-B/16-SAM.

Sharpness at convergence. Another intriguing question is as follows. Can augmentations also smooth the loss geometry similarly to SAM? To answer it, we also plot the landscape of ViT-B/16-AUG (see Figure 5 in the Appendix) and compute its Hessian λ_{max} in Table 4. Besides, we calculate the training error under Gaussian perturbations $L_{train}^{\mathcal{N}} = \mathbb{E}_{\epsilon\sim\mathcal{N}}[L_{train}(w+\epsilon)]$, which reveals the average flatness. Surprisingly, strong augmentations even enlarge the λ_{max}. However, like SAM, augmentations make ViT-B/16-AUG smoother and achieve a significantly smaller training error under random Gaussian perturbations than ViT-B/16. **These results show that** both SAM and augmentations make the loss landscape flat *on average*. **The difference is that** SAM enforces the smoothness by reducing the largest curvature via a minimax formulation to optimize the *worst-case* scenario, while augmentations ignore the worse-case curvature and instead smooth the landscape over the directions concerning the inductive biases induced by the augmentations.

Table 4: Comparison between ViT-B/16-SAM and ViT-B/16-AUG. We average the results for 1,000 random noises when calculating $L_{train}^{\mathcal{N}}$. R denotes the missing rate under linear interpolation.

Model	λ_{max}	L_{train}	$L_{train}^{\mathcal{N}}$	$R(\downarrow)$
ViT-B/16	738.8	0.65	6.66	57.9%
ViT-B/16-SAM	20.9	0.82	0.96	39.6%
ViT-B/16-AUG	1659.3	0.85	1.23	21.4%

Interestingly, besides the similarity in smoothing the loss curvature on average, we also discover that SAM-trained models possess "linearality" resembling the property manually injected by the mixup augmentation. Following Zhang et al. (2018), we compute the prediction error in-between training data in Table 4, where a prediction y is counted as a miss if it does not belong to $\{y_i, y_j\}$ evaluated at $x = 0.5x_i + 0.5x_j$. We observe that SAM greatly reduces the missing rate (R) compared with the vanilla baseline, showing a similar effect to mixup that explicitly encourages such linearity.

5 Ablation Studies

Section 4.3 shows that ViTs outperform ResNets when trained from scratch on ImageNet with the basic Inception-style preprocessing for both clean accuracy and robustness. In this section, we provide a more comprehensive study about SAM's effect on various vision models and under different training setups (e.g., varying the amount of training data, cross-entropy loss vs. contrastive loss).

5.1 When scaling the training set size

Previous studies scale up training data to show massive pre-training trumps inductive biases (Dosovitskiy et al., 2021; Tolstikhin et al., 2021). Here we show SAM further enables ViTs and MLP-Mixers to handle small-scale training data well. We randomly sample 1/4 and 1/2 images from each ImageNet class to compose two smaller-scale training sets, i.e., i1k (1/4) and i1k (1/2) in Figure 4 with 320,291 and 640,583 images, respectively. We also include ImageNet-21k to pre-train the mod-

els with SAM, followed by fine-tuning on ImageNet-1k without SAM. The ImageNet validation set remains intact.

As expected, fewer training examples amplify the drawback of ViTs and MLP-Mixers' lack of the convolutional inductive bias — their accuracies decline much faster than ResNets' (see the top panel in Figure 4 and the corresponding numbers in Table 5). When trained with 1/4 of the ImageNe training images, ViT-B/16 has top-1 accuracy 52.4%, Mixer-B/16 gives 37.2%, but ResNet-152 maintains as high as 68.0%.

However, SAM can drastically rescue ViTs and MLP-Mixers' performance decrease on smaller training sets. Figure 4 (bottom) shows that *the improvement brought by SAM over vanilla SGD training is proportional to the number of training images.* When trained on i1k (1/4), it boosts ViT-B/16 and Mixer-B/16 by 14.4% and 25.6%, escalating their results to 66.8% and 62.8%, respectively. It also tells that ViT-B/16-SAM matches the performance of ResNet-152-SAM even with only 1/2 ImageNet training data.

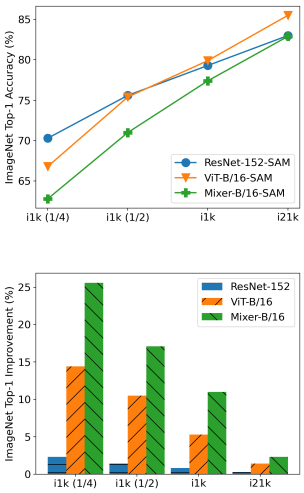


Figure 4: ImageNet accuracy (**Top**) and improvement (**Bottom**) brought by SAM on various training sets.

5.2 When SAM meets contrastive learning

In addition to data augmentations and large-scale pre-training, another notable way of improving a neural model's generalization is (supervised) contrastive learning (Chen et al., 2020; He et al., 2020; Caron et al., 2021; Khosla et al., 2020). We couple SAM with the supervised contrastive learning (Khosla et al., 2020) for 350 epochs, followed by fine-tuning the classification head by 90 epochs for both ViT-S/16 and ViT-B/16. Please see Appendix F.4 for more implementation details. Compared to the training procedure without SAM, we find considerable performance gain thanks to SAM's smoothing of the contrastive loss geometry, improving the ImageNet top-1 accuracy of ViT-S/16 from 77.0% to 78.1%, and ViT-B/16 from 77.4% to 80.0%.

5.3 When SAM meets adversarial training

Interestingly, SAM and adversarial training are both minimax problems except that SAM's inner maximization is with respect to the network weights, while the latter concerns about the input for defending contrived attack (Madry et al., 2018; Wong et al., 2020). Moreover, similar to SAM, Shafahi et al. (2019) suggest that adversarial training can flatten and smooth the loss landscape. In light of these connections, we study ViTs and MLP-Mixers under the adversarial training framework (Wu et al., 2020; Madry et al., 2018). To incorporate SAM, we formulate a three-level objective:

$$\min_{w} \max_{\epsilon \in \mathbb{S}_{sam}} \max_{\delta \in \mathbb{S}_{adv}} L_{train}(w + \epsilon, x + \delta, y), \tag{5}$$

where $\mathbb{S}_{sam}$ and $\mathbb{S}_{adv}$ denote the allowed perturbation norm balls for the model parameter w and input image x, respectively. Note that we can simultaneously obtain the gradients for computing ϵ and δ by backpropagation only once. To lower the training cost, we use fast adversarial training (Wong et al., 2020) with the l_∞ norm for δ, and the maximum per-pixel change is set as 2/255.

Table 6 (see Appendices) evaluates the models' clean accuracy, real-world robustness, and adversarial robustness (under 10-step PGD attack (Madry et al., 2018)). It is clear that the landscape smoothing significantly improves the convolution-free architectures for both clean and adversarial accuracy. However, we observe a slight accuracy decrease on clean images for ResNets despite gain for robustness. Similar to our previous observations, *ViTs surpass similar-size ResNets when adversarially trained on ImageNet with Inception-style preprocessing for both clean accuracy and adversarial robustness.*

6 Conclusion

This paper presents a detailed analysis of the convolution-free ViTs and MLP-Mixers from the lens of the loss landscape geometry, intending to reduce the models' dependency on massive pre-training

and/or strong data augmentations. We arrive at the sharpness-aware minimizer (SAM) after observing sharp local minima of the converged models. By explicitly regularizing the loss geometry through SAM, the models enjoy much flatter loss landscapes and improved generalization regarding accuracy and robustness. The resultant ViT models outperform ResNets of comparable size and throughput when learned with no pre-training or strong augmentations. Further investigation reveals that the smoothed loss landscapes attribute to much sparser activated neurons in the first few layers. Moreover, ViTs after SAM offer perceptive attention maps. Last but not least, we discover that SAM and strong augmentations share certain similarities to enhance the generalization. They both smooth the average loss curvature and encourage linearity.

REFERENCES

Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *arXiv preprint arXiv:2104.11178*, 2021.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Irwan Bello, William Fedus, Xianzhi Du, Ekin D. Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies, 2021.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 557–565. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/botev17a.html.

Rebekka Burkholz and Alina Dubatovka. Initialization of relus for dynamical isometry. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d9731321ef4e063ebbee79298fa36f56-Paper.pdf.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations*, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four GPU hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=Cnon5ezMHtu.

Xiangning Chen and Cho-Jui Hsieh. Stabilizing differentiable architecture search via perturbation-based regularization. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1554–1565. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/chen20f.html.

Xiangning Chen, Cihang Xie, Mingxing Tan, Li Zhang, Cho-Jui Hsieh, and Boqing Gong. Robust and accurate object detection via adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16622–16631, June 2021b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers, 2021c.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3008–3017, 2020. doi: 10.1109/CVPRW50498.2020.00359.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler (eds.), *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. URL http://auai.org/uai2017/proceedings/papers/173.pdf.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2020.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization, 2020.

Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amost Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkgEaj05t7.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does SGD escape local minima? In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2698–2707. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/kleinberg18a.html.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 491–507, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58558-7.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf.

Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Luke Melas-Kyriazi. Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. *arXiv preprint arXiv:2105.02723*, 2021.

Y. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/7503cfacd12053d309b6bed5c89de212-Paper.pdf.

Yeonjong Shin and George Em Karniadakis. Trainability of relu networks and data-dependent initialization. *Journal of Machine Learning for Modeling and Computing*, 1(1):39–74, 2020. ISSN 2689-3967.

Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJij4yg0Z.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 843–852, 2017. doi: 10.1109/ICCV.2017.97.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training, 2021a.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention, 2021b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2958–2969. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1ef91c212e30e14bf125e9374262401f-Paper.pdf.

Lechao Xiao, Jeffrey Pennington, and Samuel Schoenholz. Disentangling trainability and generalization in deep neural networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10462–10472. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/xiao20b.html.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Ge Yang and Samuel Schoenholz. Mean field residual networks: On the edge of chaos. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/81c650caac28cdefce4de5ddc18befa0-Paper.pdf.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Syx4wnEtvH.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet, 2021.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019. doi: 10.1109/ICCV.2019.00612.

Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1gDNyrKDS.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

APPENDICES

A ARCHITECTURES

Table 7 specifies the ViT (Dosovitskiy et al., 2021; Vaswani et al., 2017) and MLP-Mixer (Tolstikhin et al., 2021) architectures used in this paper. "S" and "B" denote the small and base model scales following (Dosovitskiy et al., 2021; Touvron et al., 2021b; Tolstikhin et al., 2021), followed by the size of each image patch. For instance, "B/16" means the model of base scale with non-overlapping image patches of resolution 16×16. We use the input resolution 224×224 throughout the paper. Following Tolstikhin et al. (2021), we sweep the batch sizes in $\{32, 64, \ldots, 8192\}$ on TPU-v3 and report the highest throughput for each model.

Table 6: Comparison under the adversarial training framework on ImageNet (numbers in the parentheses denote the improvement over the standard adversarial training without SAM). With similar model size and throughput, ViTs-SAM can still outperform ResNets-SAM for clean accuracy and adversarial robustness.

Model	#params	Throughput (img/sec/core)	ImageNet	Real	V2	PGD-10	ImageNet-R	ImageNet-C
ResNet								
ResNet-50-SAM	25M	2161	70.1 (-0.7)	77.9 (-0.3)	56.6 (-0.8)	54.1 (+0.9)	27.0 (+0.9)	42.7 (-0.1)
ResNet-101-SAM	44M	1334	73.6 (-0.4)	81.0 (+0.1)	60.4 (-0.6)	58.8 (+1.4)	29.5 (+0.6)	46.9 (+0.3)
ResNet-152-SAM	60M	935	75.1 (-0.4)	82.3 (+0.2)	62.2 (-0.4)	61.0 (+1.8)	30.8 (+1.4)	49.1 (+0.6)
Vision Transformer								
ViT-S/16-SAM	22M	2043	73.2 (+1.2)	80.7 (+1.7)	60.2 (+1.4)	58.0 (+5.2)	28.4 (+2.4)	47.5 (+1.6)
ViT-B/32-SAM	88M	2805	69.9 (+3.0)	76.9 (+3.4)	55.7 (+2.5)	54.0 (+6.4)	26.0 (+3.0)	46.4 (+3.0)
ViT-B/16-SAM	87M	863	76.7 (+3.9)	82.9 (+4.1)	63.6 (+4.3)	62.0 (+7.7)	30.0 (+4.9)	51.4 (+5.0)
MLP-Mixer								
Mixer-S/16-SAM	18M	4005	67.1 (+2.2)	74.5 (+2.3)	52.8 (+2.5)	50.1 (+4.1)	22.9 (+2.6)	37.9 (+2.5)
Mixer-B/32-SAM	60M	4209	69.3 (+9.1)	76.4 (+10.2)	54.7 (+9.4)	54.5 (+13.9)	26.3 (+8.0)	43.7 (+8.8)
Mixer-B/16-SAM	59M	1390	73.9 (+11.1)	80.8 (+11.8)	60.2 (+11.9)	59.8 (+17.3)	29.0 (+10.5)	45.9 (+12.5)

Table 7: Specifications of the ViT and MLP-Mixer architectures used in this paper. We train all the architectures with image resolution 224×224.

Model	#params	Throughput (img/sec/core)	Patch Resolution	Sequence Length	Hidden Size	#heads	#layers	Token MLP Dimension	Channel MLP Dimension
ViT-S/32	23M	6888	32×32	49	384	6	12	–	–
ViT-S/16	22M	2043	16×16	196	384	6	12	–	–
ViT-S/14	22M	1234	14×14	256	384	6	12	–	–
ViT-S/8	22M	333	8×8	784	384	6	12	–	–
ViT-B/32	88M	2805	32×32	49	768	12	12	–	–
ViT-B/16	87M	863	16×16	196	768	12	12	–	–
Mixer-S/32	19M	11401	32×32	49	512	–	8	256	2048
Mixer-S/16	18M	4005	16×16	196	512	–	8	256	2048
Mixer-S/8	20M	1498	8×8	784	512	–	8	256	2048
Mixer-B/32	60M	4209	32×32	49	768	–	12	384	3072
Mixer-B/16	59M	1390	16×16	196	768	–	12	384	3072
Mixer-B/8	64M	466	8×8	784	768	–	12	384	3072

B TRANSFERABILITY

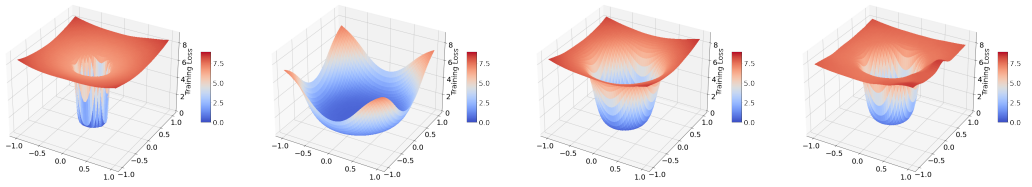
We also study the role of smoothed loss geometry in transfer learning. We select four datasets to test ViTs and MLP-Mixers' transferabilities: CIFAR-10/100 (Krizhevsky, 2009), Oxford-IIIT

Table 8: Hyperparameters for downstream tasks. All models are fine-tuned with 224×224 resolution, a batch size of 512, cosine learning rate decay, no weight decay, and grad clipping at global norm 1.

Dataset	Total steps	Warmup steps	Base LR
CIFAR-10	10K	500	{0.001, 0.003, 0.01, 0.03}
CIFAR-100	10K	500	
Flowers	500	100	
Pets	500	100	

Table 9: Accuracy on downstream tasks of the models pre-trained on ImageNet. SAM improves ViTs and MLP-Mixers' transferabilities to the tasks. ViTs transfer better than ResNets of similar sizes.

%	ResNet-50-SAM	ResNet-152-SAM	ViT-S/16	ViT-S/16-SAM	ViT-B/16	ViT-B/16-SAM	Mixer-S/16	Mixer-S/16-SAM	Mixer-B/16	Mixer-B/16-SAM
CIFAR-10	97.4	98.2	97.6	98.2	98.1	98.6	94.1	96.1	95.4	97.8
CIFAR-100	85.2	87.8	85.7	87.6	87.6	89.1	77.9	82.4	80.0	86.4
Flowers	90.0	91.1	86.4	91.5	88.5	91.8	83.3	87.9	82.8	90.0
Pets	91.6	93.3	90.4	92.9	91.9	93.1	86.1	88.7	86.1	92.5
Average	91.1	92.6	90.0	92.6	91.5	93.2	85.4	88.8	86.1	91.7



(a) ViT (b) ViT-SAM (c) ViT-AUG (d) ViT-21k

Figure 5: Cross-entropy loss landscapes of ViT-B/16, ViT-B/16-SAM, ViT-B/16-AUG, and ViT-B/16-21k. Strong augmentations and large-scale pre-training can also smooth the curvature.

Pets (Parkhi et al., 2012), and Oxford Flowers-102 (Nilsback & Zisserman, 2008). We use image resolution 224×224 during fine-tuning on downstream tasks, other settings exactly follow Dosovitskiy et al. (2021); Tolstikhin et al. (2021) (see Table 8). Note that we do not employ SAM during fine-tuning. We perform a grid search over the base learning rates on small sub-splits of the training sets (10% for Flowers and Pets, 2% for CIFAR-10/100). After that, we fine-tune on the entire training sets and report the results on the respective test sets. For comparison, we also include ResNet-50-SAM and ResNet-152-SAM in the experiments. Table 9 summarizes the results, which confirm that the enhanced models also perform better after fine-tuning and that MLP-Mixers gain the most from the sharpness-aware optimization.

C Visualization

C.1 Loss landscape

We use the "filter normalization" method (Li et al., 2018) to visualize the loss function curvature in Figure 1 and 5. For a fair comparison, we use the cross-entropy loss when plotting the landscapes for all architectures, although the original training objective is the sigmoid loss for ViTs and MLP-Mixers. Note that their sigmoid loss geometry is even sharper. We equally sample 2,500 points on the 2D projection space and compute the losses using 10% of the ImageNet training images (Chen et al., 2020), i.e., the i1k (1/10) subset in the main text to save computation.

C.2 Attention map

The visualization of the ViT's attention maps (Figure 3 in the main text) follows (Caron et al., 2021). We average the self-attention scores of the "classification token" from the last MSA layer to obtain a matrix $A \in \mathbb{R}^{H/P \times W/P}$, where H, W, P are the image height, width, and the patch resolution, respectively. Then we upsample A to the image shape $H \times W$ before generating the figure.

D Hessian Eigenvalue

The Hessian matrix requires second-order derivative, so we compute the Hessian (and all the sub-diagonal Hessian) λ_{max} using 10% of the ImageNet training images (i.e., i1k (1/10)) via power iteration [1], where we use 100 iterations to ensure its convergence.

[1] https://en.wikipedia.org/wiki/Power_iteration

Table 10: The SAM perturbation strength ρ for training on ImageNet. ViTs and MLP-Mixers favor larger ρ than ResNets does. Larger models with longer patch sequences need stronger strengths.

Model	**Task**	**SAM ρ**
ResNet		
ResNet-50-SAM	supervised	0.02
ResNet-101-SAM	supervised	0.05
ResNet-152-SAM	supervised	0.02
ResNet-50x2-SAM	supervised	0.05
ResNet-101x2-SAM	supervised	0.05
ResNet-152x2-SAM	supervised	0.05
ResNet-50-SAM	adversarial	0.05
ResNet-101-SAM	adversarial	0.05
ResNet-152-SAM	adversarial	0.05
ViT		
ViT-S/32-SAM	supervised	0.05
ViT-S/16-SAM	supervised	0.1
ViT-S/14-SAM	supervised	0.1
ViT-S/8-SAM	supervised	0.15
ViT-B/32-SAM	supervised	0.15
ViT-B/16-SAM	supervised	0.2
ViT-B/16-AUG-SAM	supervised	0.05
ViT-S/16-SAM	adversarial	0.1
ViT-B/32-SAM	adversarial	0.1
ViT-B/16-SAM	adversarial	0.1
ViT-S/16-SAM	supervised contrastive	0.02
ViT-B/16-SAM	supervised contrastive	0.02
MLP-Mixer		
Mixer-S/32-SAM	supervised	0.1
Mixer-S/16-SAM	supervised	0.15
Mixer-S/8-SAM	supervised	0.2
Mixer-B/32-SAM	supervised	0.35
Mixer-B/16-SAM	supervised	0.6
Mixer-B/8-SAM	supervised	0.6
Mixer-B/16-AUG-SAM	supervised	0.2
Mixer-S/16-SAM	adversarial	0.05
Mixer-B/32-SAM	adversarial	0.25
Mixer-B/16-SAM	adversarial	0.25

E NTK Condition Number

We approximate the neural tangent kernel on the i1k (1/10) subset by averaging over block diagonal entries (with block size 48×48) in the full NTK. Notice that the computation is based on the architecture at initialization without training. As the activation plays an important role when computing NTK — we find that smoother activation functions enjoy smaller condition numbers, we replace the GELU in ViT and MLP-Mixer with ReLU for a fair comparison with ResNet.

F Training Details

We use image resolution 224×224 during fine-tuning on downstream tasks, other settings exactly follow (Dosovitskiy et al., 2021; Tolstikhin et al., 2021) (see Table 8). Note that we do not employ SAM during fine-tuning. We perform a grid search over the base learning rates on small sub-splits of the training sets (10% for Flowers and Pets, 2% for CIFAR-10/100). After that, we fine-tune on the entire training sets and report the results on the respective test sets.

Except for the experiments in Section 4.5 (SAM with strong data augmentations) and Section 5.2 (contrastive learning), we train all the models from scratch on ImageNet with the basic Inception-style preprocessing (Szegedy et al., 2016), i.e., a random image crop and a horizontal flip with probability 50%. Please see Table 11 for the detailed training settings. We simply follow the original training settings of ResNet and ViT (Kolesnikov et al., 2020; Dosovitskiy et al., 2021). For MLP-Mixer, we remove the strong augmentations in its original training pipeline and perform a grid search over the learning rate in $\{0.003, 0.001\}$, weight decay in $\{0.3, 0.1, 0.03\}$, Dropout rate in $\{0.1, 0.0\}$, and stochastic depth in $\{0.1, 0.0\}$. Note that training for 90 epochs is enough for ResNets

Table 11: Hyperparameters for training from scratch on ImageNet with basic Inception-style preprocessing and 224×224 image resolution.

	ResNet	**ViT**	**MLP-Mixer**
Data augmentation		Inception-style	
Input resolution		224×224	
Batch size		4096	
Epoch	90	300	300
Warmup steps	5K	10K	10K
Peak learning rate	$0.1 \times \frac{\text{batch size}}{256}$	3e-3	3e-3
Learning rate decay	cosine	cosine	linear
Optimizer	SGD	AdamW	AdamW
SGD Momentum	0.9	–	–
Adam (β_1, β_2)	–	(0.9, 0.999)	(0.9, 0.999)
Weight decay	1e-3	0.3	0.3
Dropout rate	0.0	0.1	0.0
Stochastic depth	–	–	0.1
Gradient clipping	–	1.0	1.0

Table 12: ImageNet top-1 accuracy (%) of ViT-B/16 and Mixer-B/16 when trained from scratch with different perturbation strength ρ in SAM.

SAM ρ	0.0	0.05	0.1	0.2	0.25	0.35	0.4	0.5	0.6	0.65
ViT-B/16	74.6	77.5	78.8	**79.9**	79.3	–	–	–	–	–
Mixer-B/16	66.4	69.5	–	–	74.1	74.7	75.6	76.9	**77.4**	77.1

to converge, and longer schedule brings almost no effect. For all the experiments, we use 128 TPU-v3 cores (2 per chip), resulting in 32 images per core. The SAM computation for $\hat{\epsilon}$ is conducted on each core independently.

F.1 Perturbation strength in SAM

Different architecture species favor different strengths of perturbation ρ. We perform a grid search over ρ and report the best results — Table 10 reports the corresponding strengths used in our ImageNet experiments. Besides, we show the results when varying ρ in Table 12. Similar to (Foret et al., 2021), we also find that a relative small $\rho \in [0.02, 0.05]$ works the best for ResNets. However, larger ρ gives rise to the best results for ViTs and MLP-Mixers. We also observe that architectures with larger capacities and longer input sequences prefer stronger perturbation strengths. Interestingly, the choice of ρ coincides with our previous observations. Since MLP-Mixers suffer the sharpest landscapes, they need the largest perturbation strength. As strong augmentations and contrastive learning already improve generalization, the suitable ρ becomes significantly smaller. Note that we do not re-tune any other hyperparameters when using SAM.

F.2 Training on ImageNet subsets

In Section 5.1, we train the models on ImageNet subsets, and the hyperparameters have to be adjusted accordingly. We simply change the batch size to maintain similar total iterations and keep all other settings the same, i.e., 2048 for i1k (1/2), 1024 for i1k (1/4), and 512 for i1k (1/10). We do not scale the learning rate as we find the scaling harms the performance.

F.3 Training with strong augmentations

We tune the learning rate and regularization when using strong augmentations (mixup with probability 0.5, RandAugment with two layers and magnitude 15) in Section 4.5 following (Tolstikhin et al., 2021). For ViT, we use 1e-3 peak learning rate, 0.1 weight decay, 0.1 Dropout, and 0.1 stochastic depth; For MLP-Mixer, those hyperparameters are exactly the same as (Tolstikhin et al., 2021), peak learning rate as 1e-3, weight decay as 0.1, Dropout as 0.0, and stochastic depth as 0.1. Other settings are unchanged (Table 11).

F.4 CONTRASTIVE LEARNING

In Section 5.2, we train ViTs under the supervised contrastive learning framework (Khosla et al., 2020). We take the classification token output from the last layer as the encoded representation and retain the structures of the projection and classification heads (Khosla et al., 2020). We employ a batch size 2048 without memory bank (He et al., 2020) and use AutoAugment (Cubuk et al., 2019) with strength 1.0 following Khosla et al. (2020). For the 350-epoch pretraining stage, the contrastive loss temperature is set as 0.1, and we use the LAMB optimizer (You et al., 2020) with learning rate $0.001 \times \frac{\text{batch size}}{256}$ along with a cosine decay schedule. For the second stage, we train the classification head for 90 epochs via a RMSProp optimizer (Tieleman & Hinton, 2012) with base learning rate 0.05 and exponential decay. The weight decays are set as 0.3 and 1e-6 for the first and second stages, respectively. We use a small SAM perturbation strength $\rho = 0.02$.

F.5 ADVERSARIAL LEARNING

We use the fast adversarial training (Wong et al., 2020) (FGSM with random start) with the l_∞ norm and maximum per-pixel change 2/255 during training. All the hyperparameters remain the same as the vanilla supervised training. When evaluating the adversarial robustness, we use the PGD attack (Madry et al., 2018) with the same maximum per-pixel change 2/255. The total number of attack steps is 10, and the step size is 0.25/255.

G LONGER SCHEDULE OF VANILLA SGD

Since SAM needs another forward and backward propagation to compute $\hat{\epsilon}$, its training overhead is $\sim 2\times$ of the vanilla baseline. We also experiment with $2\times$ schedule vanilla training (600 epochs). We observe that training longer brings no effect on both clean accuracy and robustness, indicating that the current 300 training epochs for ViTs and MLP-Mixers are enough for them to converge.