How Do Vision Transformers Work?

Namuk Park, Songkuk Kim
Yonsei University
`{namuk.park,songkuk}@yonsei.ac.kr`

Abstract

The success of multi-head self-attentions (MSAs) for computer vision is now indisputable. However, little is known about how MSAs work. We present various explanations to help better understand the nature of MSAs. In particular, we demonstrate the following properties of MSAs and Vision Transformers (ViTs): ❶ MSAs improve not only accuracy but also generalization by flattening the loss landscapes. Such improvement is primarily attributable to their data specificity, not long-range dependency. On the other hand, ViTs suffer from non-convex losses. Large datasets and loss landscape smoothing methods alleviate this problem; ❷ MSAs and Convs exhibit opposite behaviors. For example, MSAs are low-pass filters, but Convs are high-pass filters. Therefore, MSAs and Convs are complementary; ❸ Multi-stage neural networks behave like a series connection of small individual models. In addition, MSAs at the end of a stage play a key role in prediction. Based on these insights, we propose AlterNet, a model in which Conv blocks at the end of a stage are replaced with MSA blocks. AlterNet outperforms CNNs not only in large data regimes but also in small data regimes.

1 Introduction

There is limited understanding of multi-head self-attentions (MSAs), although they are now ubiquitous in computer vision. The most widely accepted explanation for the success of MSAs is their weak inductive bias and capture of long-range dependencies (See, e.g., (Dosovitskiy et al., 2021; Naseer et al., 2021; Tuli et al., 2021; Yu et al., 2021; Mao et al., 2021; Chu et al., 2021)). Yet because of their over-flexibility, Vision Transformers (ViTs)—neural networks (NNs) consisting of MSAs—have been known to have a tendency to overfit training datasets, consequently leading to poor predictive performance in small data regimes, e.g., image classification on CIFAR. However, we show that the explanation is poorly supported.

1.1 Related Work

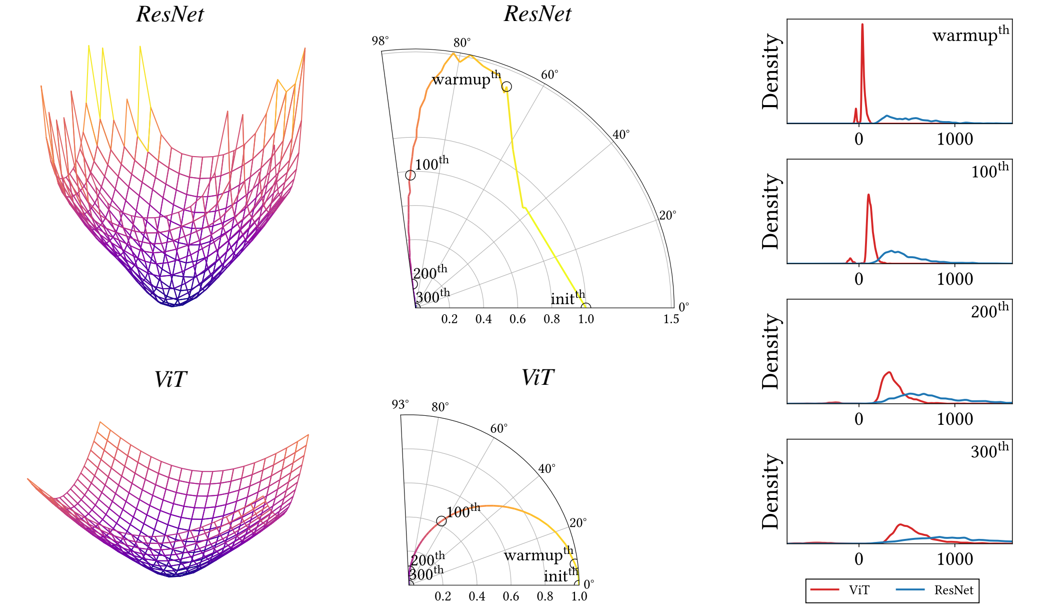
Self-attentions (Vaswani et al., 2017; Dosovitskiy et al., 2021) aggregate (spatial) tokens with normalized importances:

$$\boldsymbol{z}_j = \sum_i \texttt{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}}{\sqrt{d}}\right)_i \boldsymbol{V}_{i,j} \tag{1}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are query, key, and value, respectively. d is the dimension of query and key, and $\boldsymbol{z}_j$ is the j-th output token. From the perspective of convolutional neural networks (CNNs), MSAs are a transformation of all feature map points with large-sized data-specific kernels. Therefore, MSAs are at least as expressive as convolutional layers (Convs) (Cordonnier et al., 2020), although this does not guarantee that MSAs will behave like Convs.

Is the weak inductive bias of MSA, such as modeling long-range dependencies, beneficial for the predictive performance? To the contrary, appropriate constraints may actually help a model learn strong representations. For example, local MSAs (Yang et al., 2019; Liu et al., 2021; Chu et al., 2021), which calculate self-attention only within small windows, achieve better performance than global MSAs not only on small datasets but also on large datasets, e.g., ImageNet-21K.

In addition, prior works observed that MSAs have the following intriguing properties: ① MSAs improve the predictive performance of CNNs (Dai et al., 2021; Guo et al., 2021; Srinivas et al., 2021),



(a) Loss landscape visualizations (b) Trajectories in polar coordinates (c) Hessian max eigenvalue spectra

Figure 1: **Three different aspects consistently show that MSAs flatten loss landscape**. *Left*: Loss landscape visualizations show that ViT has a flatter loss (NLL + ℓ_2 regularization) than ResNet. *Middle*: ViT converges to the optimum along a smooth trajectory. In the polar coordinate, $r_t = \frac{||\Delta \boldsymbol{w}_t||}{||\Delta \boldsymbol{w}_{\text{init}}||}$ and $\theta_t = \cos^{-1}\left(\frac{\Delta \boldsymbol{w}_t \cdot \Delta \boldsymbol{w}_{\text{init}}}{||\Delta \boldsymbol{w}_t||\,||\Delta \boldsymbol{w}_{\text{init}}||}\right)$ where $\Delta \boldsymbol{w}_t = \boldsymbol{w}_t - \boldsymbol{w}_{\text{optim}}$ and $\boldsymbol{w}_t$ is NN weight at t^{th} epoch. $\boldsymbol{w}_{\text{init}}$ and $\boldsymbol{w}_{\text{optim}}$ are the weights at initialization and optimum. *Right*: The magnitude of the Hessian eigenvalues of ViT is smaller than that of ResNet. See Fig. 4 for a more detailed analysis.

and ViTs predict well-calibrated uncertainty (Minderer et al., 2021). ② ViTs are robust against data corruptions, image occlusions (Naseer et al., 2021), and adversarial attacks (Shao et al., 2021; Bhojanapalli et al., 2021; Paul & Chen, 2021; Mao et al., 2021). They are particularly robust against high-frequency noises (Shao et al., 2021). ③ MSAs closer to the last layer significantly improve predictive performance (Graham et al., 2021; Dai et al., 2021; Xiao et al., 2021).

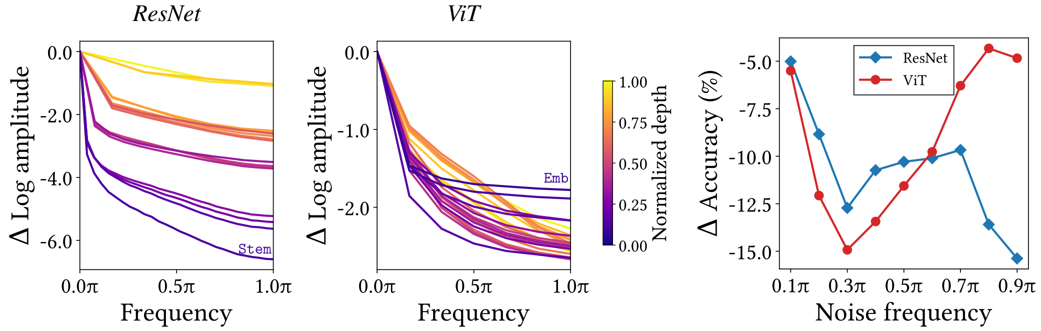
These empirical observations raise immediate questions: ❶ What properties of MSAs do we need to better optimize NNs? Do the long-range dependencies of MSAs help NNs learn? ❷ Do MSAs act like Convs? If not, how are they different? ❸ How can we harmonize MSAs with Convs? Can we just leverage their advantages?

We provide an explanation of how MSAs work by addressing them as a trainable spatial smoothing of feature maps, because Eq. (1) also suggests that MSAs average feature map values with the positive importance-weights. Even non-trainable spatial smoothings, such as a small 2×2 box blur, help CNNs see better (Zhang, 2019; Anonymous, 2022). These simple spatial smoothings not only improve accuracy but also robustness by ensembling feature map points and flattening the loss landscapes (Anonymous, 2022). Remarkably, spatial smoothings have the properties of MSAs ①–③. See Appendix B for theoretical and empirical explanations of MSAs as a spatial smoothing.

1.2 Contribution

We address the three key questions:

❶ **What properties of MSAs do we need to improve optimization?** We present various evidences to support that MSA is generalized spatial smoothing. It means that MSAs improve performance because their formulation—Eq. (1)—is an appropriate inductive bias. Their weak inductive bias disrupts NN training. In particular, *a key feature of MSAs is their data specificity*, not long-range dependency. As an extreme example, local MSAs with a 3×3 receptive field outperforms global MSA because they reduce unnecessary degrees of freedom.



(a) Relative log amplitudes of Fourier transformed feature maps. (b) Robustness for noise frequency

Figure 2: **The Fourier analysis shows that MSAs do not act like Convs**. *Left*: Relative log amplitudes of Fourier transformed feature map show that ViT tends to reduce high-frequency signals, while ResNet amplifies them. Δ Log amplitude is the difference between the log amplitude at normalized frequency 0.0π (center) and at 1.0π (boundary). See Fig. 8 for more detailed analysis. *Right*: We measure the decrease in accuracy against frequency-based random noise. ResNet is vulnerable to high-frequency noise, while ViT is robust against them. We use frequency window size of 0.1π.

How do MSAs improve performance? MSAs have their advantages and disadvantages. On the one hand, they flatten loss landscapes as shown in Fig. 1. The flatter the loss landscape, the better the performance and generalization (Li et al., 2018; Keskar et al., 2017; Santurkar et al., 2018; Foret et al., 2021; Chen et al., 2021). Thus, they improve not only accuracy but also robustness in large data regimes. On the other hand, MSAs allow negative Hessian eigenvalues in small data regimes. This means that the loss landscapes of MSAs are non-convex, and this non-convexity disturbs NN optimization (Dauphin et al., 2014). Large amounts of training data suppress negative eigenvalues and convexify losses.

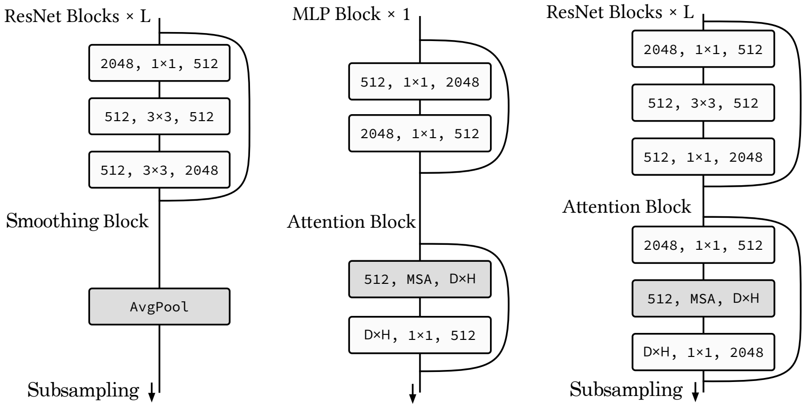
❷ **Do MSAs act like Convs?** We show that MSAs and Convs exhibit opposite behaviors. MSAs aggregate feature maps, but Convs disperse them. Moreover, as shown in Fig. 2a, the Fourier analysis of feature maps shows that MSAs reduce high-frequency signals, while Convs, conversely, amplifies high-frequency components. In other words, *MSAs are low-pass filters, but Convs are high-pass filters*. In addition, Fig. 2b indicates that Convs are vulnerable to high-frequency noise but that MSAs are not. Therefore, MSAs and Convs are complementary.

❸ **How can we harmonize MSAs with Convs?** We reveal that multi-stage NNs behave like a series connection of small individual models. Thus, applying spatial smoothing at the end of a stage improves accuracy by ensembling transformed feature map outputs from each stage (Anonymous, 2022) as shown in Fig. 3a. Based on this finding, *we propose an alternating pattern of Convs and MSAs*. NN stages using this design pattern consists of a number of CNN blocks and one (or a few) MSA block as shown in Fig. 3c. The design pattern naturally derives the structure of canonical Transformer, which has one MSA block per MLP block as shown in Fig. 3b. It also provides an explanation of how adding Convs to Transformer's MLP block improves accuracy and robustness (Yuan et al., 2021; Guo et al., 2021; Mao et al., 2021).

Surprisingly, models using this alternating pattern of Convs and MSAs outperform CNNs not only on large datasets but also on small datasets, such as CIFAR. This contrasts with canonical ViTs, models that perform poorly on small amount of data. It implies that MSAs are generalized spatial smoothings that complement Convs, not simply generalized Convs.

2 What Properties of MSAs Do We Need To Improve Optimization?

To understand the underlying nature of MSAs, we investigate the properties of the ViT family: e.g., vanilla ViT (Dosovitskiy et al., 2021); PiT (Heo et al., 2021), which is "ViT + multi-stage"; and Swin (Liu et al., 2021), which is "ViT + multi-stage + local MSA". This section shows that these additional inductive biases enable ViTs to learn strong representations. We also use ResNet (He et al., 2016a) for comparison. NNs are trained from scratch with DeiT-style data augmentation (Touvron et al., 2021)



(a) Spatial smoothing (b) Canonical Transformer (c) Alternating pattern (*ours*)

Figure 3: **Comparison of three different repeating patterns**. *Left*: Spatial smoothings are located at the end of CNN stages. *Middle*: The stages of ViTs consist of repetitions of canonical Transformers. "D" is the hidden dimension and "H" is the number of heads. *Right*: The stages using alternating pattern consists of a number of CNN blocks and an MSA block. For more details, see Fig. 11.

for 300 epochs. The NN training begins with a gradual warmup (Goyal et al., 2017) for 5 epochs. Appendix A provides more detailed configurations and background information for experiments.

The stronger the inductive biases, the stronger the representations (*not regularizations*). Do models with weak inductive biases overfit training datasets? To address this question, we provide two criteria on CIFAR-100: the error of the test dataset and the cross-entropy, or the negative log-likelihood, of the training dataset ($\text{NLL}_{\text{train}}$, the lower the better). See Fig. 5a for the results.

Contrary to our expectations, experimental results show that the stronger the inductive bias, the lower *both* the test error and the training NLL. This indicates that *ViT does not overfit training datasets*. In addition, appropriate inductive biases, such as locality constraints for MSAs, helps NNs learn strong representations. We also observe these phenomena on CIFAR-10 and ImageNet as shown in Fig. C.1. Figure C.3 also supports that weak inductive biases disrupt NN training. In this experiment, extremely small patch sizes for the embedding hurt the predictive performance of ViT.

ViT does not overfit small training datasets. We observe that ViT does not overfit even on smaller datasets. Figure 5b shows the test error and the training NLL of ViT on subsampled datasets. In this experiment, as the size of the dataset decreases, the error increases as expected, but surprisingly, $\text{NLL}_{\text{train}}$ also increases. Thanks to the strong data augmentation, ViT does not overfit even on a dataset size of 2%. This suggests that ViT's poor performance in small data regimes is *not* due to overfitting.

ViT's non-convex losses lead to poor performance. How do weak inductive biases of MSAs disturb the optimization? A loss landscape perspective provides an explanation: *the loss function of ViT is non-convex, while that of ResNet is strongly (near-)convex*. This poor loss disrupts NN training (Dauphin et al., 2014), especially in the early phase of training (Jastrzebski et al., 2020; 2021). Figure 1c and Fig. 4 provide top-5 largest Hessian eigenvalue densities (Anonymous, 2022) with a batch size of 16. The figures show that ViT has a number of negative Hessian eigenvalues, while ResNet only has a few.

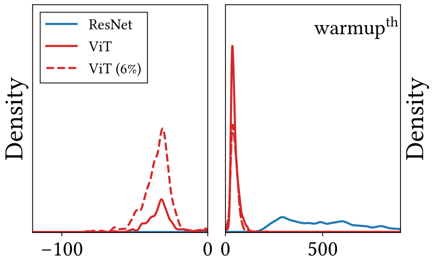
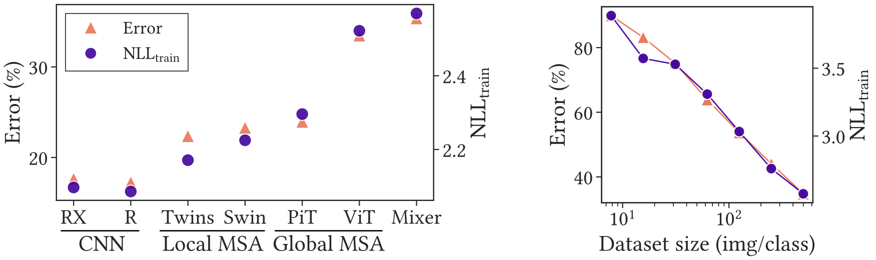


Figure 4: **Hessian max eigenvalue spectra show that MSAs have their advantages and disadvantages.** The dotted line is the spectrum of ViT using 6% dataset for training. *Left*: ViT has a number of negative Hessian eigenvalues, while ResNet only has a few. *Right*: The magnitude of ViT's positive Hessian eigenvalues is small. See also Fig. 1c for more results.



(a) Error and NLL$_{train}$ for each model. (b) Performance of ViT for dataset size.

Figure 5: **ViT does not overfit training datasets.** "R" is ResNet and "RX" is ResNeXt. *Left:* Weak inductive bias disturbs NN optimization. The lower the NLL$_{train}$, the lower the error. *Right:* The lack of dataset also disturbs NN optimization.

Figure 4 also shows that large datasets suppress negative Hessian eigenvalues in the early phase of training. Therefore, large datasets tend to help ViT learn strong representations by convexifying the loss. ResNet enjoys little benefit from large datasets because its loss is convex even on small datasets.

Loss landscape smoothing methods aids in ViT training. Loss landscape smoothing methods can also help ViT learn strong representations. In classification tasks, global average pooling (GAP) smoothens the loss landscape by strongly ensembling feature map points (Anonymous, 2022). We demonstrate how the loss smoothing method can help ViT improve performance by analyzing ViT with GAP classifier instead of CLS token on CIFAR-100.

Figure 6 shows the Hessian max eigenvalue spectrum of the ViT with GAP. As expected, the result shows that GAP classifier suppresses negative Hessian max eigenvalues, suggesting that GAP convexify the loss. Since negative eigenvalues disturb NN optimization, GAP classifier improve the accuracy by +2.7 percent point.

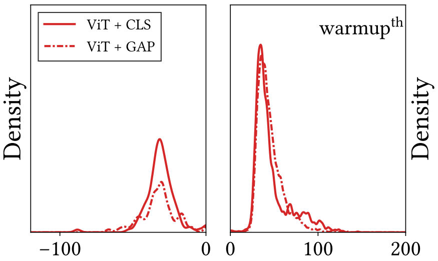
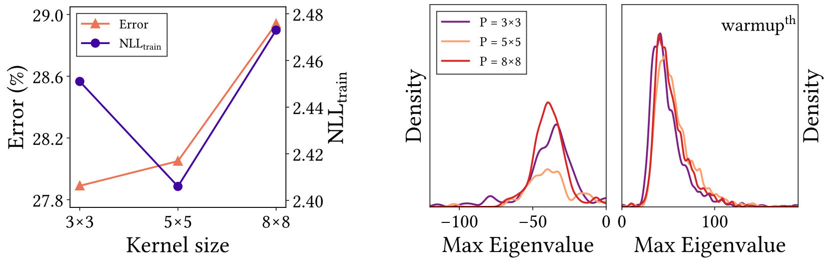


Figure 6: **GAP classifier suppresses negative Hessian max eigenvalues** in an early phase of training. We present Hessian max eigenvalue spectrum of ViT with GAP classifier instead of CLS token.

Likewise, Sharpness-Aware Minimization (SAM) (Foret et al., 2021), an optimizer that relies on the local smoothness of the loss function, also helps NNs seek out smooth minima. Chen et al. (2021) showed that SAM improves the predictive performance of ViT.

MSAs flatten the loss landscape. Another property of MSAs is that they reduces the magnitude of Hessian eigenvalues. Figure 1c and Fig. 4 shows that the eigenvalues of ViT are significantly smaller than that of CNNs. While large eigenvalues impede NN training (Ghorbani et al., 2019), *MSAs can help NNs learn better representations by suppressing large Hessian eigenvalues*. Figure 1a and Fig. 1b also support this claim. In Fig. 1a, we visualize the loss landscapes by using filter normalization (Li et al., 2018). The loss landscape of ViT is flatter than that of ResNet, and this trend is noticeable at the boundary. Similarly, Fig. 1b shows that ResNet follows an irregular trajectory, especially in the early phase of training; ViT converges to the optimum along a smooth trajectory. In large data regimes, the negative Hessian eigenvalues—the disadvantage of MSAs—disappears, and only their advantages remain. As a result, ViTs outperform CNNs on large datasets, such as ImageNet and JFT (Sun et al., 2017). PiT and Swin also flatten the loss landscapes. See Fig. C.4.

A key feature of MSAs is data specificity *(not long-range dependency)*. The two distinguishing features of MSAs are long-range dependency and data specificity. Contrary to popular belief, the long-range dependency hinders NN optimization. To demonstrate this, we analyze *convolutional ViT*, which consists of two-dimensional convolutional MSAs (Yang et al., 2019) instead of global MSAs. Convolutional MSAs calculates self-attention only between feature map points in convolutional receptive fields after unfolding the feature maps in the same way as convolutions.



(a) Error and $\mathrm{NLL}_{\mathrm{train}}$ of ViT with local MSA for kernel size

(b) Hessian negative and positive max eigenvalue spectra in early phase of training

Figure 7: **Locality constraint improves the performance of ViT**. We analyze the ViT with convolutional MSAs. Convolutional MSA with 8×8 kernel is global MSA. *Left:* Local MSAs learn stronger representations than global MSA. *Right:* Locality inductive bias suppresses the negative Hessian eigenvalues, i.e., local MSAs have convex losses.

Figure 7a shows the error and $\mathrm{NLL}_{\mathrm{train}}$ of convolutional ViTs with kernel sizes of 3×3, 5×5, and 8×8 (global MSA) on CIFAR-100. In this experiment, 5×5 kernel outperforms 8×8 kernel on both the training and the test datasets. $\mathrm{NLL}_{\mathrm{train}}$ of 3×3 kernel is worse than that of 5×5 kernel, but better than that of global MSA. Although the test accuracies of 3×3 and 5×5 kernels are comparable, the robustness of 5×5 kernel is significantly better than that of 3×3 kernel on CIFAR-100-C (Hendrycks & Dietterich, 2019).

Figure 7b shows that the strong locality inductive bias not only reduce computational complexity as originally proposed (Liu et al., 2021), but also aid in optimization by convexifying the loss landscape. 5×5 kernel has fewer negative eigenvalues than global MSA because it restricts unnecessary degrees of freedom. 5×5 kernel also has fewer negative eigenvalues than 3×3 kernel because it ensembles a larger number of feature map points (See also Fig. 6). The amount of negative eigenvalues is minimized when the two effects are balanced.

It is clear that data specificity improves NNs. MLP-Mixer (Tolstikhin et al., 2021; Yu et al., 2021), a model with an MLP kernel that does not depend on input data, underperforms compared to ViTs. Data specificity without self-attention (Bello, 2021) improves performance.

3 Do MSAs Act Like Convs?

Convs are data-agnostic and channel-specific. In contrast, MSAs are data-specific and channel-agnostic. This section shows that these differences lead to large behavioral differences. It suggests that MSAs and Convs are complementary.

MSAs are low-pass filters, but Convs are high-pass filters. As explained in Section 1.1, MSAs spatially smoothen feature maps with self-attention importances. Therefore, we expect that MSAs will tend to reduce high-frequency signals. See Proposition B.1 for a more detailed discussion.

Figure 8 shows the relative log amplitude (Δ log amplitude) of ViT's Fourier transformed feature map at high-frequency (1.0π) on ImageNet. In this figure, MSAs almost always decrease the high-frequency amplitude, and MLPs—corresponding to Convs—increase it. The only exception is in the early stages of the model. In these stages, MSAs behave like Convs, i.e., they increase the amplitude. This could serve as an evidence for a hybrid model that uses Convs in early stages and MSAs in late stages (Guo et al., 2021; Graham et al., 2021; Dai et al., 2021; Xiao et al., 2021; Srinivas et al., 2021).

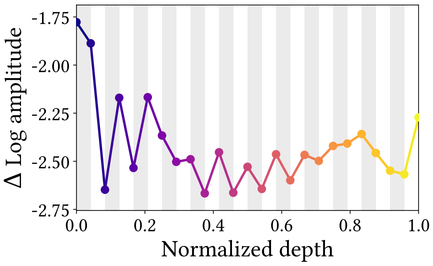


Figure 8: **MSAs (gray area) generally reduce the high-frequency component of feature map, and MLPs (white area) amplify it**. This figure provides Δ log amplitude of ViT at high-frequency (1.0π). See also Fig. 2a and Fig. D.2 for more results.

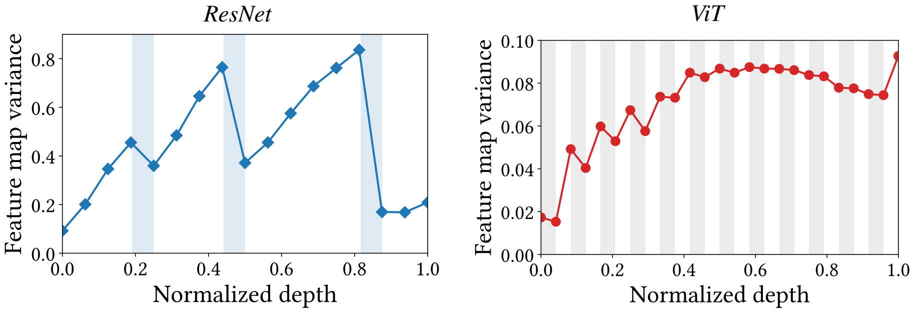


Figure 9: **MSAs (gray area) reduce the variance of feature map points, but Convs (white area) increase the variance.** The blue area is subsampling layer. The results implies that MSAs ensemble feature maps, but Convs do not.

Based on this, we can infer that *low-frequency signals and high-frequency signals are informative to MSAs and Convs, respectively*. In support of this argument, we report the robustness of ViT and ResNet against frequency-based random noise. Following Shao et al. (2021) and Anonymous (2022), we measure the decrease in accuracy with respect to data with frequency-based random noise $\boldsymbol{x}_{\text{noise}} = \boldsymbol{x}_0 + \mathcal{F}^{-1}\left(\mathcal{F}(\delta) \odot \mathbf{M}_f\right)$, where $\boldsymbol{x}_0$ is clean data, $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are Fourier transform and inverse Fourier transform, δ is Gaussian random noise, and $\mathbf{M}_f$ is frequency mask.

Figure 2b shows the results. As expected, the results reveals that ViT and ResNet are vulnerable to low-frequency noise and high-frequency noise, respectively. Low-frequency signals and the high-frequency signals each correspond to the shape and the texture of images. The results thus suggests that MSAs are shape-biased (Naseer et al., 2021), whereas Convs are texture-biased (Geirhos et al., 2019).

MSAs aggregate feature maps, but Convs do not. Since MSAs average feature maps, they will reduce variance of feature map points. This suggests that MSAs ensemble feature maps (Anonymous, 2022). To demonstrate this claim, we measure the variance of feature maps from NN layers.

Figure 9 shows the experimental results of ResNet and ViT. This figure indicates that MSAs in ViT tend to reduce the variance; conversely, Convs in ResNet and MLPs in ViT increase it. In conclusion, *MSAs ensemble feature maps, but Convs do not*. As Anonymous (2022) figured out, reducing the feature map uncertainty helps optimization by ensembling and stabilizing the feature maps. See Fig. D.1 for more results on PiT and Swin.

We observe two additional patterns for feature map variance. First, the variance accumulates in every NN layer and tends to increase as the depth increases. Second, the feature map variance in ResNet peaks at the ends of each stage. Therefore, we can improve the predictive performance of ResNet by inserting MSAs at the end of each stage. Furthermore, we also can improve the performance by using MSAs with a large number of heads in late stages.

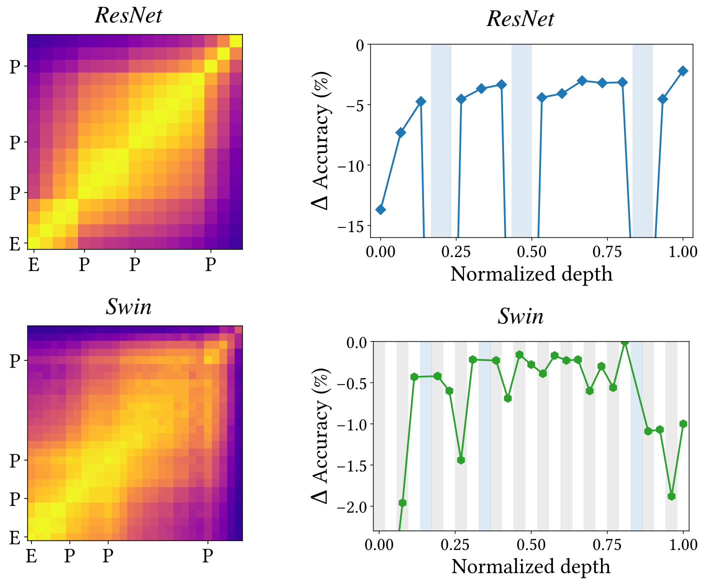
4 How Can We Harmonize MSAs With Convs?

Since MSAs and Convs are complementary, this section seeks to design a model that leverages only the advantages of the two modules. To this end, we propose the design rules described in Fig. 3c, and demonstrate that the models using these rules outperforms CNNs, not only in the large data regimes but also in the small data regimes, such as CIFAR.

4.1 Designing Architecture

We first investigate the properties of multi-stage NN architectures. Based on this investigation, we come to propose an alternating pattern, i.e., a principle for stacking MSAs based on CNNs.

Multi-stage NNs behave like individual models. In Fig. 9, we observe that the pattern of feature map variance repeats itself at every stages. This behavior is also observed in feature map similarities and lesion studies.



(a) Feature map similarity

(b) Accuracy of one-unit-removed model.

Figure 10: **Multi-stage CNNs and ViTs behave like a series connection of small individual models.** *Left:* The feature map similarities show the block structure of ResNet and Swin. "E" stands for stem/embedding and "P" for pooling (subsampling) layer. *Right:* We measure decrease in accuracy after removing one unit from the trained model. Accuracy changes periodically, and this period is one stage. White, gray, and blue areas are Conv/MLP, MSA, and subsampling layers, respectively.

Figure 10a shows the representational similarities of ResNet and Swin on CIFAR-100. In this experiment, we use mini-batch CKA (Nguyen et al., 2021) to measure the similarities. As Nguyen et al. (2021) figured out, the feature map similarities of CNNs have a block structure. Likewise, we observe that the feature map similarities of multi-stage ViTs, such as PiT and Swin, also have a block structure. Since vanilla ViT does not have this structure (Bhojanapalli et al., 2021; Raghu et al., 2021), the structure is an intrinsic characteristic of multi-stage architectures. See Fig. D.3 for more detailed results of ViT, PiT, and Swin.

Figure 10b shows the results of lesion study (Bhojanapalli et al., 2021), where one NN unit is removed from already trained ResNet and Swin during the testing phase. In this experiment, we remove one 3×3 Conv layer from the bottleneck block of ResNet, and one MSA or MLP block from Swin. In ResNet, removing an early stage layers hurts accuracy more than removing a late stage layers. More importantly, removing a layer at the beginning of a stage impairs accuracy more than removing a layer at the end of a stage. The case of Swin is even more interesting. At the beginning of a stage, removing an MLP hurts accuracy. At the end of a stage, removing an MSA seriously impairs the accuracy. These results are consistent with Fig. 8. See Fig. D.4 for the results on ViT and PiT.

Based on these findings, we expect *MSAs closer to the end of a stage* to significantly improve the performance. This is contrary to the popular belief that MSAs closer to the end of a model improve the performance (Srinivas et al., 2021; d'Ascoli et al., 2021; Graham et al., 2021; Dai et al., 2021).

Build-up rule. Considering all the insights, we propose the following design rules:

- Alternately replace Conv blocks with MSA blocks from the end of a baseline CNN model.
- If the added MSA block does not improve predictive performance, replace a Conv block located at the end of an earlier stage with an MSA block .
- Use more heads and higher hidden dimensions for MSA blocks in late stages.

We call the model that follows these rules *AlterNet*. AlterNet unifies ViTs and CNNs by adjusting the ratio of MSAs and Convs as shown in Fig. 3. Figure 11 shows AlterNet based on pre-activation ResNet-50 (He et al., 2016b) for CIFAR-100 as an example. Figure D.5 shows AlterNet for ImageNet.

Figure 12a reports the accuracy of *Alter-ResNet-50*, which replaces the Conv blocks in ResNet-50 with local MSAs (Liu et al., 2021) according to the aforementioned rules, on CIFAR-100. As expected, MSAs in the last stage (c4) significantly improve the accuracy. Surprisingly, an MSA in 2nd stage (c2) improves the accuracy, while two or more MSAs in the 3rd stage (c3) reduce it. In conclusion, MSAs at the end of a stage play an important role in prediction.

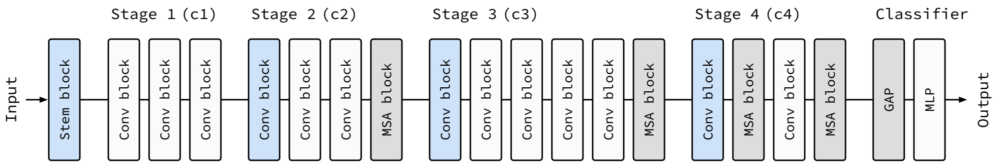
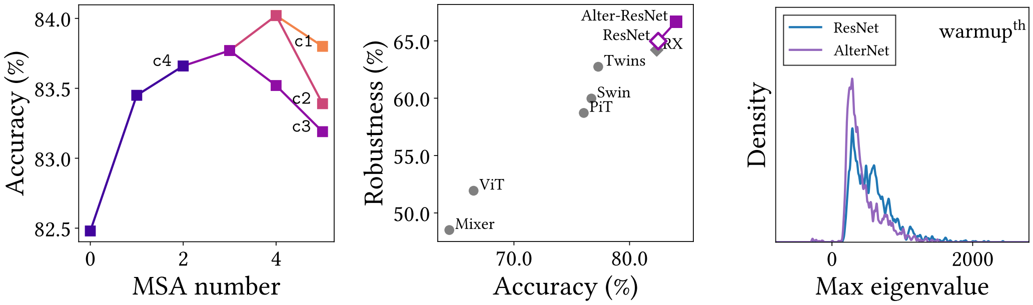


Figure 11: **Detailed architecture of Alter-ResNet-50 for CIFAR-100.** White, gray, and blue blocks mean Conv, MSA, and subsampling blocks. All stages (except stage 1) end with MSA blocks. This model is based on pre-activation ResNet-50. Following Swin, MSAs in stages 1 to 4 have 3, 6, 12, and 24 heads, respectively.



(a) Accuracy of AlterNet for MSA number

(b) Accuracy and robustness in a small data regime (CIFAR-100)

(c) Hessian max eigenvalue spectra in an early phase of training

Figure 12: **AlterNet outperforms CNNs and ViTs.** *Left:* MSAs in the late of the stages improve accuracy. We replace Convs of ResNet with MSAs one by one according to the build-up rules. c1 to c4 stands for the stages. Several MSAs in c3 harm the accuracy, but the MSA at the end of c2 improves it. *Center:* AlterNet outperforms CNNs even in a small data regime. Robustness is mean accuracy on CIFAR-100-C. "RX" is ResNeXt. *Right:* MSAs in AlterNet suppress the large eigenvalues; i.e., AlterNet has a flatter loss landscape than ResNet in the early phase of training.

Figure 12c demonstrates that MSAs suppress large eigenvalues while allowing only a few negative eigenvalues. As explained in Fig. 4, large datasets compensate for the shortcomings of MSAs. Therefore, more data allows more MSAs for a models.

4.2 Performance

Figure 12b shows the accuracy and corruption robustness of Alter-ResNet-50 and other baselines on CIFAR-100 and CIFAR-100-C. Since CIFAR is a small dataset, CNNs outperforms canonical ViTs. Surprisingly, Alter-ResNet—a model with MSAs following the appropriate build-up rule—outperforms CNNs even in the small data regimes. This suggests that MSAs complement Convs. In the same manner, this simple modification shows competitive performance on larger datasets, such as ImageNet. See Fig. E.1 for more details.

5 Discussion

Our present work demonstrates that MSAs are not merely generalized Convs, but rather generalized spatial smoothings that complement Convs. MSAs help NNs learn strong representations by ensembling feature map points and flattening the loss landscape. Since the main objective of this work is to investigate the nature of MSA for computer vision, we preserve the architectures of Conv and MSA blocks in AlterNet. Thus, AlterNet has a strong potential for future improvements. In addition, AlterNet can conveniently replace the backbone for other vision tasks such as dense prediction (Carion et al., 2020). As Anonymous (2022) pointed out, global average pooling (GAP) for simple classification tasks has a strong tendency to ensemble feature maps, but NNs for dense prediction do not use GAP. Therefore, we believe that MSA to be able to significantly improve the results in dense prediction tasks by ensembling feature maps. Lastly, strong data augmentation for MSA training harms uncertainty calibration as shown in Fig. F.1a. We leave a detailed investigation for future work.

Reproducibility Statement

To ensure reproducibility, we provide comprehensive resources, such as code and experimental details. Code is available at `https://github.com/xxxnell/how-do-vits-work`. Appendix A.1 provides the specifications of all models used in this work. Detailed experimental setup including hyperparameters and the structure of AlterNet are also available in Appendix A.1 and Appendix E. De-facto image datasets are used for all experiments as described in Appendix A.1.

References

Anonymous. Blur is an ensemble: Spatial smoothings to improve accuracy, uncertainty, and robustness. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=34mWBCWMxh9`. under review.

Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. *arXiv preprint arXiv:2102.08602*, 2021.

Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pretraining or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 1(2):3, 2021.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021.

Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, pp. 2933–2941, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pp. 2232–2241. PMLR, 2019.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.

Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021.

Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: Nngp and ntk for deep attention networks. In *International Conference on Machine Learning*, pp. 4376–4386. PMLR, 2020.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 2018.

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020.

Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic fisher explosion: Early phase fisher matrix impacts generalization. In *International Conference on Machine Learning*, pp. 4772–4784. PMLR, 2021.

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.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 31, 2018.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33, 2020.

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. *International Conference on Computer Vision (ICCV)*, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Shaokai Ye, Yuan He, and Hui Xue. Rethinking the design principles of robust vision transformer. *arXiv preprint arXiv:2105.07926*, 2021.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in Neural Information Processing Systems*, 32:14014–14024, 2019.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *arXiv preprint arXiv:2106.07998*, 2021.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021.

Namuk Park, Taekyu Lee, and Songkuk Kim. Vector quantized bayesian neural network inference for data streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9322–9330, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2021.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *arXiv preprint arXiv:2108.08810*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Mądry. How does batch normalization help optimization? In *Proceedings of the 32nd international conference on neural information processing systems*, pp. 2488–2498, 2018.

Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *arXiv preprint arXiv:2103.15670*, 2021.

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16519–16529, 2021.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Phil Wang. Implementation of vision transformer. https://github.com/lucidrains/vit-pytorch, 2021.

Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Combining ensembles and data augmentation can harm your calibration. In *International Conference on Learning Representations*, 2021.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers, 2021.

Baosong Yang, Longyue Wang, Derek F Wong, Lidia S Chao, and Zhaopeng Tu. Convolutional self-attention networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4040–4045, 2019.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 581–590. IEEE, 2020.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32:13276–13286, 2019.

Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. Rethinking token-mixing mlp for mlp-based vision backbone. *arXiv preprint arXiv:2106.14882*, 2021.

Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pp. 7324–7334. PMLR, 2019.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13001–13008, 2020.

A Experimental Details

This section provides experimental details, e.g., setups and background information.

A.1 Setups

We obtain the main experimental results with two kind of machines for CIFAR (Krizhevsky et al., 2009). The first one is the Intel Xeon W-2123 Processor, 32GB memory, and a single GeForce RTX 2080 Ti, and the other is four Intel Intel Broadwell CPUs, 15GB memory, and a single NVIDIA T4. For ImageNet (Russakovsky et al., 2015), we use AMD Ryzen Threadripper 3960X 24-Core Processor, 256GB memory, and four GeForce RTX 2080 Ti. Models are implemented in PyTorch (Paszke et al., 2019).

NNs are trained using categorical cross-entropy (NLL) loss and AdamW optimizer (Loshchilov & Hutter, 2019) with initial learning rate of 1.25×10^{-4} and weight decay of 5×10^{-2}. We also use cosine annealing scheduler (Loshchilov & Hutter, 2016). We train NNs for 300 epochs with batch size of 96 on CIFAR, and with batch size of 128 on ImageNet. The learning rate is gradually increased (Goyal et al., 2017) for 5 epochs. «««< HEAD Following Touvron et al. (2021), strong data augmentations, such as RandAugment (Cubuk et al., 2020), Random Erasing (Zhong et al., 2020), label smoothing (Szegedy et al., 2016), mixup (Zhang et al., 2018), and CutMix (Yun et al., 2019), are used for training. Stochastic depth (Huang et al., 2016) is also used to regularize NNs. Since this DeiT-style configuration (Touvron et al., 2021) is the de facto standard in ViT training (e.g., (Heo et al., 2021; Liu et al., 2021)), we believe the insights presented in this paper can be widely useful. See source code for detailed configurations.

We mainly report the performances of ResNet-50, ViT-Ti, PiT-Ti, and Swin-Ti. Their training throughputs on CIFAR-100 are 320, 434, 364, and 469 image/sec respectively, which are comparable to each other. In Figs. 5a and C.1a, we also report the predictive performance of ResNeXt-50 (Xie et al., 2017), Twins-S (Chu et al., 2021), and MLP-Mixer-Ti (Tolstikhin et al., 2021). In Fig. E.1, we additionally report the performance of ConViT-Ti (d'Ascoli et al., 2021), LeViT-128S (Graham et al., 2021), and CoaT-Lite-Ti (Xu et al., 2021). We use patch size of 2×2 for ViT and PiT on CIFAR. For Swin, we use patch size of 1×1 and window size of 4×4. We use patch size of 4×4 for ViT only in Fig. 7. We halve the depth of the ViT in Fig. C.5 and Fig. C.6 due to the memory limitation.

All models for CIFAR, and ResNet, ViT, and AlterNet for ImageNet are trained from scratch. We use pertained PiT and Swin from Wightman (2019) for ImageNet. The implementations of Vision Transformers are based on Wightman (2019) and Wang (2021).

For Hessian max eigenvalue spectrum (Anonymous, 2022), we use power iteration with batch size of 16 to produce the top-5 greatest eigenvalues. To this end, Yao et al. (2020)'s implementation is used. We modify the algorithm to calculate the eigenvalues with respect to ℓ_2 regularized NLL on augmented training datasets. In the strict sense, the weight decay is not ℓ_2 regularization, but we neglect the difference.

A.2 Background Information

Below, we describe the preliminaries and terms for the experiments.

Test error and training NLL. We report test error on clean test datasets and training NLL on augmented training datasets in experiments, e.g., Fig. 5 and Fig. C.1. NLL is an appropriate metric to evaluate convergence on training datasets because NN optimizes NLL. In addition, it is the most widely used as a proper scoring rule representing both accuracy and uncertainty. To represent predictive performance on test datasets, we use a well-known metric, error; however, NLL can also be used as a metric for performance evaluation on test datasets. Even if we use NLL as the metric, consistent conclusions are drawn.

If an additional inductive bias or a learning technique improves the performance of NNs, this is either a method to help the NNs learn *"strong representations"*, or a method to *"regularize"* the NNs. Improved—i.e, decreased—training NLLs suggest that the method helps NNs learn strong representations. Conversely, compromised training NLLs suggest that the method regularize NNs.

Likewise, we say that *"NN overfits training datasets"* when test error is compromised as training NLL is improved.

Learning trajectory in polar coordinates. We propose a visualization of learning trajectory in polar coordinates as shown in Fig. 1b. In this figure, $r = 0$ and $r = 1$ stand for optimized weight and initialized weight, respectively. The closer the θ is to $90°$, the further the weight is from the initialized weight, because we define θ based on cosine similarity. Since NN weights are high-dimensional, the visualization of learning trajectory in cartesian coordinates with two random vector bases appears to be a random walk (Li et al., 2018). To address this issue, we exploit the property that NN weights continuously move away from initializations during training phases. Li et al. (2018) used principal component analysis (PCA) to visualize the trajectory in cartesian coordinates; however, our visualization is easy to implement as it does not require any additional methods.

Hessian max eigenvalue spectra. Anonymous (2022) proposed *"Hessian max eigenvalue spectra"*, a feasible method to visualize Hessian eigenvalues of large sized NNs for real-world problems. It calculates tok-k Hessian eigenvalues by using power iteration mini-batch-wisely. Ghorbani et al. (2019) visualized Hessian eigenvalue spectra by using Lanczos quadrature algorithm for full batch. However, it is not feasible for practical NNs because the algorithm requires a lot of memory and computing resources.

A good loss landscape is a flat and convex loss landscape. Hessian eigenvalues represent flatness and convexity of losses. The magnitude of Hessian eigenvalues shows sharpness, and the presence of negative Hessian eigenvalues shows non-convexity. Based on these insights, we introduce *negative max eigenvalue proportion* (NEP, the lower the better) and *average of positive max eigenvalue* (APE, the lower the better) to quantitatively measure the non-convexity and the sharpness, respectively. For a Hessian max eigenvalue spectrum $p(\lambda)$, NEP is the proportion of negative eigenvalues $\int_{-\infty}^{0} p(\lambda)\, d\lambda$, and APE is the expected value of positive eigenvalues $\int_{0}^{\infty} \lambda\, p(\lambda)\, d\lambda \,/\, \int_{0}^{\infty} p(\lambda)\, d\lambda$. We use these metrics in Fig. C.5 and Fig. C.6.

Note that measuring loss landscapes and Hessian eigenvalues on clean dataset would give incorrect results, since NN training optimizes ℓ_2 regularized NLL on augmented training datasets—not NLL on clean training datasets. We visualize loss landscapes and Hessian eigenvalues with respect to ℓ_2 regularized NLL loss on augmented training datasets.

Fourier analysis of feature maps. We investigate feature maps in Fourier space to figure out that MSA is a low-pass filter as shown in Fig. 2, Fig. 8, and Fig. D.2. Fourier transform transforms feature maps into frequency domain. We represent the transformed feature maps on normalized frequency domain, so the highest-frequency components are at $f = \{-\pi, +\pi\}$, and the lowest-frequency components are at $f = 0$. We mainly report the amplitude ratio of high-frequency components and low-frequency components by using Δ log amplitude, the difference in log amplitude at $f = \pi$ and $f = 0$. Yin et al. (2019) also analyzed robustness of NNs from a Fourier perspective; however, they investigate input images—not feature maps—in Fourier spaces.

B MSA IS SPATIAL SMOOTHING

As mentioned in Section 1.1, spatial smoothings before subsampling layers help CNNs see better (Zhang, 2019; Anonymous, 2022). Anonymous (2022) showed that such improvement in performance is possible due to (Bayesian) spatial ensembles of feature map points. To this end, they used *ensemble average for proximate data points* (Park et al., 2021) which exploits *data uncertainty* (i.e., a distribution of feature maps) as well as model uncertainty (i.e., a posterior probability distribution of NN weights):

$$p(\boldsymbol{z}_j|\boldsymbol{x}_j, \mathcal{D}) \simeq \sum_i \pi(\boldsymbol{x}_i|\boldsymbol{x}_j)\, p(\boldsymbol{z}_j|\boldsymbol{x}_i, \boldsymbol{w}_i) \tag{2}$$

where $\pi(\boldsymbol{x}_i|\boldsymbol{x}_j)$ is normalized importance weight of a feature map point $\boldsymbol{x}_i$ with respect to another feature map $\boldsymbol{x}_j$, i.e., $\sum_i \pi(\boldsymbol{x}_i|\boldsymbol{x}_j) = 1$. The importance is defined as a similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $p(\boldsymbol{z}_j|\boldsymbol{x}_i, \boldsymbol{w}_i)$ and $p(\boldsymbol{z}_j|\boldsymbol{x}_j, \mathcal{D})$ are one NN prediction and output predictive distribution, respectively. $\boldsymbol{w}_i$ is a NN weight sample from posterior $p(\boldsymbol{w}|\mathcal{D})$ with respect to training dataset $\mathcal{D}$. Put shortly, Eq. (2) spatially complements a prediction with other predictions based on similarities. For instance, a 2 × 2 box blur ensembles four neighboring feature map points, each with ¼ of the same importance.

We note that the formulations for self-attention and data-complemented BNN are identical. The `Softmax` term and $\boldsymbol{V}$ in Eq. (1) exactly correspond to $\pi(\boldsymbol{x}_i|\boldsymbol{x}_j)$ and $p(\boldsymbol{z}_j|\boldsymbol{x}_i, \boldsymbol{w}_i)$ in Eq. (2). The weight samples in Eq. (2) is correspond to the multi-heads of MSAs (See also (Hron et al., 2020)).

Proposition B.1. *Self-attention is spatial smoothing of the values, i.e., the weighted average with self-attention importances is low-pass filter.*

Proof. Self-attention is convolution of the importance weights $\boldsymbol{A}$ (`Softmax` term in Eq. (1)) and the value predictions ($\boldsymbol{V}$ term in Eq. (1)):

$$\boldsymbol{A} * \boldsymbol{V} \tag{3}$$

Consider $\boldsymbol{V}$ averaged N times with $\boldsymbol{A}$:

$$\underbrace{\boldsymbol{A} * \cdots * \boldsymbol{A}}_{N \text{ times}} * \boldsymbol{V} \tag{4}$$

Since $\boldsymbol{A}$ is probability, $\boldsymbol{A} * \cdots * \boldsymbol{A}$ is the probability for the sum of N random variables from $\boldsymbol{A}$, i.e, $\boldsymbol{a} + \cdots + \boldsymbol{a} \sim \boldsymbol{A} * \cdots * \boldsymbol{A}$ where $\boldsymbol{a} \sim \boldsymbol{A}$. By definition, an operator is low-pass filter if and only if the high frequency component vanishes when the operator is applied infinitely. Since $\mathbf{Var}(\boldsymbol{a} + \cdots + \boldsymbol{a}) = N\,\mathbf{Var}(\boldsymbol{a})$ and $\mathcal{F}\left[\boldsymbol{A} * \cdots * \boldsymbol{A} * \boldsymbol{V}\right] = \mathcal{F}\left[\boldsymbol{A} * \cdots * \boldsymbol{A}\right] \mathcal{F}\left[\boldsymbol{V}\right]$ where $\mathcal{F}$ is Fourier transform, the high-frequency component of the $\boldsymbol{A} * \cdots * \boldsymbol{A} * \boldsymbol{V}$ vanishes when N goes to infinity. Therefore, $\boldsymbol{A}$ is low-pass filter. □

Likewise, the properties of spatial smoothing are the same as those of MSAs (Anonymous, 2022): ① Spatial smoothing improves the accuracy of CNNs. In addition, spatial smoothing predicts well-calibrated uncertainty. ② Spatial smoothing is robust against MC dropout (which is equivalent to image occlusion), data corruption, and adversarial attacks, and particularly robust against high-frequency noise. ③ Spatial smoothing layers closer to the output layer significantly improves the predictive performance.

Taking all these observations together, *we provide an explanation of how MSAs work by addressing them as a general form of spatial smoothing or an implementation of data-complemented BNN.* Spatial smoothing improves performance in the following ways (Anonymous, 2022): ❶ Spatial smoothing helps in NN optimization by flattening the loss landscapes. Even a small 2×2 blur filter significantly improves performance. ❷ Spatial smoothing is a low-pass filter. CNNs are vulnerable to high-frequency noise, but spatial smoothing improves the robustness against it by reducing high-frequency noise significantly. ❸ Spatial smoothing is effective when applied at the end of a stage because it aggregates all the feature maps that are transformed in various ways. The main paper shows that these mechanisms are maintained for MSAs.

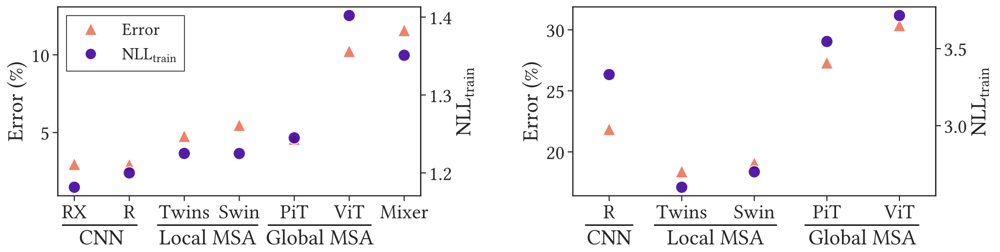
C ViTs From a Loss Landscape Perspective

This section provides further explanations of the analysis in Section 2.

The lower the NLL on training dataset, the lower the error on test dataset. Figure 5a demonstrates that low training NLL results in low test error on CIFAR-100. We observe the same pattern on CIFAR-10 and ImageNet. Figure C.1 shows the result.

In small data regimes, such as CIFAR-10 (Fig. C.1a) and CIFAR-100 (Fig. 5a), both the error and the $\text{NLL}_{\text{train}}$ of ViTs are inferior to those of CNNs. It suggests that the cons of MSA outweigh the pros. As discussed in Fig. 4, ViTs suffers from the non-convex losses, and the non-convex loss disturb ViT optimization.

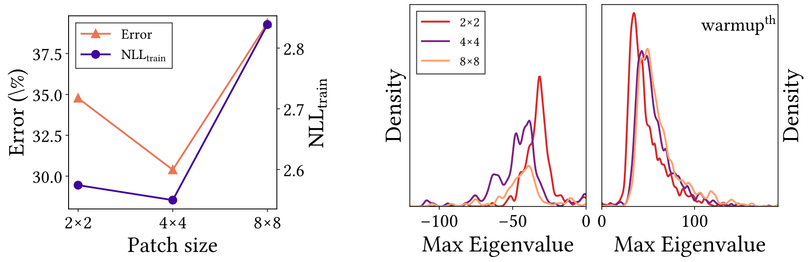
In large data regimes, such as ImageNet (Fig. C.1b), both the error and the $\text{NLL}_{\text{train}}$ of ViTs with local MSAs are superior to those of CNNs. Since large datasets convexify the loss function as discussed in Fig. 4, the pros of MSAs outweigh the cons. Therefore, MSAs help NNs learn strong representations by flattening the loss landscapes.



(a) CIFAR-10

(b) ImageNet

Figure C.1: **The lower the training NLL, the lower the test error.** "R" is ResNet and "RX" is ResNeXt. *Left:* In small data regimes, such as CIFAR-10 and CIFAR-100 (Fig. 5a), the cons of MSAs outweigh the pros, i.e., the non-convex losses disturb ViT optimization. *Right:* Large datasets convexify the loss functions. Therefore, the pros of MSAs outweigh the cons in large data regimes, i.e., MSAs help NN learn strong representations by flattening the loss landscapes.



(a) Error and $\text{NLL}_{\text{train}}$ of ViT for patch size

(b) Negative and positive Hessian max eigenvalue spectra in early phase of training

Figure C.3: **A small patch size does not guarantee better performance**. We analyze ViTs with three embedded patch sizes: 2×2, 4×4, and 8×8. Note that every MSAs have global receptive fields. *Left:* A large size of patch harms the performance as we would expect, and surprisingly, a small size of patch also harms it. *Right:* A small patch size, or weak inductive bias, allows the negative eigenvalues. This is another evidence that weak inductive bias hinders NN optimization. On the other hand, MSA with a small patch size reduces the magnitude of eigenvalues because it ensembles a large number of feature map points. The performance is optimized when these two effects are balanced.

Rigorous discussion on regularization of CNN's inductive bias. In Fig. 5a, we compare similarly sized models, such as ResNet-50 and ViT-Ti. Through the comparison, we show that a weak inductive bias hinders NN training, and CNN's inductive biases—Conv's inductive bias and multi-stage architecture—help NN learn strong representations. However, the inductive biases of CNN gives better test accuracy for the same training NLL, i.e., Convs somewhat regularize NNs. We analyze two comparable models in terms of $\text{NLL}_{\text{train}}$ on CIFAR-100. The $\text{NLL}_{\text{train}}$ of ResNet-18, a model smaller than ResNet-50, is 2.31 and the error is 22.0%. The $\text{NLL}_{\text{train}}$ of ViT-S, a larger than ViT-Ti, is 2.17 and the error is 30.4%. In summary, the inductive biases of CNNs improve accuracy for similar training NLL.

Most of the improvement come from multi-stage architecture, not the inductive bias of Conv. The $\text{NLL}_{\text{train}}$ of the PiT-Ti, multi-stage ViT-Ti, is 2.29 and the error is 24.1 %. The accuracy of PiT is only 1.9 percent point lower than that of ResNet. In addition, the small receptive field also regularize ViT. See Fig. 7.

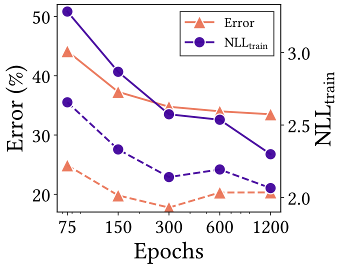
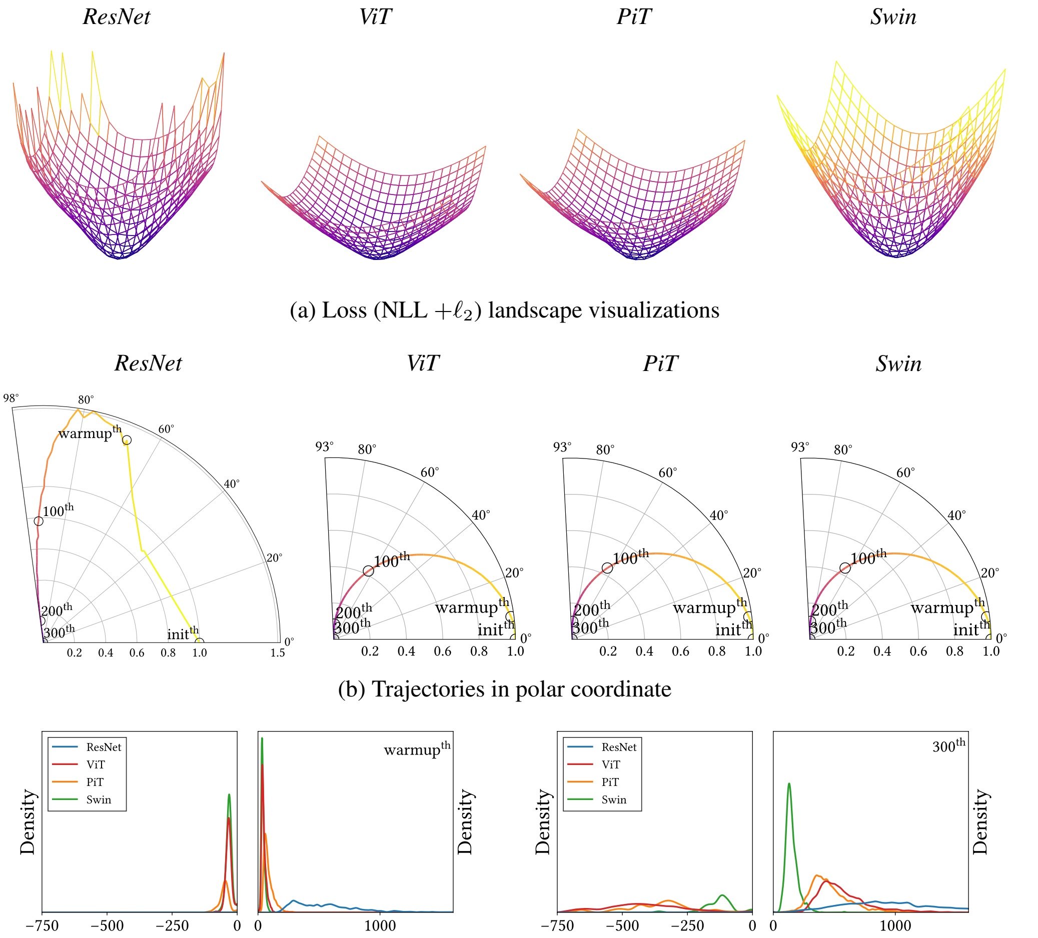


Figure C.2: **A large number of epochs does not make ViT overfit the training dataset of CIFAR**. Solid line is the predictive performance of ViT and dashed line is that of ResNet.

ViT does not overfit small training dataset even with a large number of epochs. Figure 5b shows that ViT does not overfit small training datasets such as CIFAR. We observe the same phenomenon in ViT training with a large number of epochs.



(a) Loss (NLL $+\ell_2$) landscape visualizations

(b) Trajectories in polar coordinate

(c) Negative and positive Hessian max eigenvalue spectra in early phase (*left*) and late phase (*right*) of training

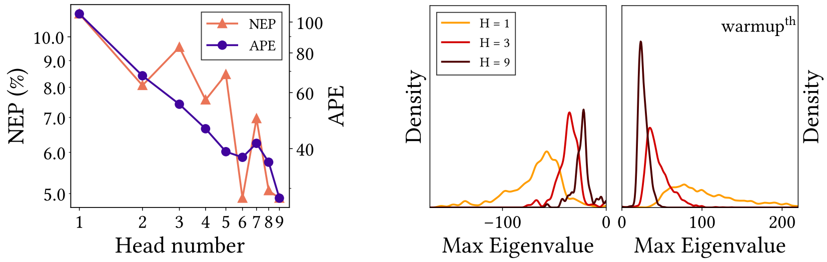
Figure C.4: **Multi-stage architecture (in PiT) and local MSA (in Swin) also flatten the loss landscapes.** *Top:* PiT has a flatter loss landscape than ViT near the optimum. Swin has an almost perfect smooth parabolic loss landscape (due to ℓ_2 regularization). *Middle:* All three ViTs converge to the optima with smooth trajectories. *Bottom:* Multi-stage in PiT suppresses negative Hessian eigenvalues. Local MSA in Swin allows the negative eigenvalues, but significantly reduces the magnitude of eigenvalues.

In Fig. C.2, we train ViT and ResNet for 75, 150, 300, 600, and 1200 epochs. The experimental results show that the $\text{NLL}_{\text{train}}$ and the error decrease as the epoch increases. However, the predictive performances of ViT is inferior to those of ResNet across a whole range of epochs.

A smaller patch size does not always imply better results. ViT splits the image into multiple patches. The smaller the patch size, the greater the flexibility of expression and the weaker the inductive bias. By analyzing ViT with three patch sizes—2×2, 4×4, and 8×8—we again demonstrate that weak inductive bias disturbs NN optimization.

Figure C.3a shows the error on test dataset and NLL on the training dataset of CIFAR-100. As expected, large patch size harms the performance on both test and training datasets. Surprisingly, a small patch size also harms the performance. An appropriate patch size helps ViT learn strong representations, not regularizes ViT.

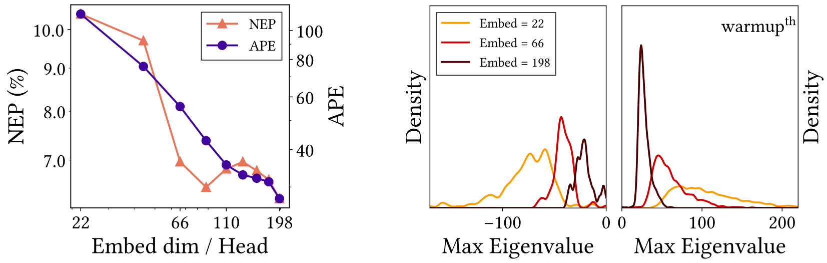
The Hessian max eigenvalue spectra in Fig. C.3b explain this observation. The results reveal that a small patch size reduces the magnitude of Hessian eigenvalues but allows negative Hessian eigenvalues. In other words, weak inductive bias makes the loss landscape flat but non-convex. A large patch



(a) NEP (non-convexity) and APE (sharpness) for the head number

(b) Hessian negative and positive max eigenvalue spectra in early phase of training

Figure C.5: **Multi-heads convexify and flatten the loss landscape**. *Left:* We use negative max eigenvalue proportion (NEP) and average of positive max eigenvalue (APE) to quantify the convexity and flatness of the loss landscape, respectively. As the number of heads increases, the loss landscape becomes convex and flatter. *Right:* Hessian max eigenvalue spectra also show that multi-head suppress negative eigenvalues and reduce the magnitude of eigenvalues.



(a) NEP (non-convexity) and APE (sharpness) for the embedding dim

(b) Hessian negative and positive max eigenvalue spectra in early phase of training

Figure C.6: **High embedding dimensions per head convexify and flatten the loss landscape**. *Left:* As the number of embedding dimensions per head increases, the loss landscape becomes convex and flatter. *Right:* Hessian max eigenvalue spectra also show that the high embedding dimensions suppress negative eigenvalues and reduce the magnitude of eigenvalues as in Fig. C.5.

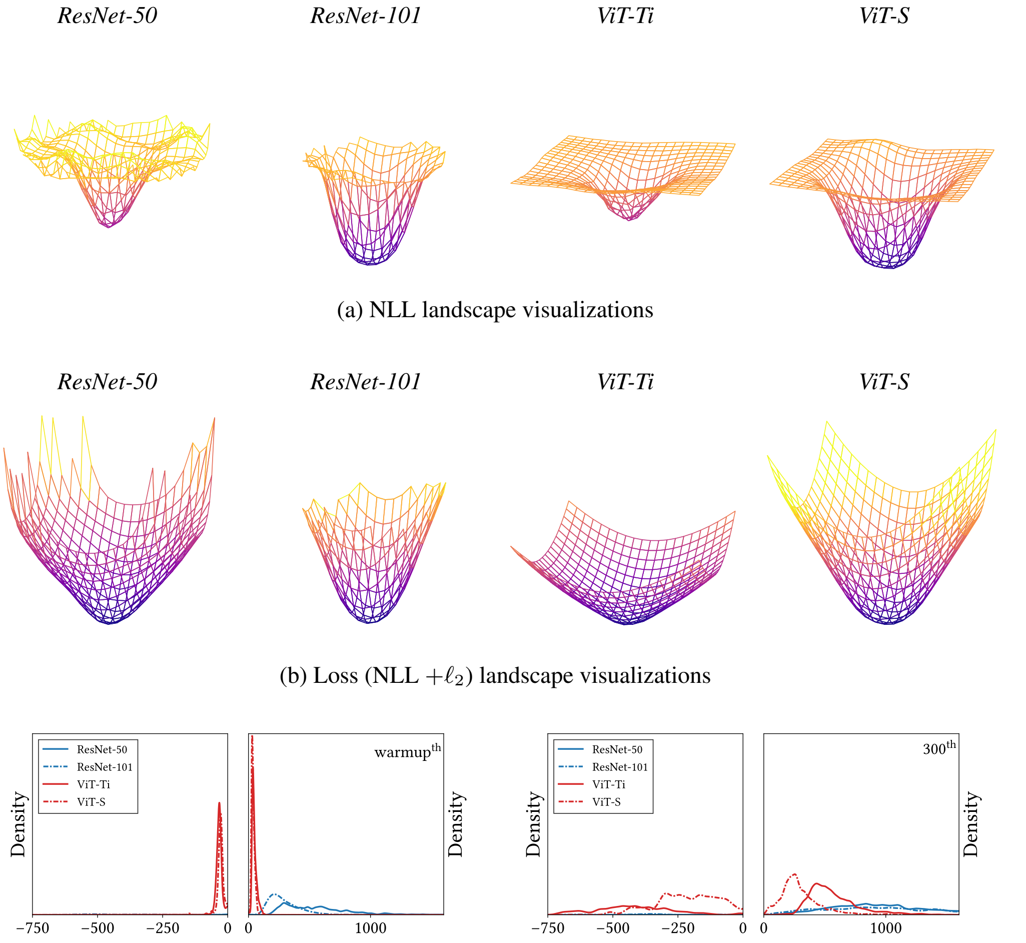
size suppresses negative eigenvalues. On the other hand, it not only limits the model expression but also sharpens the loss landscape. The performance is optimized when these two effects are balanced.

Multi-stage architecture in PiT and local MSA in Swin also flatten the loss landscapes. As explained in Fig. 1, MSA smoothens the loss landscape. Similarly, multi-stage architecture in PiT and local MSA in Swin also help NN learn strong representations by smoothing the loss landscapes.

Figure C.4 provides loss landscape visualizations, learning trajectories, and Hessian eigenvalue spectra of ResNet, ViT, PiT, and Swin. Figure C.4a visualizes the global geometries of the loss functions. The loss landscape of PiT is flatter than that of ViT near the optimum. Since Swin has more parameters than ViT and PiT, ℓ_2 regularization leads the loss landscape. All the loss surfaces of ViTs are smoother than that of ResNet. Figure C.4b demonstrates that all the ViTs converge to the optima with smooth trajectories. These support that MSA smoothens the loss landscape. Figure C.4c shows the local geometries of the loss function by representing Hessian eigenvalues. In the early phase of training, multi-stage in PiT helps training by suppressing negative Hessian eigenvalues. Local MSA in Swin allows the negative eigenvalues, but significantly reduces the magnitude of eigenvalues. Moreover, the magnitude of Swin's Hessian eigenvalue does not significantly increases in the late phase of learning.

Lack of heads may lead to non-convex loss. Neural tangent kernel (NTK) (Jacot et al., 2018) theoretically implies that the loss landscape of ViT is convex and flat when the number of heads or the number of embedding dimensions per head goes to infinity (Hron et al., 2020; Liu et al., 2020). In particular, from Liu et al. (2020), we easily derive the following proposition:

$$|\mathcal{O}(^1/\sqrt{m}) - c_\infty| \leq ||H|| \leq \mathcal{O}(^1/\sqrt{m}) + c_\infty \tag{5}$$



(a) NLL landscape visualizations

(b) Loss (NLL $+\ell_2$) landscape visualizations

(c) Negative and positive Hessian max eigenvalue spectra in early phase (*left*) and late phase (*right*) of training

Figure C.7: **The loss landscapes of large models.** ResNet-50 and ResNet-101 are comparable to ViT-Ti and ViT-S, respectively. *Top:* Large models explore low NLLs. *Middle:* The loss landscape visualizations show that the global geometry of large models is sharp. *Bottom:* The Hessian eigenvalues of large models are smaller than that of small models. It suggests that large models have flat local geometry in early phase of training, and the flat loss helps NN learn strong representations. In late phase of training, large ViT has flat minimum while large ResNet has sharp minimum.

where $||H||$ is the Hessian spectral norm, m is the number of heads or the number of emgedding dimensions per head, and c_∞ is a small constant. Therefore, in practical situations, insufficient heads may cause non-convex and sharp loss.

In Fig. C.5, we empirically show that a lot of heads in MSA convexify and flatten the loss landscape (c.f. Michel et al. (2019)). In this experiment, we use NEP and APE to measure the non-convexity and the sharpness as introduced in Appendix A.2. The results show that both NEP and APE decrease as the number of heads increases. Likewise, Fig. C.6 shows that high embedding dimensions per head also convexify and flatten the loss. The exponents of APE are -0.562 and -0.796 for the number of heads and the embedding dimensions, which are in close agreement with the value predicted by the theory of $-1/2$.

A large model has a flat loss in an early phase of training. Figure C.7 analyze the loss landscapes of large models, such as ResNet-101 and ViT-S. As shown in Fig. C.7a, large models explore low NLLs. These can be a surprising result, because the loss landscapes of the large model is globally sharp as shown in Fig. C.7b.

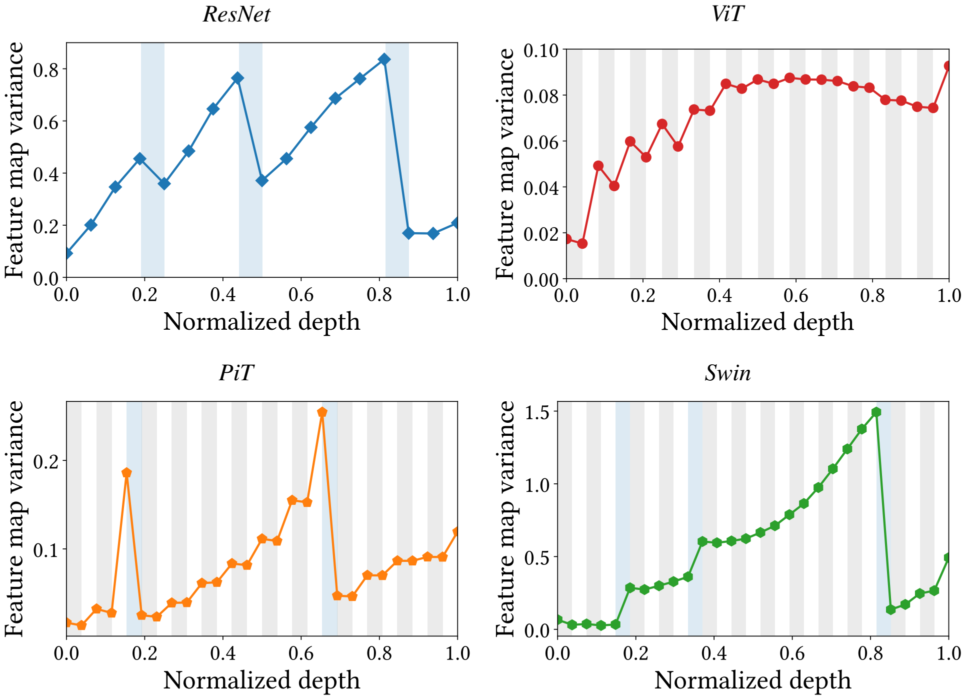


Figure D.1: **MSAs in PiT and Swin also reduce feature map variance** except in 3rd stage of Swin. White, gray, and blue areas are Conv/MLP, MSA, and subsampling layers, respectively.

Hessian eigenvalue spectra in Fig. C.7c solve the problem: The Hessian eigenvalues of large models are smaller than that of small models in early phase of training. It indicates that large models have flat loss functions locally.

D ViTs From a Feature Map Perspective

This section provides further explanations of the analysis in Section 3 and Section 4.1.

MSAs in PiT and Swin also ensemble feature maps. In Fig. 9, we show that MSAs in ViT reduce feature map variance. We observe same pattern in PiT and Swin. Figure D.1 demonstrates that MSAs in PiT and Swin also reduce the feature map variance, suggesting that they also ensemble feature maps. One exception is the 3rd stage in Swin. In the early of this stage, MSAs suppresses the increase in variance, but at the end of the stage it does not suppress it.

MSAs in PiT and Swin are also low-pass filters. As discussed in Fig. 8, MSAs in ViTs are low-pass filters, while MLPs in ViT and Convs in ResNet are high-pass filters. Likewise, we demonstrate that MSAs in PiT and Swin are also low-pass filters.

Figure D.2 shows the relative log amplitude of Fourier transformed feature maps. As in the case of ViT, MSAs in PiT and Swin generally decrease the amplitude of high-frequency signals; in contrast, MLPs increases the amplitude.

Multi-stage ViTs have the block structures. The feature map similarities of CNNs shows a block structure (Nguyen et al., 2021). As Raghu et al. (2021) figured out, ViT has uniform representations across all layers. By investigating multi-stage ViTs, we demonstrate that subsampling layers create the characteristic block structure of the representation. See Fig. D.3.

Conv in early phase of stages and MSA in late phase of stages play an important role. Figure D.4 shows the results of lesion study for ResNet and ViTs. In this experiment, we remove one 3×3 Conv layer from bottleneck block of ResNet, and one MSA or MLP block from Swin. We observe

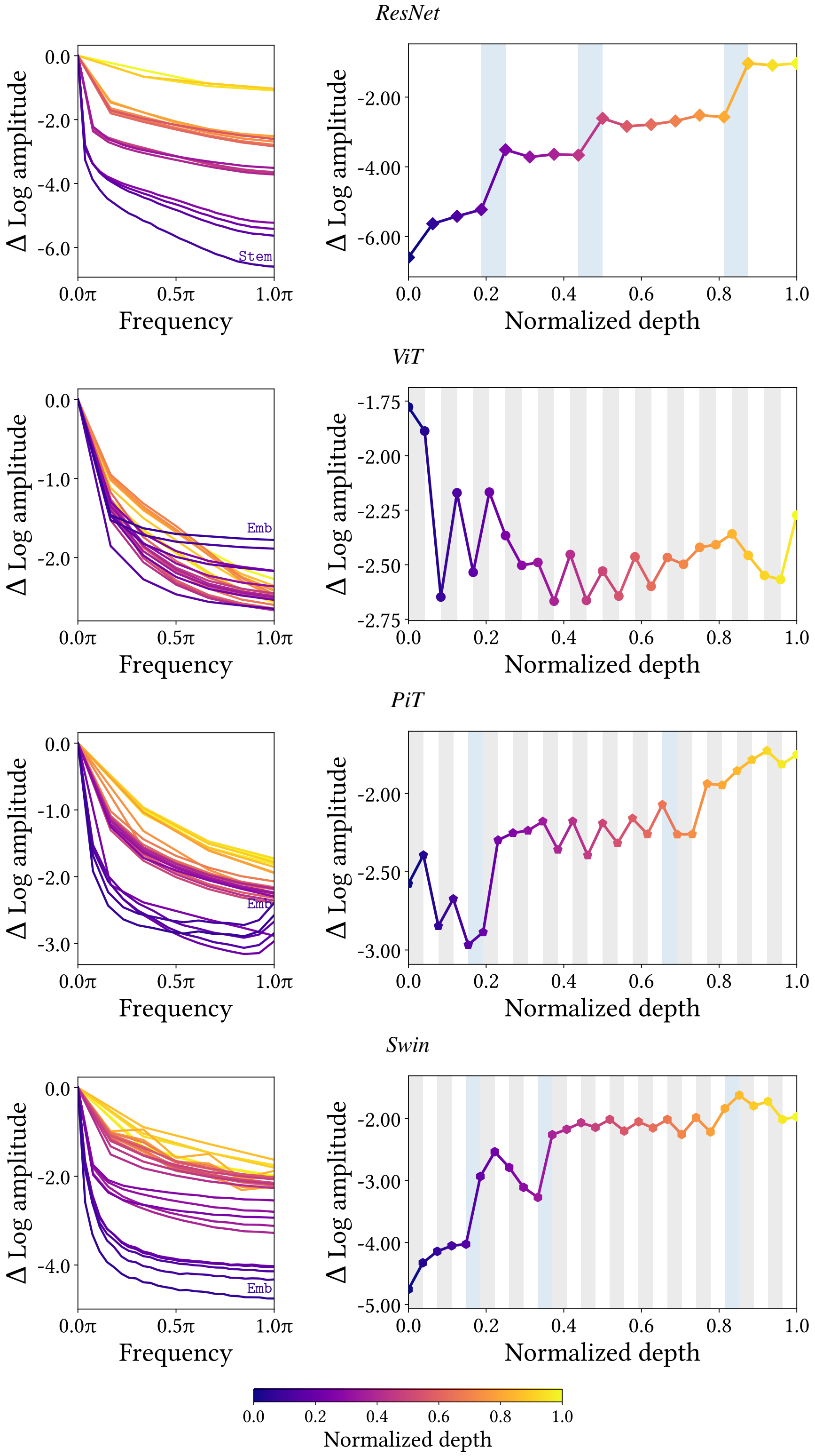


Figure D.2: **MSAs in PiT and Swin also reduce high-frequency signals**. *Left:* Δ log amplitude of Fourier transformed feature map. We only provide the diagonal components. *Right:* The high-frequency (1.0π) Δ log amplitude. White, gray, and blue areas are Conv/MLP, MSA, and subsampling layers, respectively.

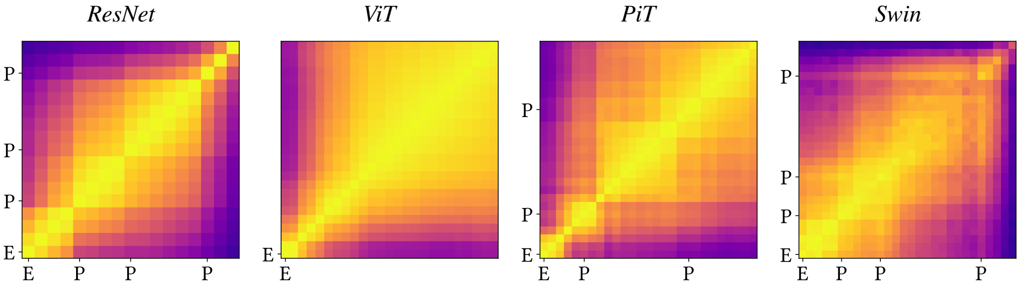


Figure D.3: **Multi-stage ViTs have block structures in representational similarities**. We observe the block structures in all multi-stage NNs, namely, ResNet, PiT, and Swin. "E" is stem/embedding and "P" is pooling (subsampling) layer.

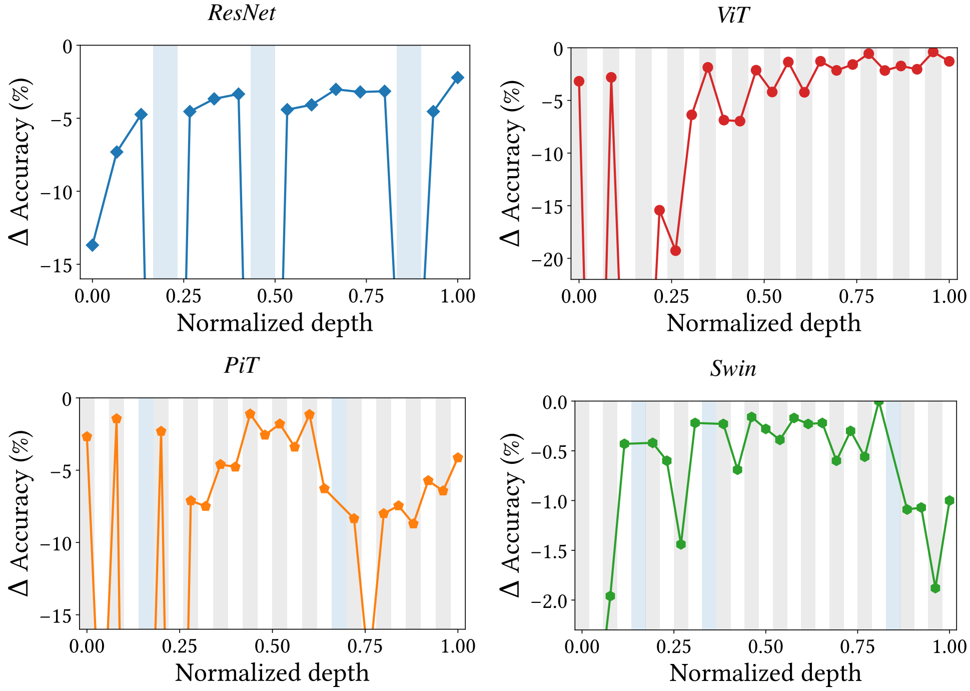


Figure D.4: **Lesion study shows that Conv in early phase of stages and MSA in late phase of stages are important for prediction**. We measure the decrease in accuracy after removing one unit from the trained model. In this experiment, we observe that the accuracy changes periodically. White, gray, and blue areas are Conv/MLP, MSA, and subsampling layers, respectively.

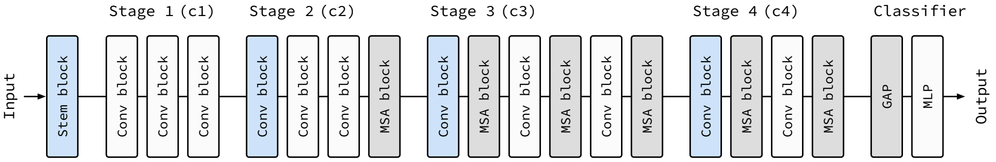


Figure D.5: **Detailed architecture of Alter-ResNet-50 for ImageNet-1K.** White, gray, and blue blocks mean Conv, MSA, and subsampling blocks. This model alternately replaces Conv blocks with MSA blocks from the end of stages. Following Swin, MSAs in stages 1 to 4 use 3, 6, 12, and 24 heads, respectively. We use 6 MSA blocks for ImageNet since large amount of data alleviates the drawback of MSA. See Fig. 11 to compare with the model for CIFAR-100.

consistent results for all models: Removing Convs in early phase of stages and removing MSAs in late phase of stages significantly harm the accuracy. As a result, the accuracy varies periodically.

E Extended Information of AlterNet

This section provides further informations on AlterNet.

Detailed architecture of AlterNet. Section 4 introduces AlterNet to harmonize Convs with MSAs. Since most MSAs take pre-activation arrangement, pre-activation ResNet is used as a baseline for consistency. We add one CNN block to the last stage of ResNet to make the number of blocks even. A local MSA with relative positional encoding from Swin is used for AlterNet. However, for simplicity of implementation, we do not implement detailed techniques, such as a cyclic shift and layer-specific initialization. For CIFAR, the patch size of the MSA is 1×1 and the window size is 4×4. If all Conv blocks are alternately replaced with MSA, AlterNet becomes a Swin-like model.

In order to achieve better performance, NNs should strongly aggregate feature maps at the end of models as discussed in Section 3 and Section 4. To this end, AlterNet use 3, 6, 12, 24 heads for MSA in each stage.

The computational costs of Conv block and MSA block are almost the same. The training throughput of Alter-ResNet-50 is 473 image/sec on CIFAR-100, which is 20% faster than that of pre-activation ResNet-50.

The optimal number of MSAs depends on the model and dataset, so we empirically determine the number of MSAs as shown in Fig. 12a. A large dataset allows a large number of MSAs. For ImageNet, we use 6 MSAs as in Fig. D.5, because large datasets alleviate the shortcoming of MSA.

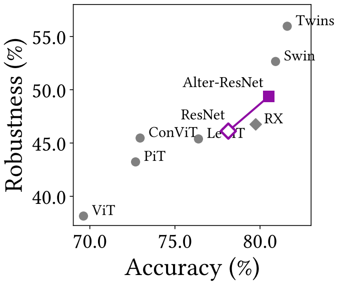


Figure E.1: **MSA with the appropriate build-up rules significantly improves ResNet** on ImageNet. Robustness is mean accuracy on ImageNet-C. “RX” is ResNeXt.

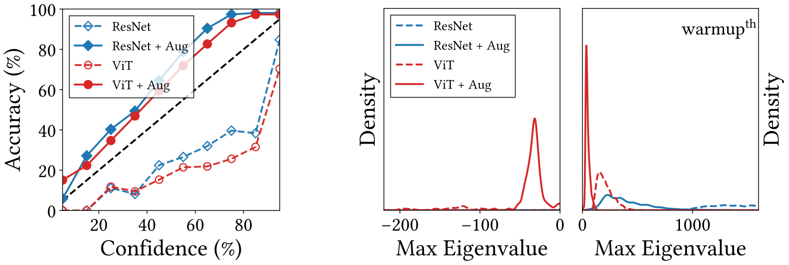
MSAs improve the performance of CNN on ImageNet. Since MSA complements Conv, MSA improves the predictive performance of CNN with the appropriate build-up rules as shown in Section 4.1. Figure E.1 reports the accuracy and robustness—mean accuracy on ImageNet-C—of CNNs and ViTs on ImageNet-1K. Since ImageNet is large dataset, a number of ViTs outperform CNNs. MSAs with the build-up rules significantly improves ResNet, and the predictive performance of AlterNet is on par with that of Swin in terms of accuracy without heavy modifications, e.g., the shifted windowing scheme (Liu et al., 2021). AlterNet is easy-to-implement and has a strong potential for future improvements. In addition, the build-up rules not only improve ResNet, but also other NNs. We observe that the build-up rule also improved the performance of vanilla post-activation ResNet and ResNeXt; we do not report the results because of the visualization simplicity.

F Distinctive Properties of Data Augmentation

This section empirically demonstrates that NN training with data augmentation is different from training on large datasets. We compare DeiT-style strong data augmentation with weak data augmentation, i.e., resize and crop.

F.1 Data Augmentation Can Harm Uncertainty Calibration

Figure F.1a shows the reliability diagrams of NNs with and without strong augment on CIFAR-100. In the results, both ResNet and ViT without data augmentation predict overconfident results. We show that data augmentation makes the predictive results under-confident (cf. Wen et al. (2021)). These are unexpected results because the predictions are not under-confident without data augmentation on large datasets such as ImageNet. We leave a detailed investigation for future work.



(a) Reliability diagram

(b) Hessian max eigenvalue spectra

Figure F.1: **Distinctive properties of strong data augmentation.** "Aug" stands for strong data augmentation. *Left:* Strong data augmentation makes predictions underconfident on CIFAR-100. We observe the same phenomenon on ImageNet-1K. *Right:* Strong data augmentation significantly reduces the magnitude of Hessian max eigenvalues. It means that the data augmentation helps NNs converge to better optima by flattening the loss landscapes. On the other hand, the data augmentation allows a lot of negative Hessian eigenvalues, i.e., it makes the loss non-convex.

F.2 Data Augmentation Reduces the Magnitude of Hessian Eigenvalues

How does data augmentation help MSA avoid overfitting on training dataset and achieve better accuracy on test dataset? Figure F.1b shows the Hessian max eigenvalue spectra of NNs with and without strong data augmentation. First of all, data augmentation reduces the magnitude of Hessian eigenvalues, i.e., data augmentation flattens the loss landscapes in an early phase of training. The flat loss leads to better generalization. On the other hand, data augmentation allows a lot of negative Hessian eigenvalues, i.e., data augmentation makes the loss non-convex. This prevents NN from converging to a low loss on training datasets. It is clearly different from the effect of large datasets discussed in Fig. 4—large datasets convexify the loss landscapes. We leave a detailed investigation for future work.