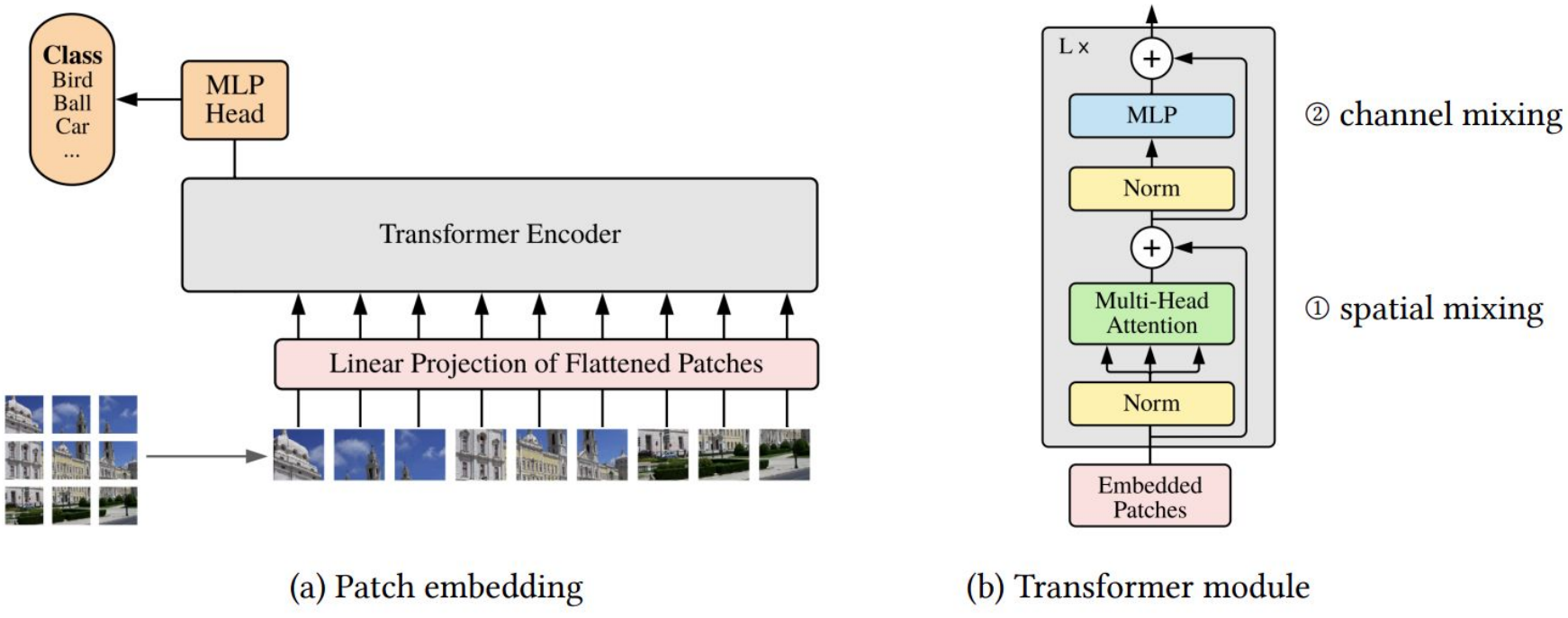
How do Vision Transformer work?

Hoàng Minh Thanh (21C11029)
Nguyễn Thành Thái (21C110026)
Nguyễn Trần Khánh Nguyên (21C11017)
Lê Trọng Tài (21C11022)
Hoàng Ngọc Thạch (21C11025)

Team 15

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Vision Transfomer



(a) Patch embedding

(b) Transformer module

Overview of ViTs. ***Left***: Vision Transformers (ViTs) split image into multiple patches. This patch embedding corresponds to the `stem` module of CNNs. ***Right***: A module of ViTs (a Transformer) consists of one multi-head self-attention (MSA) block and one MLP block.

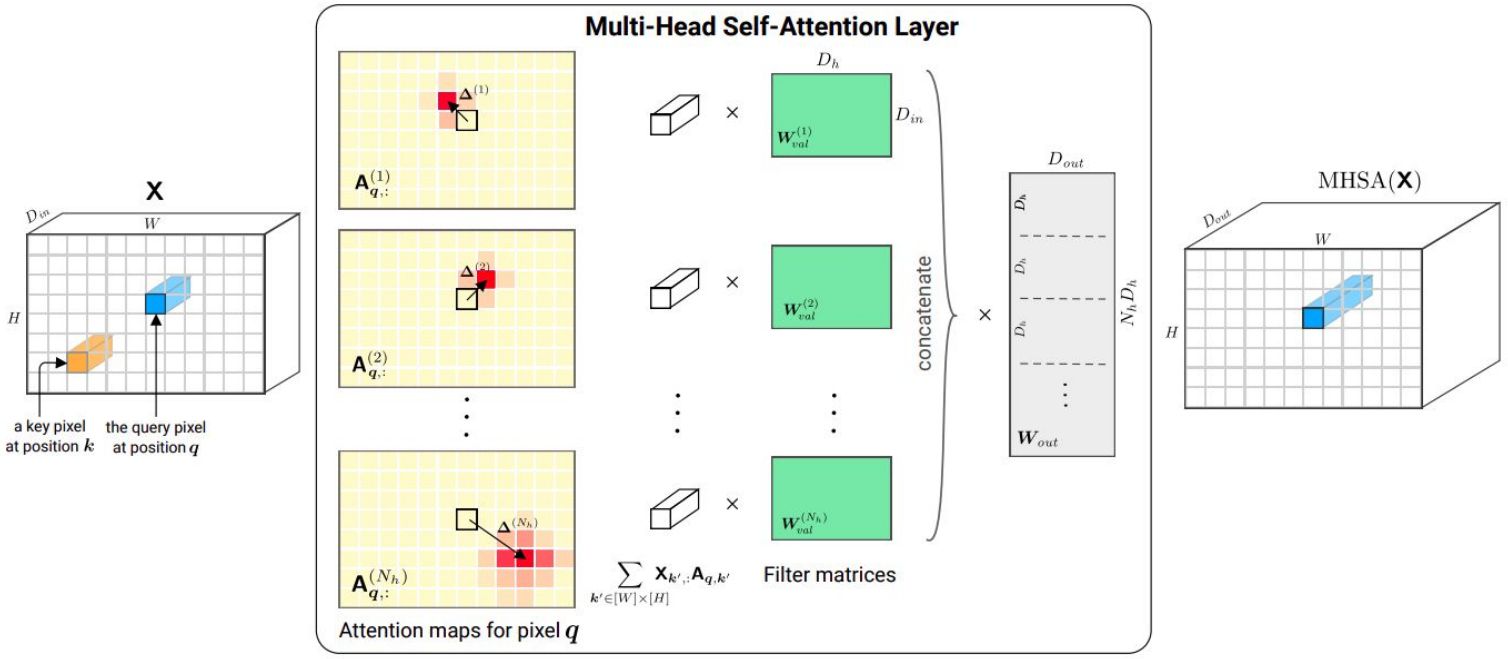
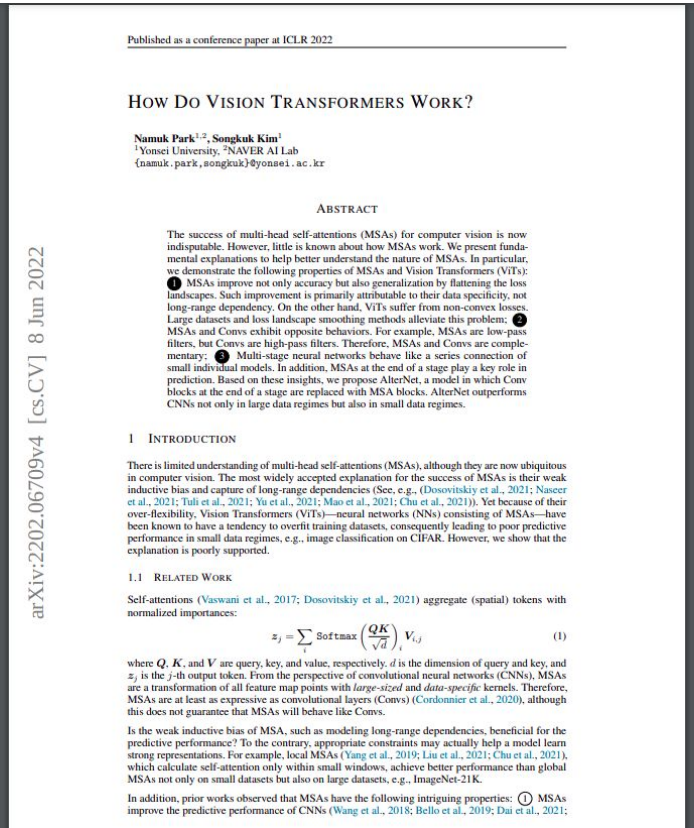


Illustration of a ***"multi-head self-attention"*** (MSA) layer applied to a tensor image ***X***.
We represent self-attention maps (red boxes)—similarities between a query pixel and a key pixel.

How do Vision Transformer work?

1. What properties of MSAs do we need to improve optimization?
2. Do Self-Attentions act like Conv?
3. How can we harmonize MSAs with Convs?



Published as a conference paper at ICLR 2022

How Do Vision Transformers Work?

Namuk Park[1,2], **Songkuk Kim**[1]
[1]Yonsei University, [2]NAVER AI Lab
{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 fundamental 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:

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

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are query, key, and value, respectively. d is the dimension of query and key, and 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* and *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 (Wang et al., 2018; Bello et al., 2019; Dai et al., 2021;

Content

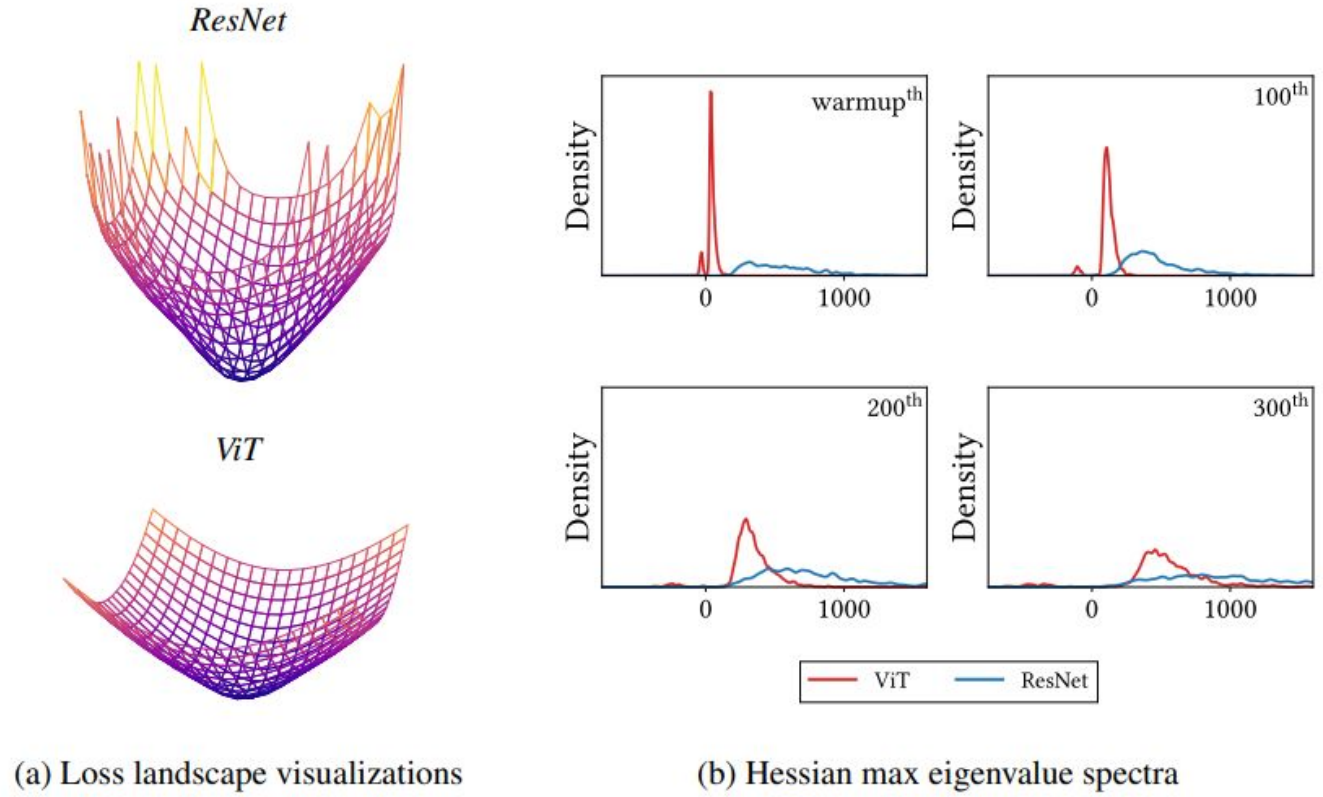
1. What properties of MSAs do we need to improve optimization?
2. Do Self-Attentions act like Conv?
3. How can we harmonize MSAs with Convs?
4. Experiment
5. Demo

1. What properties of MSAs do we need to improve optimization?

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

1. What properties of MSAs do we need to improve optimization?

- **MSAs làm phẳng không gian hàm loss**

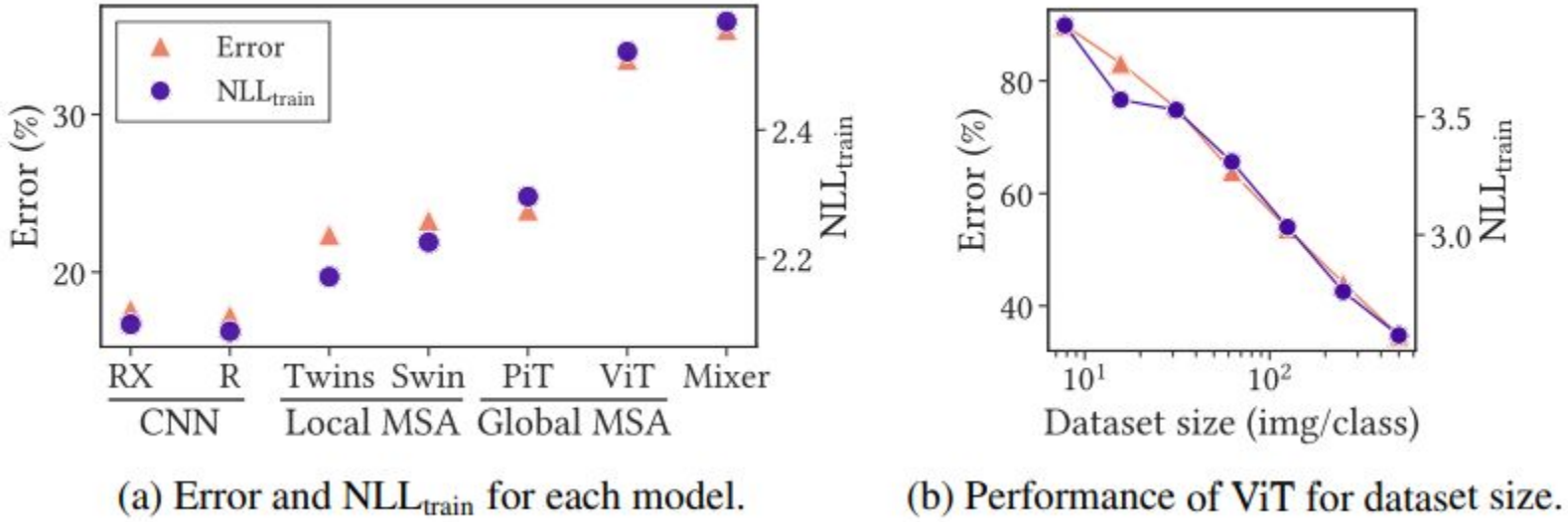


(a) Loss landscape visualizations (b) Hessian max eigenvalue spectra

Hình 1.1 Ảnh so sánh loss landscape của ResNet và ViT model

1. What properties of MSAs do we need to improve optimization?

- **ViT không bị overfit trên dataset nhỏ**

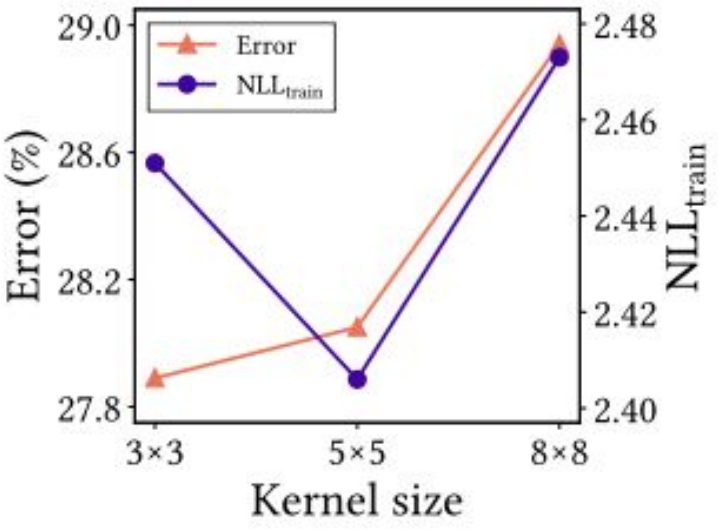


Hình 1.2 Biểu đồ so sánh error và train loss của các model trên dataset CIFAR-100 [1]

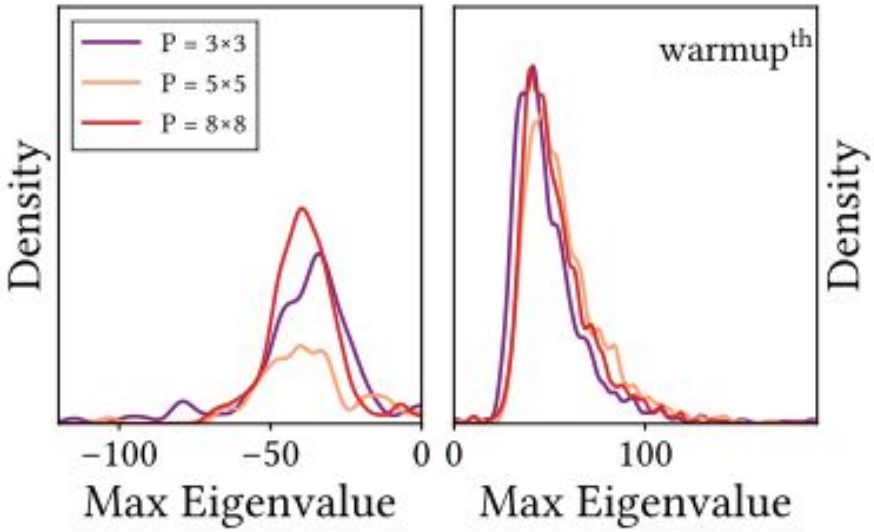
[1]: (CIFAR-100) Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

1. What properties of MSAs do we need to improve optimization?

- **MSAs có tính data specificity**



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



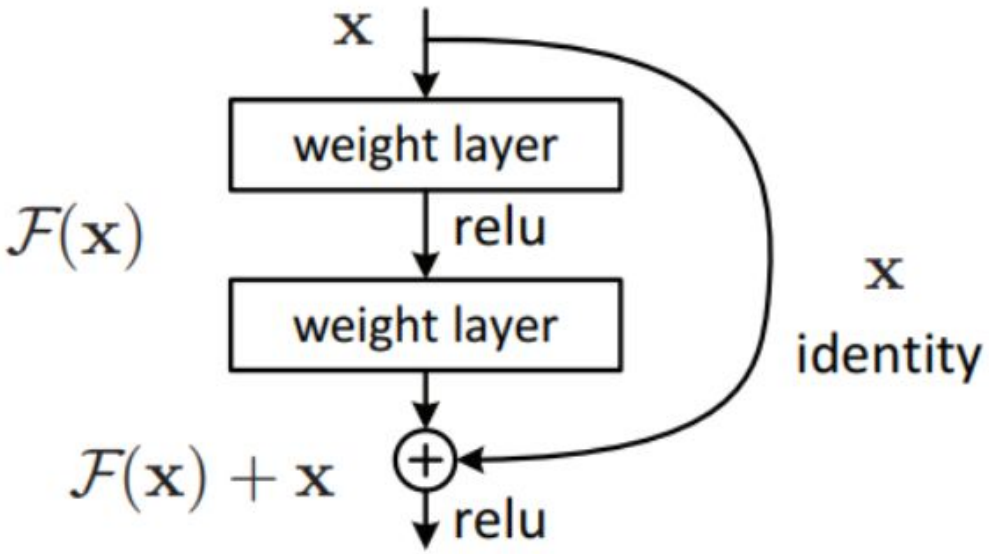
(b) Hessian negative and positive max eigenvalue spectra in early phase of training

Hình 1.3 Biểu đồ so sánh error của ViT model với các kernel size khác nhau

2. Do Self-Attentions act like Convs?

2. Do Self-Attentions Act Like Convs?

- MSA (Multi-head self-attention)
- Convs (CNN); Resnet



Skip (Shortcut) connection

2. Do Self-Attentions Act Like Convs?

- MSAs and Convs are complementary
- MSAs are **low-pass** filters, but Convs are **high-pass** filters
- MSAs aggregate feature maps, but Convs do not

Low and High pass filters

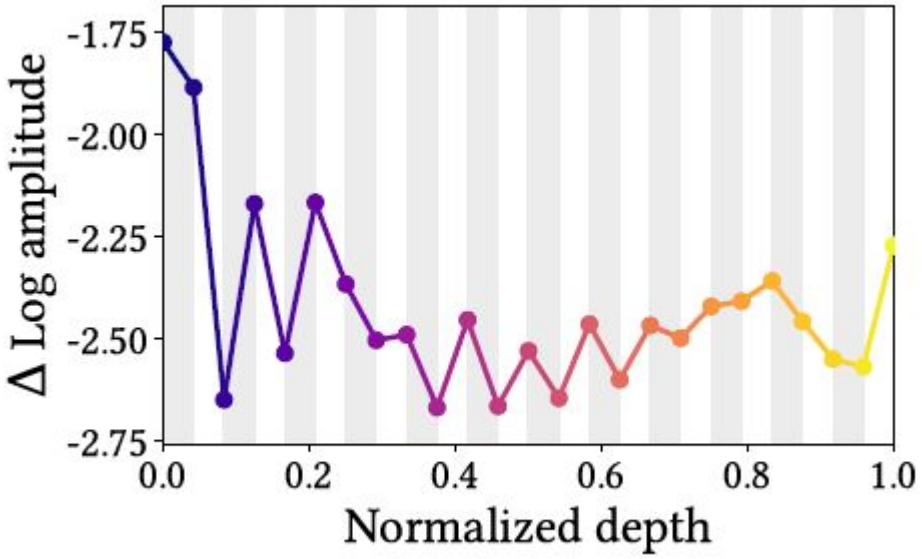
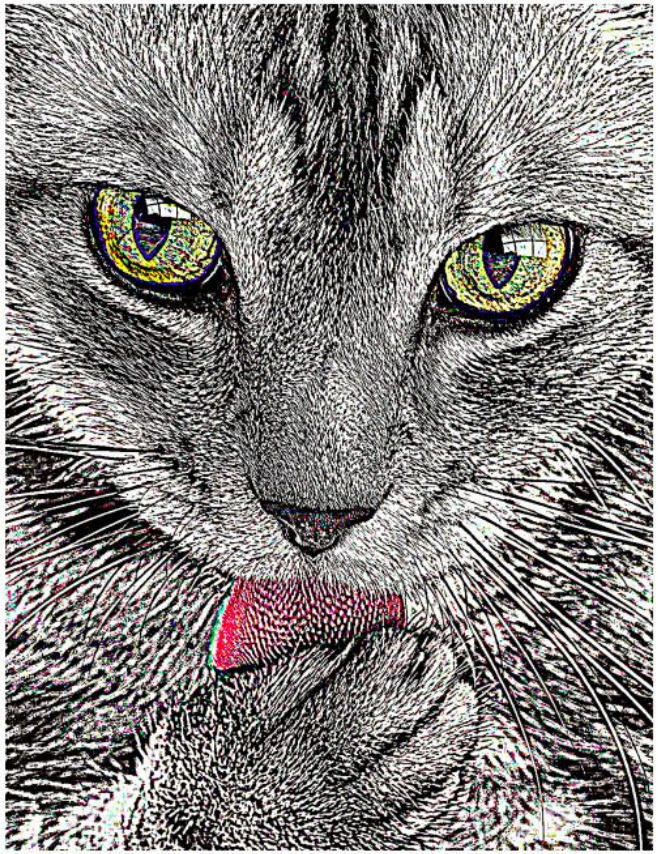


Figure 2.1 MSAs (gray area) generally reduce the high-frequency component of feature map, and MLPs (white area) amplify it.

- Images in frequency domain
- Fourier transform is used to convert them into frequency domain
- MSAs > low-pass > **Smooth**
- Convs > high-pass > **Edge**

Low and High pass filters

MSA is shape-biased.



Conv is texture-biased.

Figure 2.2 Low-frequency signals and the high-frequency signals each correspond to the shape and the texture of images

MSAs aggregate feature maps, but Convs do not

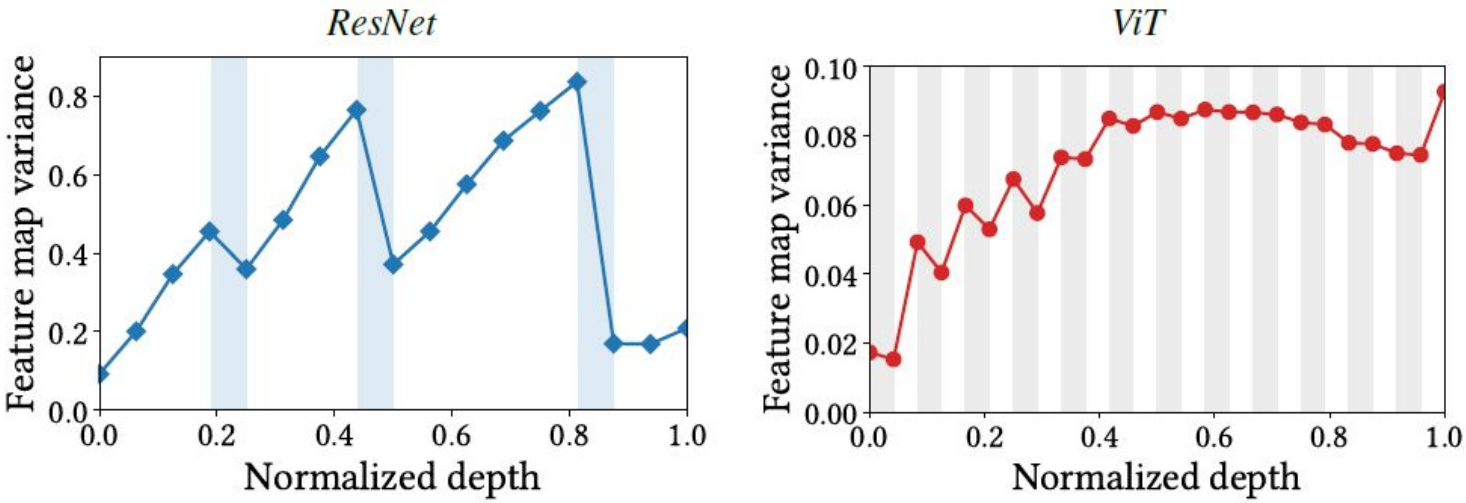


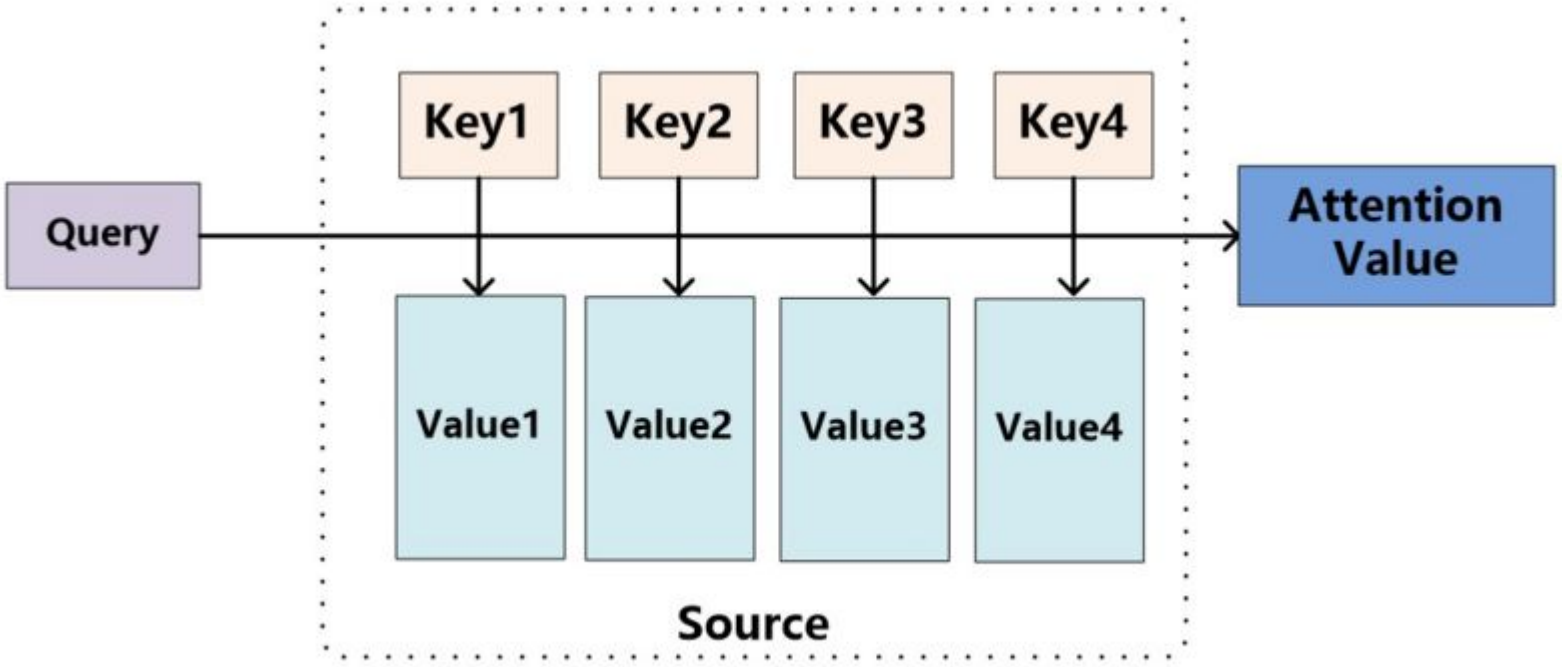
Figure 2.3 This figure indicates that MSAs in ViT tend to reduce the variance; conversely, Convs in ResNet and MLPs in ViT increase it.

- Since MSAs average feature maps, they will reduce variance of feature map points
- Measuring the variance of feature maps from NN layers to demonstrate

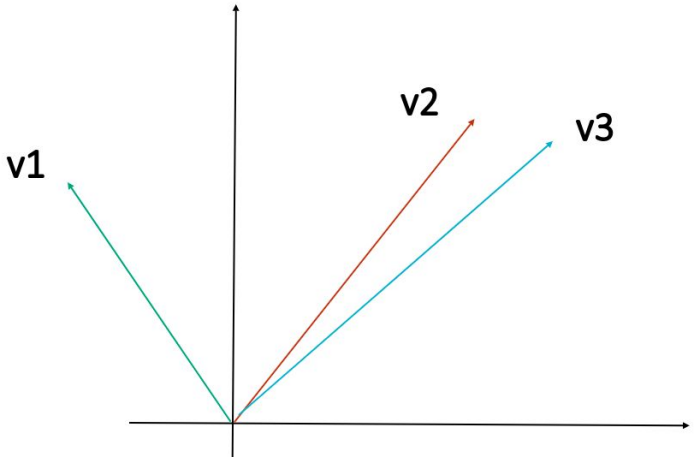
MSAs aggregate feature maps, but Convs do not

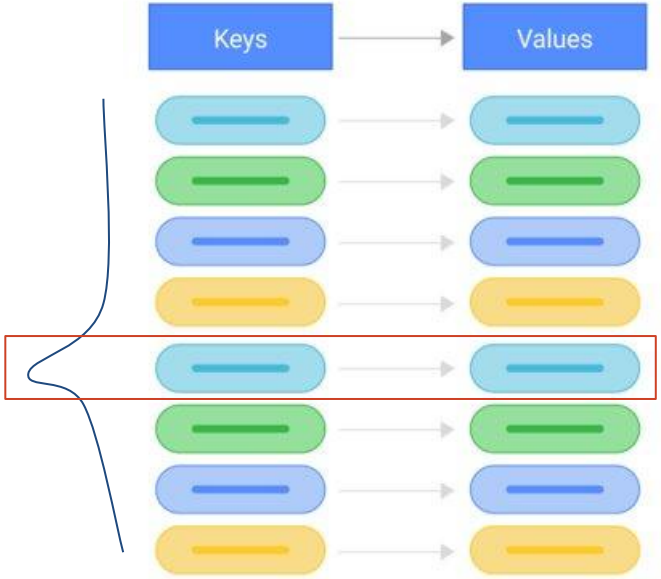
- Can improve the predictive performance of ResNet by inserting MSAs at the end of each stage
- Furthermore, also can improve the performance by using MSAs with a large number of heads in late stages.

3. How can we harmonize MSAs with Convs?


Key1
Key2
Key3
Key4
Query
Attention Value
Value1
Value2
Value3
Value4
Source

Self-Attention?

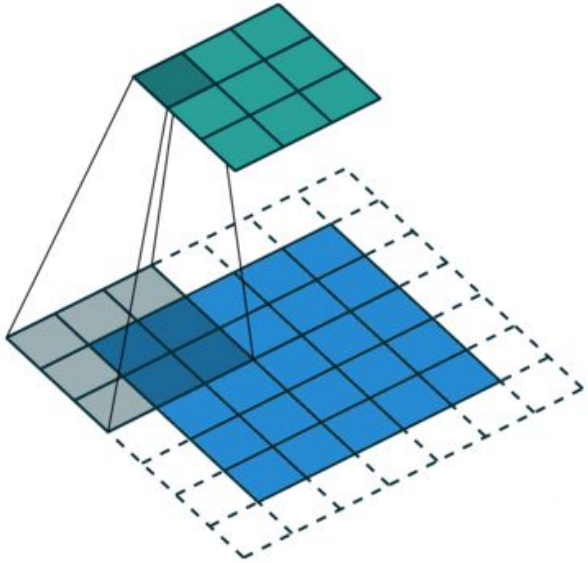




$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

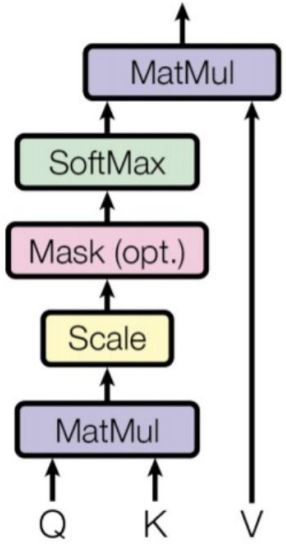
Convolution vs Self-attention

Convolution



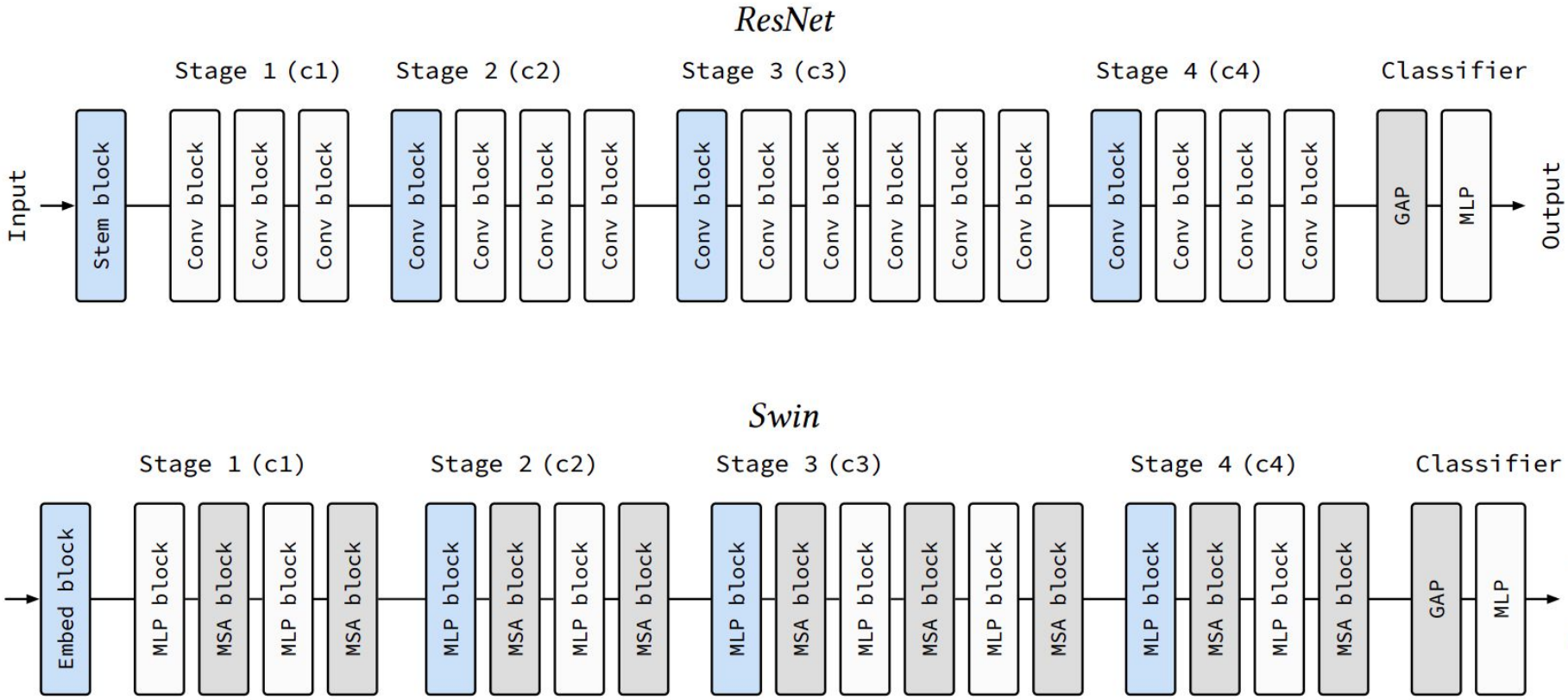
$$y[n] = x[n] ** h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

Self-Attention

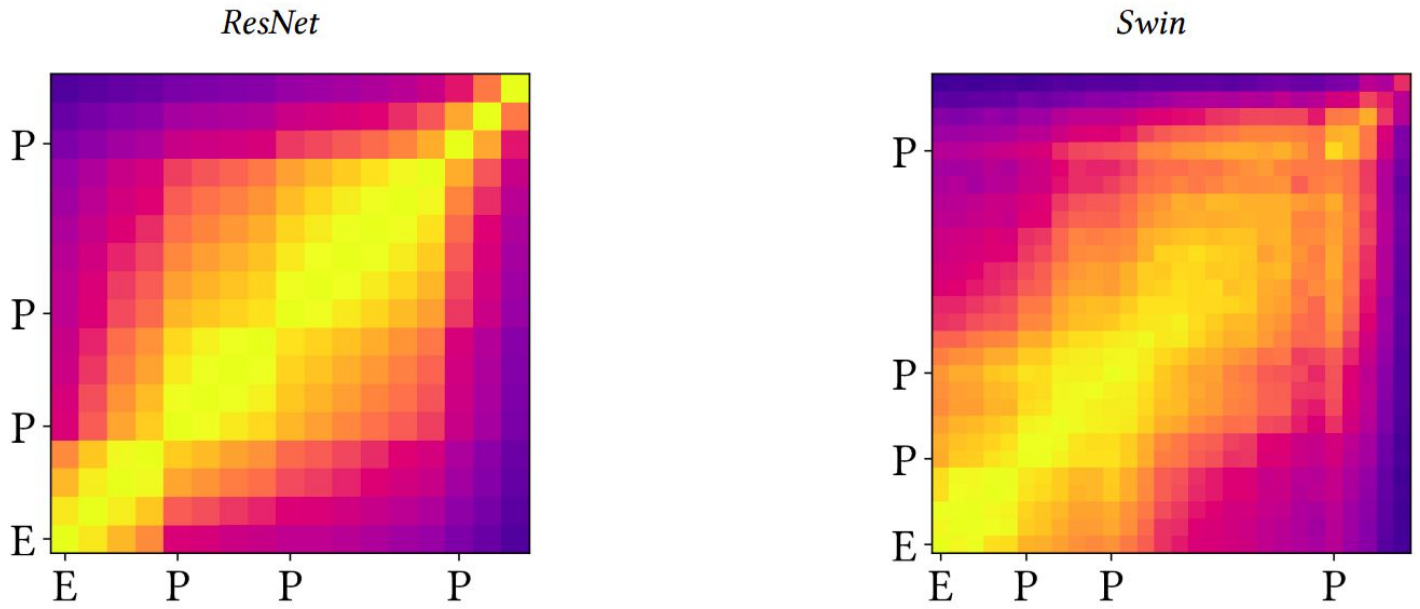


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Architecture of Swin Transformer

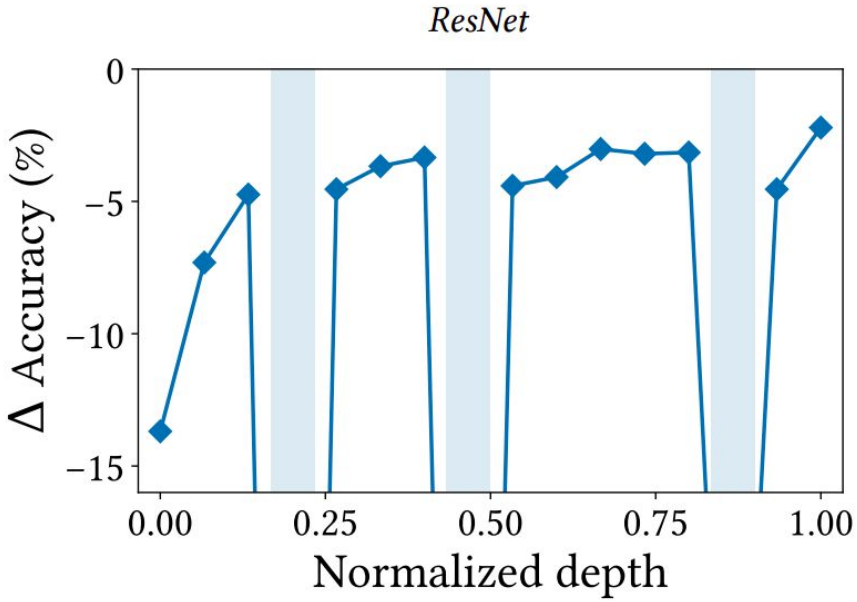


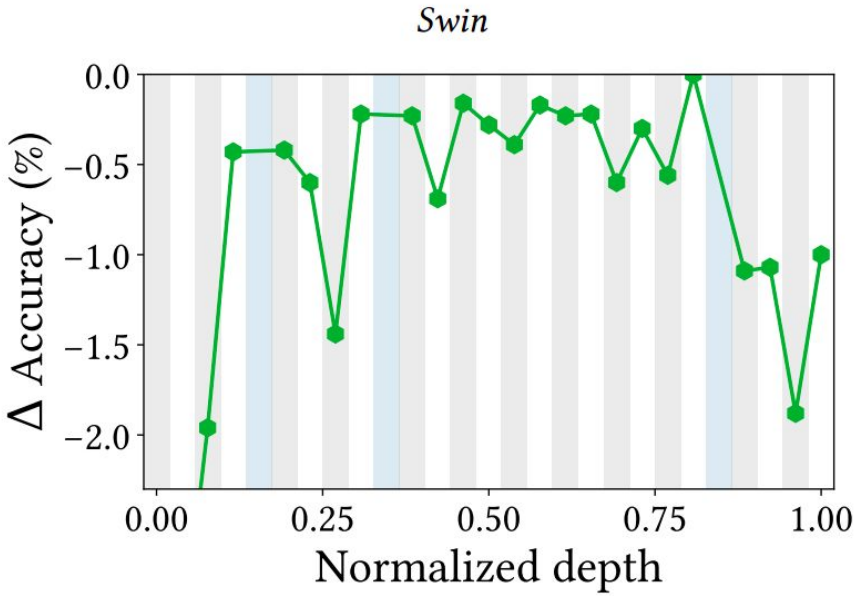
Multi-Stage NNs Have Representational Block Structures



The feature map similarities show the block structure of ResNet and Swin. "E" stands for stem/ embedding and "P" for pooling (sub- sampling) layer. Since vanilla ViT does not have this structure the structure is an intrinsic characteristic of multi-stage architectures.

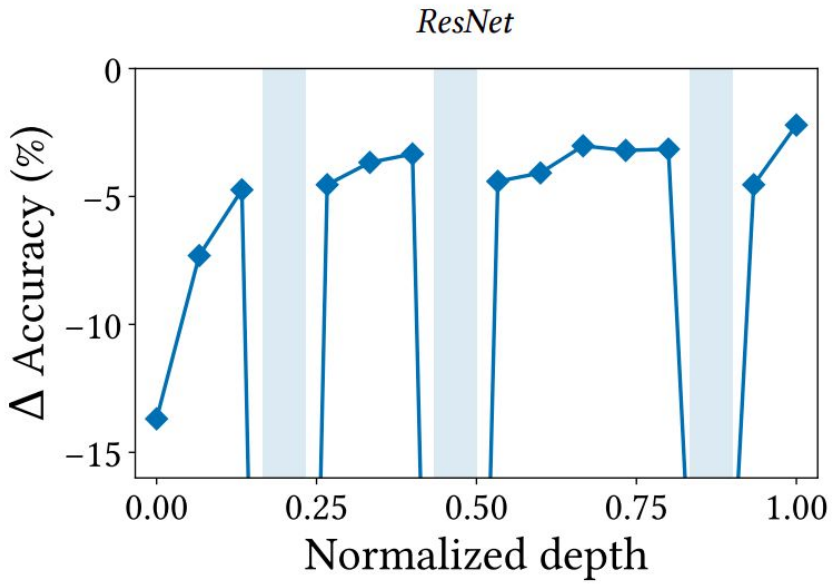
Sự suy giảm độ chính xác có tính chu kỳ

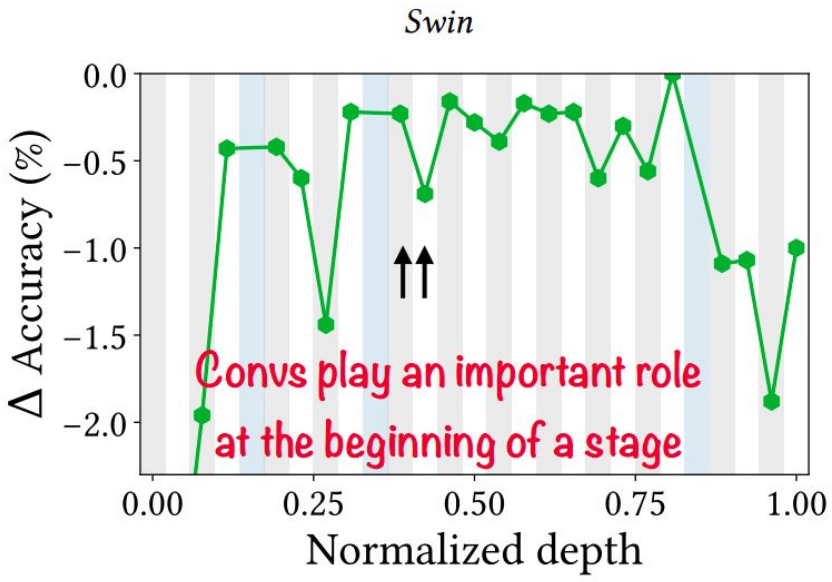




MSAs closer to the end of a stage significantly improve the performance. We measure decrease in accuracy after removing one unit from the trained model. Accuracy changes periodically, and this period is one stage.

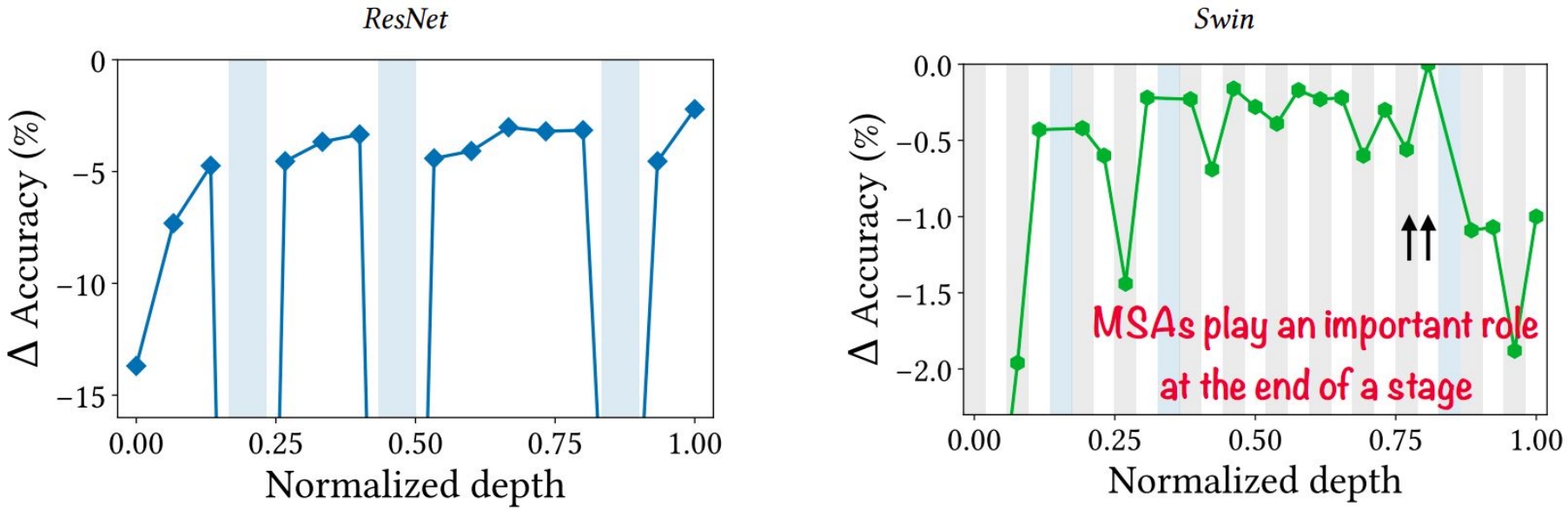
Sự suy giảm độ chính xác có tính chu kỳ





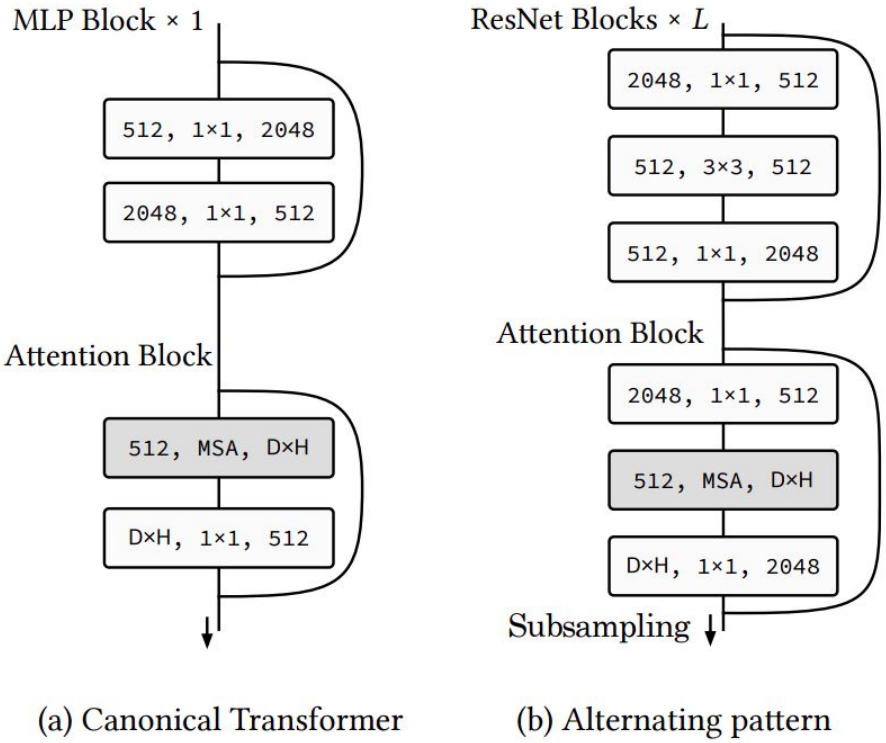
MSAs closer to the end of a stage significantly improve the performance. We measure decrease in accuracy after removing one unit from the trained model. Accuracy changes periodically, and this period is one stage.

Sự suy giảm độ chính xác có tính chu kỳ



MSAs closer to the end of a stage significantly improve the performance. We measure decrease in accuracy after removing one unit from the trained model. Accuracy changes periodically, and this period is one stage.

Alternating Pattern of Convs and MSAs

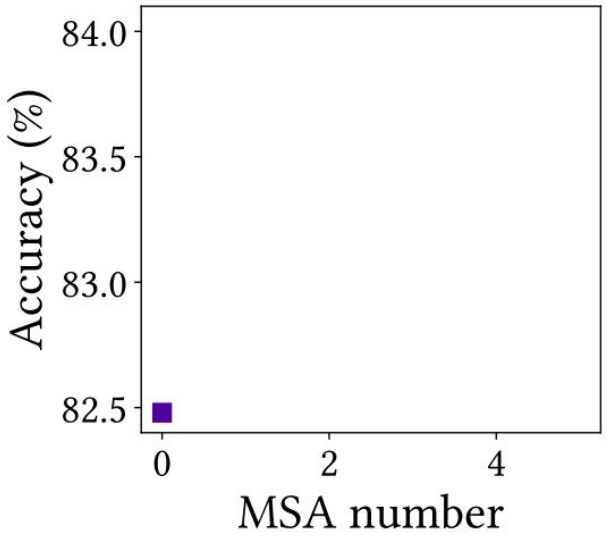


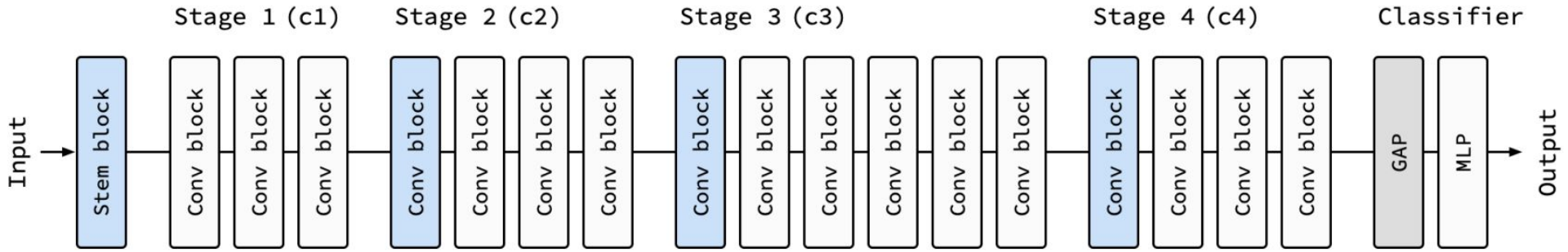
(a) Canonical Transformer (b) Alternating pattern

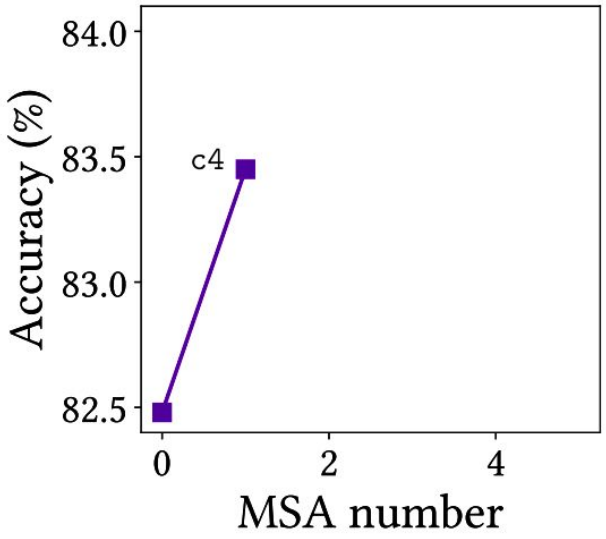
Alternating pattern replace Conv blocks at the end of a stage with MSA blocks. ***Left***: The stages of ViTs consist of repetitions of canonical Transformers. "D" is the hidden dimension and "H" is the number of heads. ***Right***: The stages using alternating pattern consists of a number of CNN blocks and anMSA block

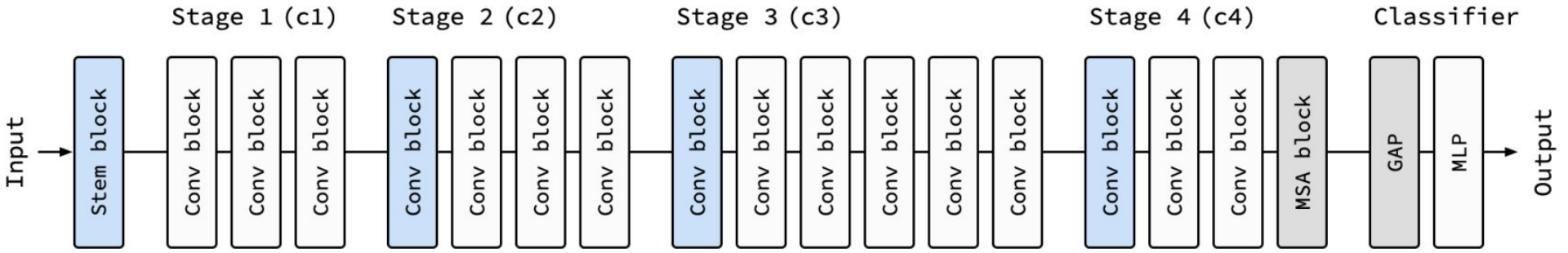
Cách áp dụng MSA cho CNN để xây dựng mô hình khác

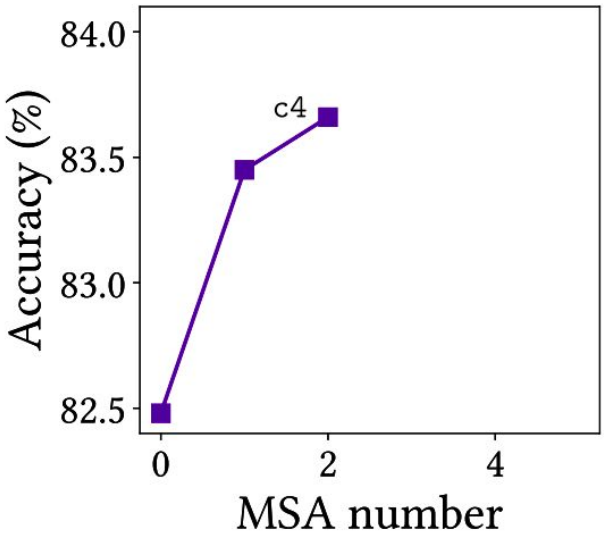
- Thay thế luân phiên các khối Conv bằng các khối MSA từ phần cuối của mô hình CNN ban đầu.
- Nếu khối MSA được thêm vào không cải thiện hiệu suất dự đoán, hãy thay thế khối Conv nằm ở cuối giai đoạn trước đó bằng khối MSA.
- Sử dụng nhiều head hơn và nhiều hidden dimensions hơn cho các khối MSA trong giai đoạn cuối.

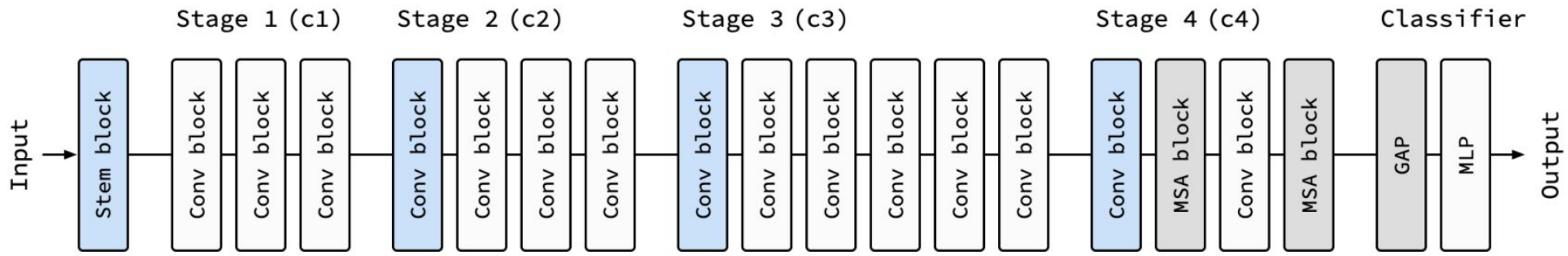

84.0
83.5
83.0
82.5
Accuracy (%)
0
2
4
MSA number

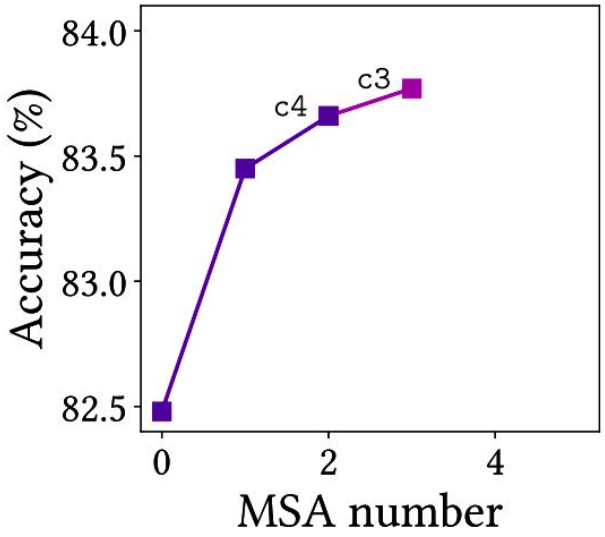

Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
GAP
MLP
Output

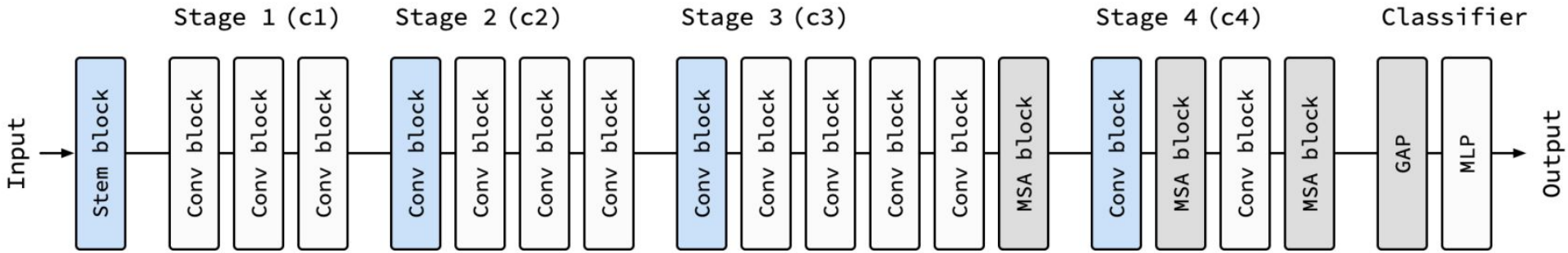

84.0
83.5
83.0
82.5
Accuracy (%)
c4
0
2
4
MSA number

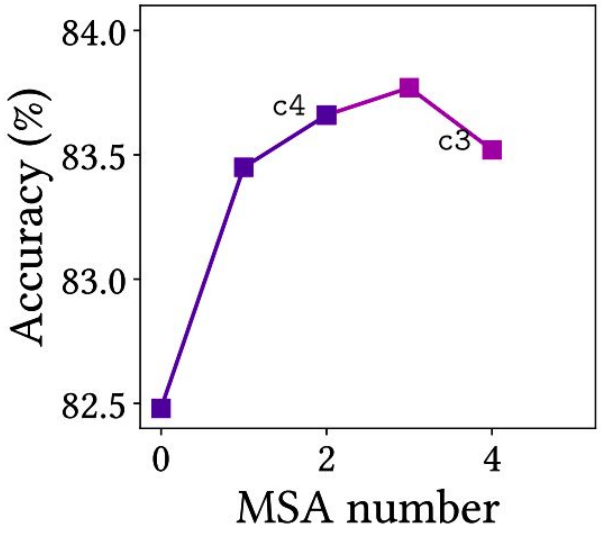

Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
MSA block
GAP
MLP
Output


84.0
83.5
83.0
82.5
Accuracy (%)
c4
0
2
4
MSA number

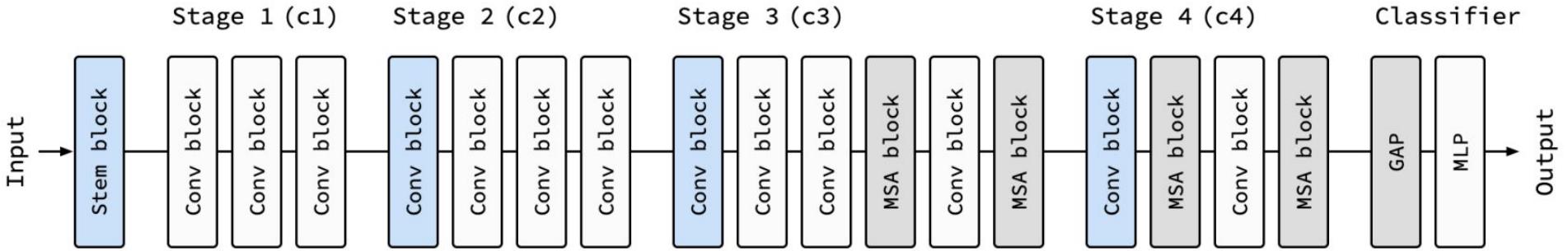

Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
MSA block
GAP
MLP
Output

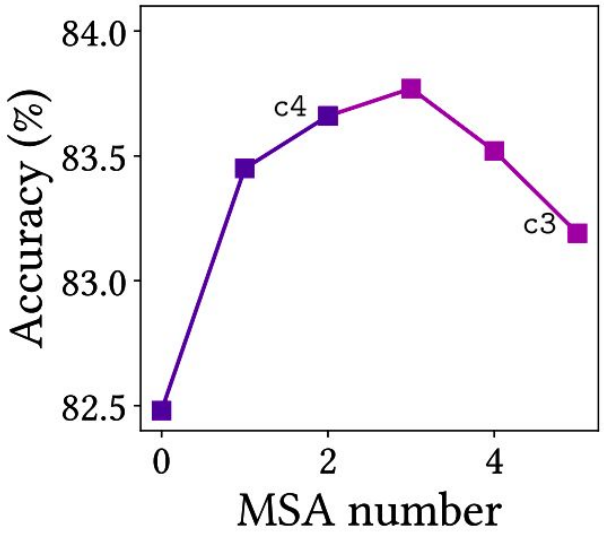

84.0
83.5
83.0
82.5
Accuracy (%)
c4
c3
0
2
4
MSA number


Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
MSA block
Conv block
MSA block
GAP
MLP
Output

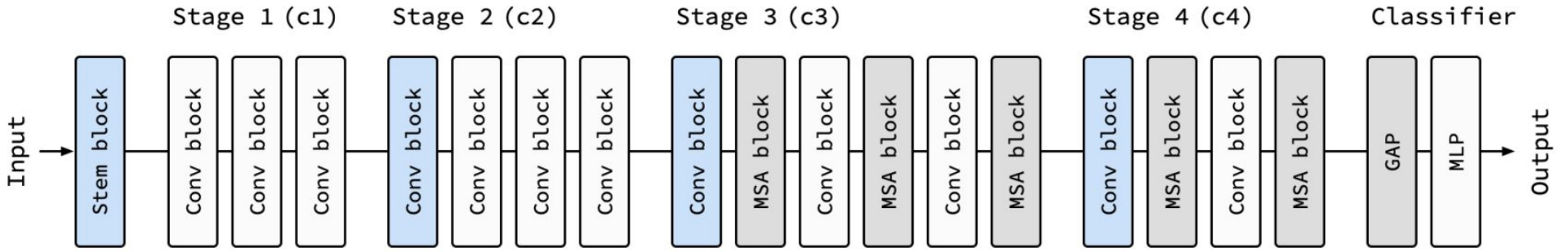


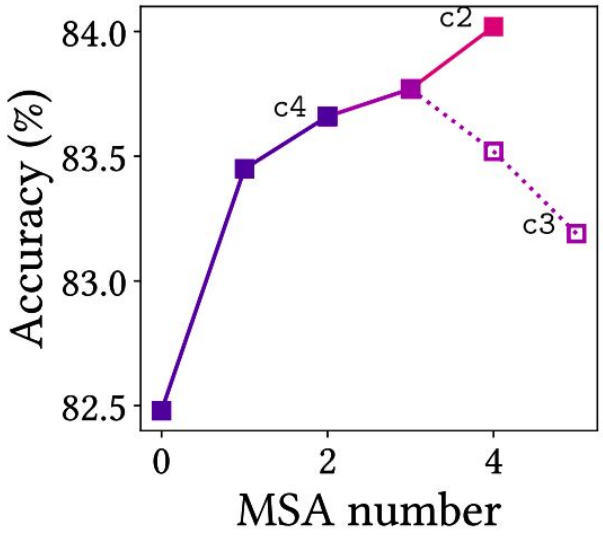
As expected,
MSAs in c3 harm accuracy.



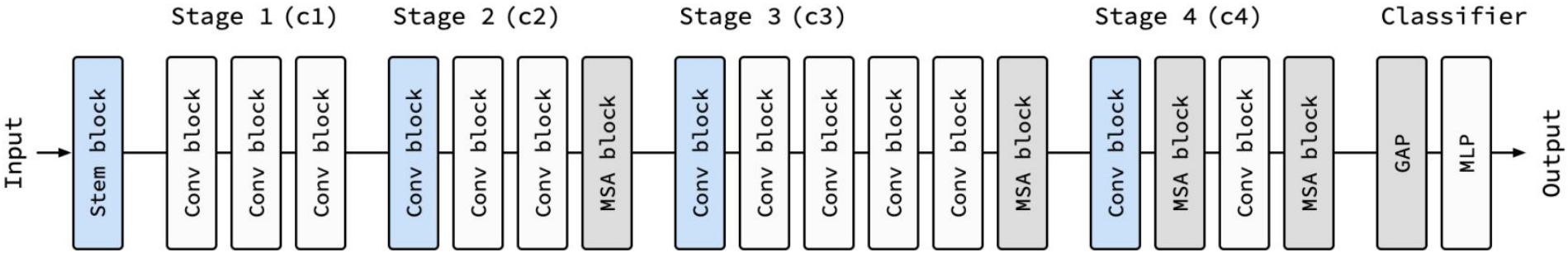


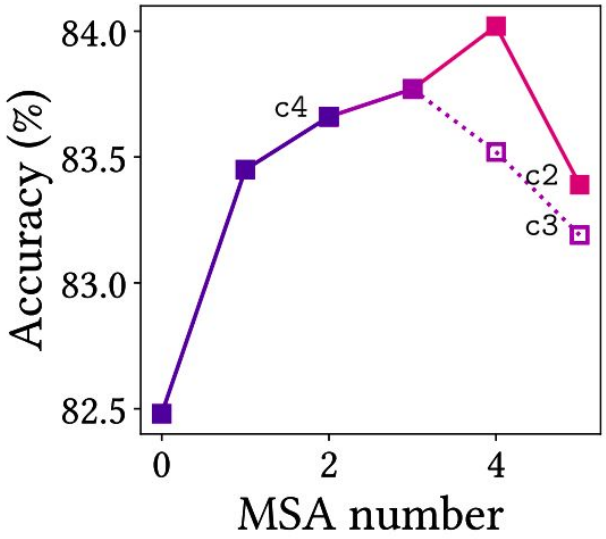
As expected,
MSAs in c3 harm accuracy.

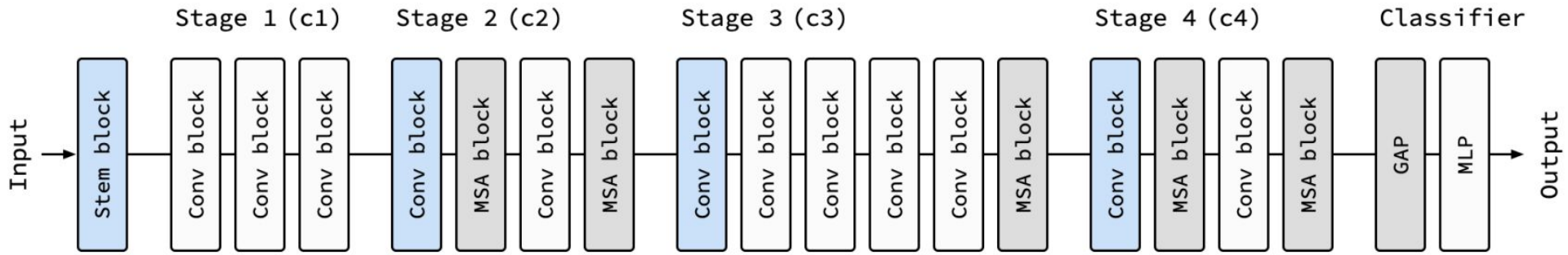


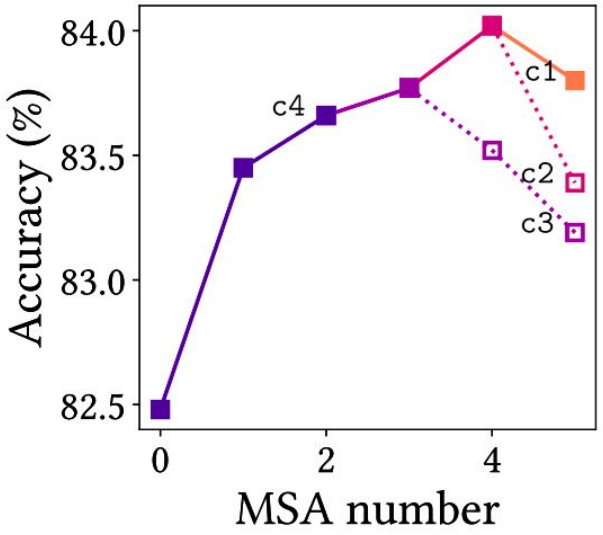


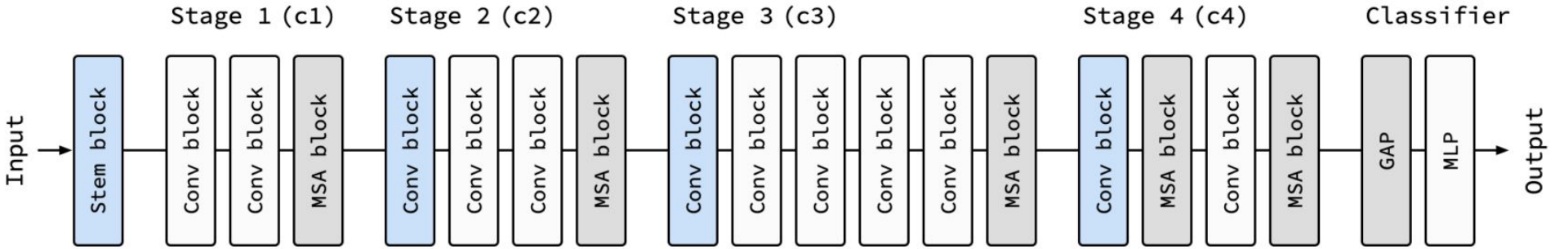
Surprisingly,
a MSA in c2 improves accuracy!

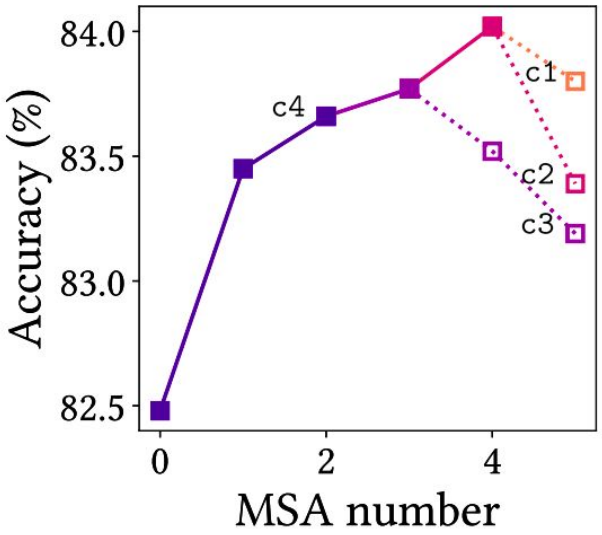


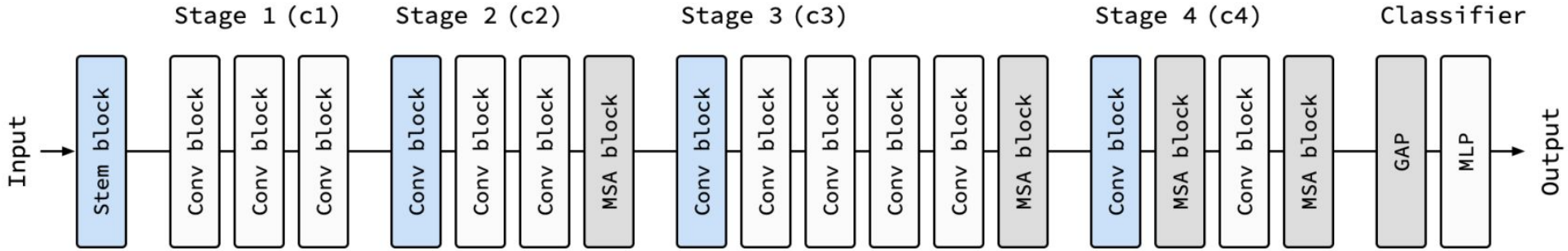

84.0
83.5
83.0
82.5
Accuracy (%)
c4
c2
c3
0
2
4
MSA number


Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
MSA block
Conv block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
MSA block
Conv block
MSA block
GAP
MLP
Output

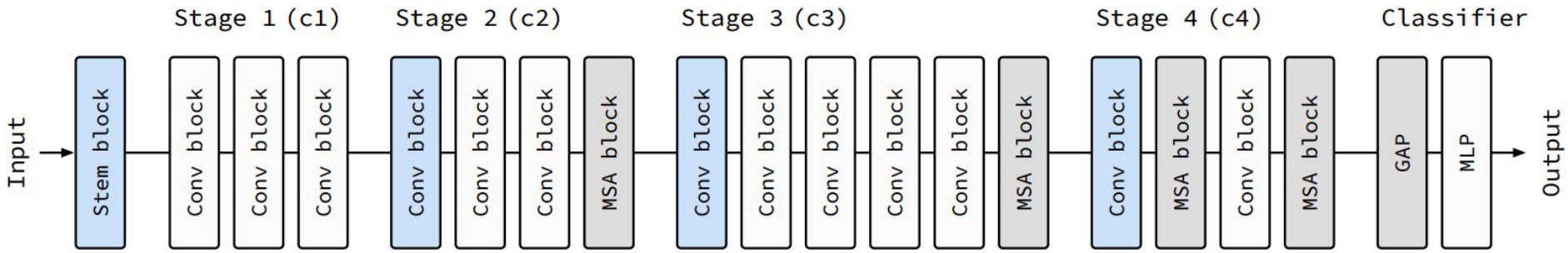

84.0
83.5
83.0
82.5
Accuracy (%)
0
2
4
MSA number
c1
c2
c3
c4


Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
Conv block
MSA block
Conv block
Conv block
Conv block
MSA block
Conv block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
MSA block
Conv block
MSA block
GAP
MLP
Output


84.0
83.5
83.0
82.5
Accuracy (%)
0
2
4
MSA number
c4
c1
c2
c3

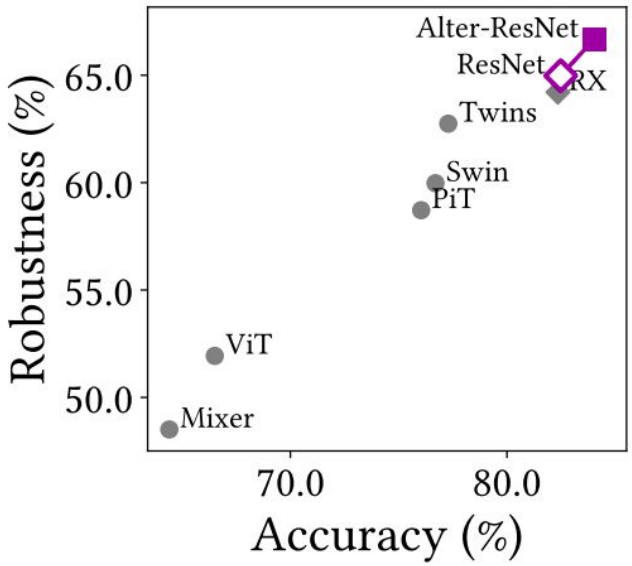

Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
MSA block
Conv block
MSA block
GAP
MLP
Output

AlterNet Based on ResNet-50

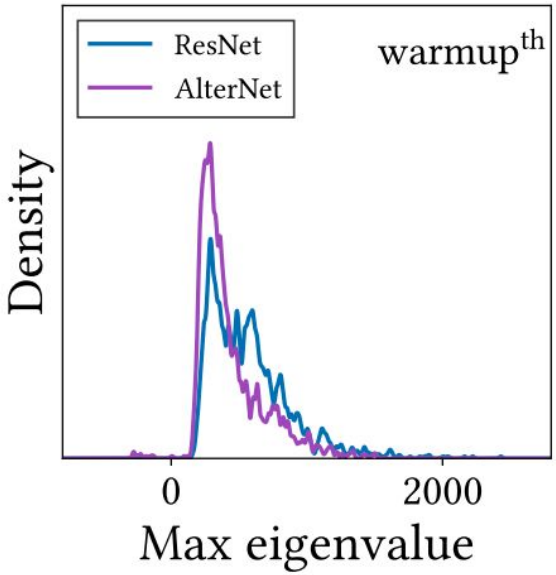


Detailed architecture of Alter-ResNet-50 for CIFAR-100. White, gray, and blue blocks mean Conv, MSA, and subsampling blocks. All stages (except stage 1) end with MSA blocks. This model is based on pre-activation ResNet-50. Following Swin, MSAs in stages 1 to 4 have 3, 6, 12, and 24 heads.

AlterNet Outperforms CNNs By Flattening Losses



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



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

AlterNet outperforms CNNs. Left: AlterNet outperforms CNNs even in a small data regime. "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.

Summary

In summary, appropriate inductive biases improves NN optimization, and self-attentions have a spatial smoothing inductive bias, which has the following properties.

	Self-Attention	Convolution
Loss Landscape	Flat but non-convex	Convex but sharp
Fourier Analysis	Low-pass filter (shape-biased)	High-pass filter (texture-biased)
Best Practice	The end of a stage	The beginning of a stage

How Do Vision Transformers Work?

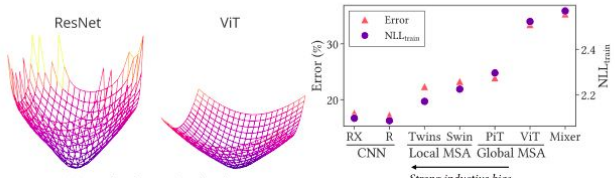
HMThanh, N.T.K.Nguyên, N.T.Tài
N.T.Thái, H.N.Thạch

Slides : https://hmthanh.github.io/how-vits-work

Code : https://github.com/hmthanh/how-vits-work

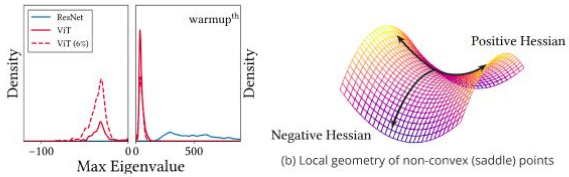
Đặc điểm nào Self-attention cần cải thiện?

MSAs (multi-head self-attention) thì làm phẳng hóa nhưng không làm lồi hàm loss. Ngược lại, Convs thì làm lồi nhưng sắc nét hàm loss.



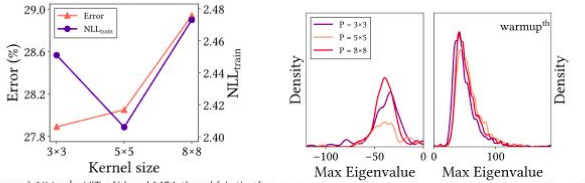
(a) Loss landscape visualization (b) Test error and training NLL for each model

Figure 1: *Hình trái* hàm thể hiện rằng **ViT** đã làm phẳng hoá hàm loss hơn so với mô hình Resnet. *Hình phải*: Weak inductive bias (e.g. long-range dependency) làm nhiễu quá trình tối ưu cho NN.



(a) Hessian max eigenvalue spectrum (b) Local geometry of non-convex (saddle) points

Figure 2: **Hessian max eigenvalue spectra** chỉ ra rằng **MSAs** có ưu và nhược điểm. **ViT** có một số negative Hessian eigenvalues, trong khi ResNet chỉ có một ít. Kích thước của **ViT**'s positive Hessian eigenvalues thì nhỏ.



(a) Error và NLL của ViT với local MSA theo kích thước kernal (b) Hessian max eigenvalue miền với kích thước kernel

Figure 3: Đặc trưng của **MSA** là data specificity, không phải long-range dependency. **Trái**: Convolutional ViT chứng minh rằng các ràng buộc cục bộ cải thiện ViT. **Phải**: Locality inductive bias bắt dừng lại the negative Hessian eigenvalues.

Self-Attention có hoạt động như Convolution?

MSAs là một hàm lọc low-pass, nhưng **Convs** là hàm lọc high-pass. Chứng tỏ rằng MSAs thì shape-biased, nhưng trái lại Convs thì texture-biased.

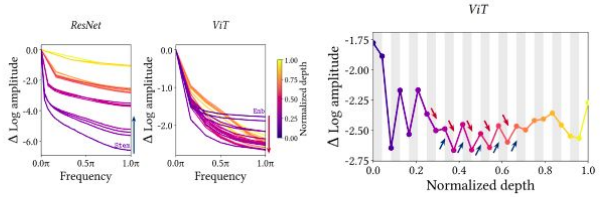
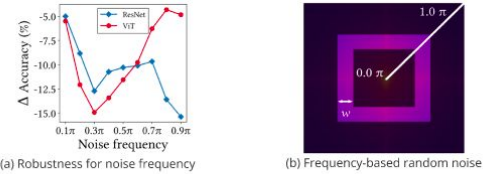
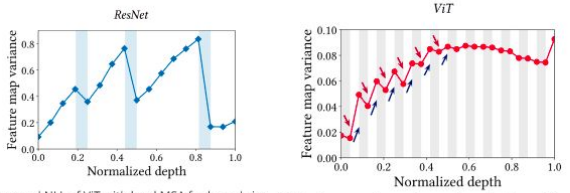


Figure 4: Biên độ log tương đối của **Đặc trưng fourier transformed** map chỉ ra rằng **ViT** có xu hướng giảm các tín hiệu (đặc trưng) high-frequency, trong khi **ResNet** tăng cường chúng. **Phải**: Trong ViT, **MSAs** (vùng xám) generally giảm thành phần high-frequency (1.0π) của feature map, và Conv/MLPs (vùng trắng) tăng cường chúng.



(a) Robustness for noise frequency (b) Frequency-based random noise

Figure 5: Chúng tôi đo lường **sự giảm độ chính xác đối với nhiễu ngẫu nhiên dựa trên tần số (frequency-based random noise)**. **ViT** được tăng cường nhiều high-frequency, trong khi **ResNet** is dễ bị tổn thương bởi chúng. Điều này chỉ ra rằng tín hiệu low-frequency và high-frequency được thông tin đến MSAs và Convs.



(a) Error and NLL of ViT with local MSA for kernel size (b) Hessian max eigenvalue spectrum for kernel size

Figure 6: **MSAs** (vùng xám) giảm phương sai của các điểm feature map, nhưng **Convs**/ MLPs (vùng trắng) tăng phương sai. Vùng xanh là lớp subsampling. Kết quả ngụ ý rằng **MSAs** tổng hợp feature maps, và **Convs** biến đổi chúng.

Làm cách nào hòa hợp giữa Conv và Attention

MSAs nên ở cuối giai đoạn của mô hình và **Convs** ở đầu giai đoạn của mô hình sẽ cải thiện đáng kể hiệu năng.

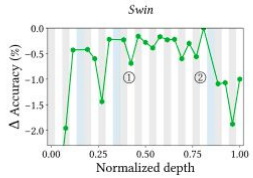
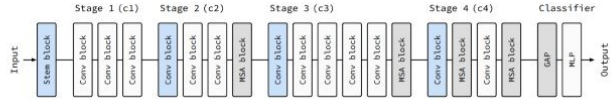
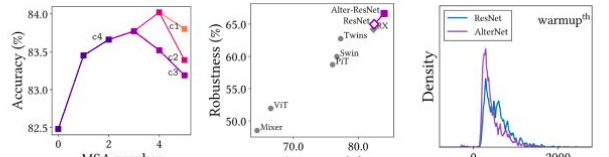


Figure 7: Chúng tôi đo lương **decrease trong độ chính xác sau khi loại bỏ một phần tử** ra khỏi mô hình đã được. Độ chính xác thay đổi định kỳ, và đây là một giai đoạn. Trong Swin, ① **Convs** (vùng trắng) đóng một vai trò quan trọng ở đầu giai đoạn, và ② **MSAs** (vùng xám) đóng một vai trò quan trọng ở giai đoạn cuối chu kỳ.



(a) Architecture of Alter-ResNet-50 for CIFAR-100



(b) Accuracy of AlterNet for MSA number (c) Accuracy and robustness in a small data regime (CIFAR-100) (d) Hessian max eigenvalue spectra

Figure 8: Chúng tôi đề xuất mô hình **AlterNet**, là mô hình mà **Conv** blocks ở cuối chu kỳ được thay thế bằng **MSA** blocks. **AlterNet** vượt trội hơn CNNs dù là ở những tập dữ liệu nhỏ.

Tổng kết, những inductive biases thích hợp cải thiện quá trình tối ưu NN, và self-attentions có một chức năng như một spatial smoothing inductive bias.

	Self-Attention	Convolution
Loss Landscape	Flat but not-convex	Convex but sharp
Fourier Analysis	Low-pass filter (shape-biased)	Low-pass filter (shape-biased)
Best practice	Ở cuối mô hình	Ở đầu mô hình

4. Experiment

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

4. Experiment Results - Statements

- Training NN need minimizes of highly non-convex loss func
- Lead-out MSAs not only generalized Convs
  - spatial smoothings that complement Convs
  - ensembling feature map points and flattening the loss landscape
  - preserve the architectures of Conv and MSA blocks => in AlterNet (harmonize)

=> backbone for vision task, improve results predict

4. Experiment Results - Setup Env

- Two sets of machines for CIFAR
  - Intel Xeon W2123, 32GB, GeForce RTX2080Ti
  - 4 Intel Intel Broadwell, 15GB, NVIDIA T4
- ImageNet
  - AMD Ryzen Threadripper 3960X 24-Core, 256GB, GeForce RTX 2080 Ti

4. Experiment Results - Setup Env

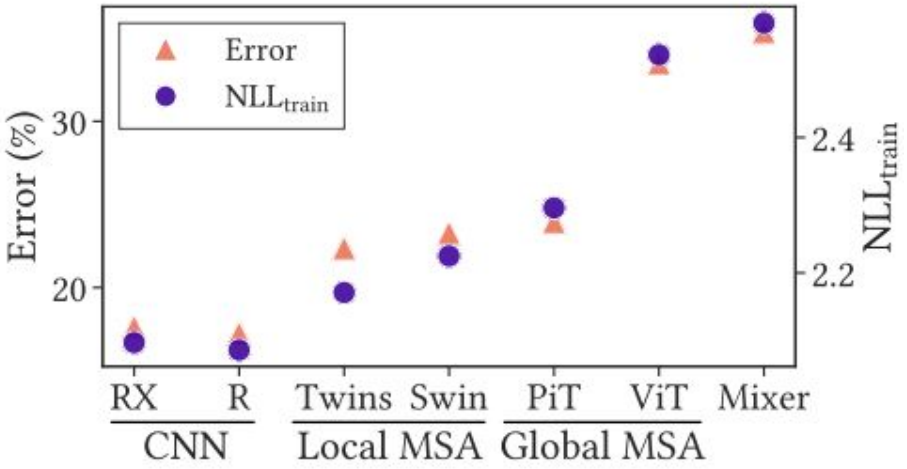
- Train NN configs
  - cross-entropy (NLL) loss and AdamW optimizer
  - initial learning rate of 1.25×10^{-4}
  - weight decay of 5×10^{-2}
  - batch size of 96 on CIFAR
  - batch size of 128 on ImageNet
  - strong data augmentations—such as RandAugment RandomErasing (resize, crop, erasing, mixup…)
  - Vision Transformers are based on Wightman (2019) and Wang (2021).

4. Experiment Results - Statements

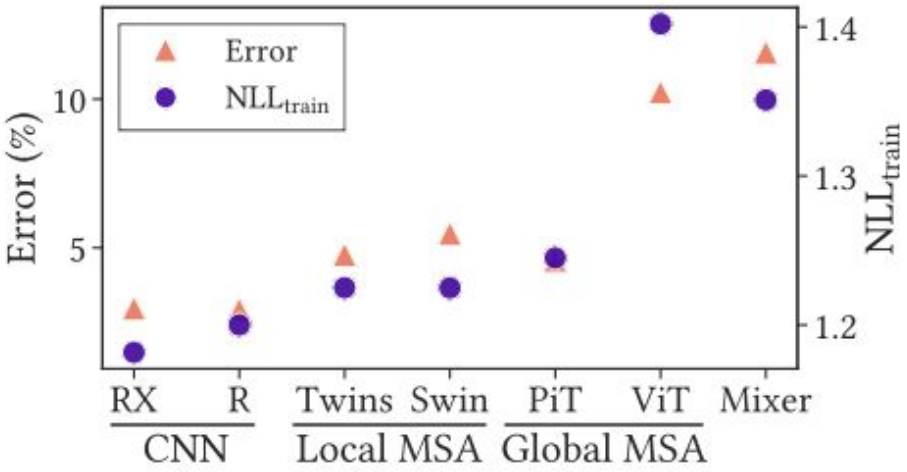
- Training NN need minimizes of highly non-convex loss func
- Lead-out MSAs not only generalized Convs
  - spatial smoothings that complement Convs
  - ensembling feature map points and flattening the loss landscape
  - preserve the architectures of Conv and MSA blocks => in AlterNet (harmonize)

=> backbone for vision task, improve results predict

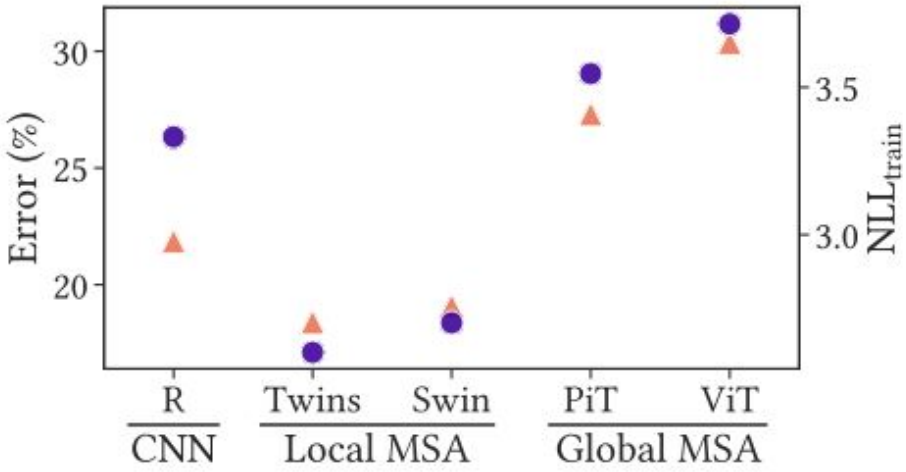
4. Experiment Results - Compare



(a) Error and NLL_{train} for each model.

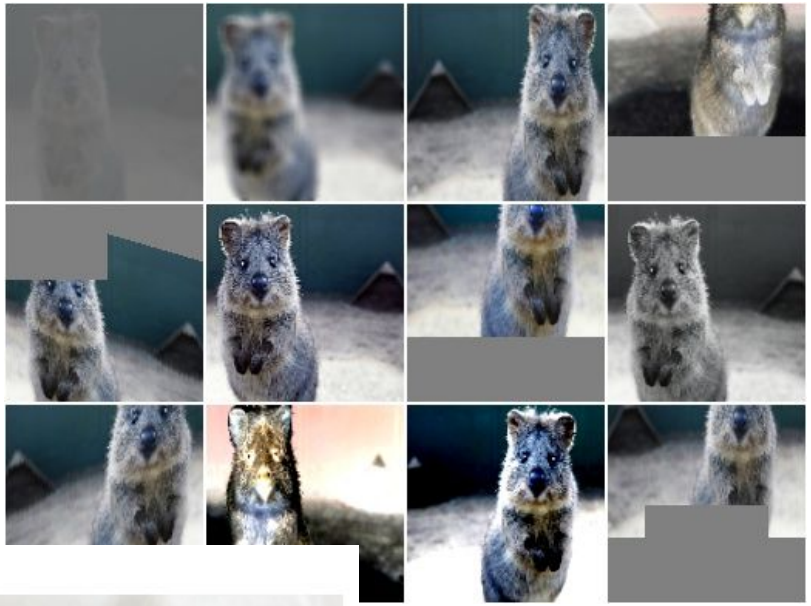


(a) CIFAR-10



(b) ImageNet

4. Experiment Results - Setup Env



ResNet-50 Mixup [47] Cutout [3] CutMix

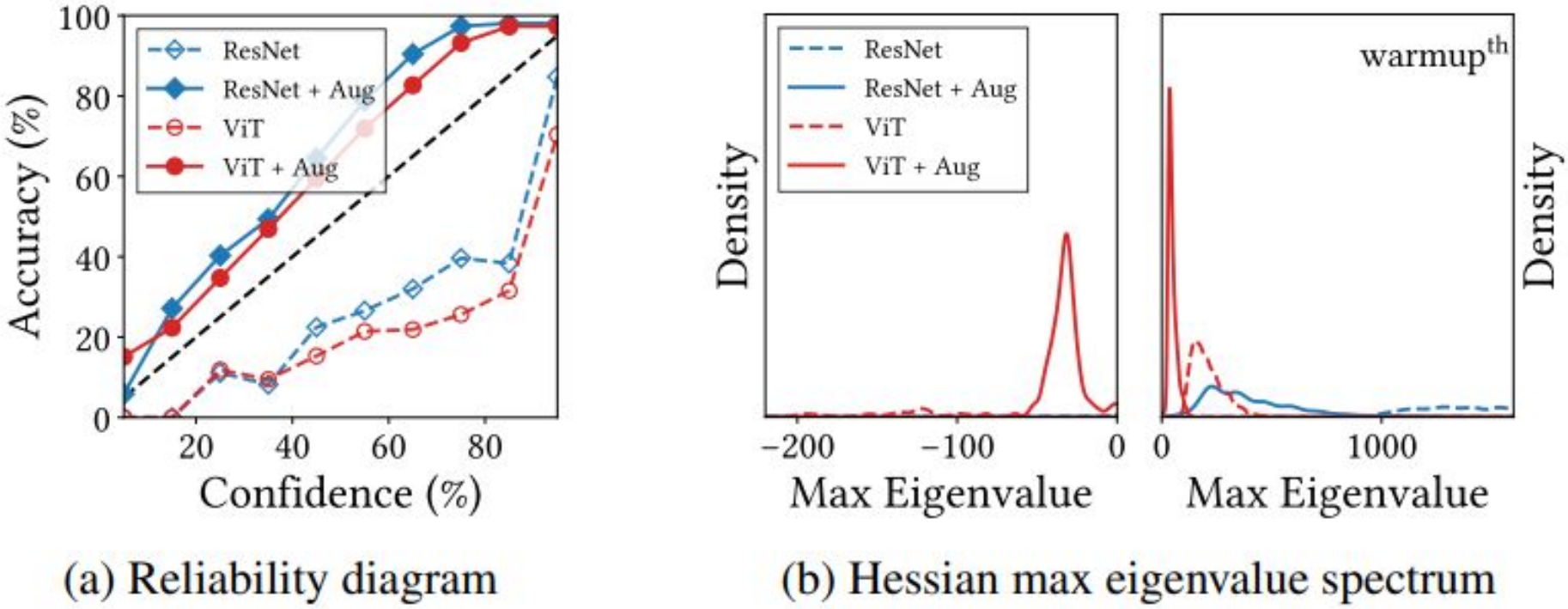
lambda=0.5

[0, 1]

[0.5, 0.5]

4. Experiment Results - Compare

- Strong data augmentation for MSA training harms uncertainty calibration



(a) Reliability diagram

(b) Hessian max eigenvalue spectrum

CIFAR-100

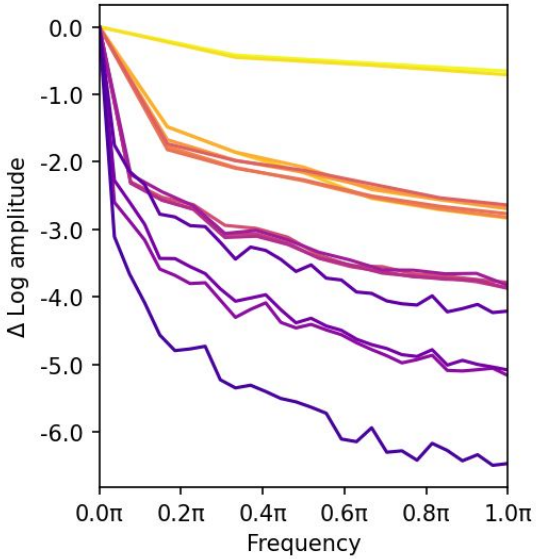
5. Demo

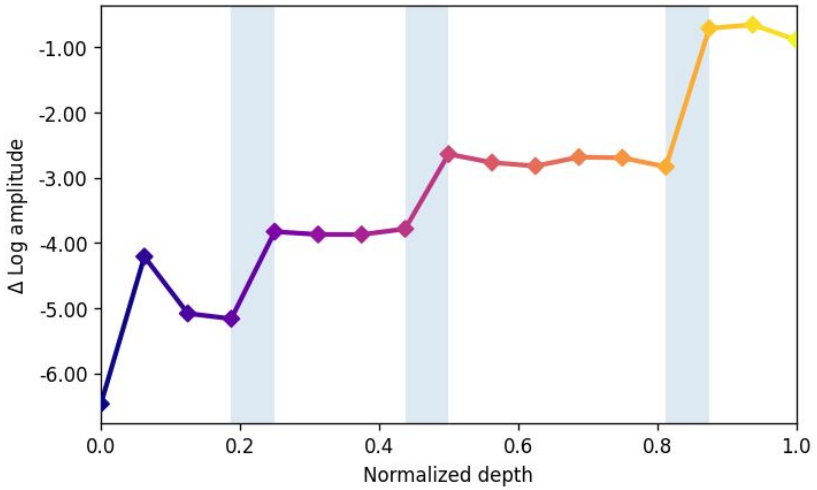
KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

5. Demo

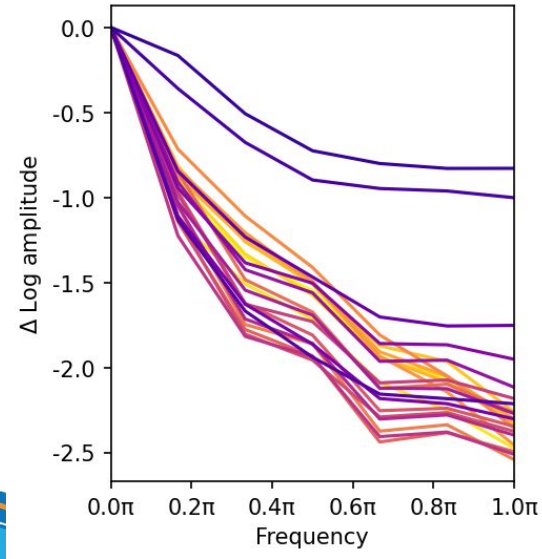
Fourier Analysis

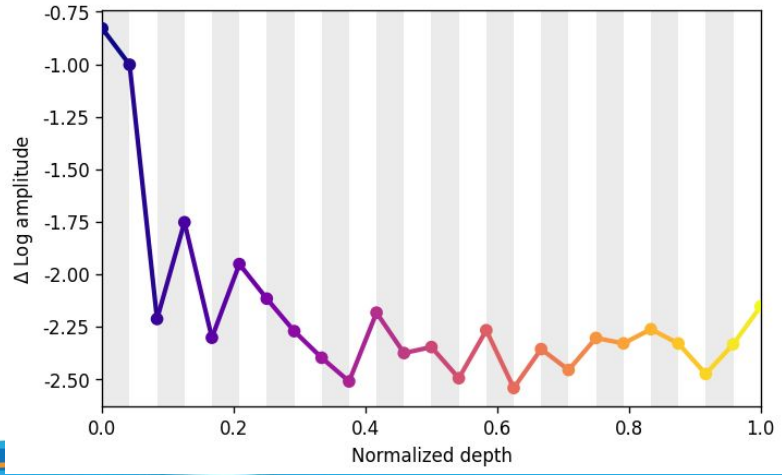
RESNET





VIT





5. Demo

Fourier Analysis

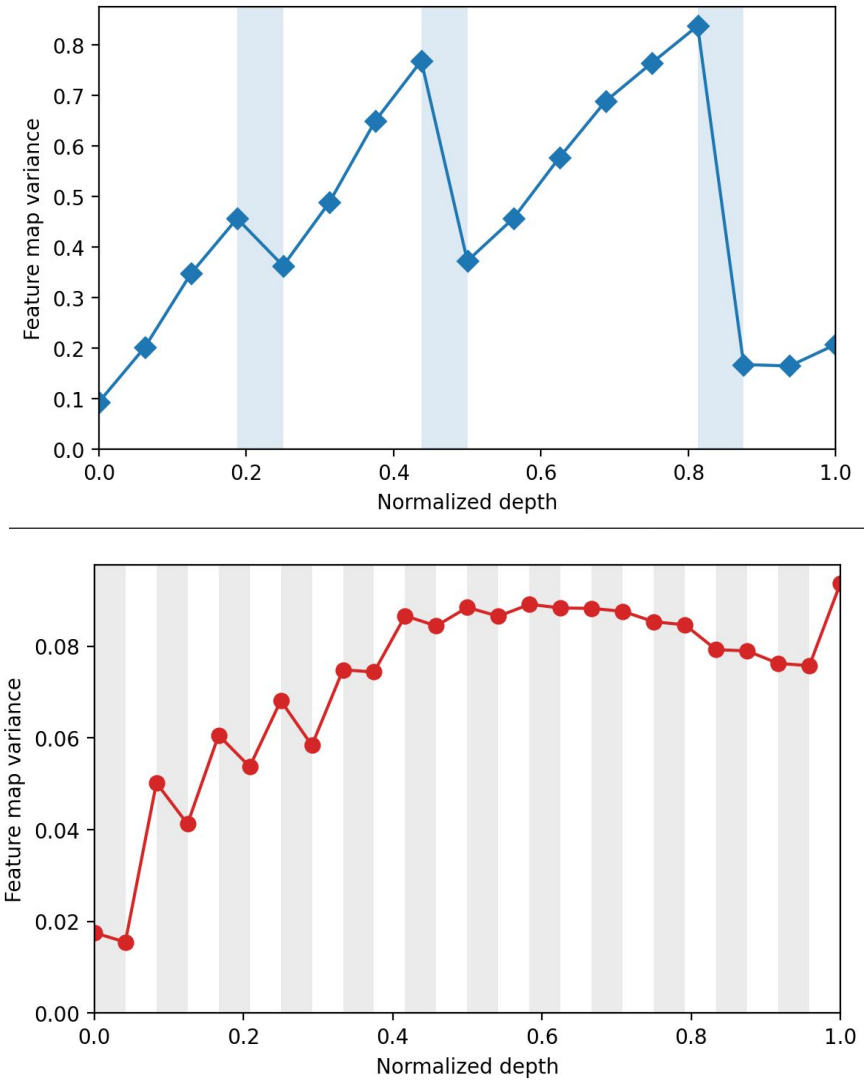
```
# Fourier transform feature maps
fourier_latents = []
for latent in latents:  # `latents` is a list of hidden feature maps in latent spaces
    latent = latent.cpu()

    if len(latent.shape) == 3:  # for ViT
        b, n, c = latent.shape
        h, w = int(math.sqrt(n)), int(math.sqrt(n))
        latent = rearrange(latent, "b (h w) c -> b c h w", h=h, w=w)
    elif len(latent.shape) == 4:  # for CNN
        b, c, h, w = latent.shape
    else:
        raise Exception("shape: %s" % str(latent.shape))
    latent = fourier(latent)
    latent = shift(latent).mean(dim=(0, 1))
    latent = latent.diag()[int(h/2):]  # only use the half-diagonal components
    latent = latent - latent[0]  # visualize 'relative' log amplitudes
                                 # (i.e., low-freq amp - high freq amp)
    fourier_latents.append(latent)
```

5. Demo

ResNET

ViT

Featuremap Variance



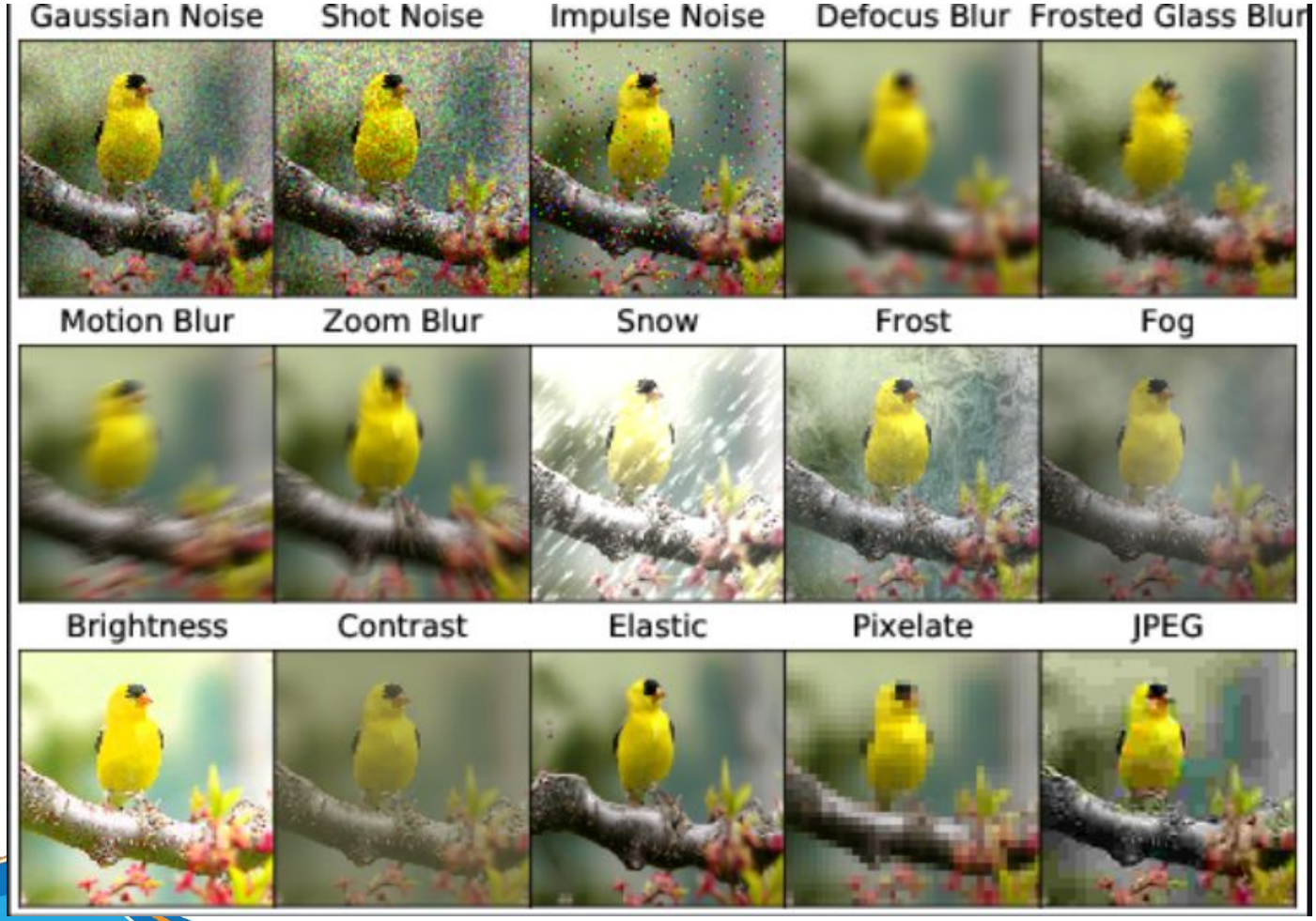
5. Demo

Featuremap Variance

```
# aggregate feature map variances
variances = []
for latent in latents:  # `latents` is a list of hidden feature maps in latent spaces
    latent = latent.cpu()

    if len(latent.shape) == 3:  # for ViT
        b, n, c = latent.shape
        h, w = int(math.sqrt(n)), int(math.sqrt(n))
        latent = rearrange(latent, "b (h w) c -> b c h w", h=h, w=w)
    elif len(latent.shape) == 4:  # for CNN
        b, c, h, w = latent.shape
    else:
        raise Exception("shape: %s" % str(latent.shape))

    variances.append(latent.var(dim=[-1, -2]).mean(dim=[0, 1]))
```

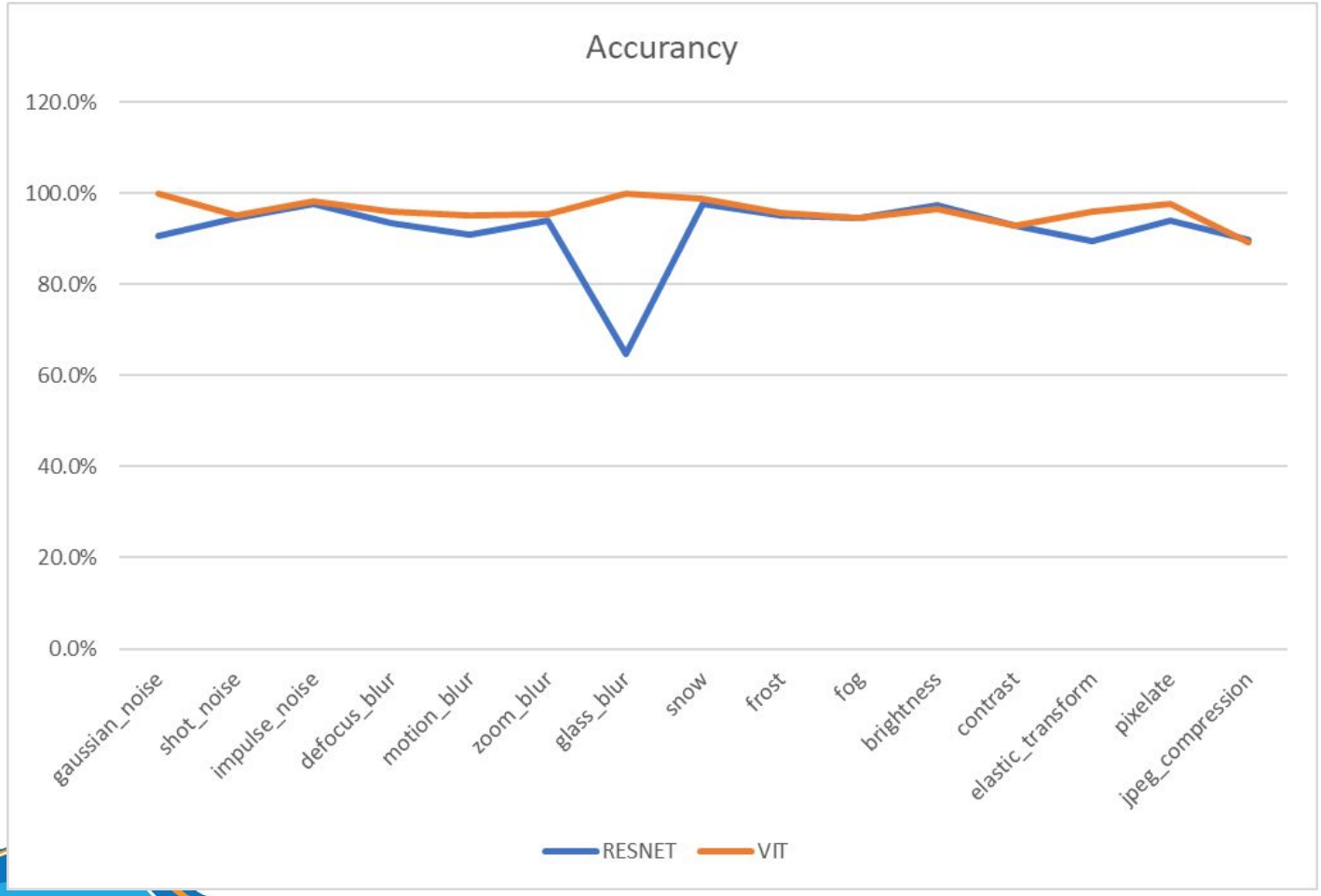
5. Demo

Common Corruptions



5. Demo

Common Corruptions



Resource

Bản tóm tắt nội dung A4 :

https://www.figma.com/proto/nosufU0fha81EnQslSM6e5/How-Do-Vision-Transformers-Work?node-id=21%3A177&scaling=min-zoom&page-id=0%3A1&hide-ui=1

Github : https://github.com/hmthanh/how-vits-work

Report : https://docs.google.com/document/d/1jZTbRF4kJoJIksyO5F91x_YoWmuBbMoS/edit?rtpof=true&sd=true

Q & A

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN