How Do Vision Transformers Work?

Namuk Park, Songkuk Kim
namuk.park@yonsei.ac.kr

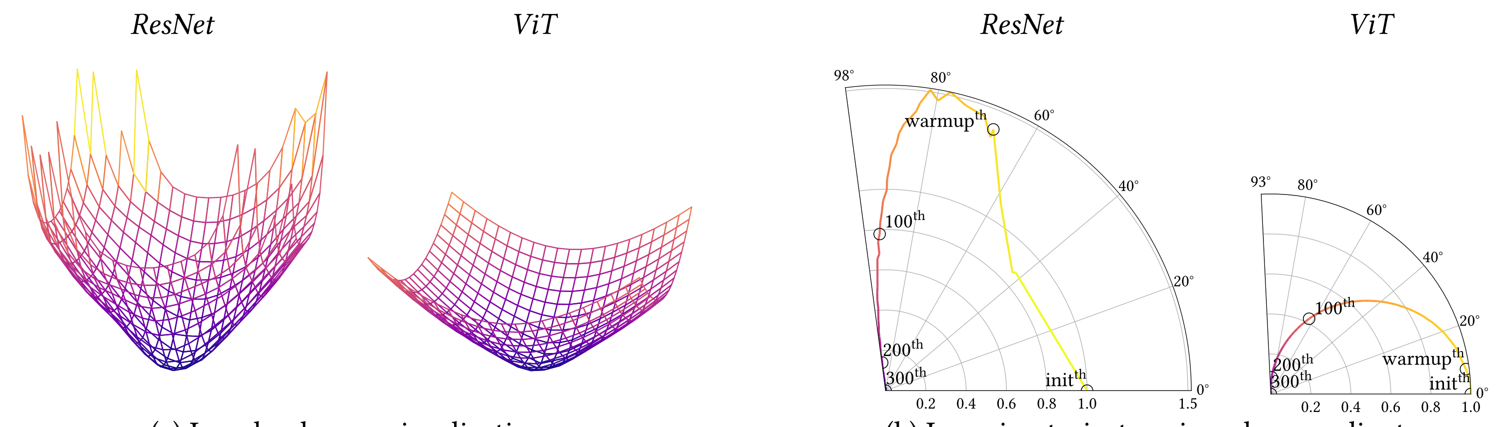
How Do Vision Transformers Work?

Namuk Park[1,2], Songkuk Kim[1]
[1]Yonsei University, [2]NAVER AI Lab

Paper: https://arxiv.org/abs/2202.06709
Code: https://github.com/xxxnell/how-do-vits-work

What Properties of Self-Attentions Do We Need?

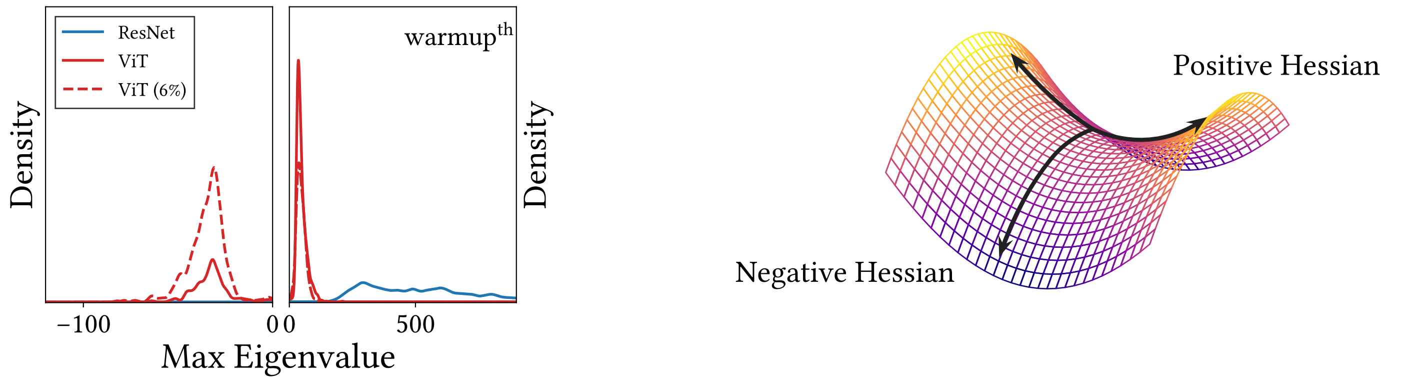
MSAs (multi-head self-attention) have flat but non-convex losses. In contrast, Convs have convex but sharp losses.



(a) Loss landscape visualization

(b) Learning trajectory in polar coordinate

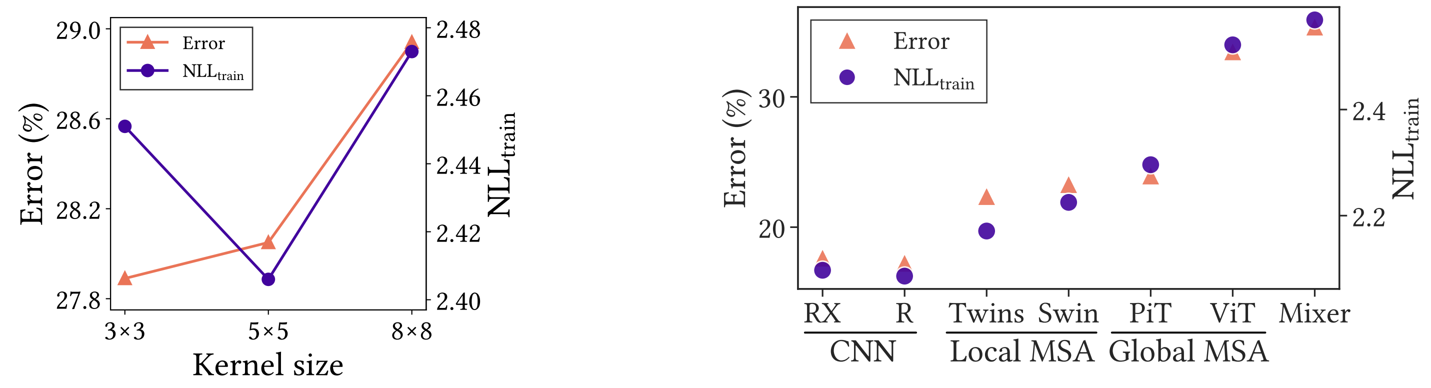
Loss landscape visualization (*Left*) show that ViT has a flatter loss than ResNet. **Learning trajectory visualization (*Right*)** shows that ViT converges to the optimum along a smooth trajectory.



(a) Hessian eigenvalue spectrum

(b) Local geometry of non-convex (saddle) points

Hessian max eigenvalue spectra show that MSAs have their pros and cons. ViT has a number of negative Hessian eigenvalues, while ResNet only has a few. The magnitude of ViT's positive Hessian eigenvalues is small.



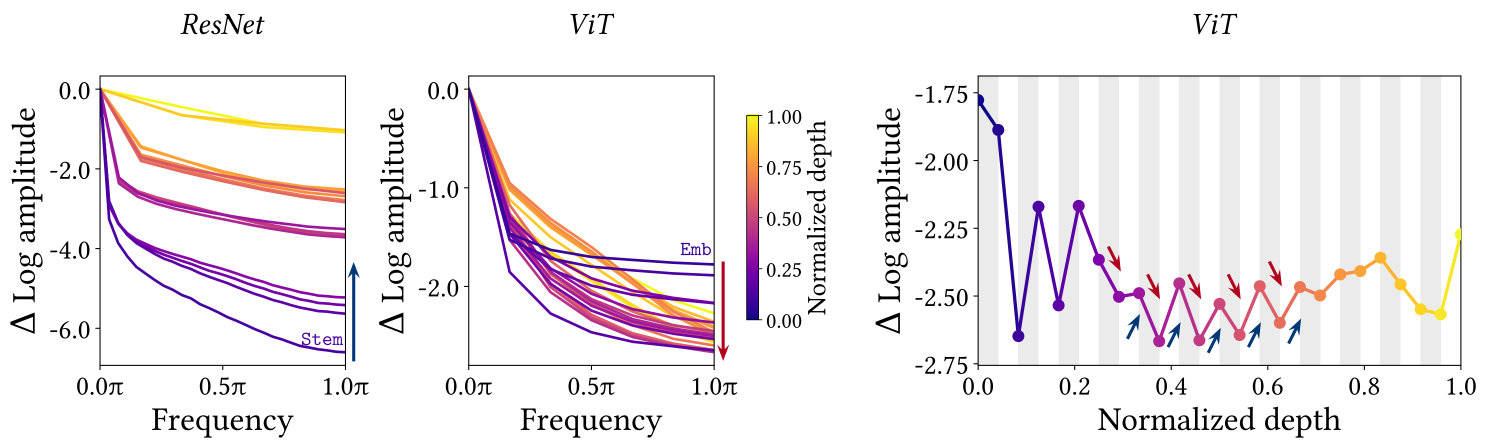
(a) Error and NLL of ViT with local MSA for kernel size

(b) Test error and training NLL for each model

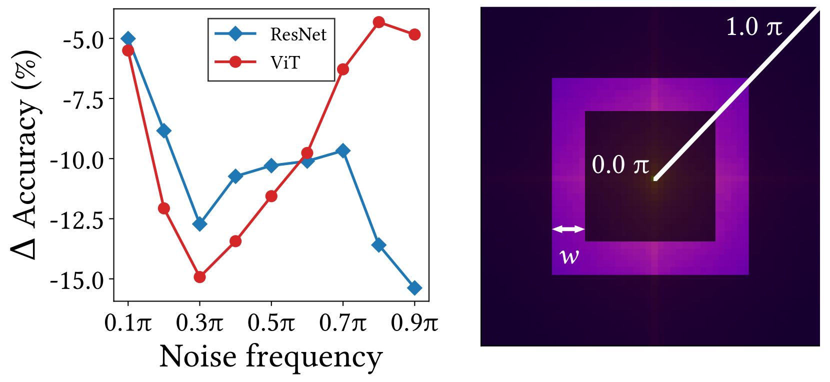
The key feature of MSA is data specificity, not long-range dependency. *Left*: Convolutional ViT demonstrates that locality constraint improves ViT. *Right*: Weak inductive bias (e.g. long-range dependency) disturbs NN optimization.

Do Self-Attentions Act Like Convs?

MSAs are low-pass filter, but Convs are high-pass filter. It suggests that MSAs are shape-biased, whereas Convs are texture-biased.



Relative log amplitudes of **Fourier transformed feature map** show that ViT tend to reduces high-frequency signals, while ResNet amplify them. *Left*: In ViT, MSAs (gray area) generally reduce the high-frequency (1.0π) component of feature map, and Conv/MLPs (white area) amplify it.

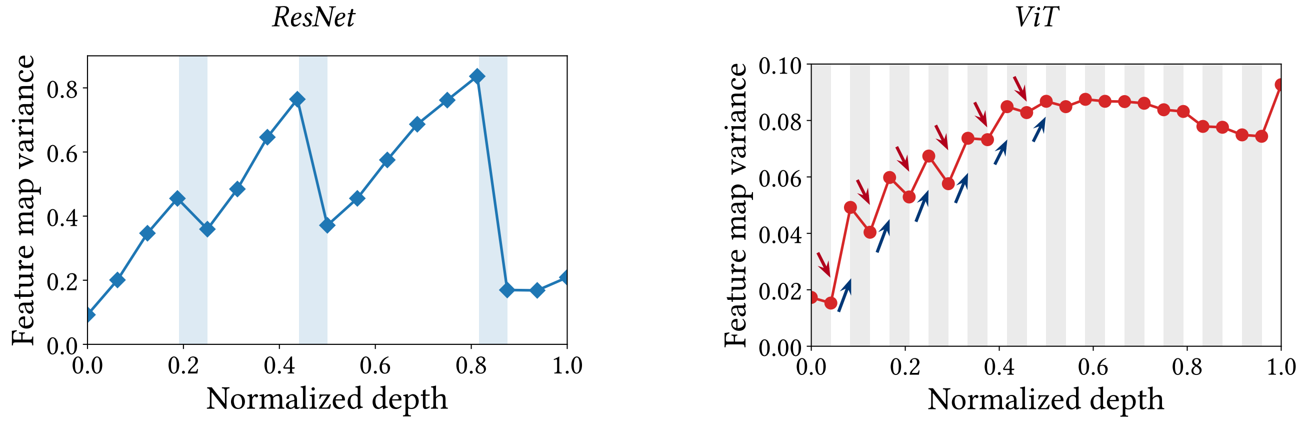


(a) Robustness for noise frequency

(b) Frequency-based random noise

We measure the **decrease in accuracy against frequency-based random noise**. ViT is robust against high-frequency noise, while ResNet is vulnerable to them.

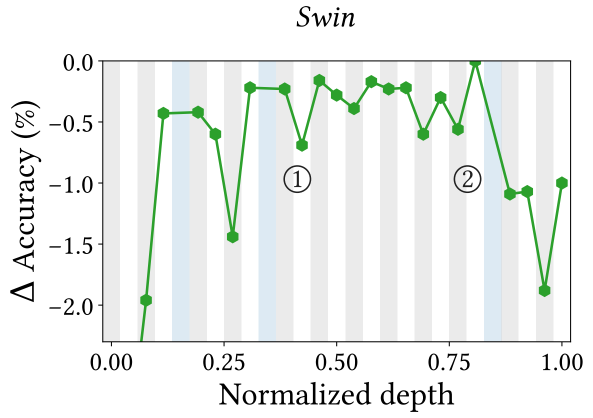
It suggests that low-frequency signals and high-frequency signals are informative to MSAs and Convs.



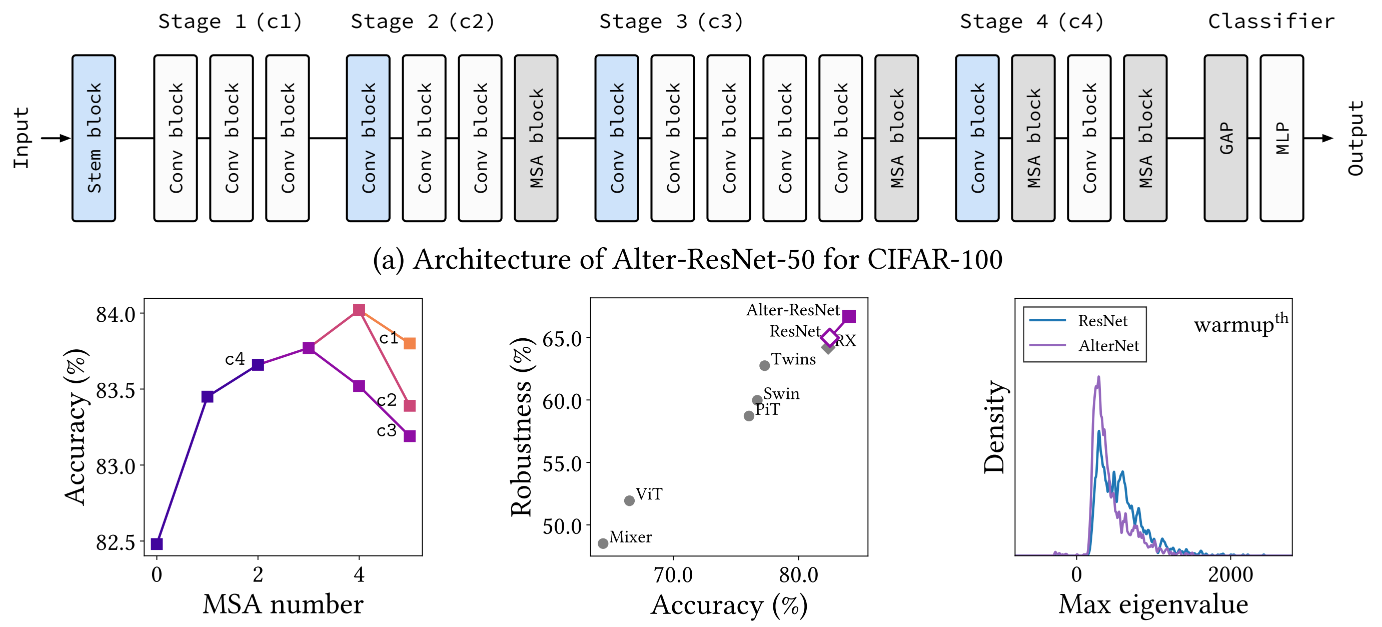
MSAs (gray area) reduce **the variance of feature map points**, but Convs/MLPs (white area) increase the variance. The blue area is subsampling layer. The results implies that MSAs aggregate feature maps, and Convs convert them.

How Can We Harmonize Self-Attentions with Convs?

MSAs closer to the end of a stage (not a model) and Convs at the beginning 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. In Swin, ① Convs (white area) play an important role at the beginning of a stage, and ② MSAs (gray area) play an important role at the end of a stage.



(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

We propose ***AlterNet***, a model in which Conv blocks at the end of *a stage* are replaced with MSA blocks. AlterNet outperforms CNNs even in small data regimes.

In summary, appropriate inductive biases improves NN optimization, and self-attentions have a spatial smoothing inductive bias.

	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

Three Key Questions

Q1. What properties of MSAs do we need to improve optimization?

▸

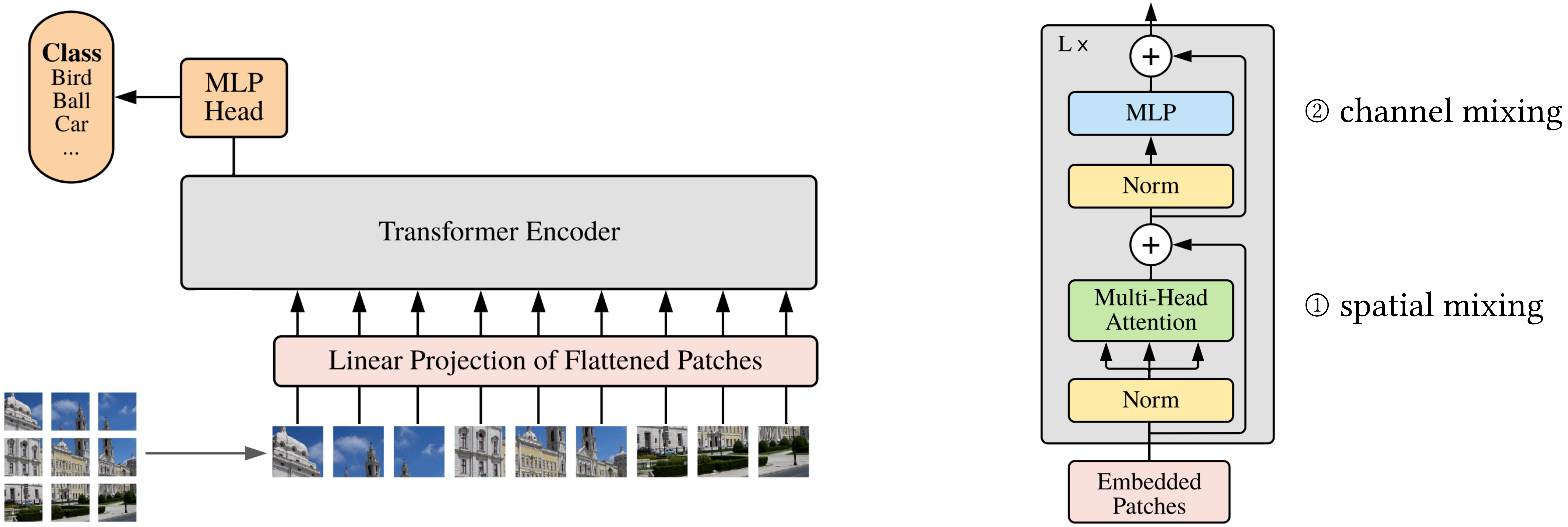
Q2. Do MSAs act like Convs?

▸

Q3. How can we harmonize MSAs with Convs?

▸

Vision Transformers Patchify Images



(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.

Dosovitskiy, Alexey, et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." ICLR (2020).

MSAs Are Convs With Large & Data-Specific Kernels

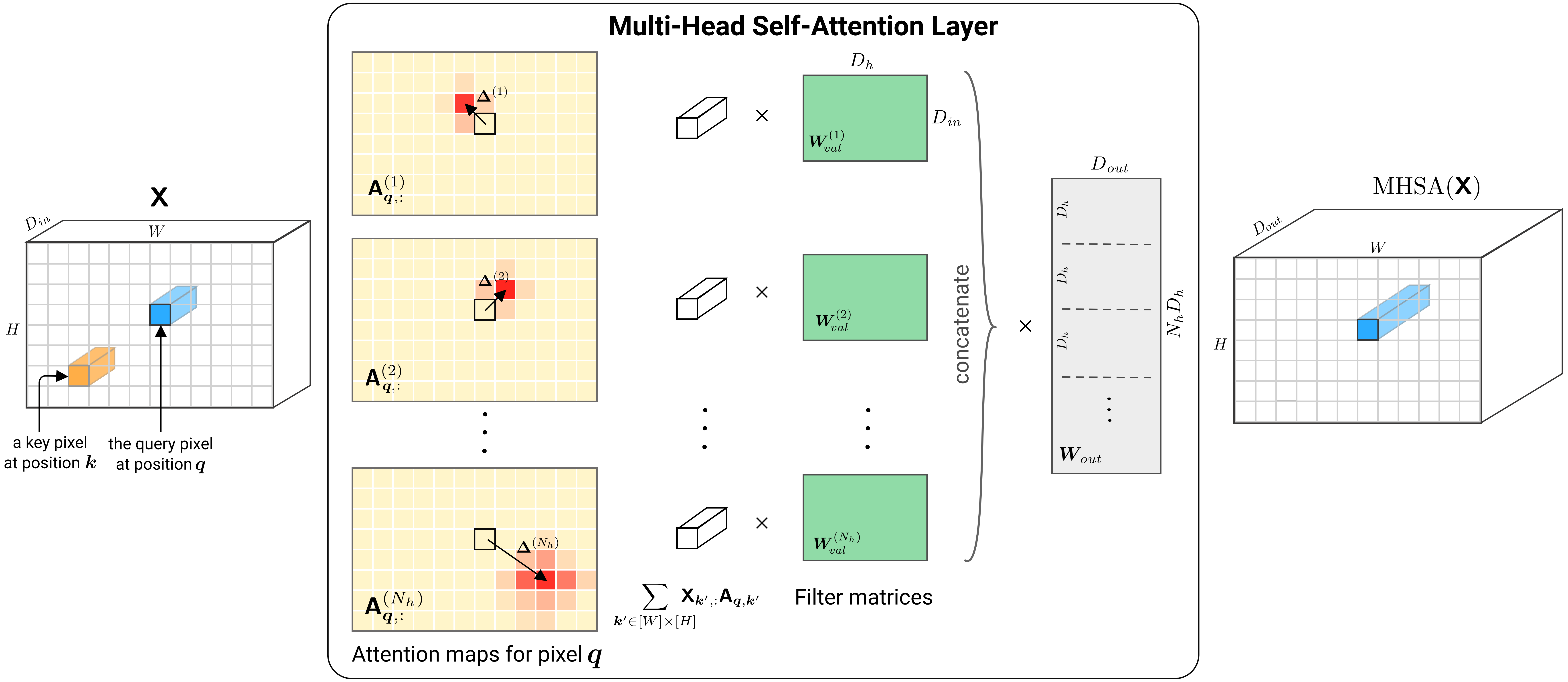


Illustration of a ***"multi-head self-attention"*** (MSA) layer applied to a tensor image ***X***.
We represent self-attention maps (red boxes)—similarities between a query pixel and a key pixel.

Cordonnier, Jean-Baptiste, Andreas Loukas, and Martin Jaggi. "On the Relationship between Self-Attention and Convolutional Layers." ICLR (2020).

Three Key Questions

Q1. What properties of MSAs do we need to improve optimization?

- Modeling long-range dependencies improve the accuracy and generalization.

Q2. Do MSAs act like Convs?

- MSAs are at least as expressive as Convs, so MSAs behave like generalized Convs.

Q3. How can we harmonize MSAs with Convs?

- MSAs closer to the end of a model improve accuracy, since they capture abstractions.

Three Key Questions

Q1. What properties of MSAs do we need to improve optimization?

- ~~Modeling long-range dependencies improve the accuracy and generalization~~.

Q2. Do MSAs act like Convs?

- ~~MSAs are at least as expressive as Convs, so MSAs behave like generalized Convs~~.

Q3. How can we harmonize MSAs with Convs?

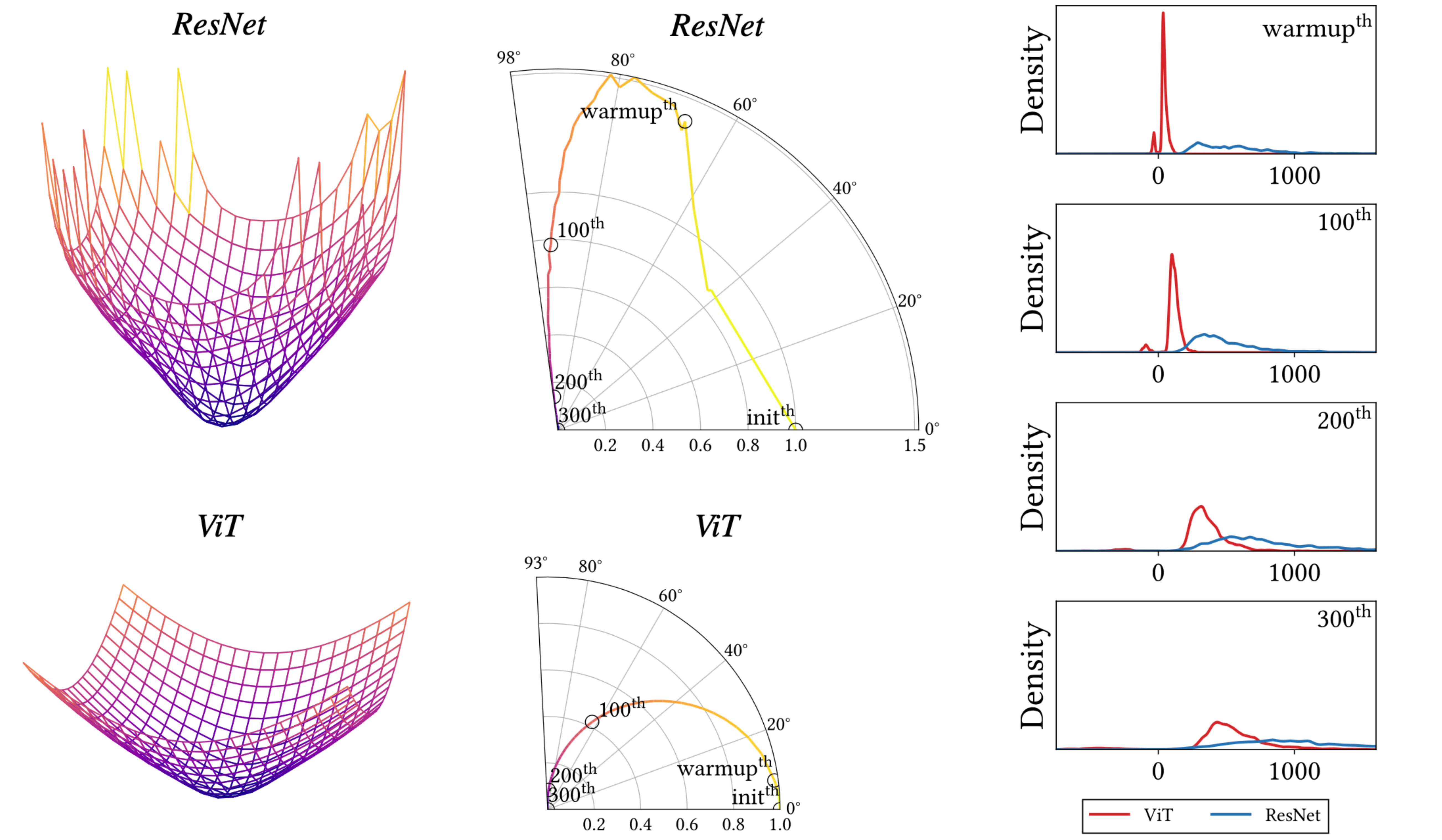
- ~~MSAs closer to the end of a model improve accuracy, since they capture abstractions~~.

All explanations and intuitions are poorly supported!

Question One

What Properties of MSAs Do We Need To Improve Optimization?

MSAs Flatten Loss Landscapes

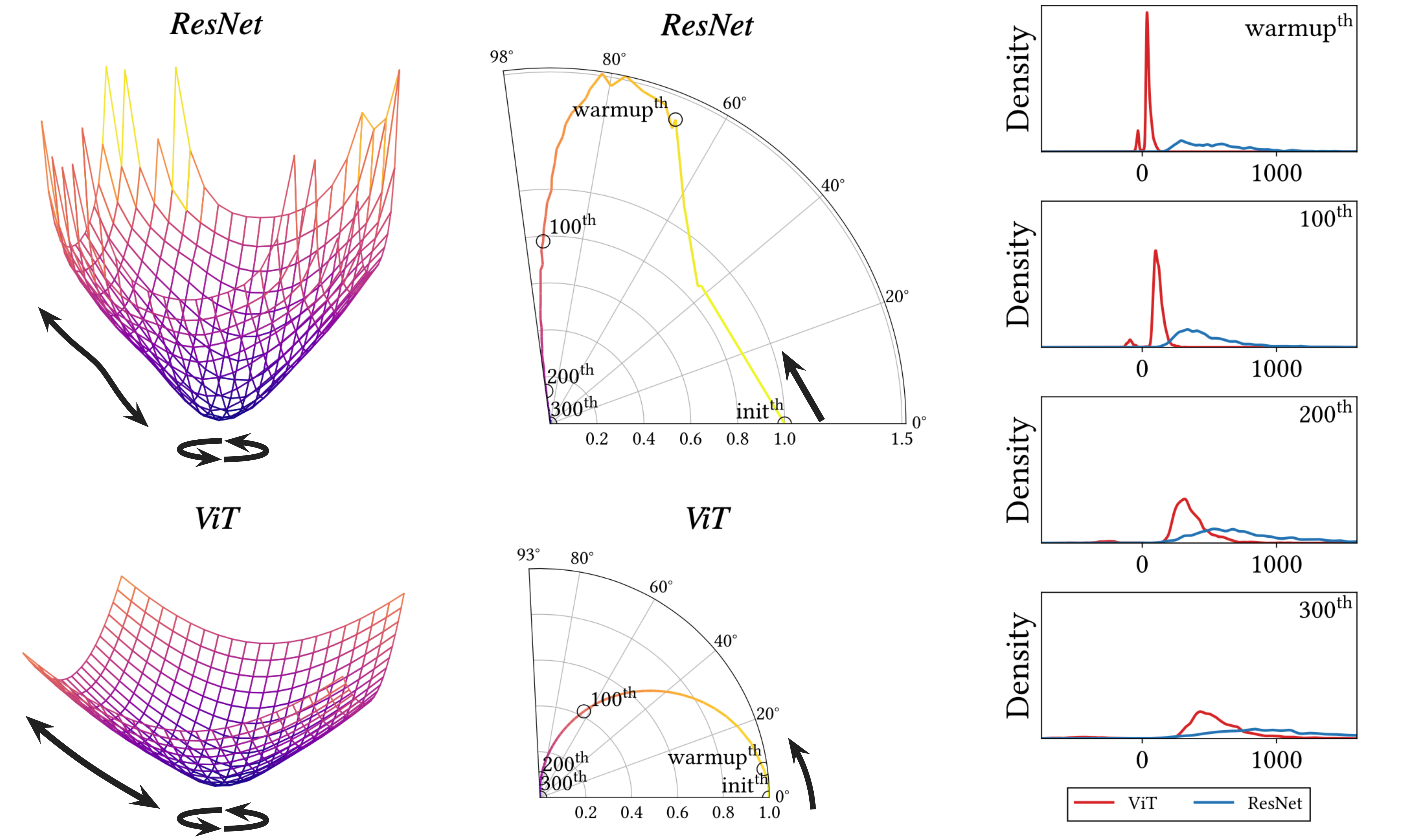


(a) Loss landscape visualizations (b) Trajectories in polar coordinate (c) Hessian max eigenvalue spectra

Three different aspects consistently show that MSAs flatten loss landscape. *Left*: Loss landscape visualizations show that ViT has a flatter loss than ResNet. ***Middle***: ViT converges to the optimum along a smooth trajectory. ***Right***: The magnitude of the Hessian eigenvalues of ViT is smaller than that of ResNet.

For loss landscape visualization, see arxiv.org/abs/1712.09913. Learning trajectory and Hessian max eigenvalue spectra are **first proposed** in these works. Compare to arxiv.org/abs/1901.10159.

MSAs Flatten Loss Landscapes



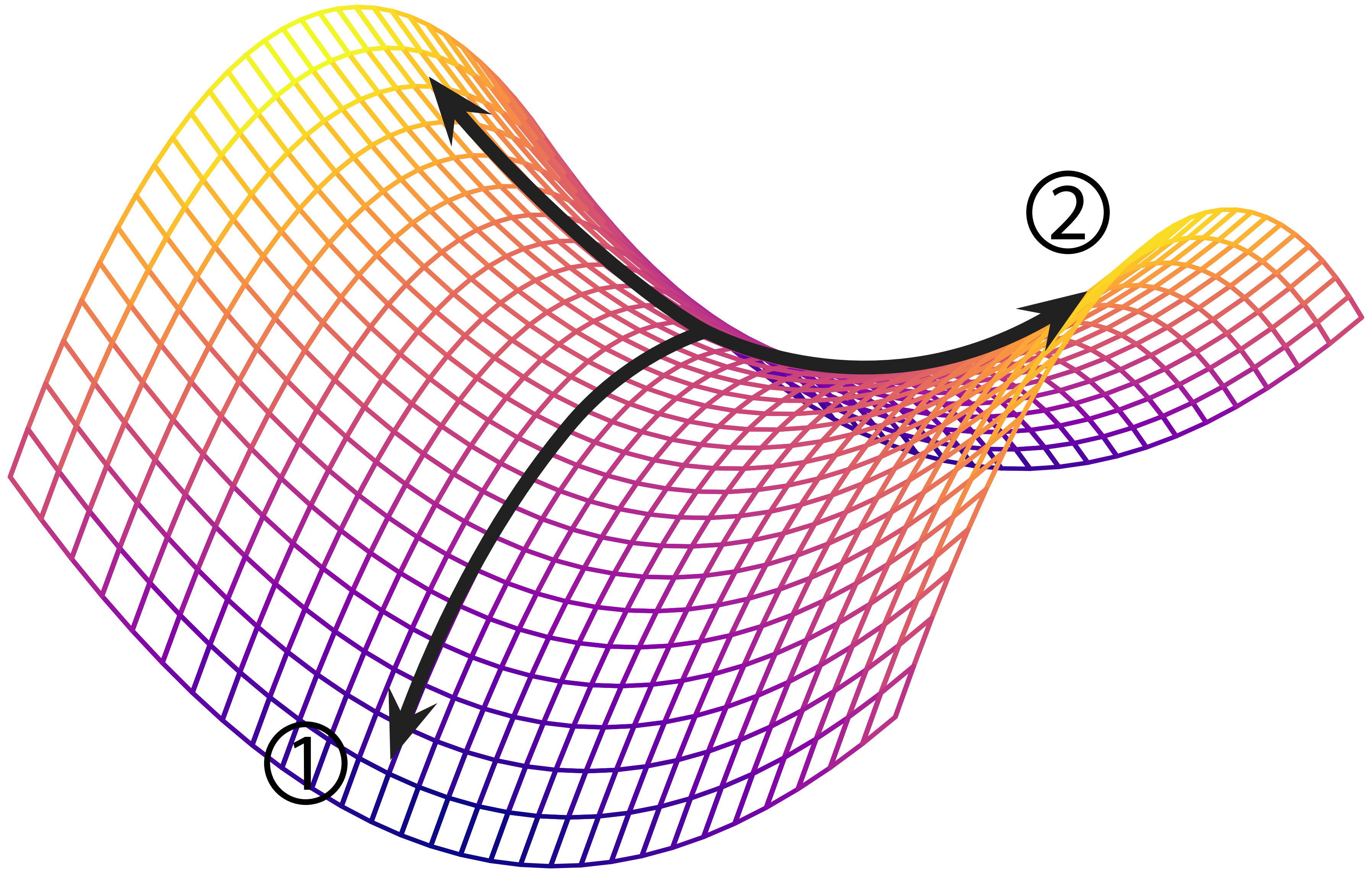
(a) Loss landscape visualizations (b) Trajectories in polar coordinate (c) Hessian max eigenvalue spectra

Three different aspects consistently show that MSAs flatten loss landscape. *Left*: Loss landscape visualizations show that ViT has a flatter loss than ResNet. ***Middle***: ViT converges to the optimum along a smooth trajectory. ***Right***: The magnitude of the Hessian eigenvalues of ViT is smaller than that of ResNet.

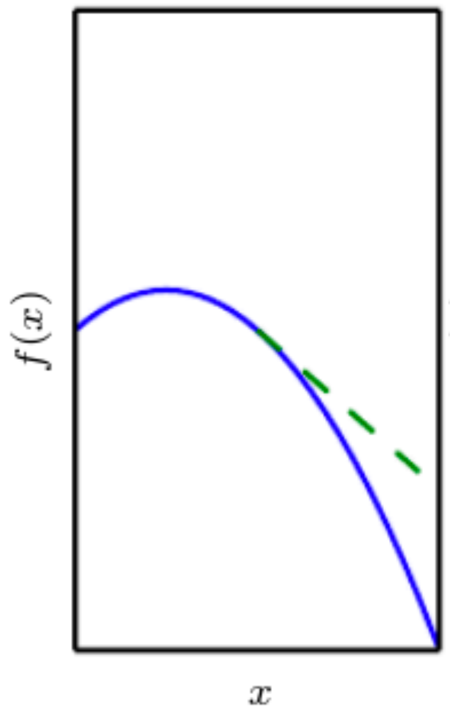
For loss landscape visualization, see arxiv.org/abs/1712.09913. Learning trajectory and Hessian max eigenvalue spectra are **first proposed** in these works. Compare to arxiv.org/abs/1901.10159.

Local Geometry in NN Landscapes

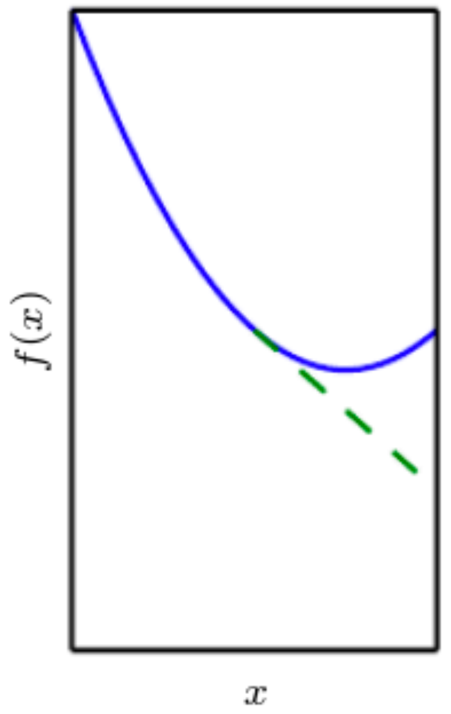
Saddle point
(non-convex)



① Negative curvature
(concave)



② Positive curvature
(convex)

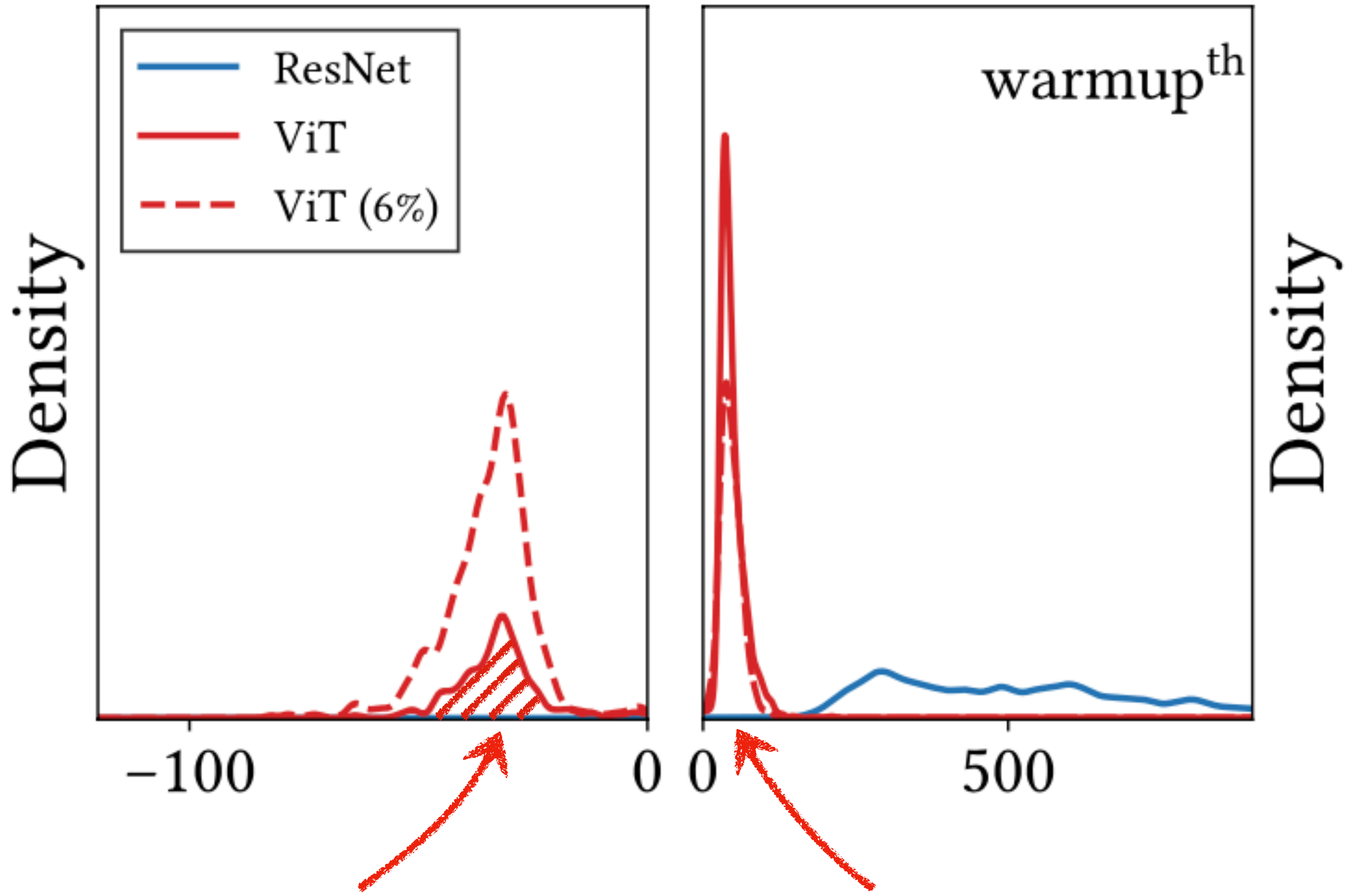


Hessian represents the local geometry of the loss landscapes.

- If Hessian is positive: convex
- If Hessian is indefinite: non-convex (e.g., saddle point)

Source: https://rohanvarma.me/optimizing-dnn/

Criteria For Sharpness and Non-convexity



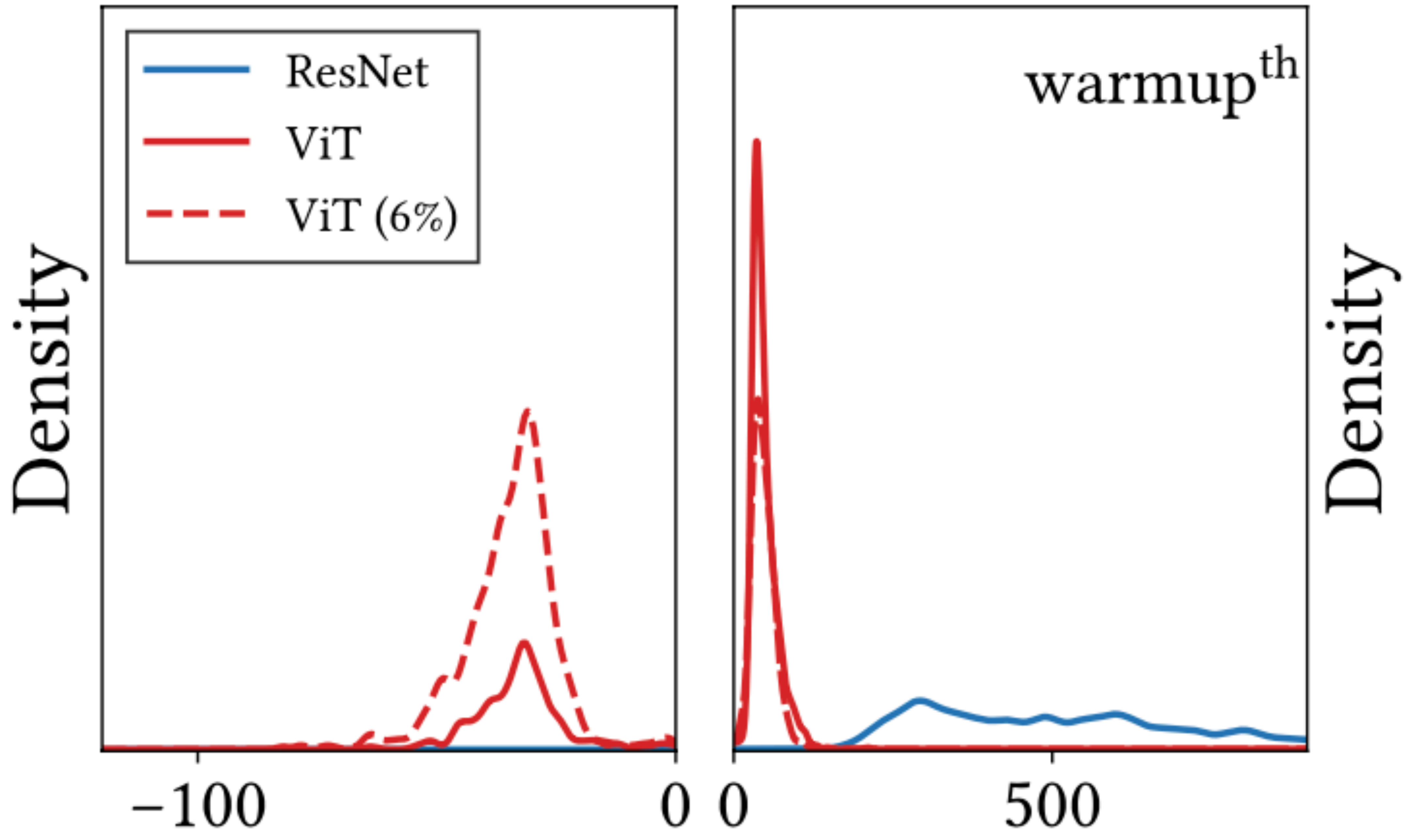
Non-convexity

Negative max Eigenvalue proportion

Sharpness

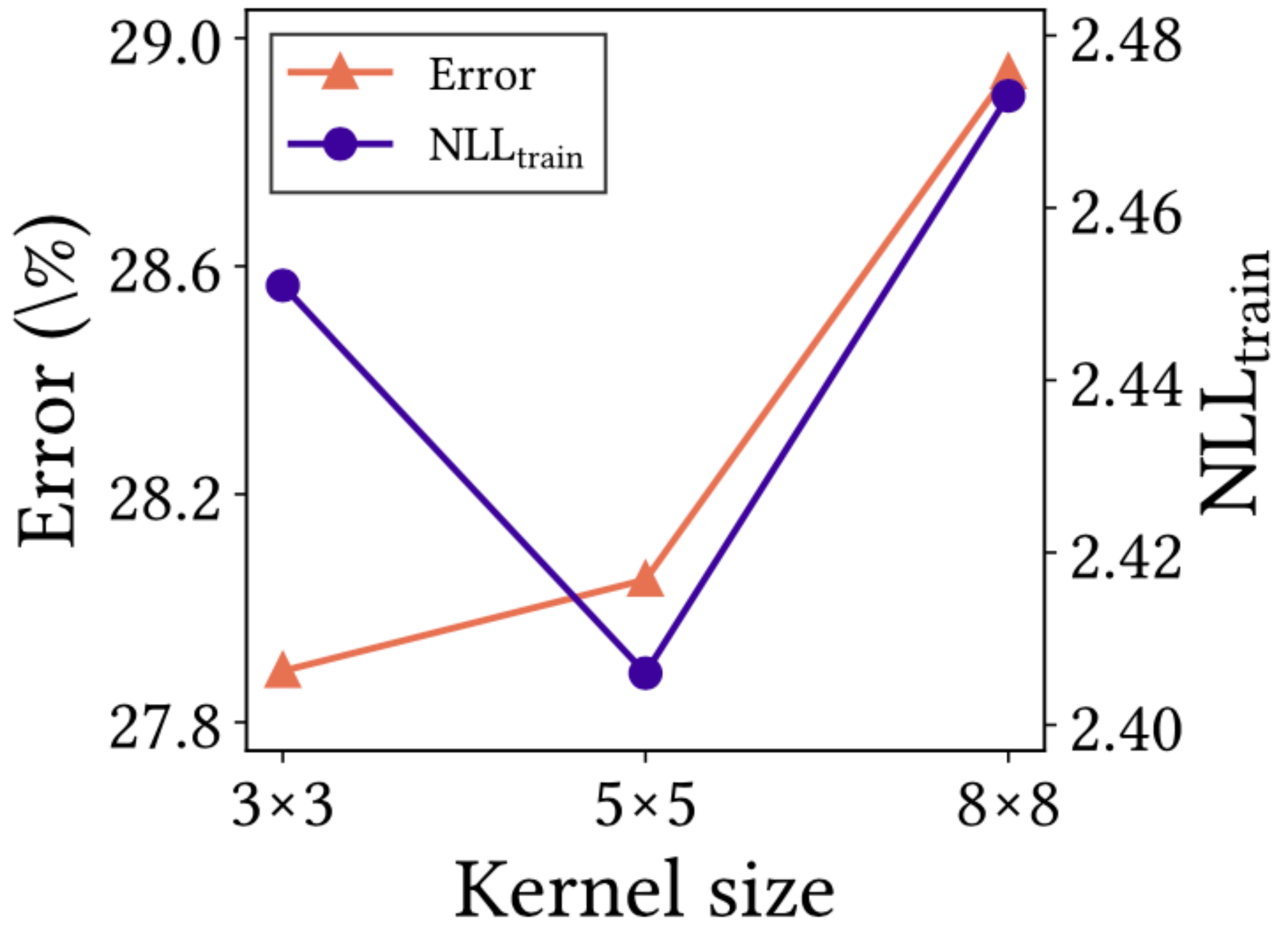
Average of Positive max Eigenvalues

ViT Has a Flat but Non-convex Loss Landscape

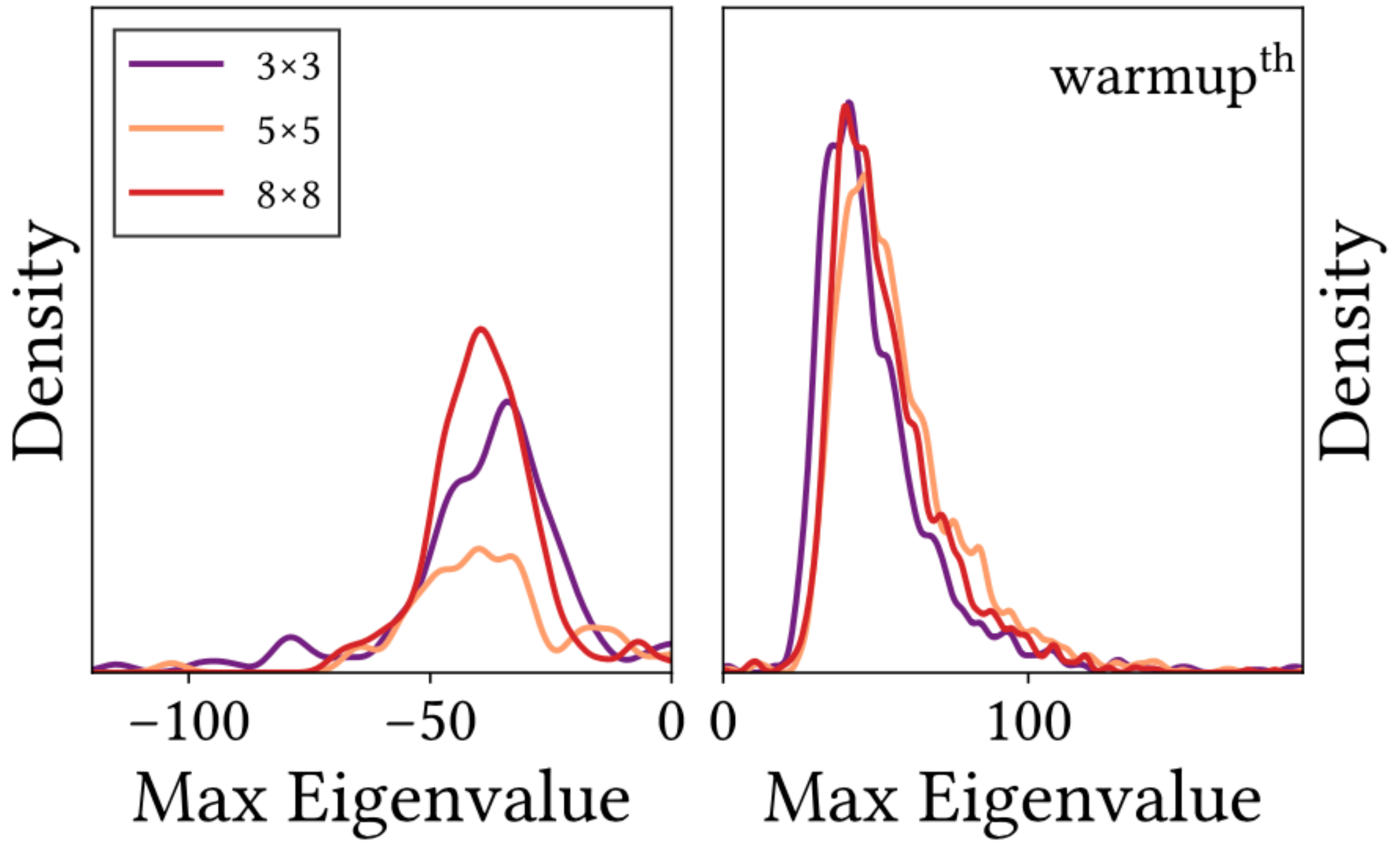


Hessian max eigenvalue spectra show that MSAs have their pros and cons. The dotted line is the spectrum of ViT using 6% dataset for training. ***Left***: ViT has a number of negative Hessian eigenvalues, while ResNet only has a few. ***Right***: The magnitude of ViT's positive Hessian eigenvalues is small.

A Key Feature Is *Not* Long-Range Dependency



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

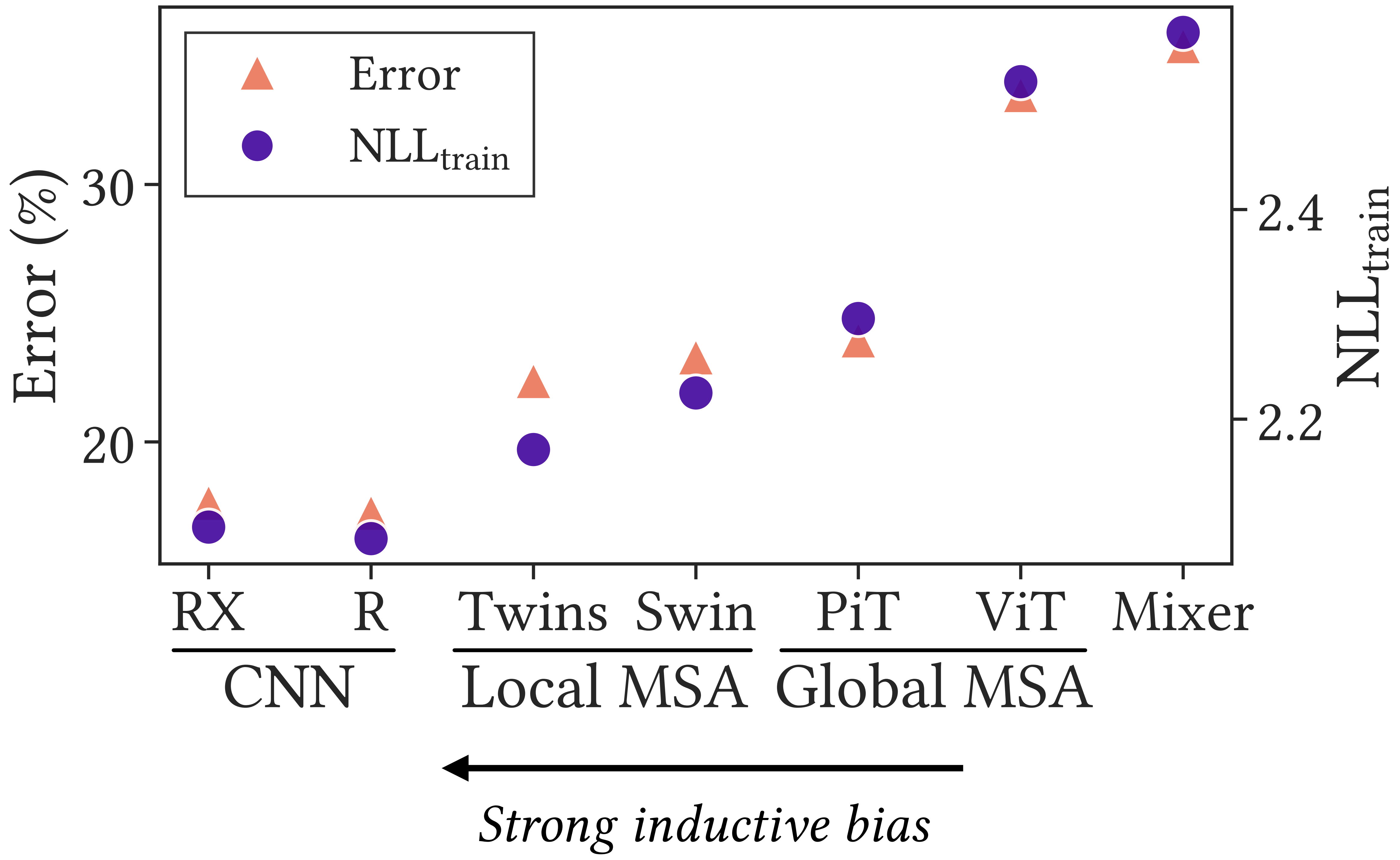


(b) Hessian negative and positive max eigenvalue spectra in early phase of training

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.

The Stronger the Inductive Biases, the Stronger the Reprs

NOT Regularizations!

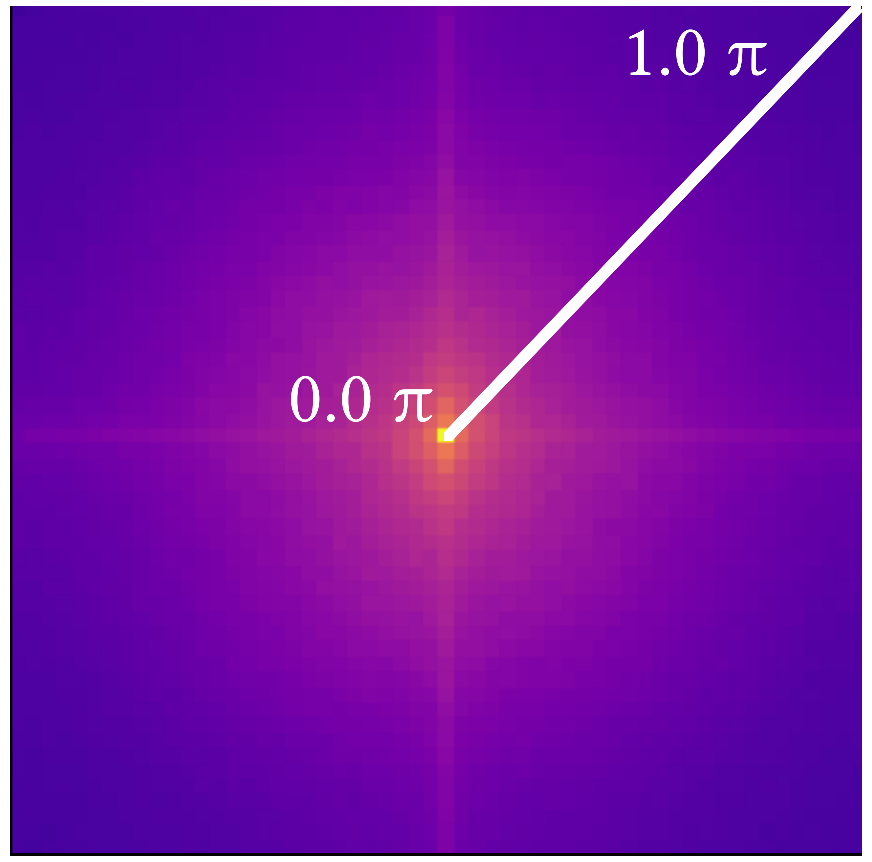


ViT does not overfit training datasets. The lower the $\mathrm{NLL}_{\mathrm{train}}$, the lower the error.
It implies that "weak inductive bias (e.g. long-range dependency) disturbs NN optimization".

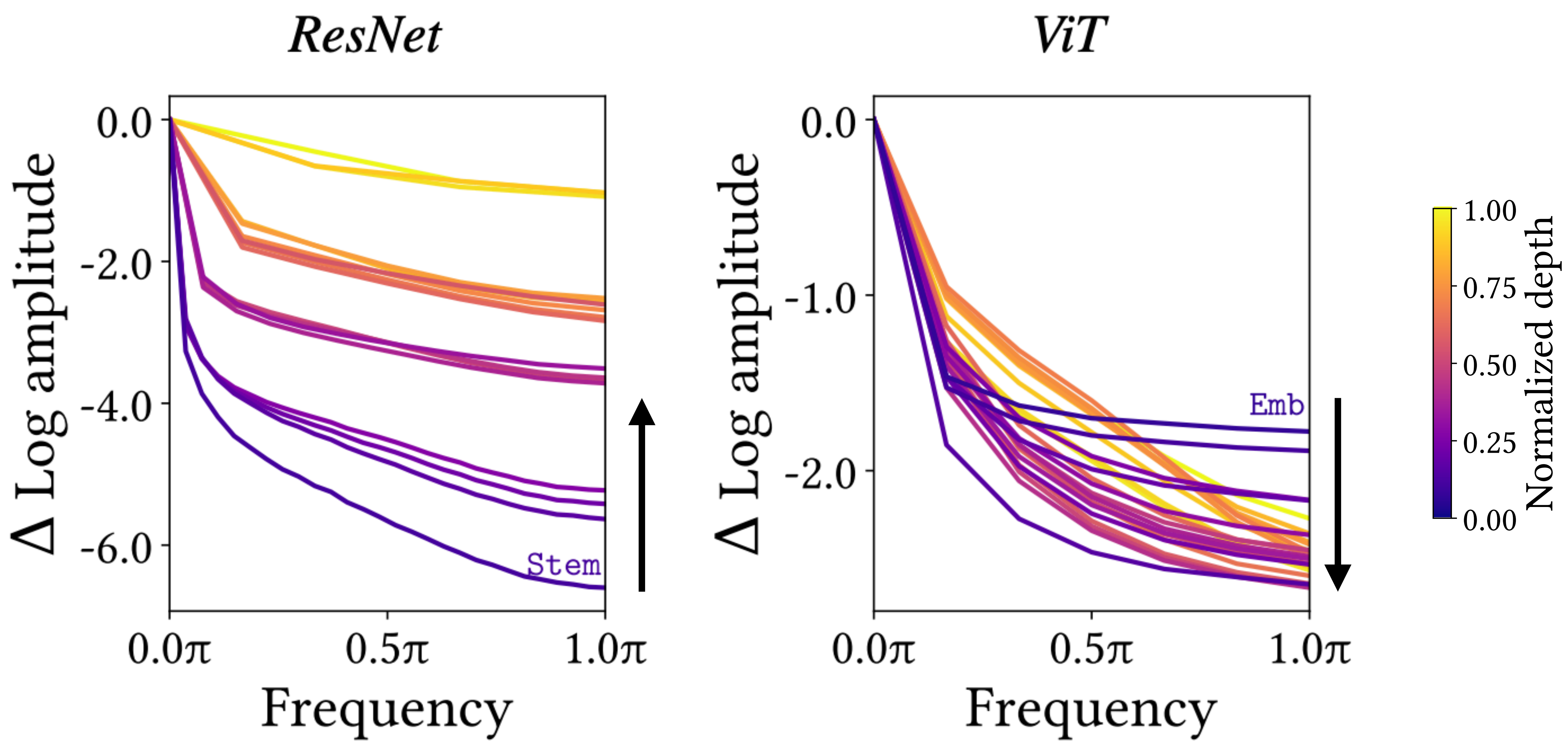
Question Two

Do MSAs Act Like Convs?

MSAs Are Low-Pass Filters, but Convs Are High-Pass Filters



(a) Fourier transformed feature maps

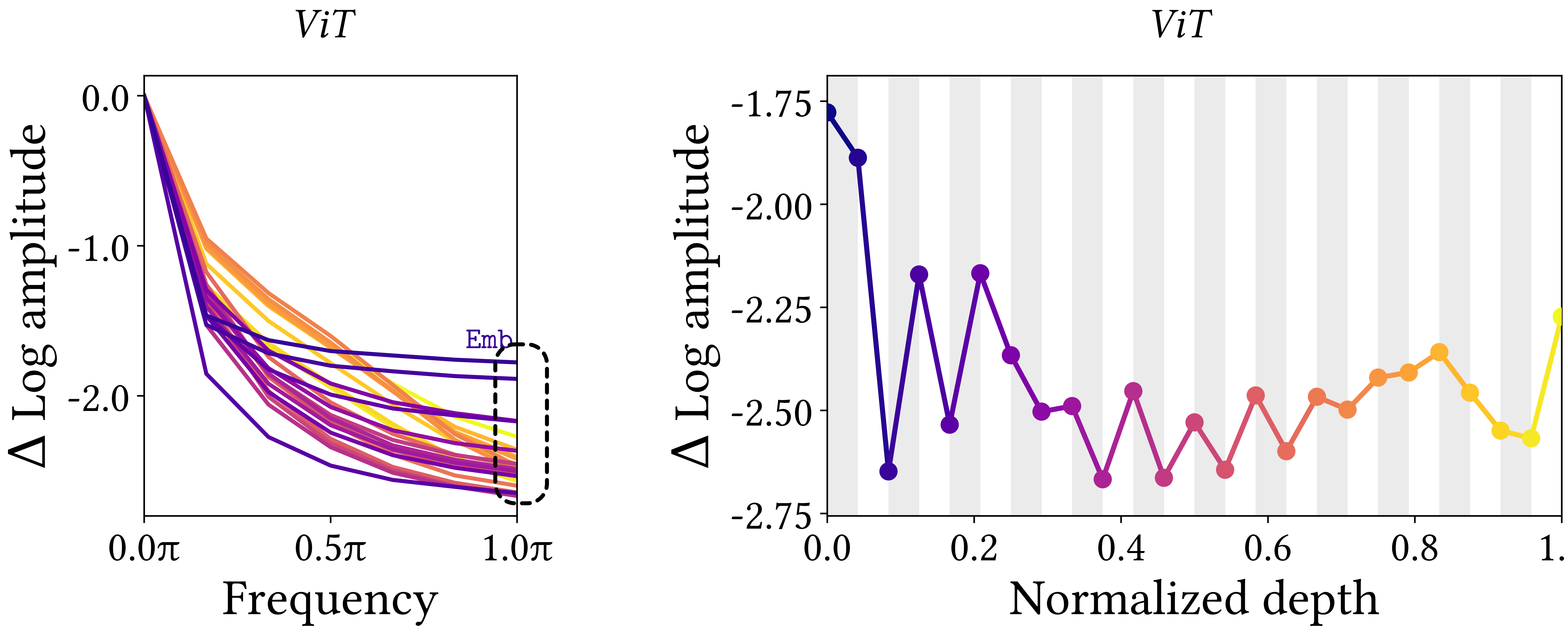


(b) Relative log amplitudes of Fourier transformed feature maps

The Fourier analysis shows that MSAs do not act like Convs. Relative log amplitudes of Fourier transformed feature map show that ViT tends to reduce high-frequency signals, while ResNet amplifies them.

Fourier analysis of feature maps is **first proposed** in these works. Compare to arxiv.org/abs/1906.08988 & arxiv.org/abs/2105.03824.

MSAs Are Low-Pass Filters, but Convs Are High-Pass Filters

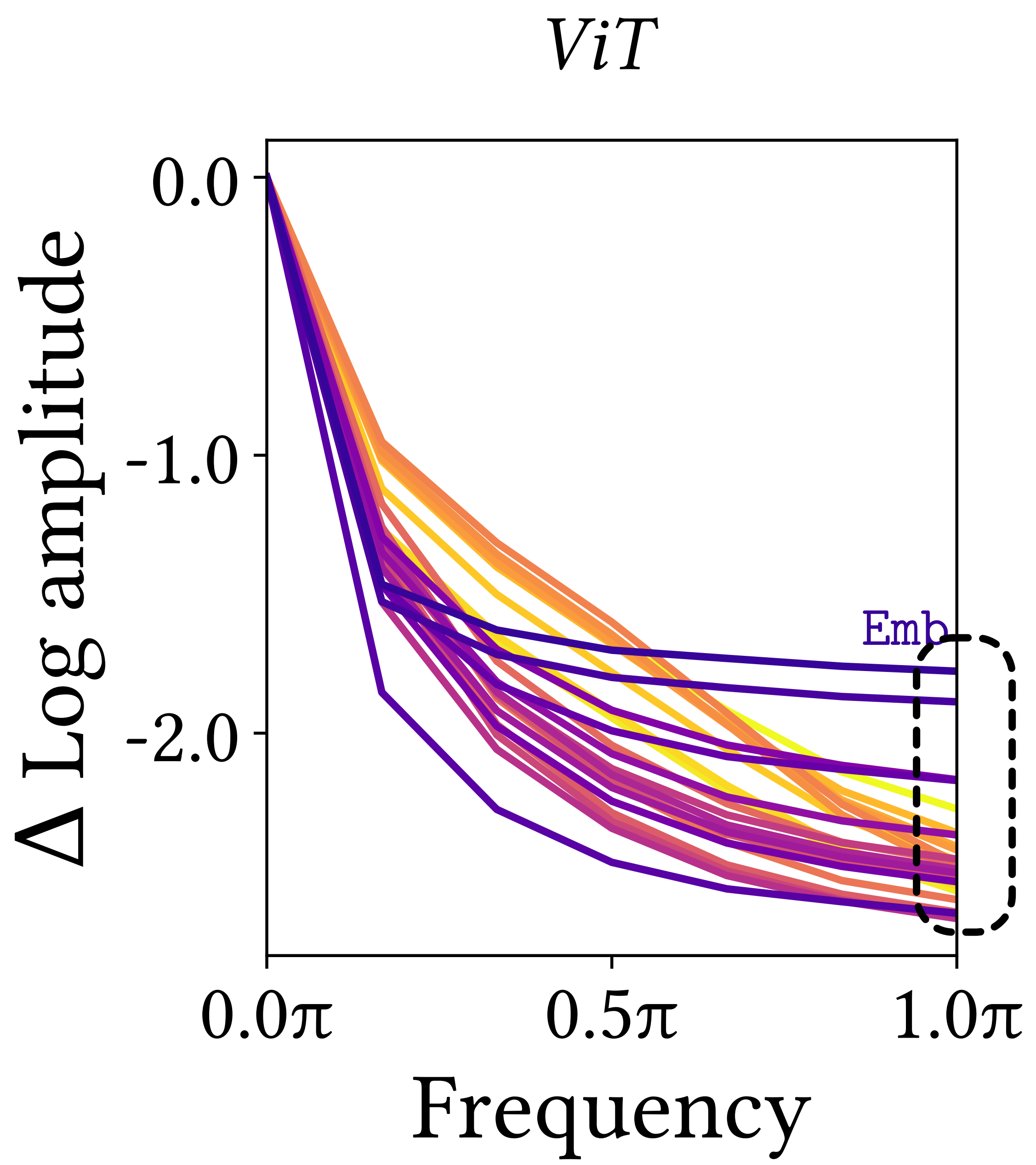


(a) Δ Log amplitude for frequency.

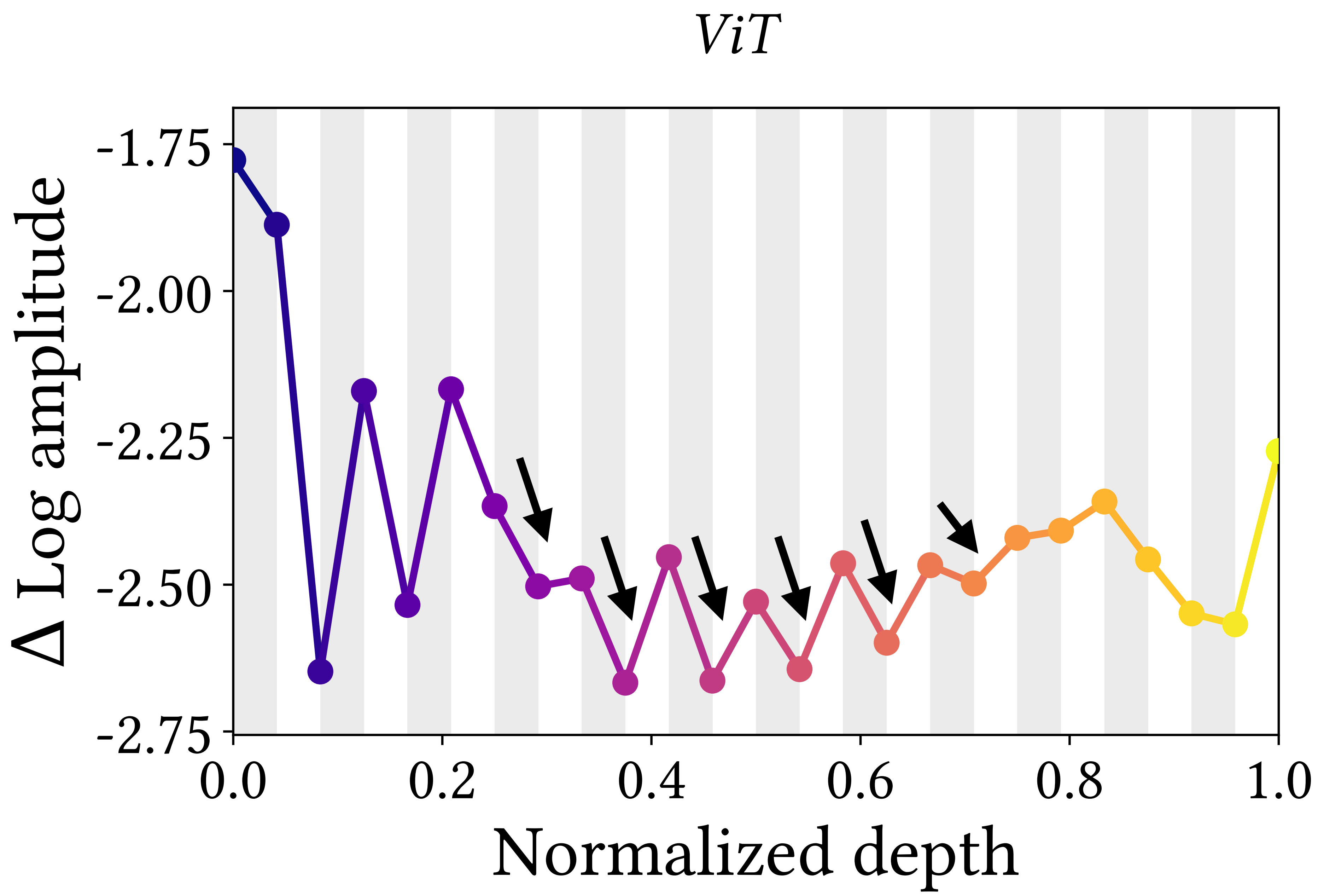
(b) Δ Log amplitude at high-frequency 1.0 π.

MSAs (gray area ▨) generally reduce the high-frequency component, and MLPs (white area □) amplify it. It implies that low-frequency signals and high-frequency signals are informative to MSAs and Convs, respectively.

MSAs Are Low-Pass Filters, but Convs Are High-Pass Filters



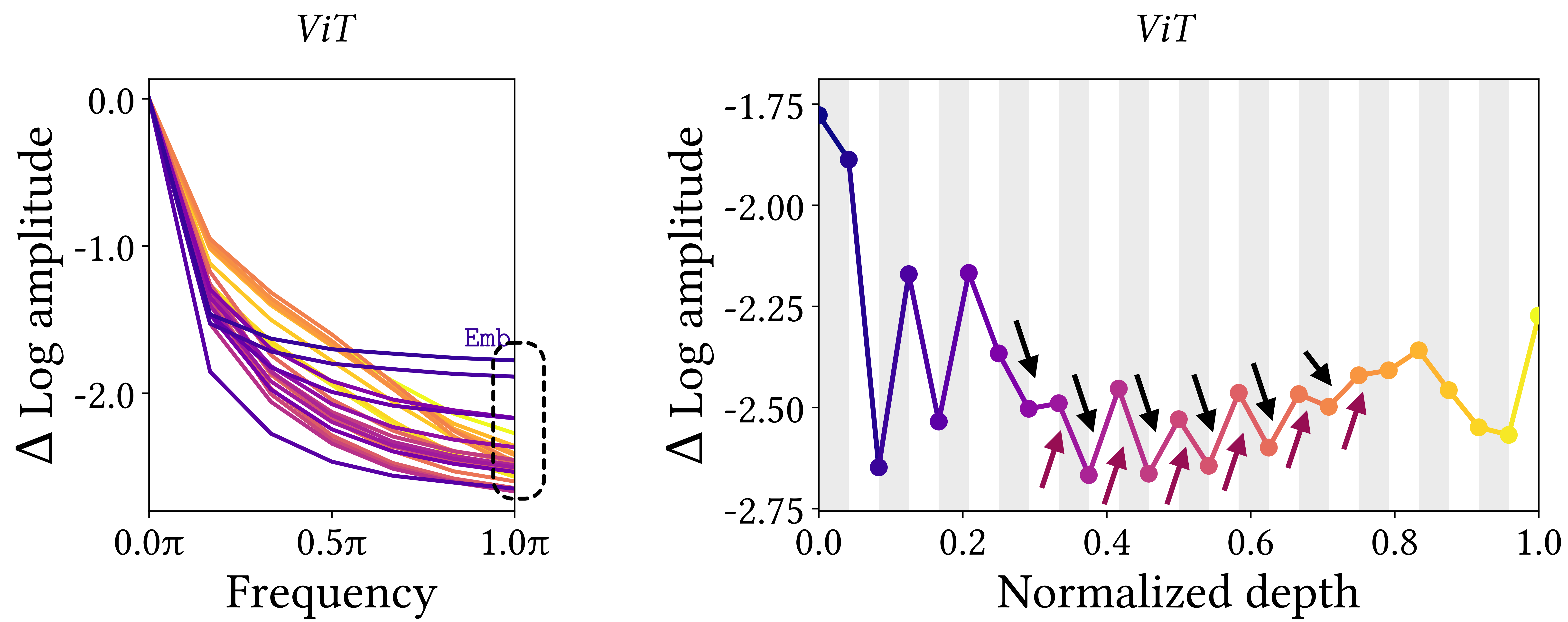
(a) Δ Log amplitude for frequency.



(b) Δ Log amplitude at high-frequency 1.0 π.

MSAs (gray area ▨) generally reduce the high-frequency component, and MLPs (white area □) amplify it. It implies that low-frequency signals and high-frequency signals are informative to MSAs and Convs, respectively.

MSAs Are Low-Pass Filters, but Convs Are High-Pass Filters

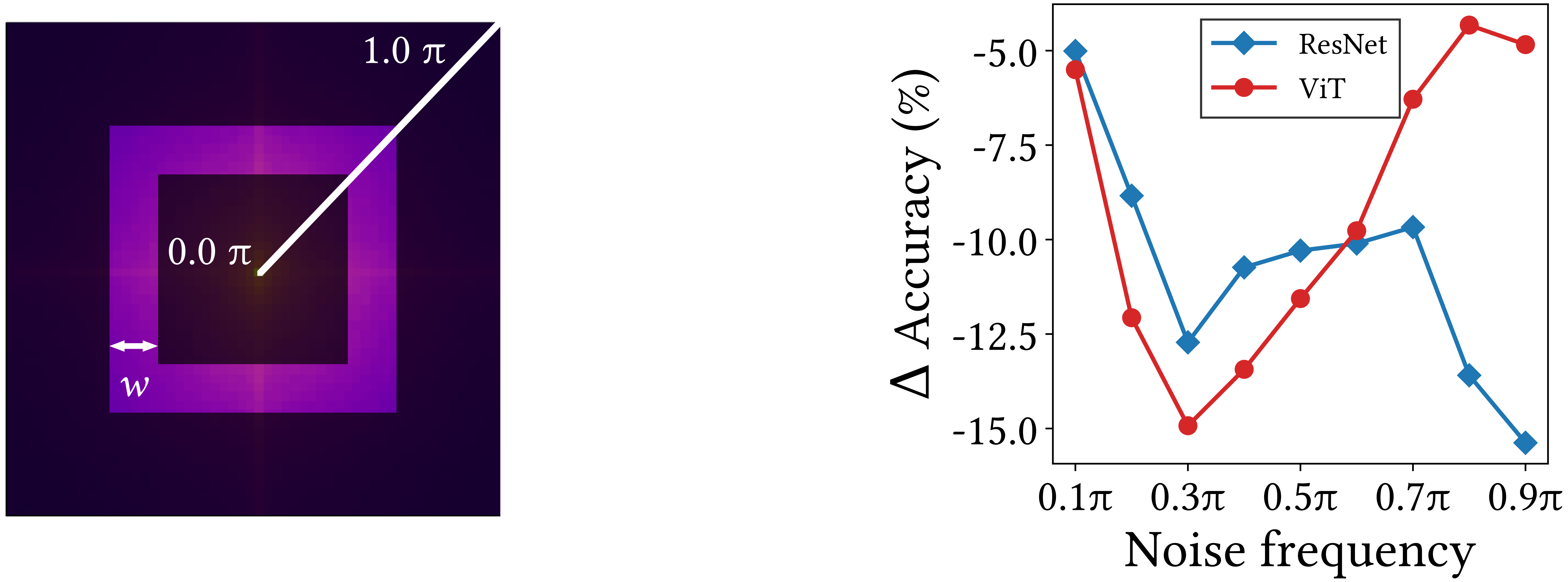


(a) Δ Log amplitude for frequency.

(b) Δ Log amplitude at high-frequency 1.0 π.

MSAs (gray area □) generally reduce the high-frequency component, and MLPs (white area □) amplify it. It implies that low-frequency signals and high-frequency signals are informative to MSAs and Convs, respectively.

ViT Is Robust Against High-Frequency Noise

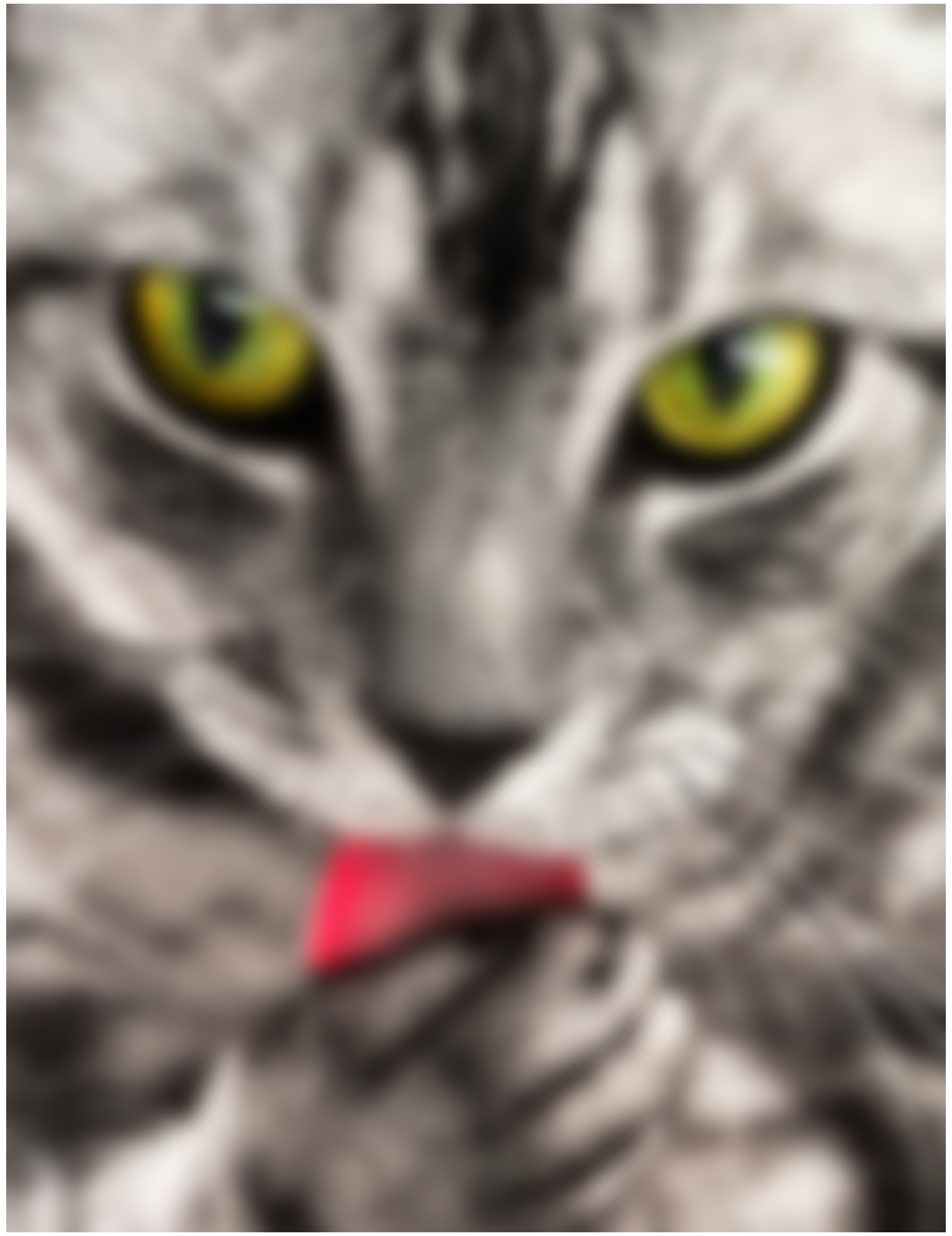


(a) Frequency-based random noise

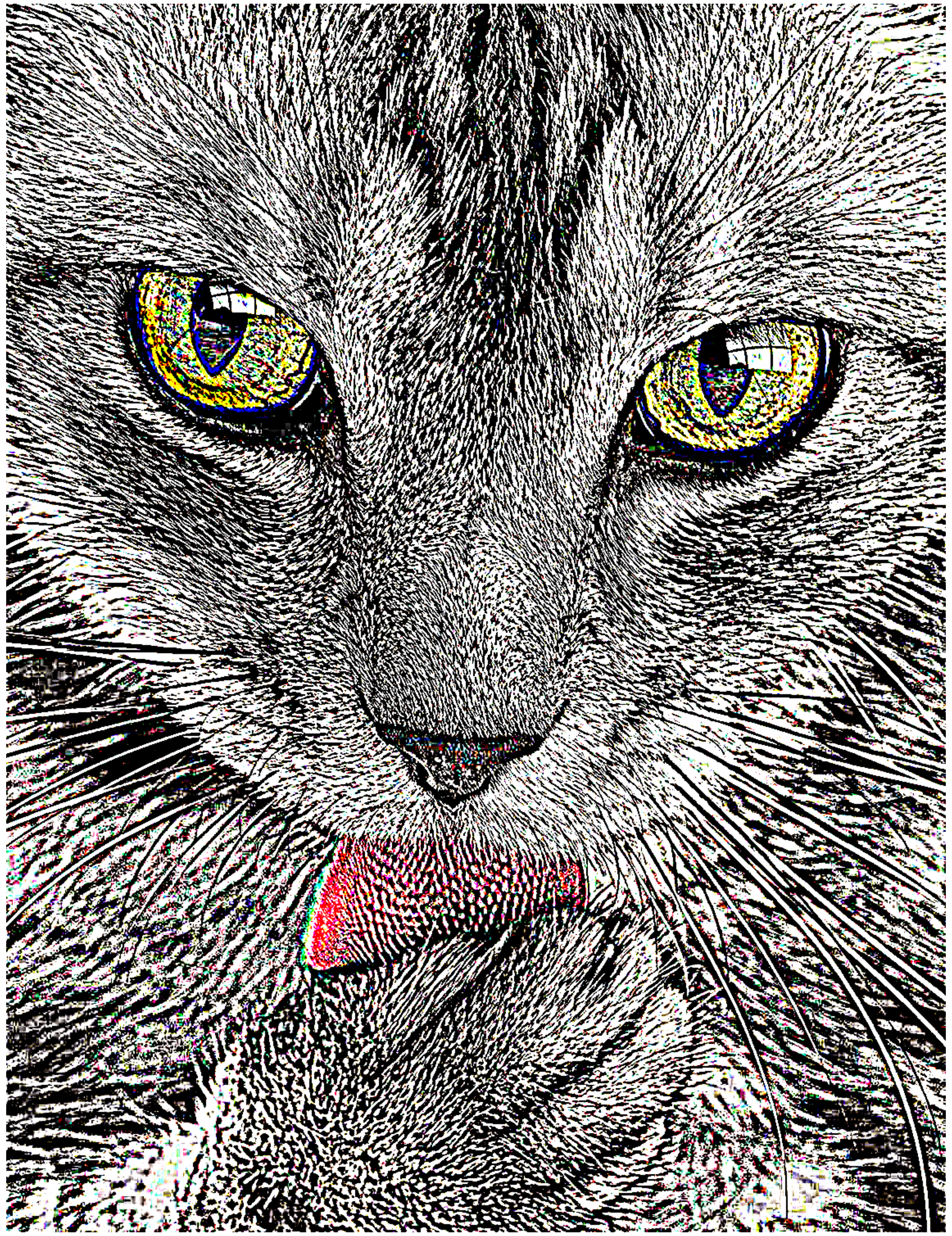
(b) Robustness for noise frequency

The Fourier analysis shows that MSAs do not act like Convs. We measure the decrease in accuracy against frequency-based random noise. ResNet is vulnerable to high-frequency noise, while ViT is robust against them.

arxiv.org/abs/2103.15670 also figured out that ViTs are robust against high-frequency adversarial attacks.

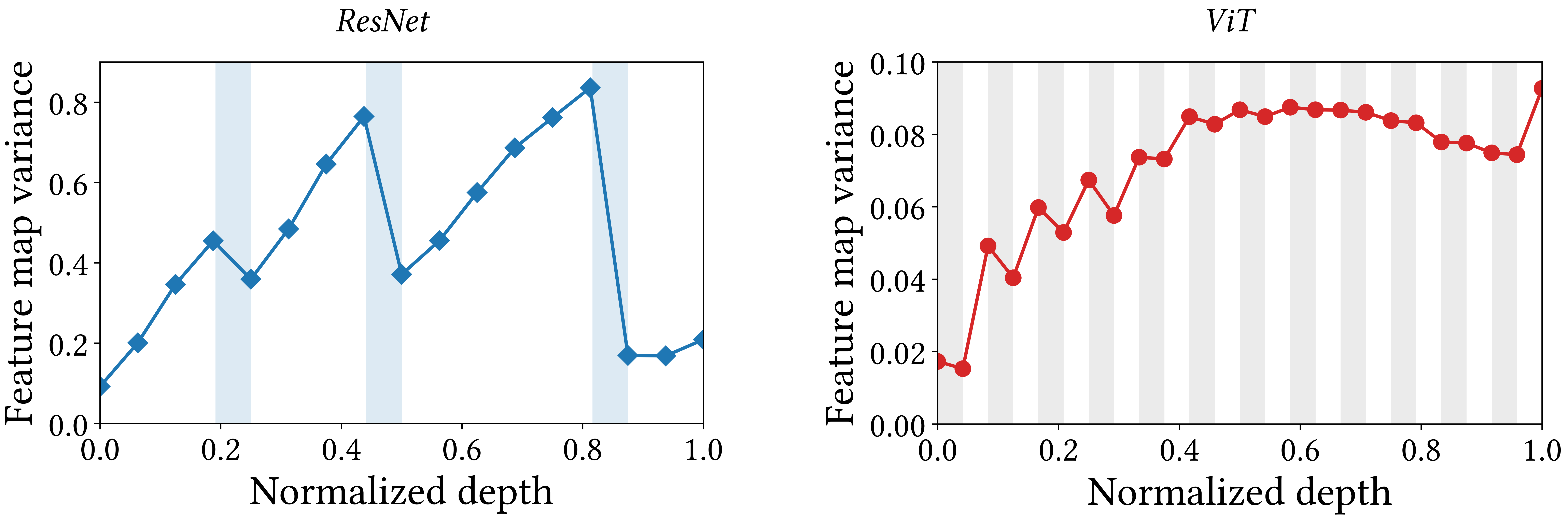


MSA is shape-biased.



Conv is texture-biased.

MSAs Aggregate Feature Maps, but Convs Do Not



MSAs (gray area ▨) reduce the variance of feature map points, but Convs/MLPs (white area □) increase the variance. The blue area (🟦) is subsampling layer.

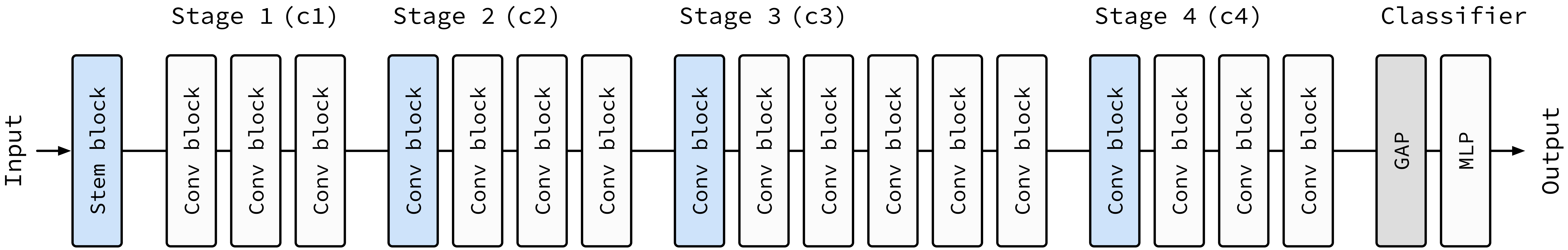
The results implies that MSAs aggregate feature maps, and Convs transform them.

Question Three

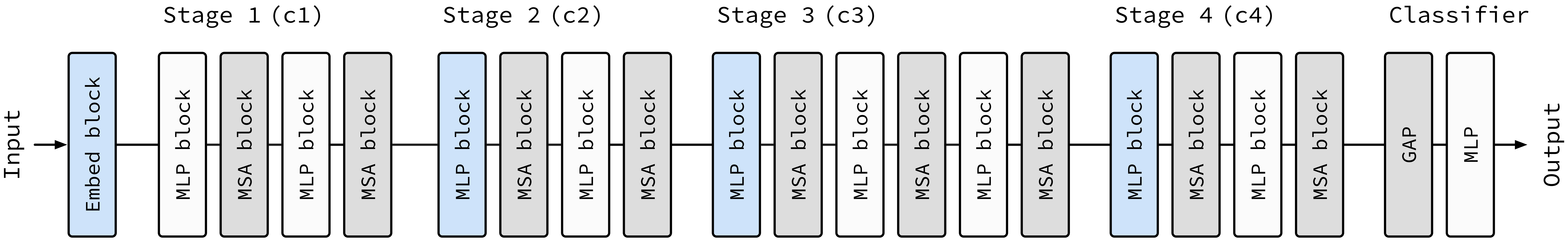
How Can We Harmonize MSAs With Convs?

Architecture of Swin Transformer

ResNet

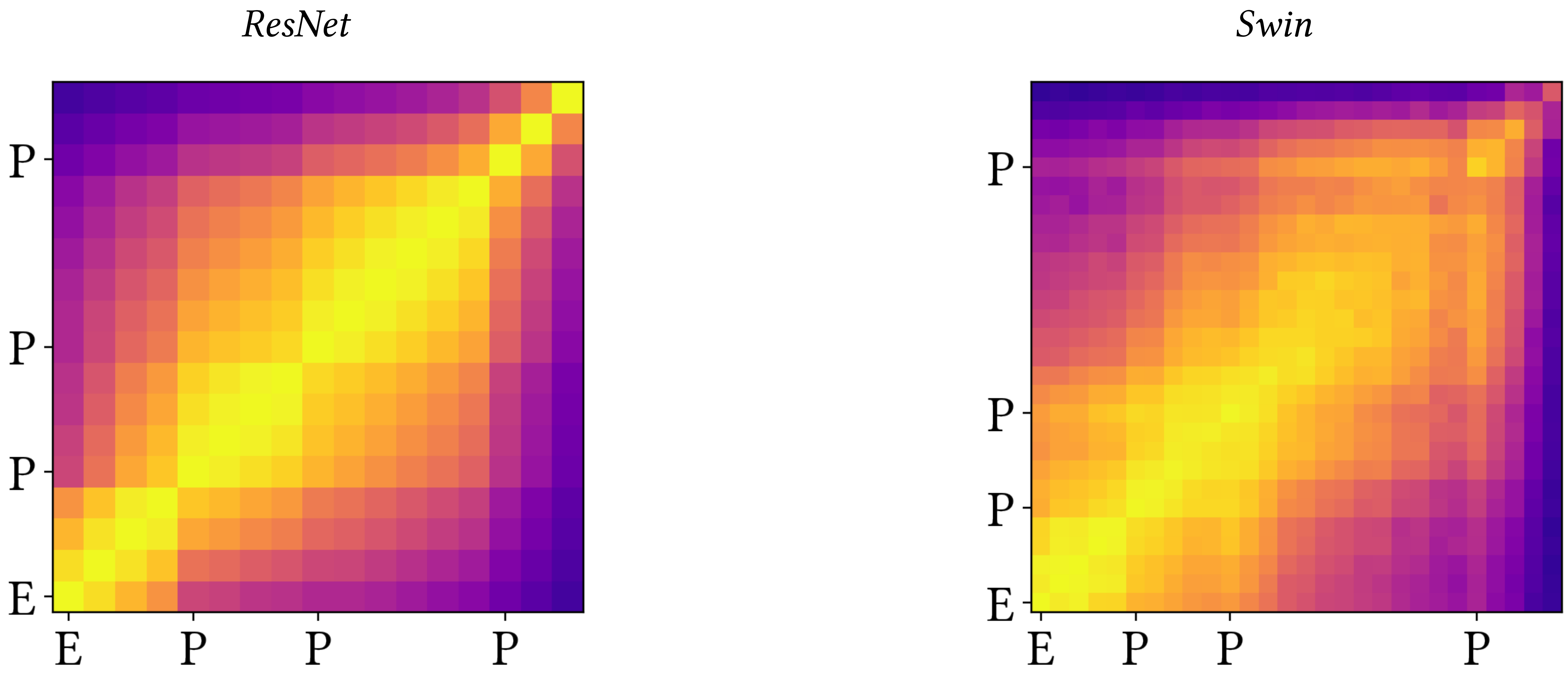


Swin



See arxiv.org/abs/2103.14030 for more details on Swin Transformer.

Multi-Stage NNs Have Representational Block Structures



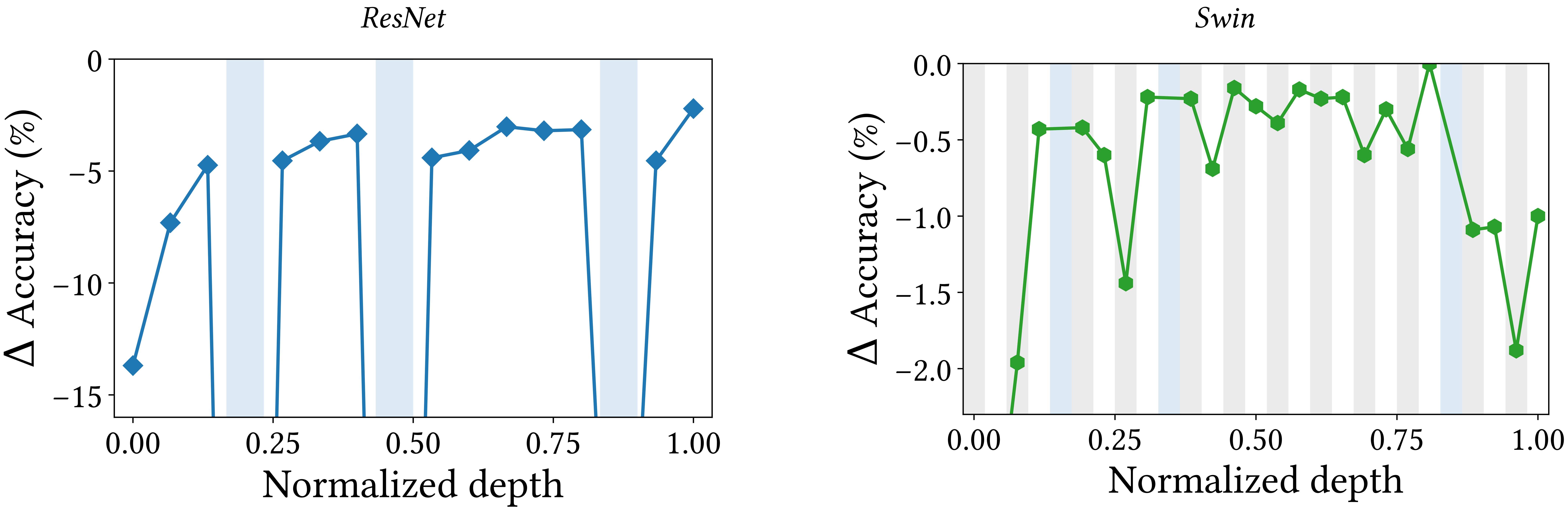
ResNet

Swin

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.

See arxiv.org/abs/2010.15327 for more details on representational similarity.

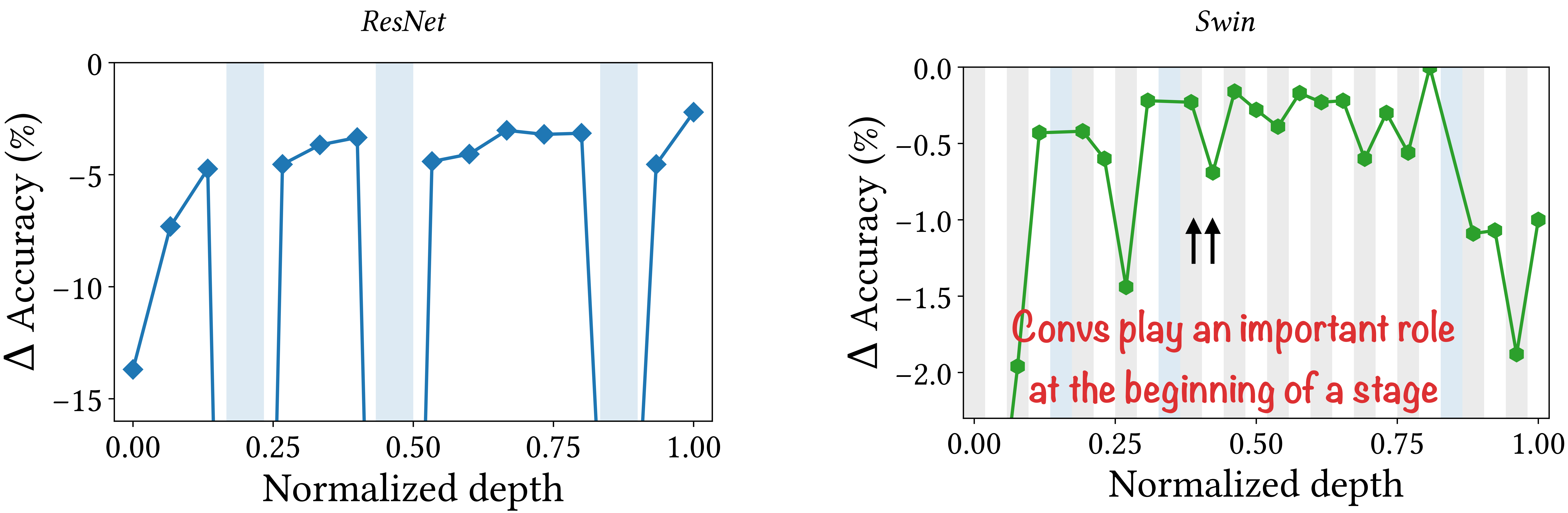
Lesion Studies Show the Periodicity of the Accuracy Drop



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.

https://arxiv.org/abs/2103.14586 is first reported the results of lesion studies on ViTs. We expand the analysis.

Lesion Studies Show the Periodicity of the Accuracy Drop

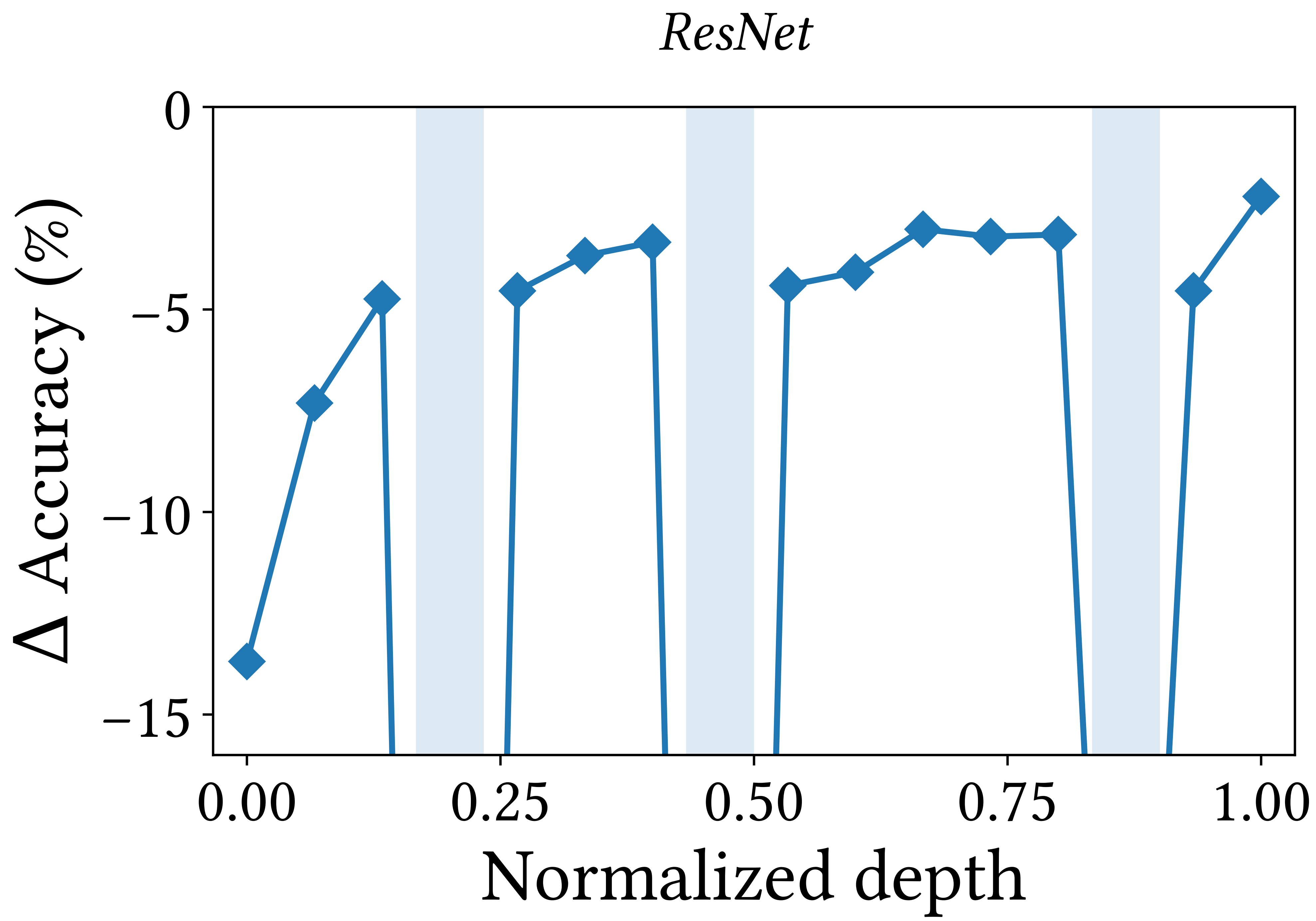


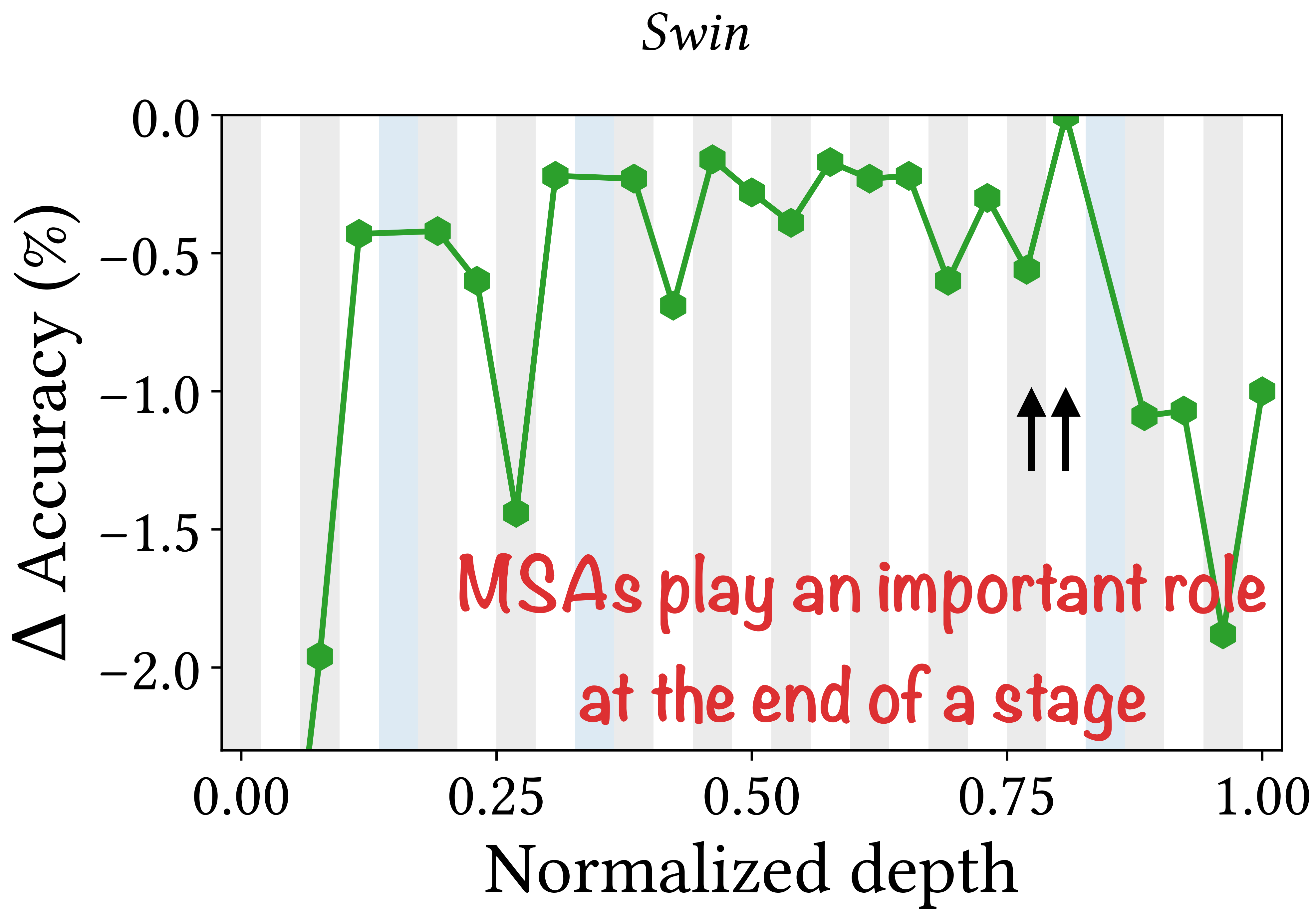
Convs play an important role at the beginning of a stage

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.

https://arxiv.org/abs/2103.14586 is first reported the results of lesion studies on ViTs. We expand the analysis.

Lesion Studies Show the Periodicity of the Accuracy Drop

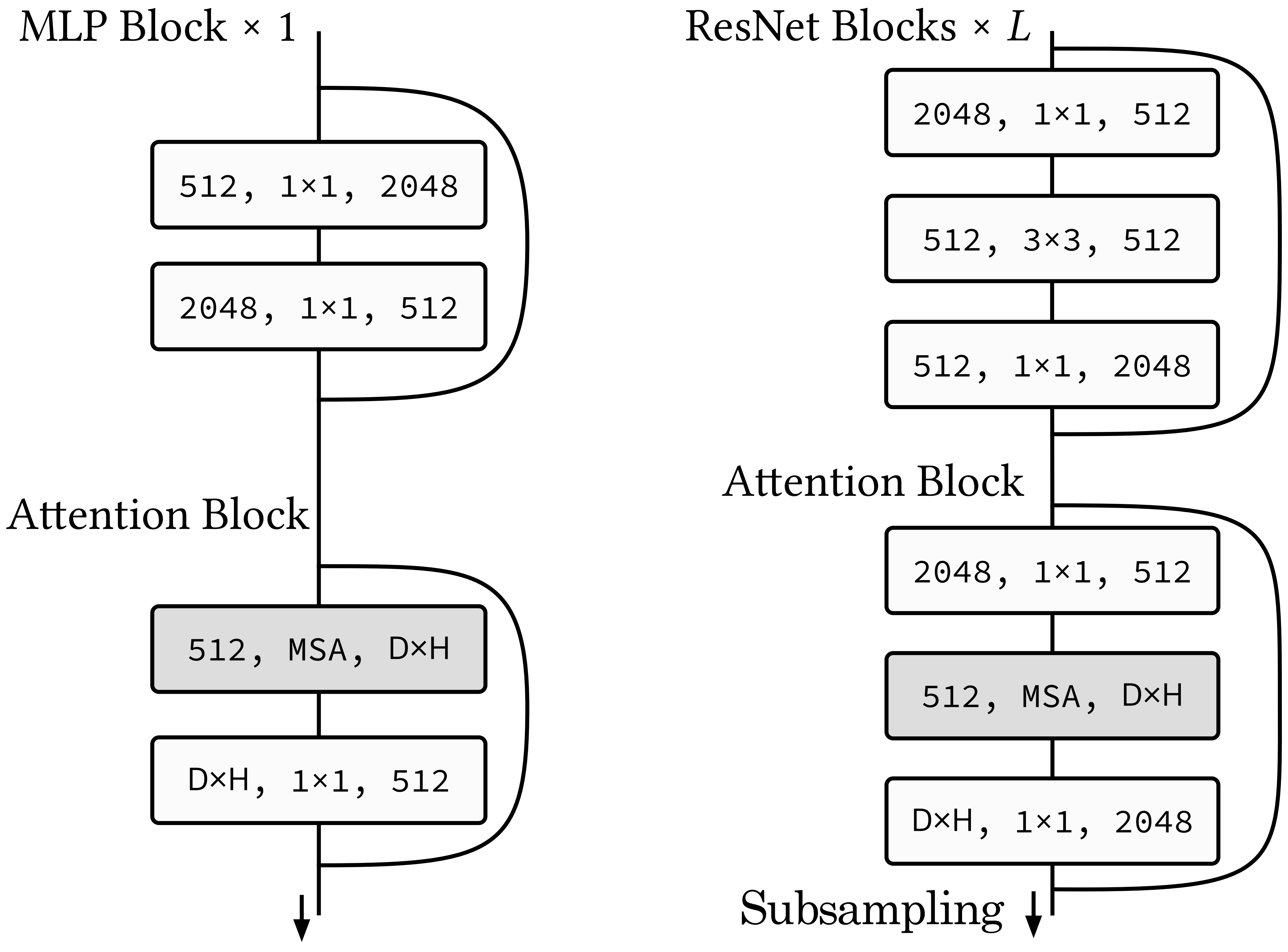




MSAs closer to the end of a stage significantly improve the performance. We measure decrease in accuracy after removing one unit from the trained model. Accuracy changes periodically, and this period is one stage.

https://arxiv.org/abs/2103.14586 is first reported the results of lesion studies on ViTs. We expand the analysis.

Alternating Pattern of Convs and MSAs

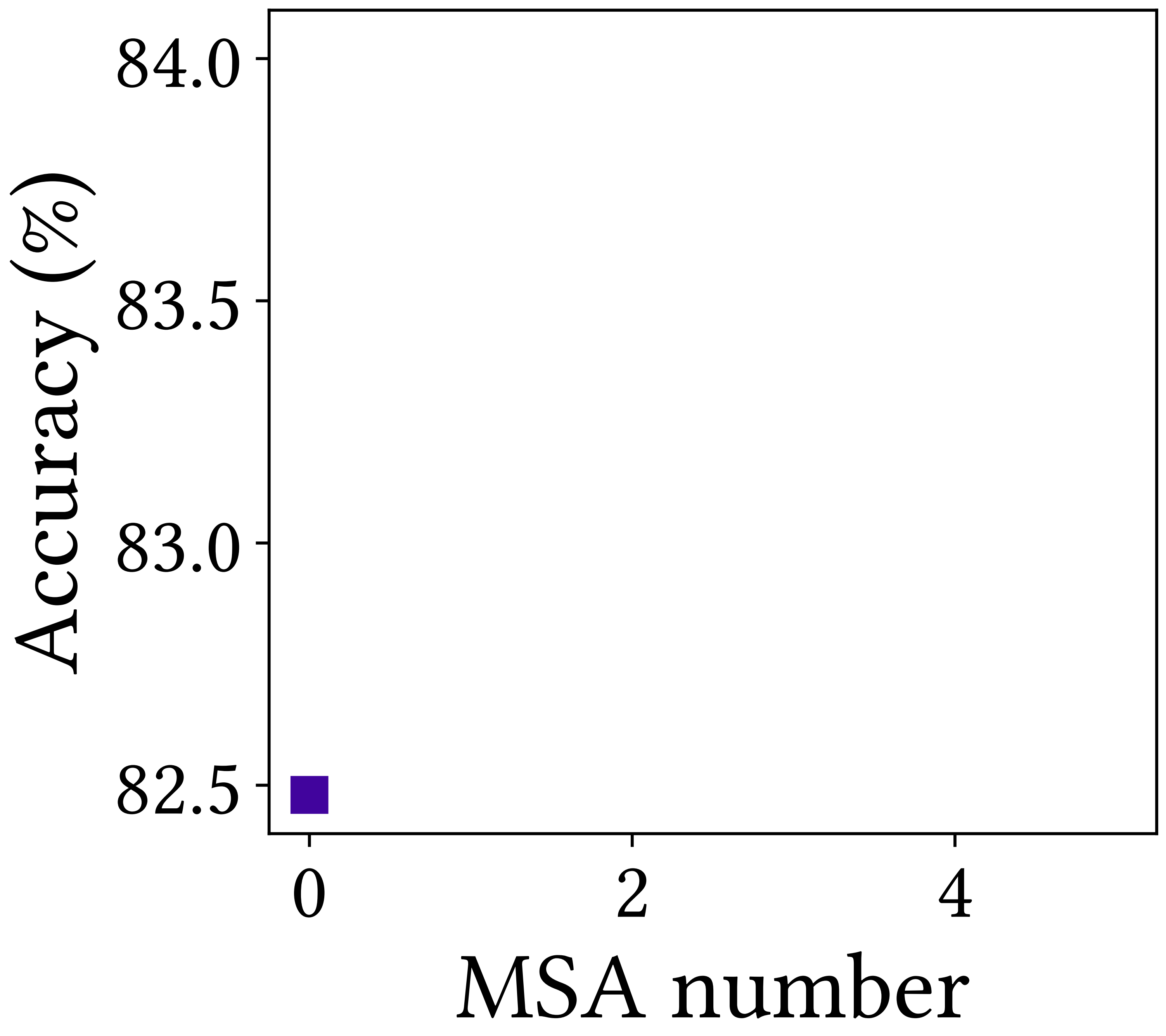


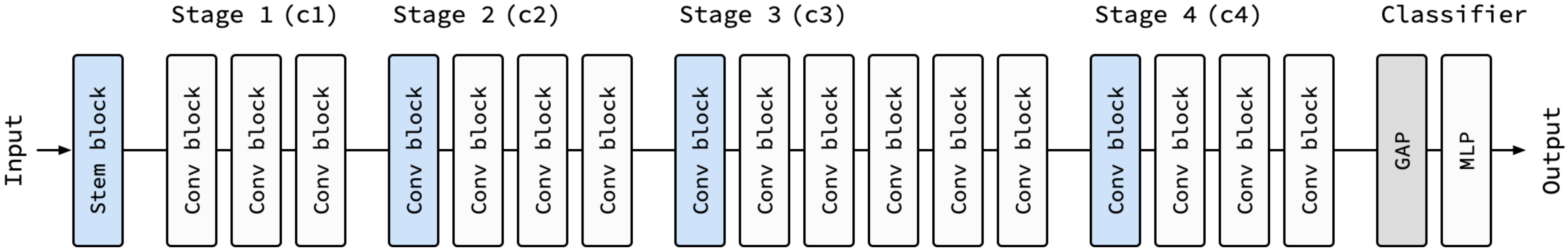
(a) Canonical Transformer (b) Alternating pattern

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

Build-Up Rule: How to Apply MSA to Your Own CNNs

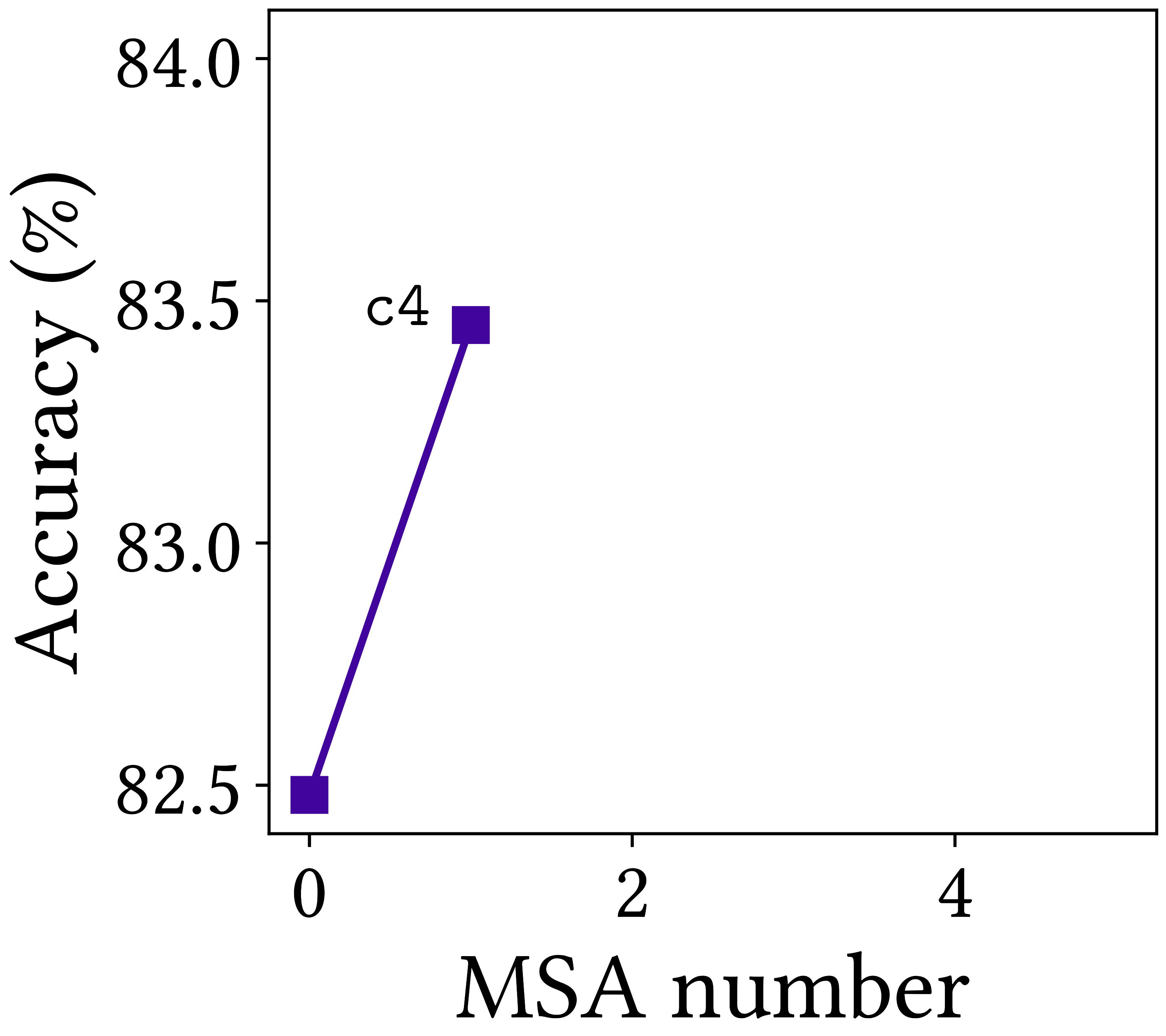
- Alternately replace Conv blocks with MSA blocks from the end of a baseline CNN model.
- If the added MSA block does not improve predictive performance, replace a Conv block located at the end of an earlier stage with an MSA block .
- Use more heads and higher hidden dimensions for MSA blocks in late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

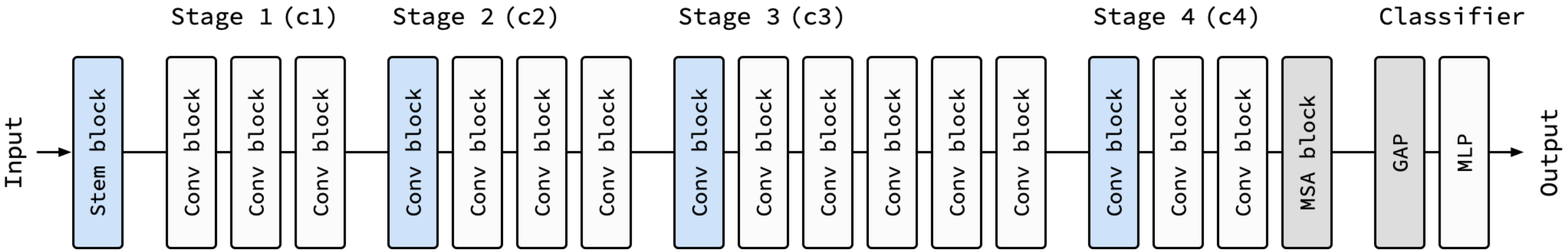




Stage 1 (c1) Stage 2 (c2) Stage 3 (c3) Stage 4 (c4) Classifier

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 — Conv block — Conv block — Conv block — GAP — MLP → Output





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

Conv block

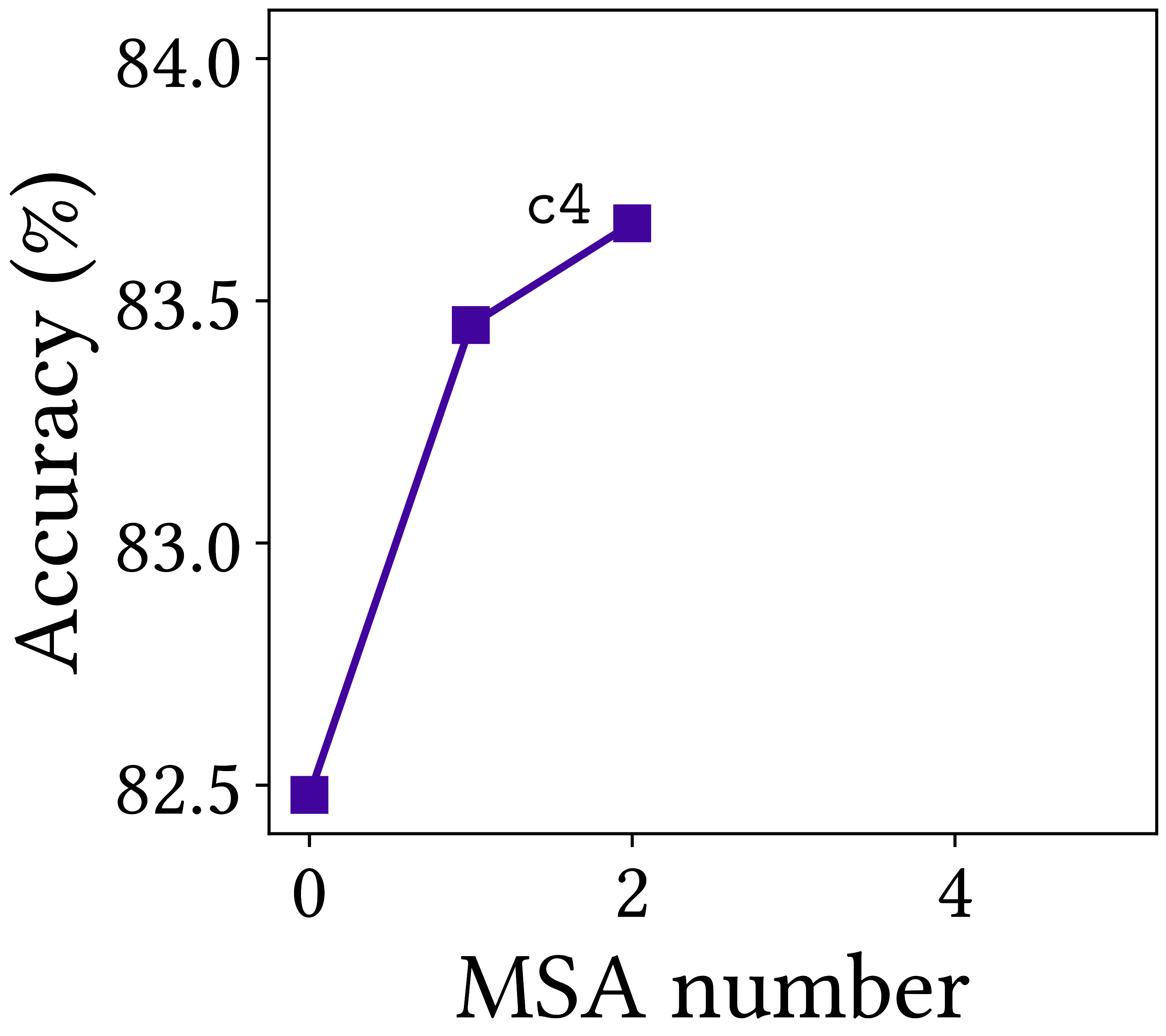
Conv block

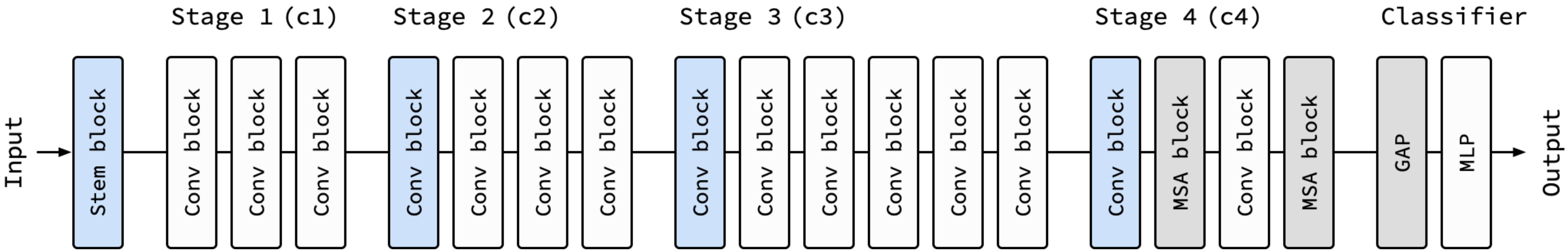
MSA block

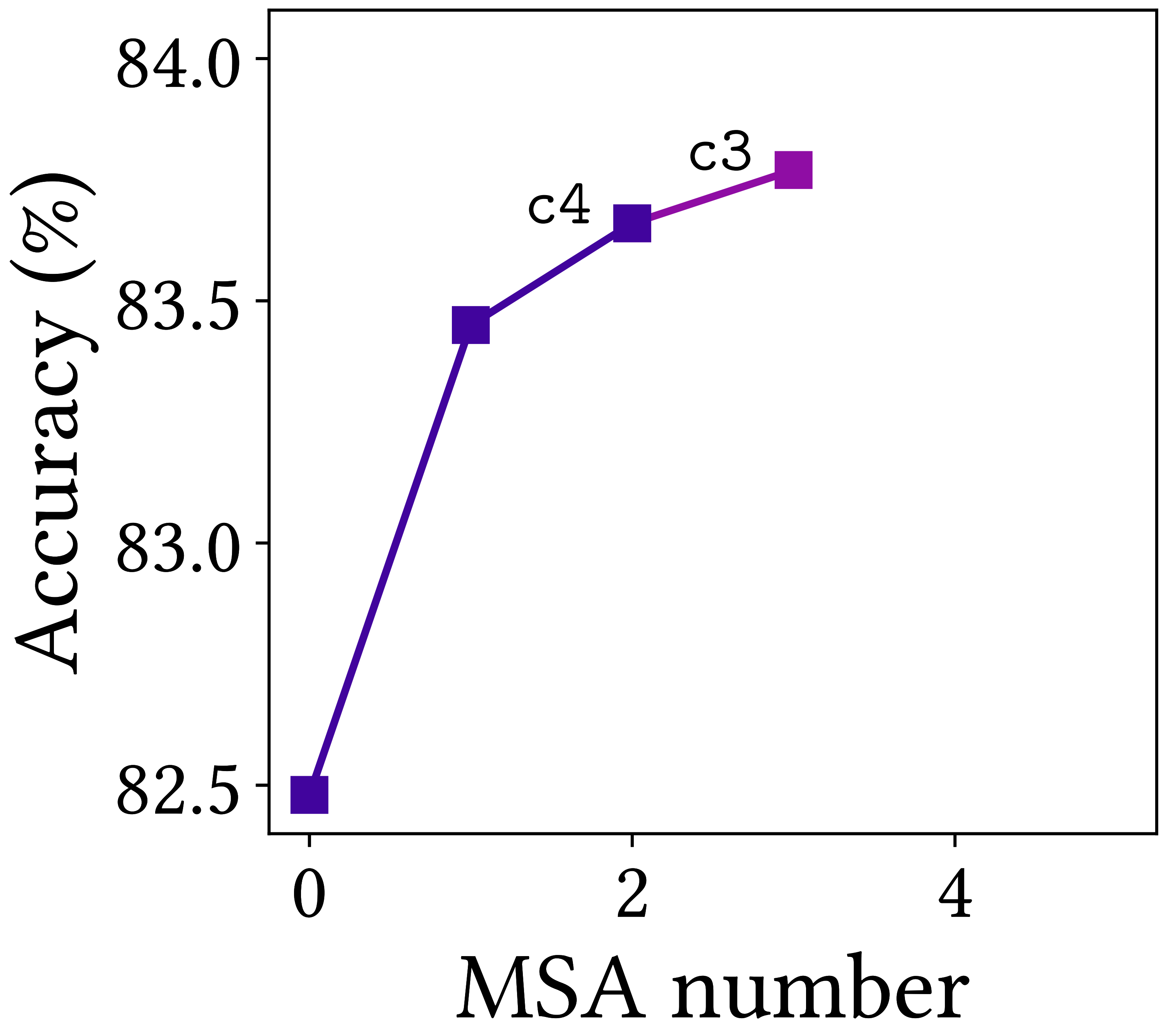
GAP

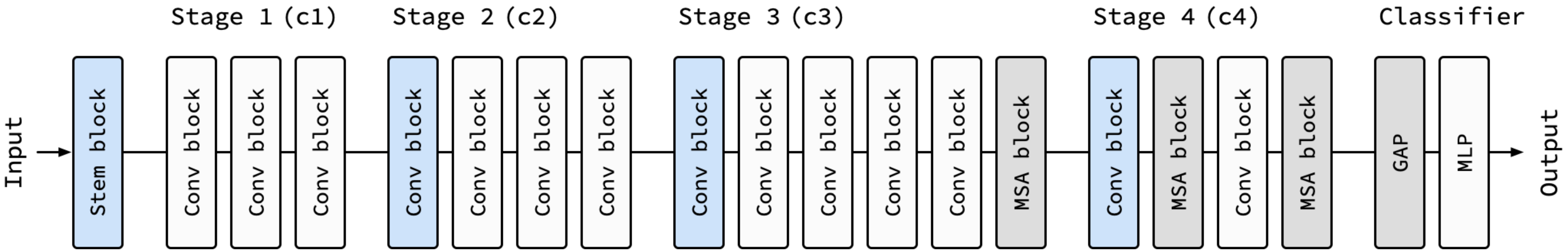
MLP

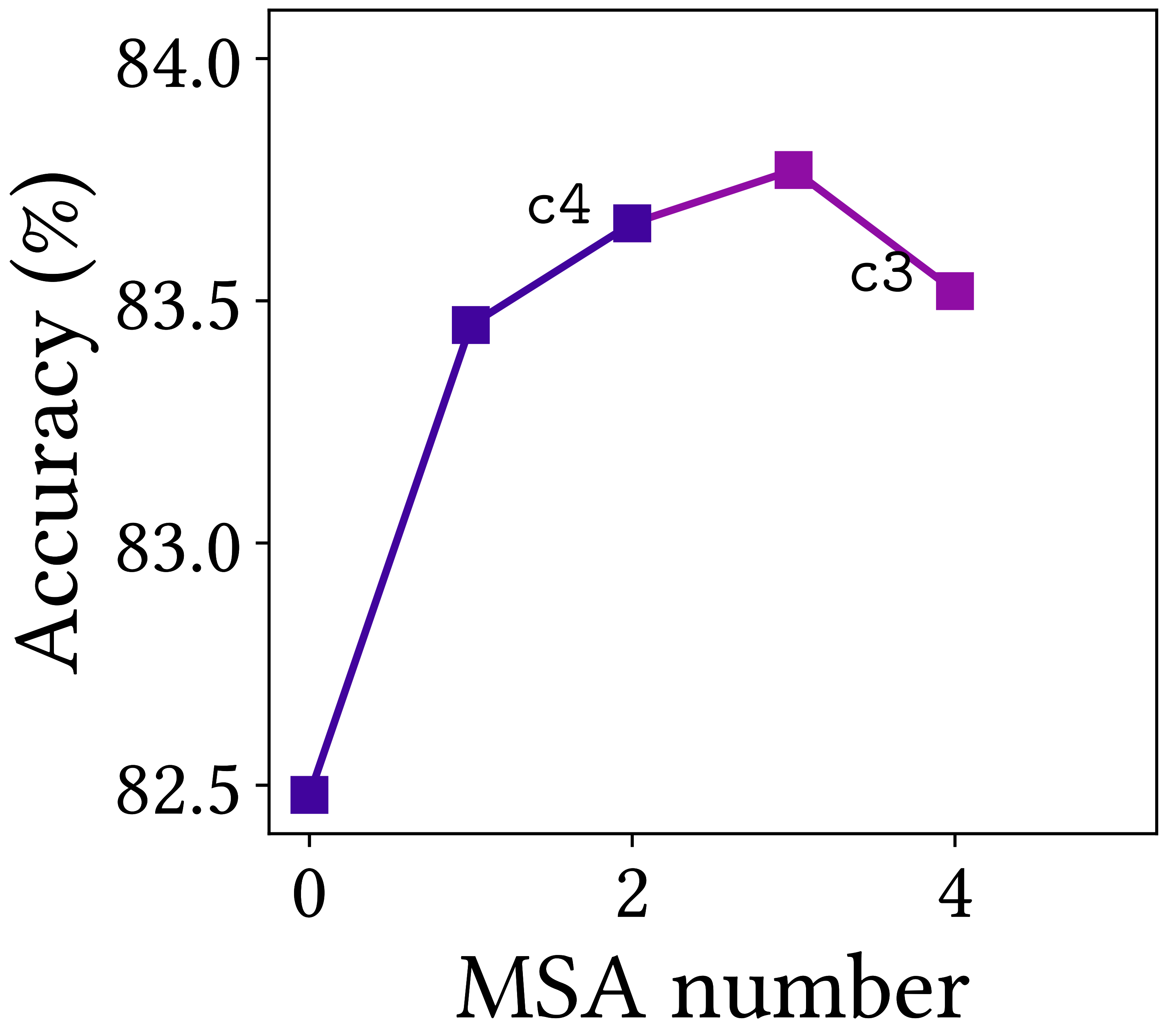
Output


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

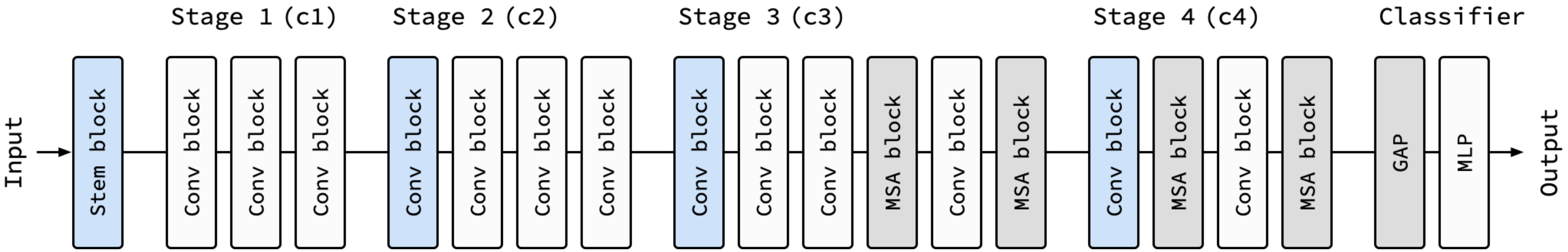

Stage 1 (c1)
Stage 2 (c2)
Stage 3 (c3)
Stage 4 (c4)
Classifier
Input
Stem block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
Conv block
MSA block
Conv block
MSA block
GAP
MLP
Output

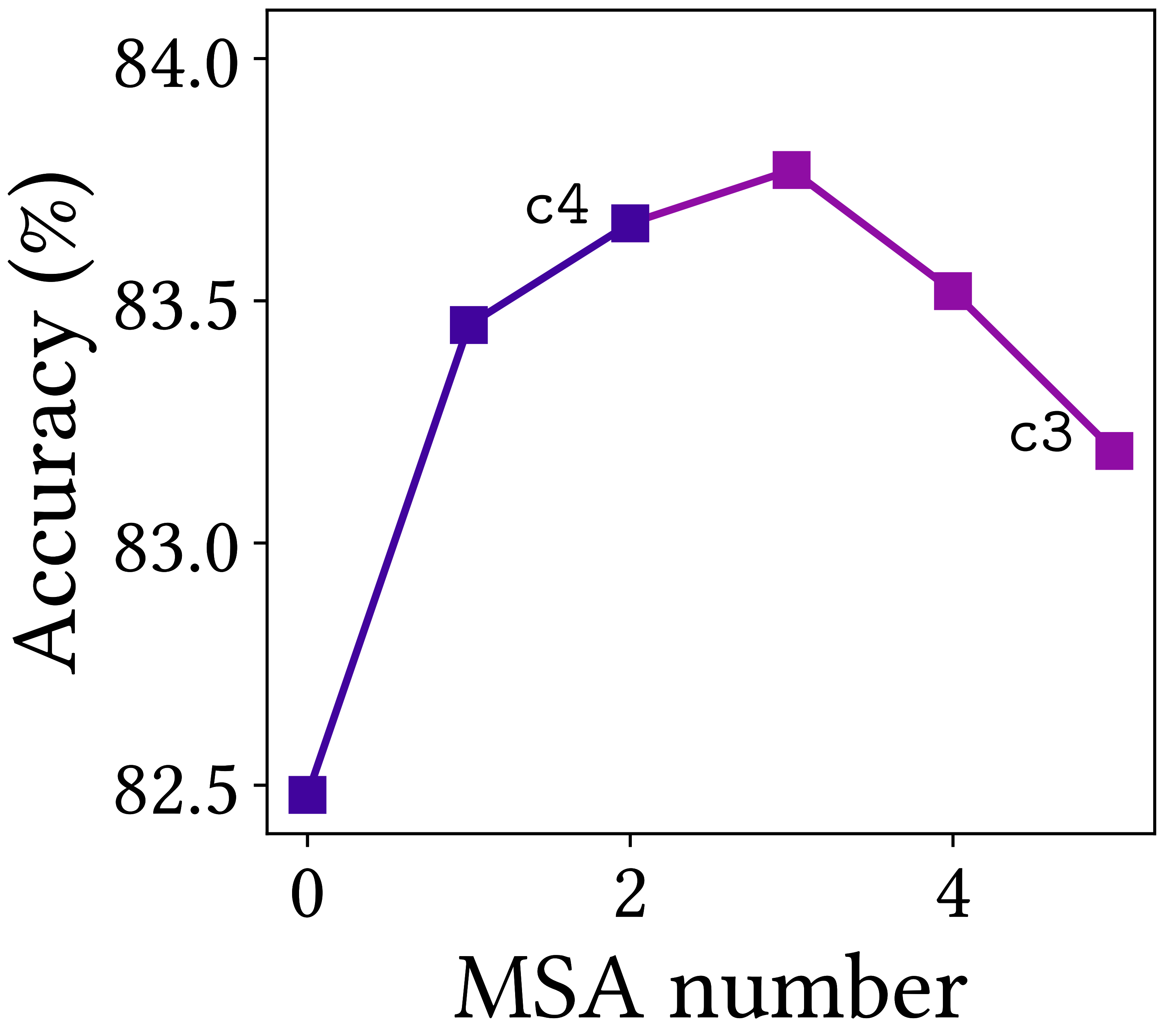

Accuracy (%)
84.0
83.5
83.0
82.5
c3
c4
0
2
4
MSA number


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
MSA block
Conv block
MSA block
Conv block
MSA block
GAP
MLP
Output

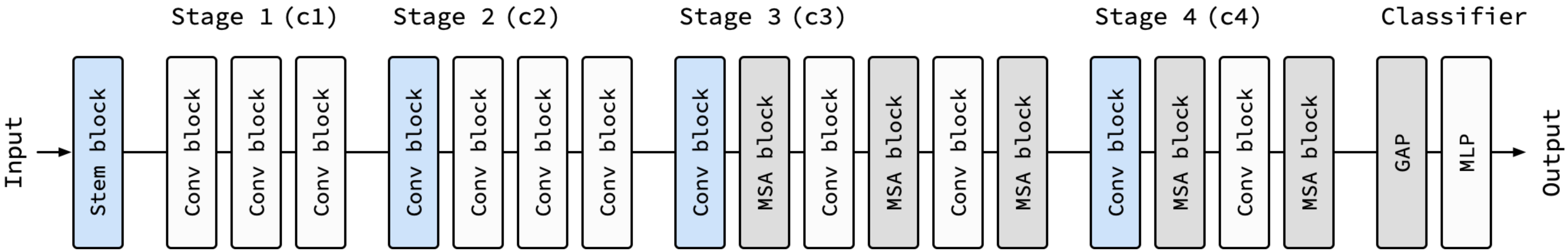


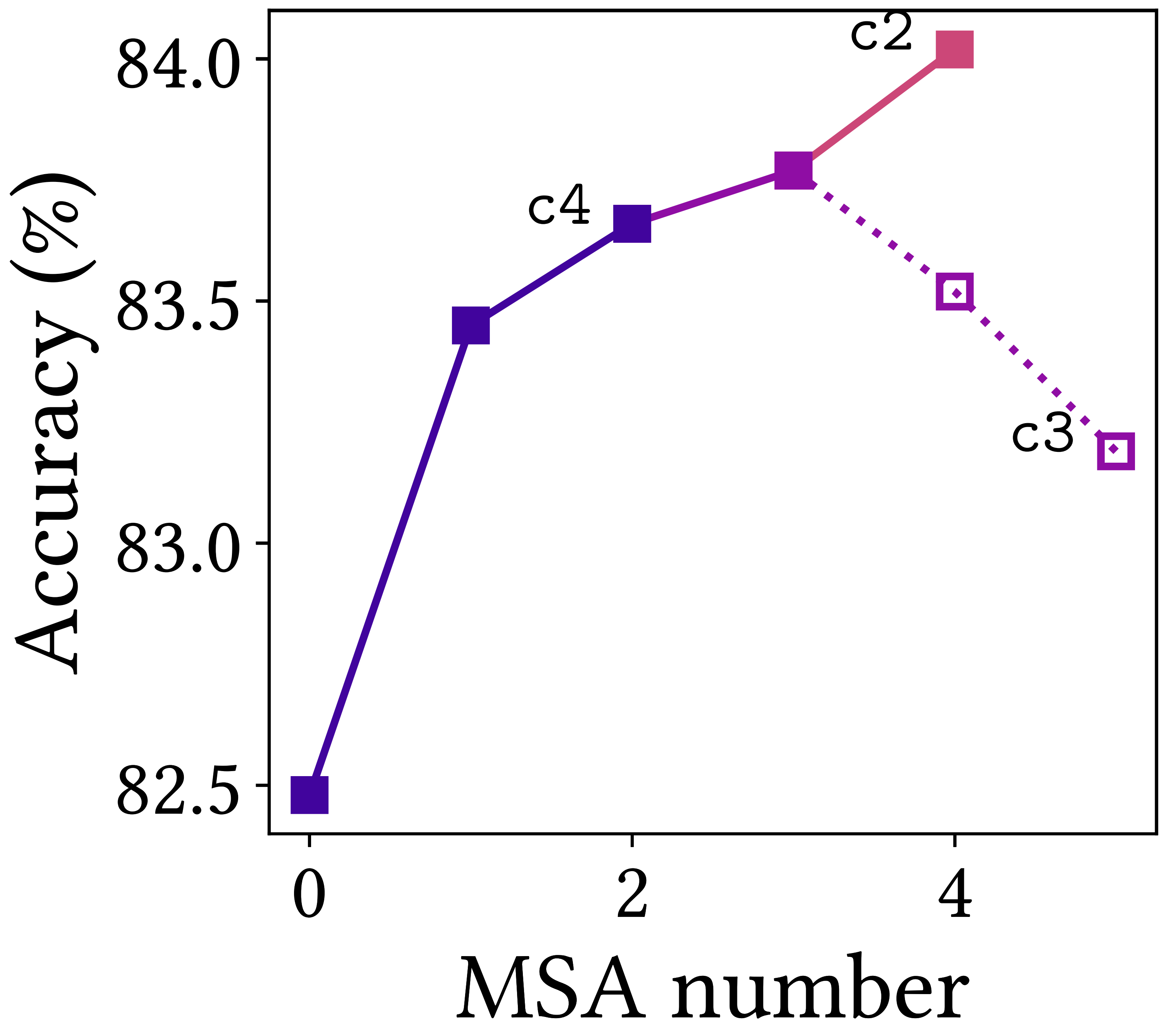
As expected,
MSAs in **c3** harm accuracy.



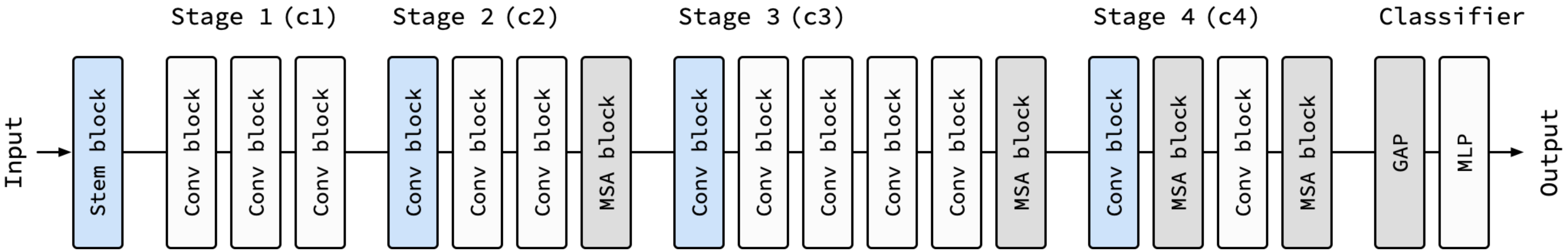


As expected,
MSAs in c3 harm accuracy.





Surprisingly,
a MSA in c2 improves accuracy!


Input

Stem block

Stage 1 (c1)

Conv block
Conv block
Conv block

Stage 2 (c2)

Conv block
Conv block
Conv block
MSA block

Stage 3 (c3)

Conv block
Conv block
Conv block
Conv block
Conv block
MSA block

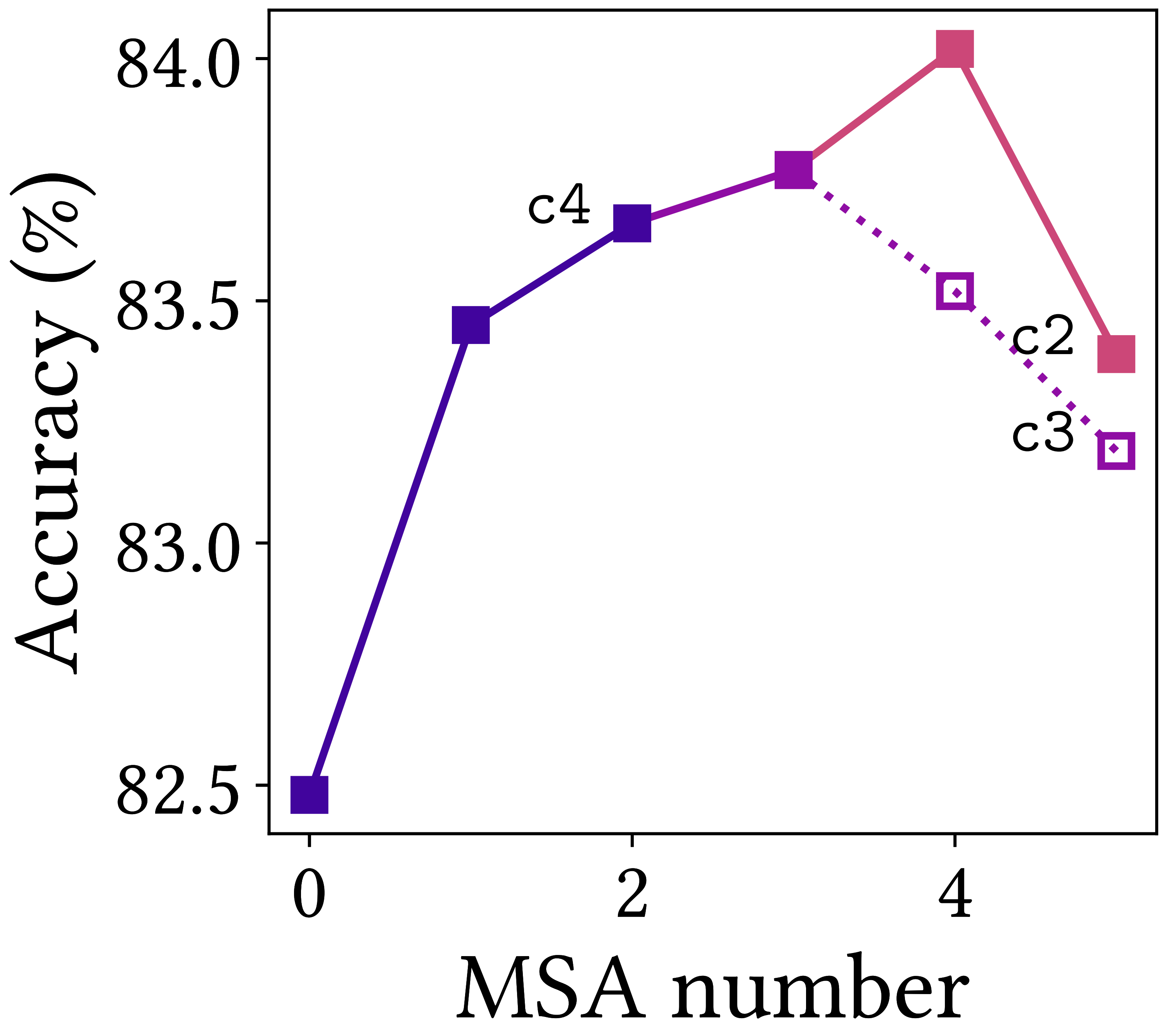
Stage 4 (c4)

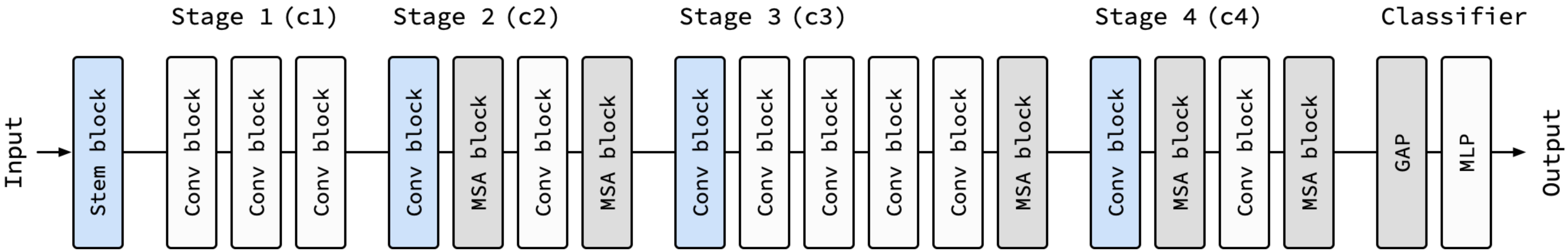
Conv block
MSA block
Conv block
MSA block

Classifier

GAP
MLP

Output





Input

Stem block

Stage 1 (c1)

Conv block

Conv block

Conv block

Stage 2 (c2)

Conv block

MSA block

Conv block

MSA block

Stage 3 (c3)

Conv block

Conv block

Conv block

Conv block

Conv block

MSA block

Stage 4 (c4)

Conv block

MSA block

Conv block

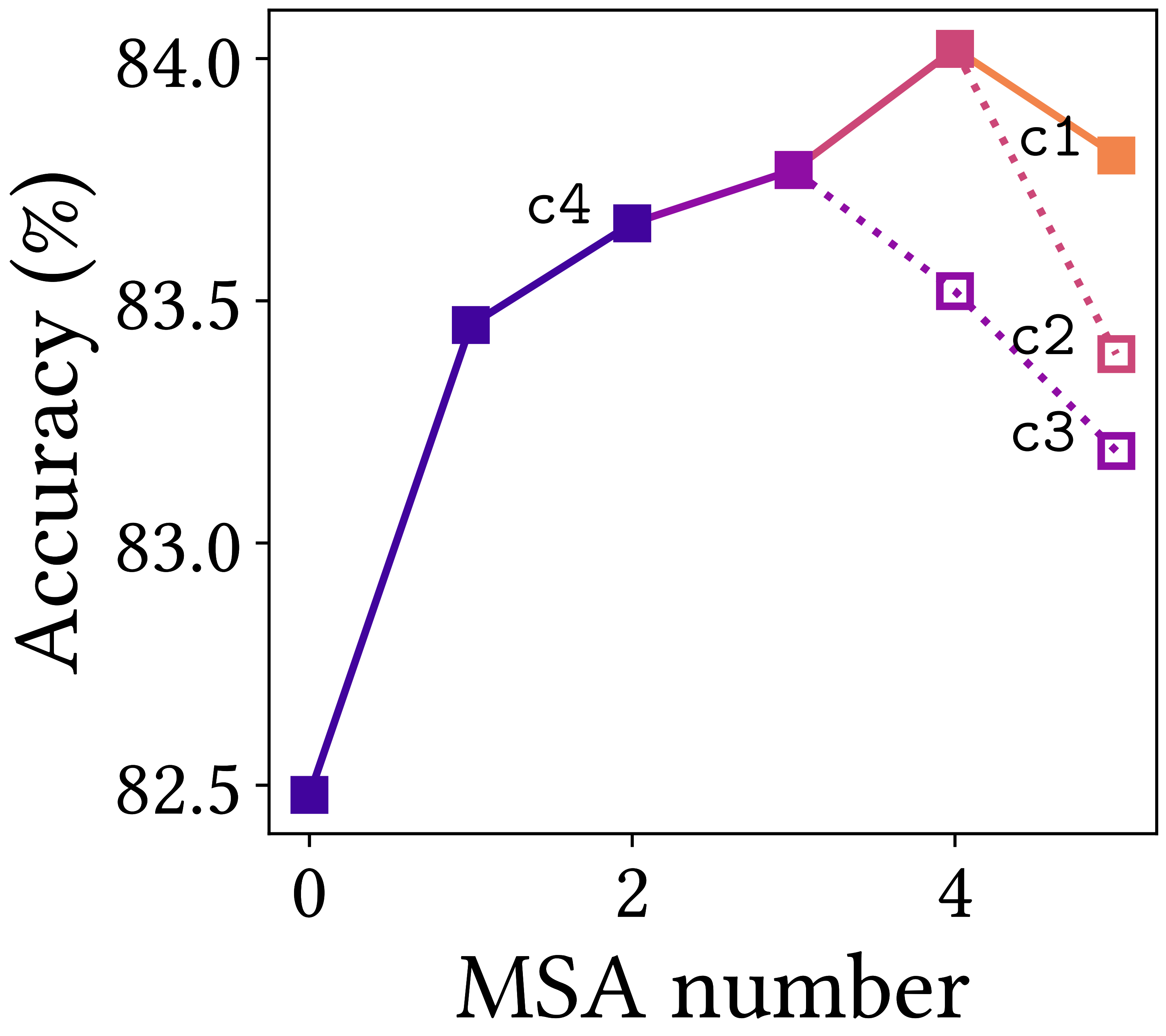
MSA block

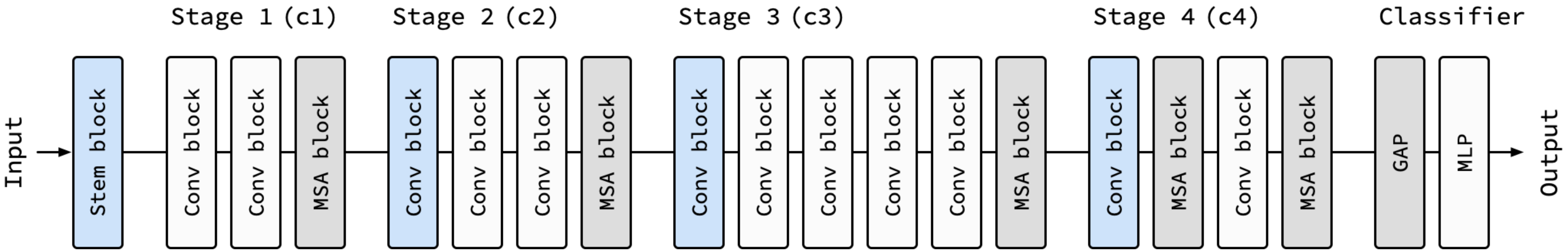
Classifier

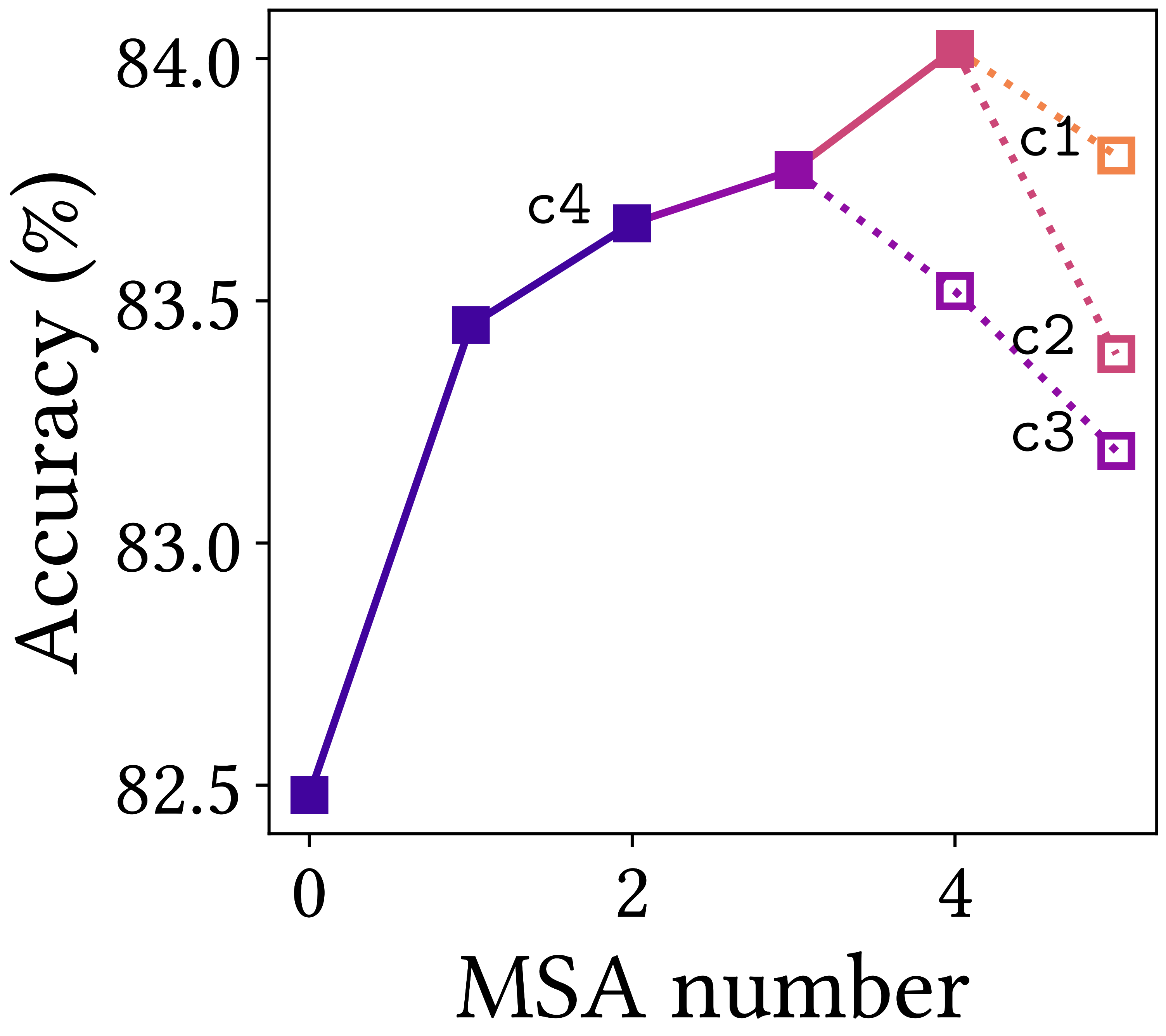
GAP

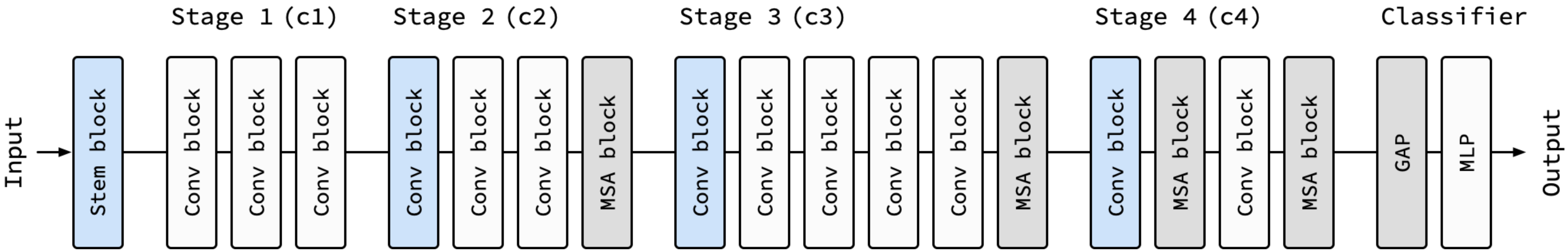
MLP

Output


Accuracy (%)
84.0
83.5
83.0
82.5
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




Input

Stem block

Stage 1 (c1)

Conv block

Conv block

Conv block

Stage 2 (c2)

Conv block

Conv block

Conv block

MSA block

Stage 3 (c3)

Conv block

Conv block

Conv block

Conv block

Conv block

MSA block

Stage 4 (c4)

Conv block

MSA block

Conv block

MSA block

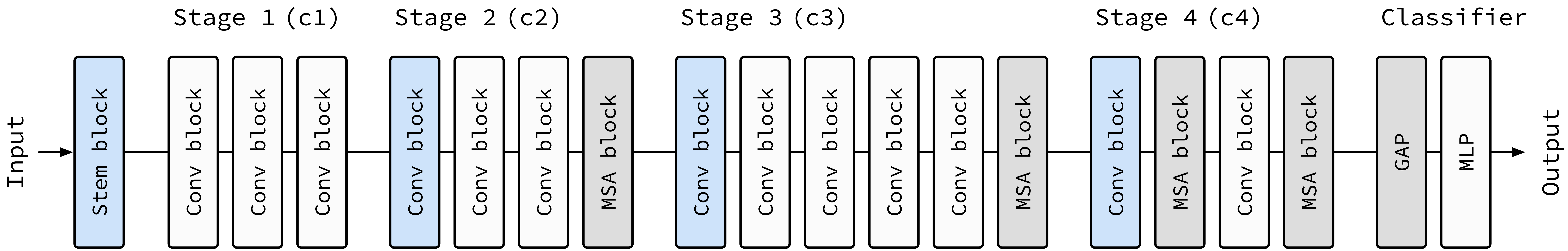
Classifier

GAP

MLP

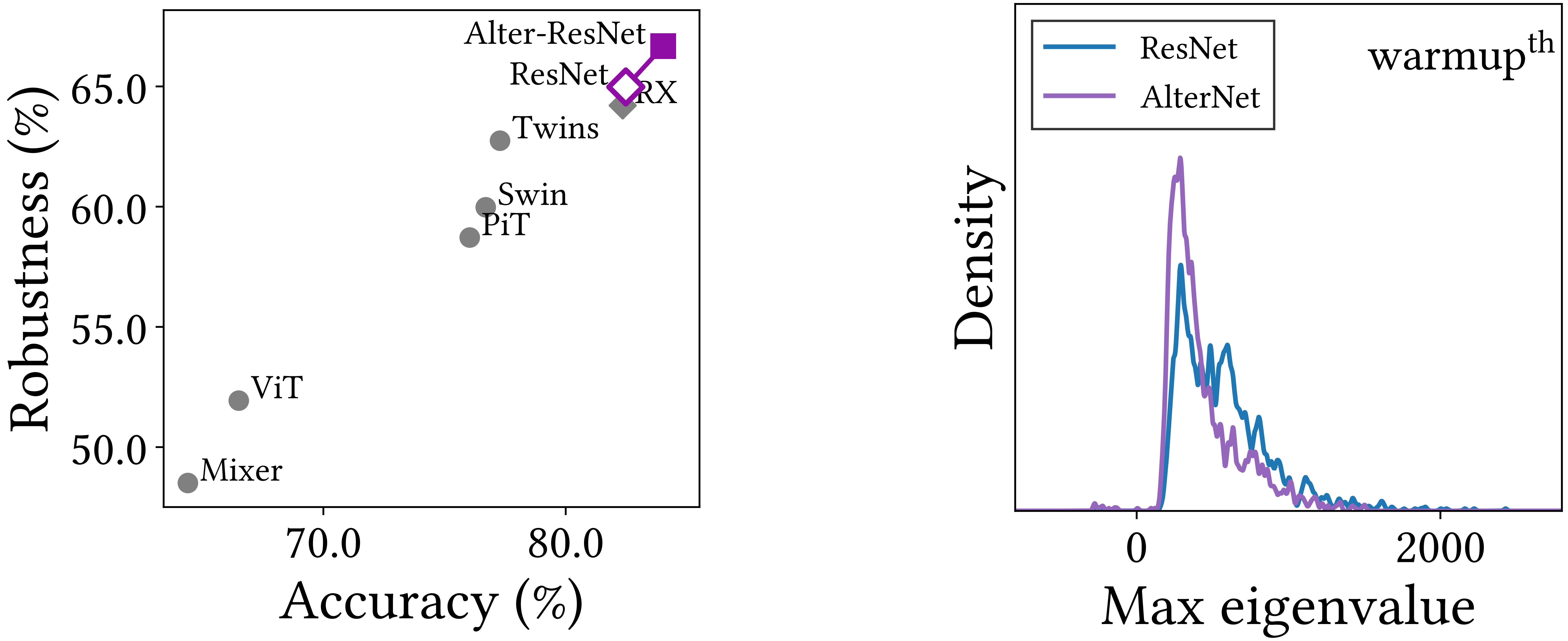
Output

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.

How Do Vision Transformers Work?

Q1. What properties of MSAs do we need to improve optimization?

‣ MSAs flatten loss landscapes. Long-range dependency disrupts NN optimization.

Q2. Do MSAs act like Convs?

‣ MSAs and Convs exhibit opposite behaviors; e.g., MSAs are low-pass filters.

Q3. How can we harmonize MSAs with Convs?

‣ MSAs *at the end of a stage* play a key role in prediction.

Finale

So, Why Do Vision Transformers Work That Way?

MSAs Are Trainable Spatial Smoothings

$$p(\boldsymbol{z}_j|\boldsymbol{x}_j, \mathcal{D}) \simeq \sum_i \overbrace{\pi(\boldsymbol{x}_i|\boldsymbol{x}_j)}^{\text{Importance}} \, \overbrace{p(\boldsymbol{z}_j|\boldsymbol{x}_i, \boldsymbol{w}_i)}^{\text{Prediction}}$$

(box blur)

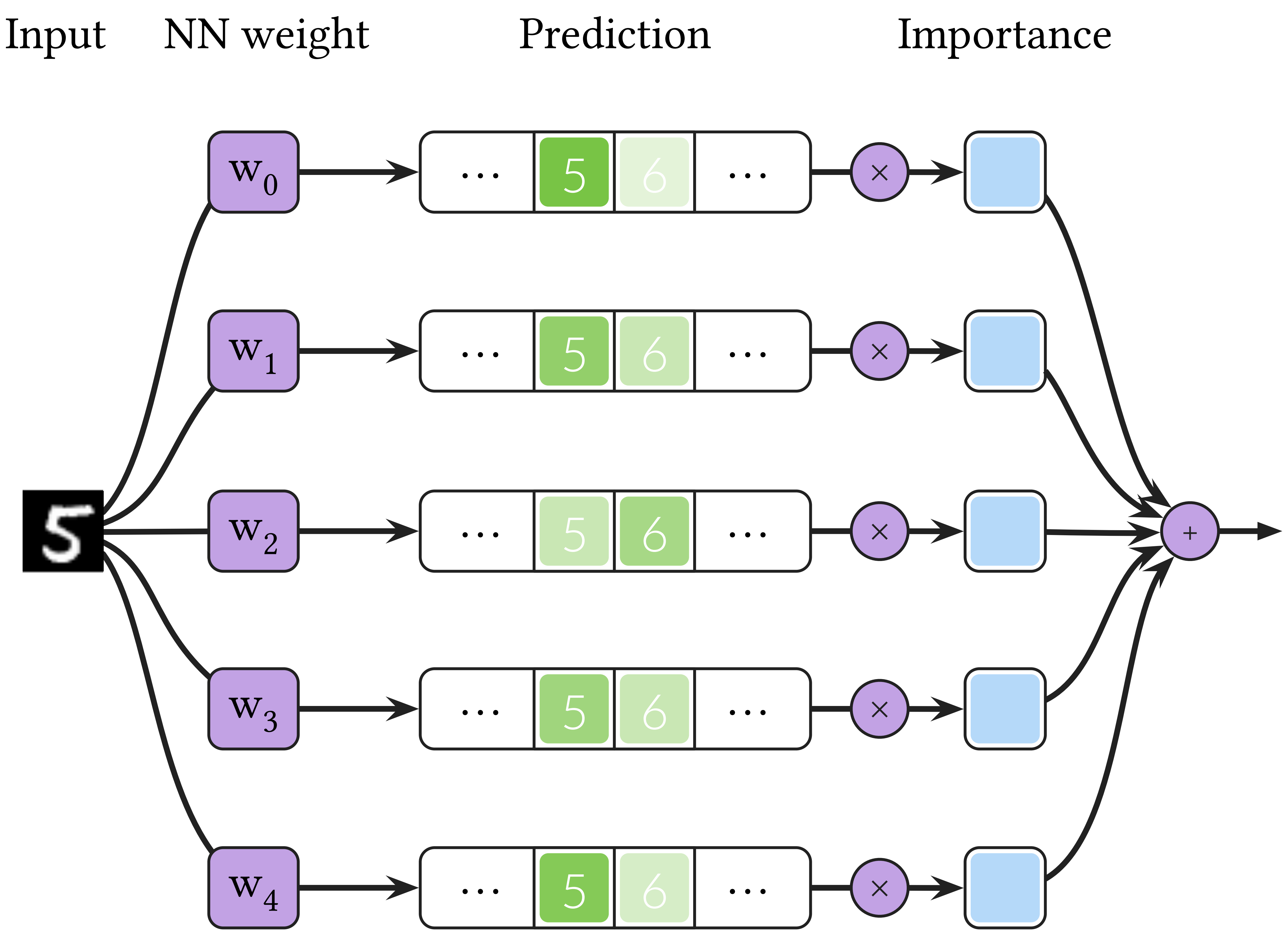
$$\text{where} \quad \pi(\boldsymbol{x}_i|\boldsymbol{x}_j) = \begin{cases} 1/4, & \text{if } \boldsymbol{x}_i \text{ is a neighbour of } \boldsymbol{x}_j \\ 0, & \text{otherwise} \end{cases}$$

$$\boldsymbol{z}_j = \sum_i \underbrace{\texttt{Softmax}\left(\frac{\boldsymbol{QK}}{\sqrt{d}}\right)_i}_{\text{Importance}} \underbrace{\boldsymbol{V}_{i,j}}_{\text{Prediction}}$$

(self-attention)

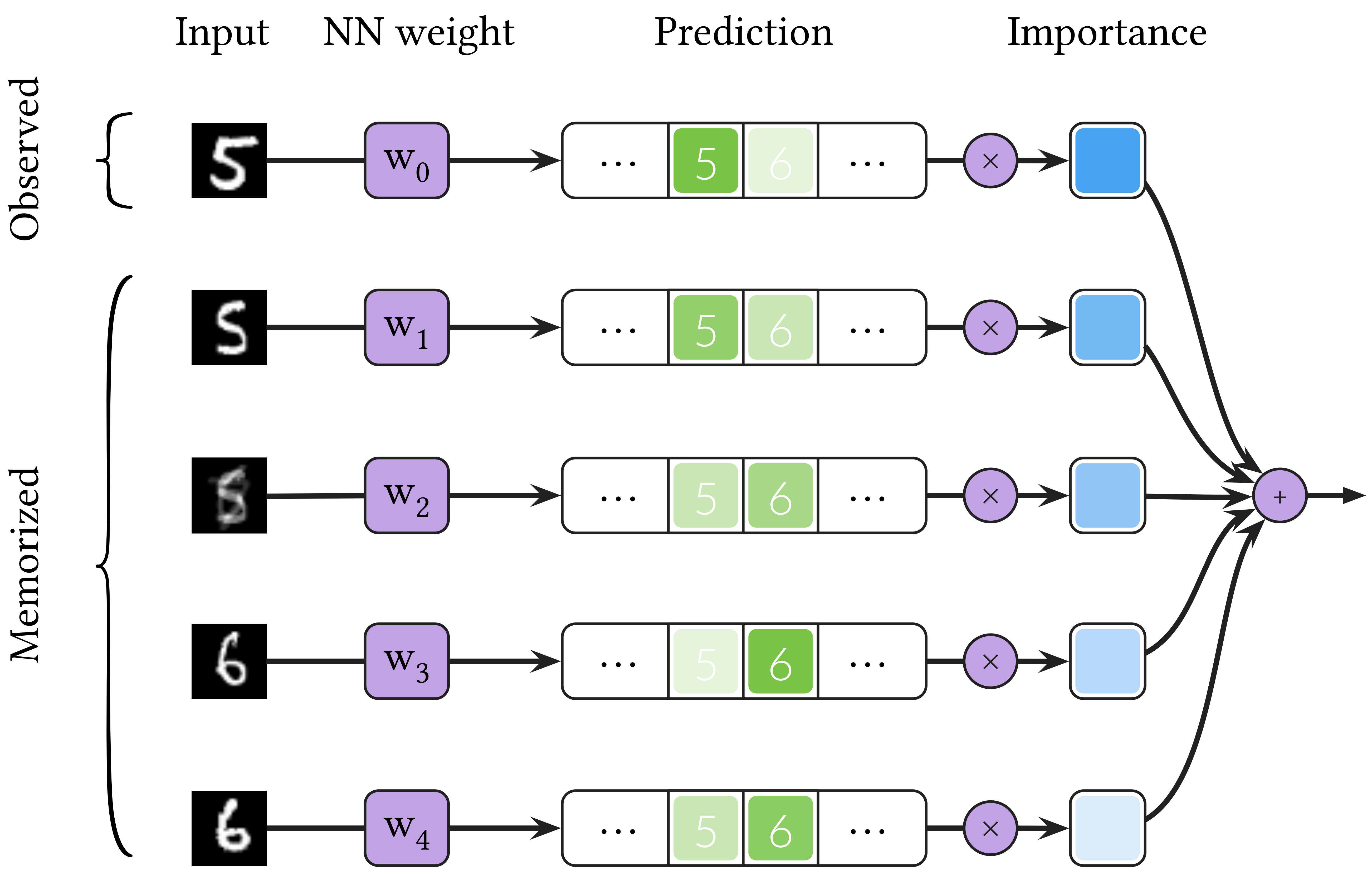
Blur is a spatial ensemble

Ensemble Average Require Multiple NN Executions



BNN inference is ensemble average of NN predictions for one observed data point. Using *N* neural networks in the ensemble would requires *N times more* computational complexity than one NN execution. The NN weights are sampled from ***weight distribution***.

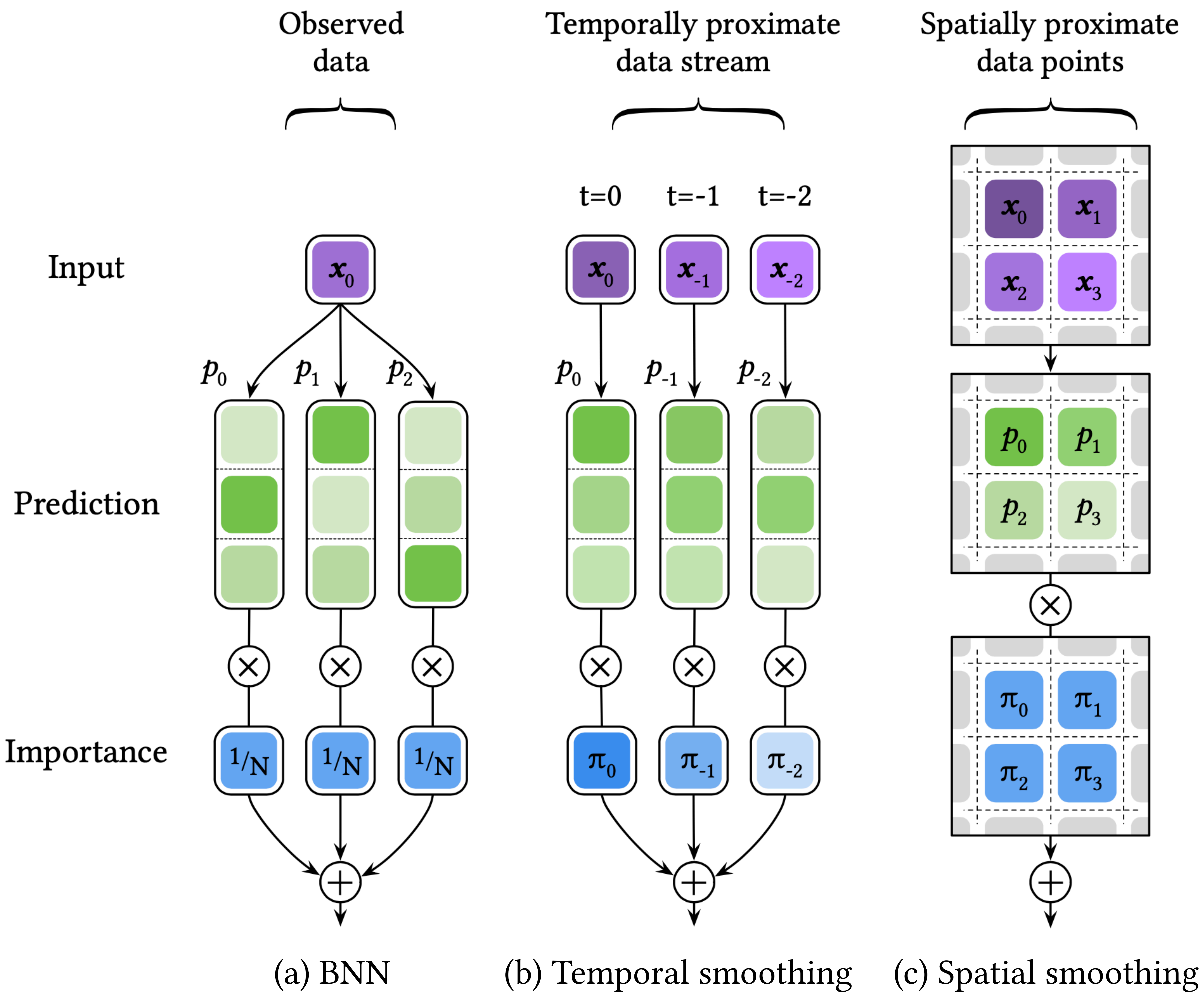
Memorized Results Can Speed up the Ensemble



We ensemble average the prediction for the observed input and memorized predictions with importance-weights. This method executes NN for an observed data only once, and complements the result with previously calculated predictions for other data.

Importance is similarity between the observed data and the memorized data. In other words, memorized data is sampled from ***data distribution***, and the probability is the importance.

Spatial Smoothing: Ensemble of Feature Maps



Comparison of three different Bayesian neural network inferences: canonical BNN inference, temporal smoothing, and spatial smoothing. Spatial smoothing aggregates neighboring feature map points.

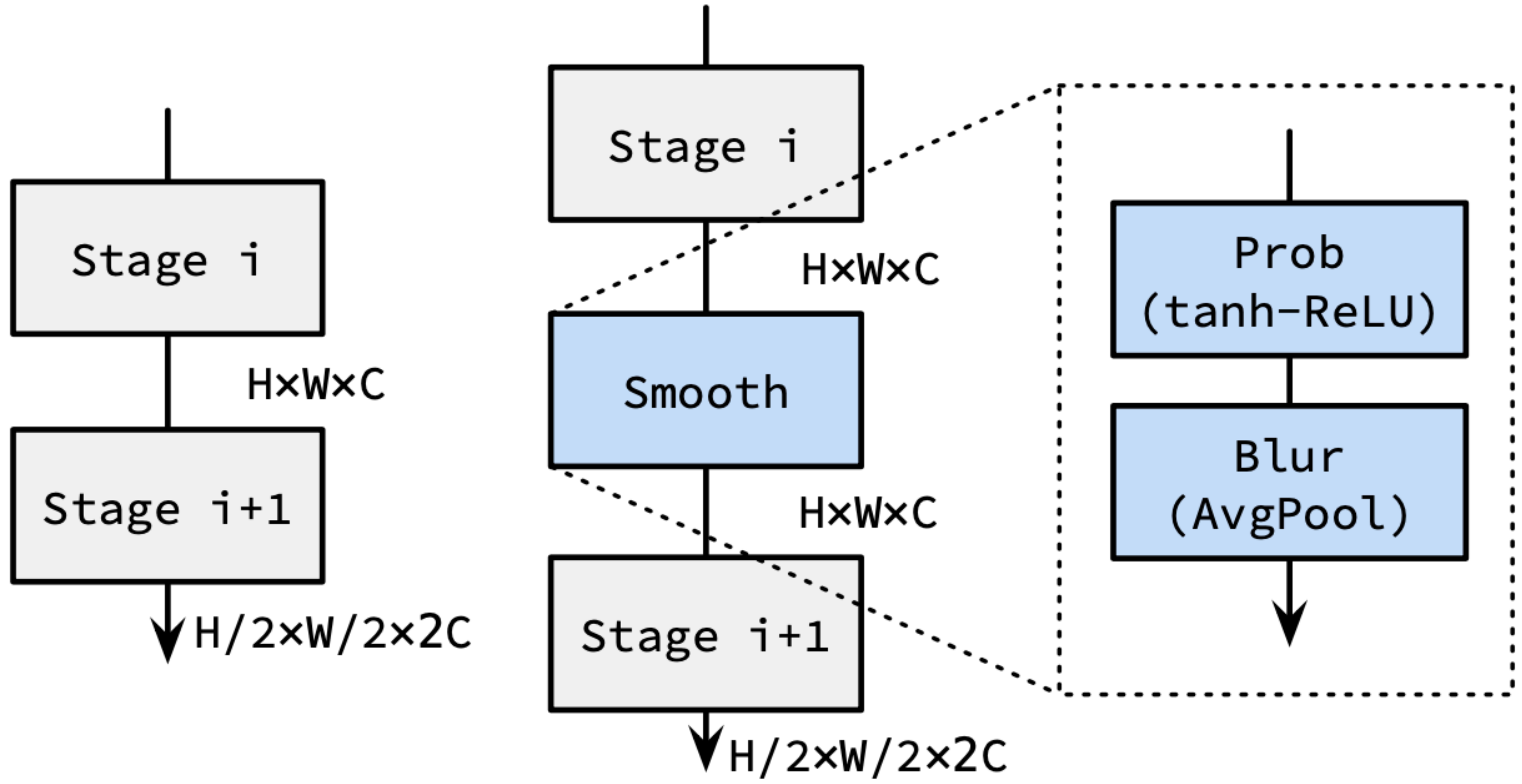
Ensemble Averaging for Proximate Data Points

proximate data points

= neighboring feature maps

$$p(\boldsymbol{z}_j|\boldsymbol{x}_j,\mathcal{D}) \simeq \sum_i \underbrace{\pi(\boldsymbol{x}_i|\boldsymbol{x}_j)}_{\text{Importance}}\, \underbrace{p(\boldsymbol{z}_j|\boldsymbol{x}_i,\boldsymbol{w}_i)}_{\text{Prediction}}$$

Module Architecture of Spatial Smoothing

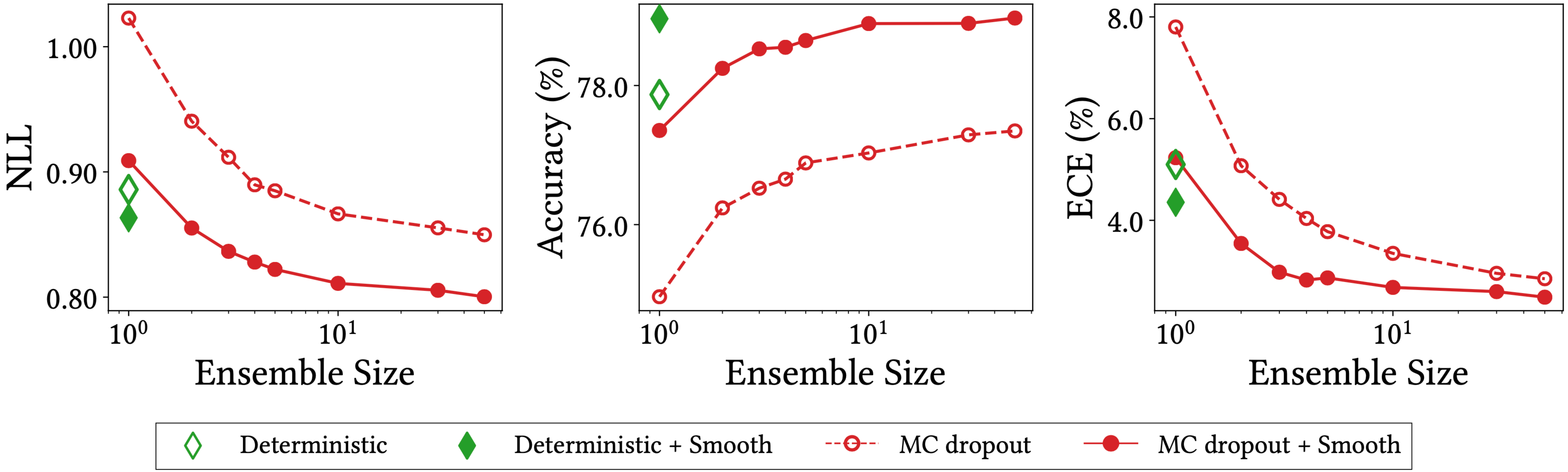


(a) ResNet

(b) ResNet + Smooth

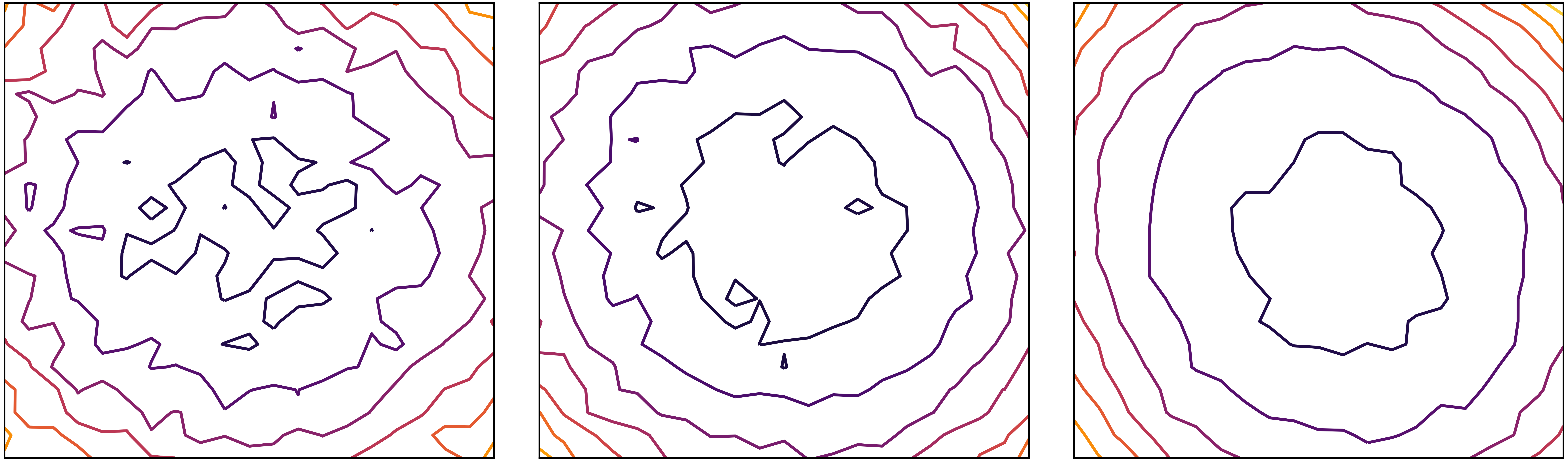
Stages of CNNs such as ResNet (left) and the stages incorporating spatial smoothing layer (right). "`Prob`" in spatial smoothing transforms a real-valued feature map into probability. "`Blur`" aggregates the probabilities from feature maps.

Spatial Smoothing Improves the Predictive Performances



Spatial smoothing improves accuracy and uncertainty estimation across a whole range of ensemble sizes. In particular, spatial smoothing achieve high predictive performance merely with a handful of ensembles. We report the predictive performance of ResNet-18 on CIFAR-100.

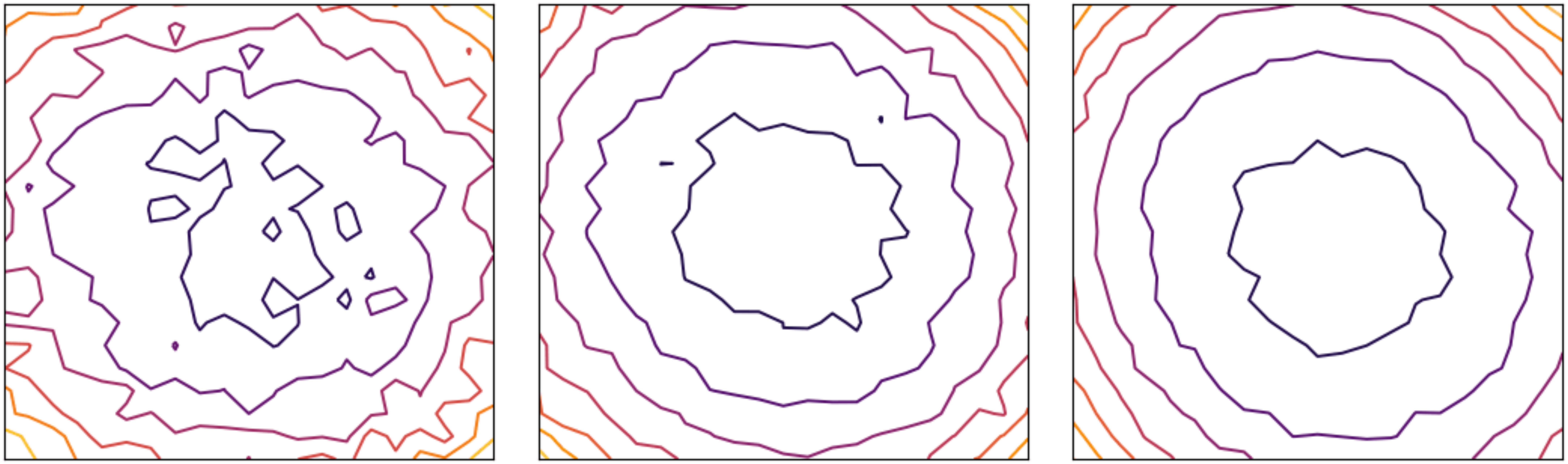
Spatial Smoothings Smoothen the Loss Landscapes



(a) MLP classifier (b) GAP classifier (*vanilla*) (c) GAP classifier + Smooth

The loss landscape of the MLP classifier is the most irregular. Both GAP and spatial smoothing flatten the loss landscapes. We present the loss landscape visualizations of ResNet-18 models with MC dropout on CIFAR-100.

Spatial Smoothings Smoothen the Loss Landscapes



(a) MLP classifier (b) GAP classifier (*vanilla*) (c) GAP classifier + Smooth

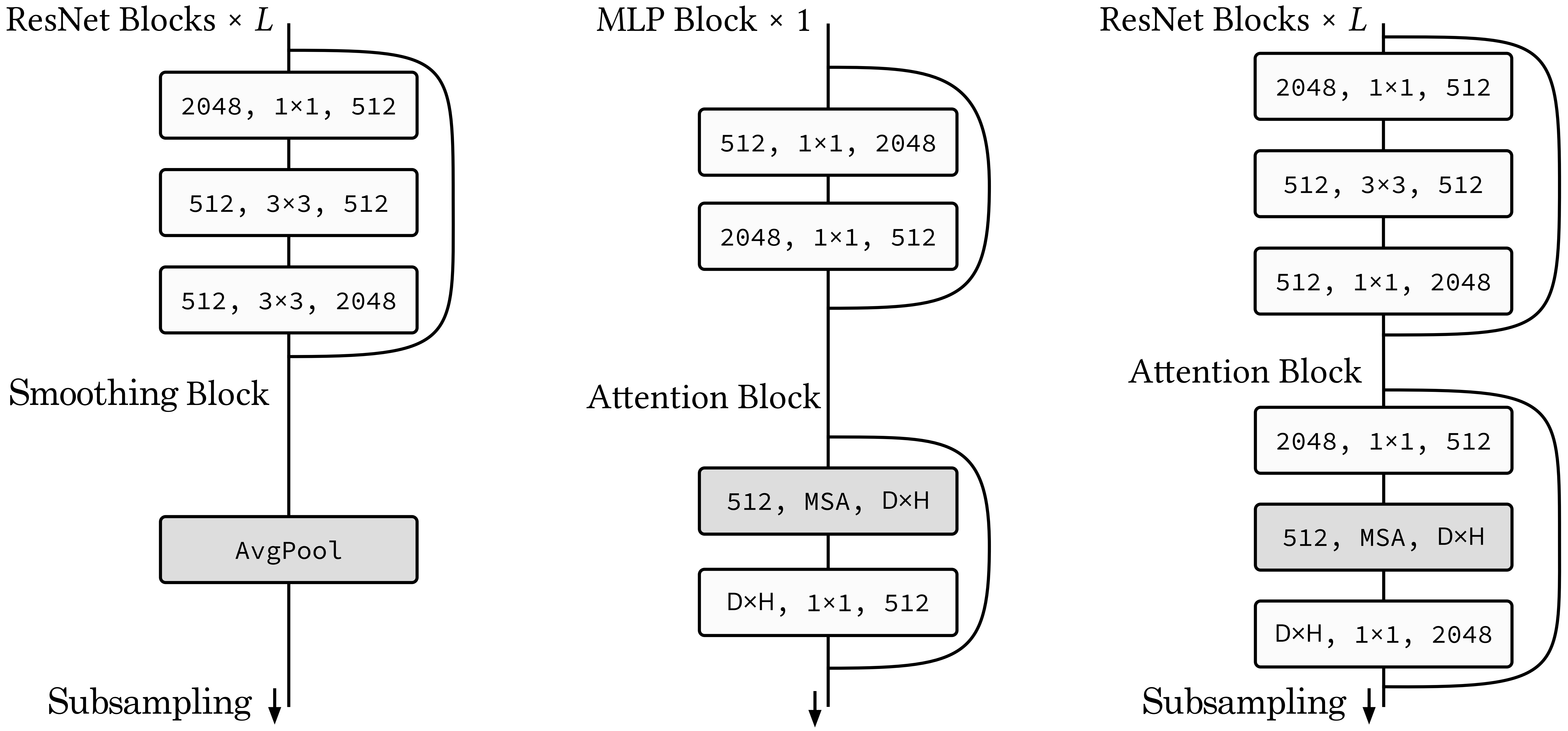
The loss landscape of the MLP classifier is the most irregular. Both GAP and spatial smoothing flatten the loss landscapes. We present the loss landscape visualizations of ResNet-18 models with MC dropout on CIFAR-100.

This is an animation! Please visit: https://github.com/xxxnell/spatial-smoothing/

MSAs Are a Trainable Spatial Smoothing

- Spatial smoothing helps in NN optimization by flattening the loss landscapes.
- Spatial smoothing is a low-pass filter. Spatial smoothing also improves the robustness against high-frequency noise significantly.
- Spatial smoothing is effective when applied at the end of a stage. This is because it aggregates the transformed feature map predictions.

Comparison of Three Different Repeating Patterns



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

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.

Conclusion

MSAs are not merely generalized Convs, but rather **generalized spatial smoothings** that complement Convs. They inherit the properties of spatial smoothings:

- **Optimization:** MSAs flatten loss landscapes. It improves not only accuracy but also generalization. Such improvement is primarily attributable to their formulation including data specificity, ***NOT*** long-range dependency.

- **Behavior:** MSAs and Convs exhibit opposite behaviors. For example, MSAs are low-pass filters, but Convs are high-pass filters. MSAs aggregate feature maps, but Convs transform them.

- **Architecture:** Multi-stage neural nets behave like a series connection of small individual models. MSAs at the end of a stage play a key role in prediction.

In short, **self-attention formulation is an appropriate inductive bias** that complements Convs.

Further Information

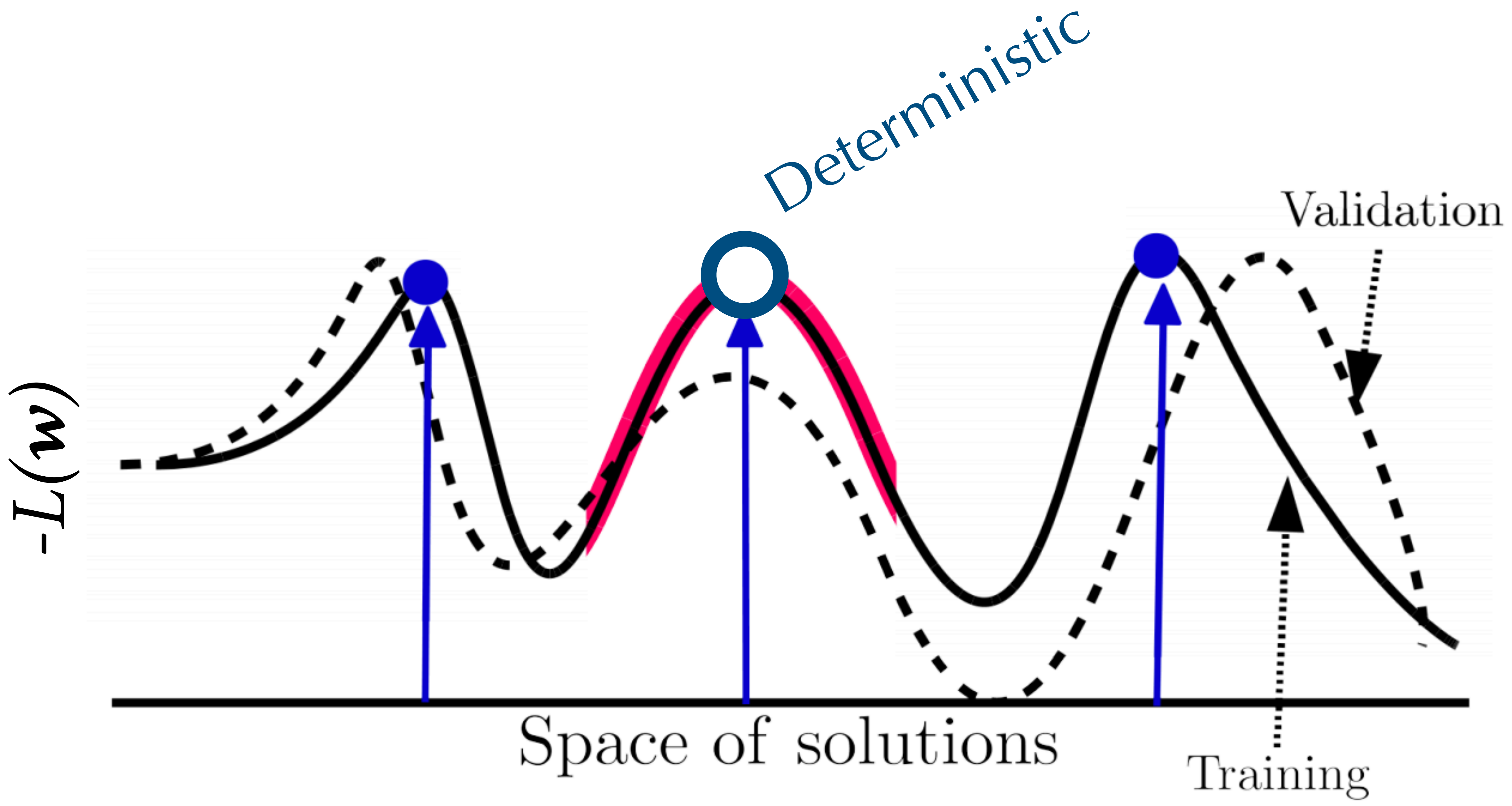
- **Researcher Homepage**: https://www.namukpark.com
- **“How Do Vision Transformers Work?”** (paper): https://openreview.net/forum?id=D78Go4hVcxO
- **“How Do Vision Transformers Work?”** (code and summary): https://github.com/xxxnell/how-do-vits-work
- **“Blurs Behave Like Ensembles: Spatial Smoothings to Improve Accuracy, Uncertainty, and Robustness”** (prior work): https://arxiv.org/abs/2105.12639
- **“Blurs Behave Like Ensembles: Spatial Smoothings to Improve Accuracy, Uncertainty, and Robustness”** (code and summary): https://github.com/xxxnell/spatial-smoothing

Appendix

Summary: Self-Attentions Have Their Pros and Cons

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

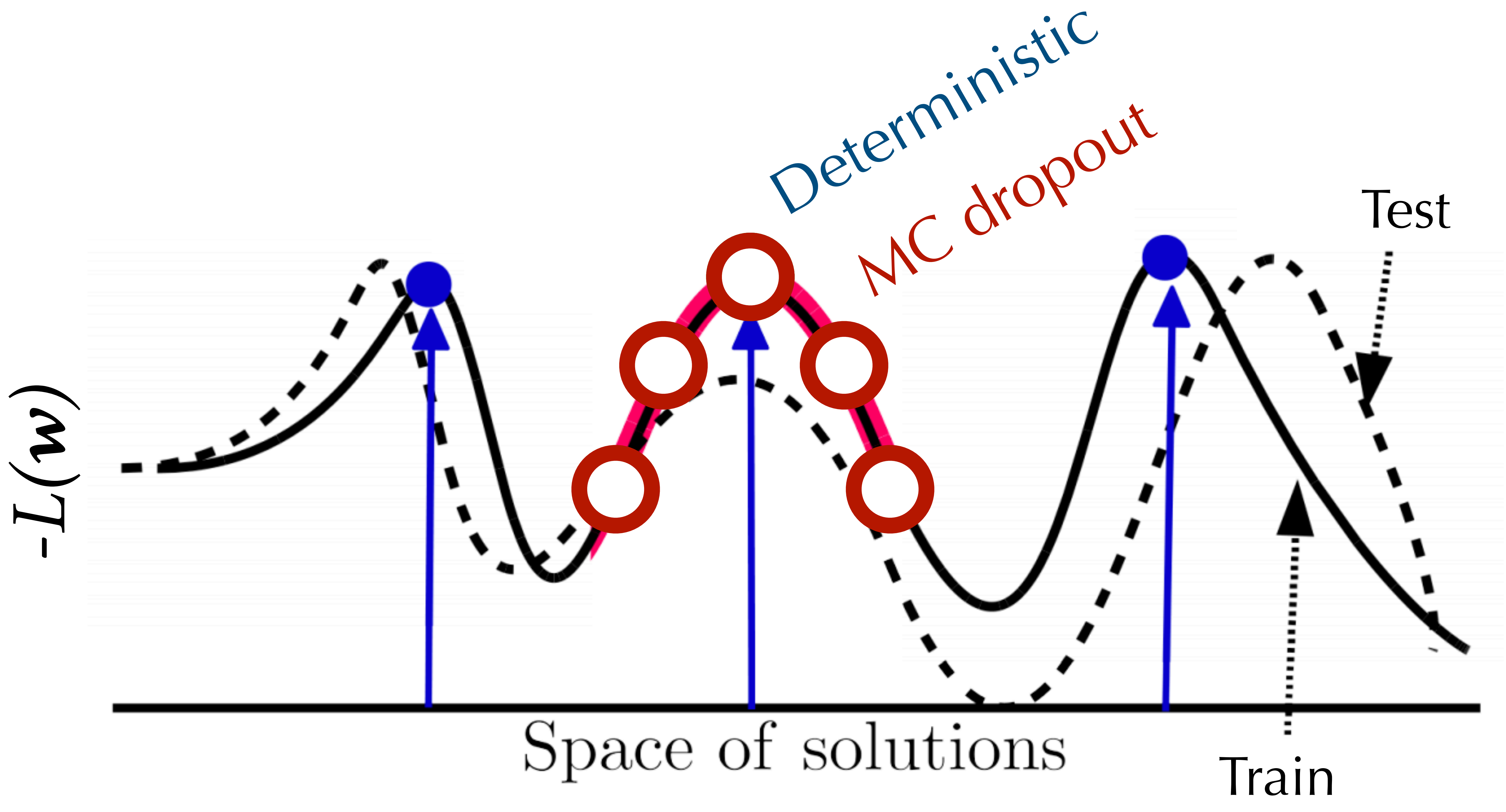
BNNs: A Loss Landscape Perspective



BNNs capture global geometries of the loss landscapes. x-axis indicates parameter values and y-axis plots the *negative* loss on train and validation (test) data.

Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan. "Deep ensembles: A loss landscape perspective." NeurIPS (2019).

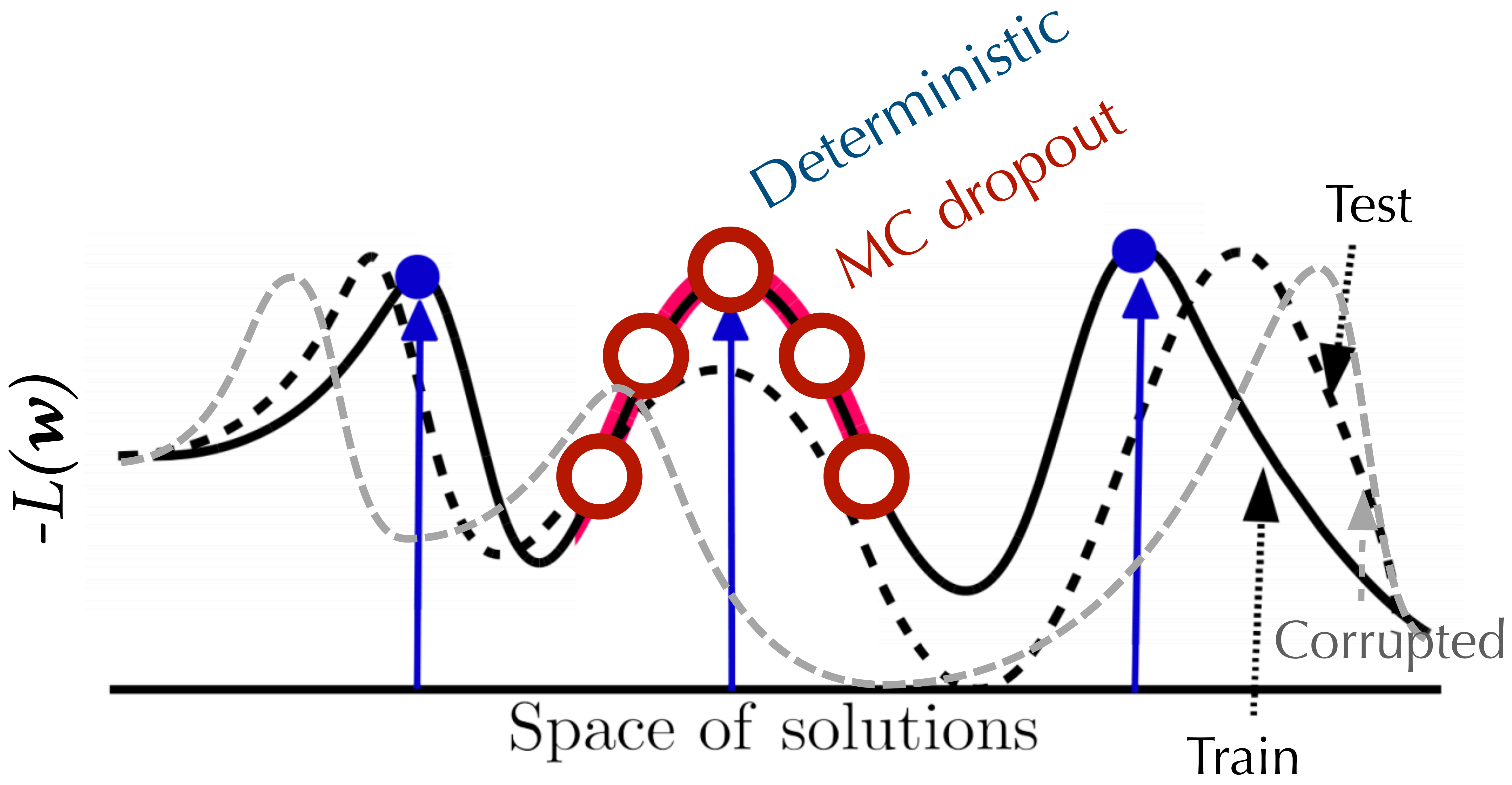
BNNs: A Loss Landscape Perspective



BNNs capture global geometries of the loss landscapes. x-axis indicates parameter values and y-axis plots the *negative* loss on train and test data.

Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan. "Deep ensembles: A loss landscape perspective." NeurIPS (2019).

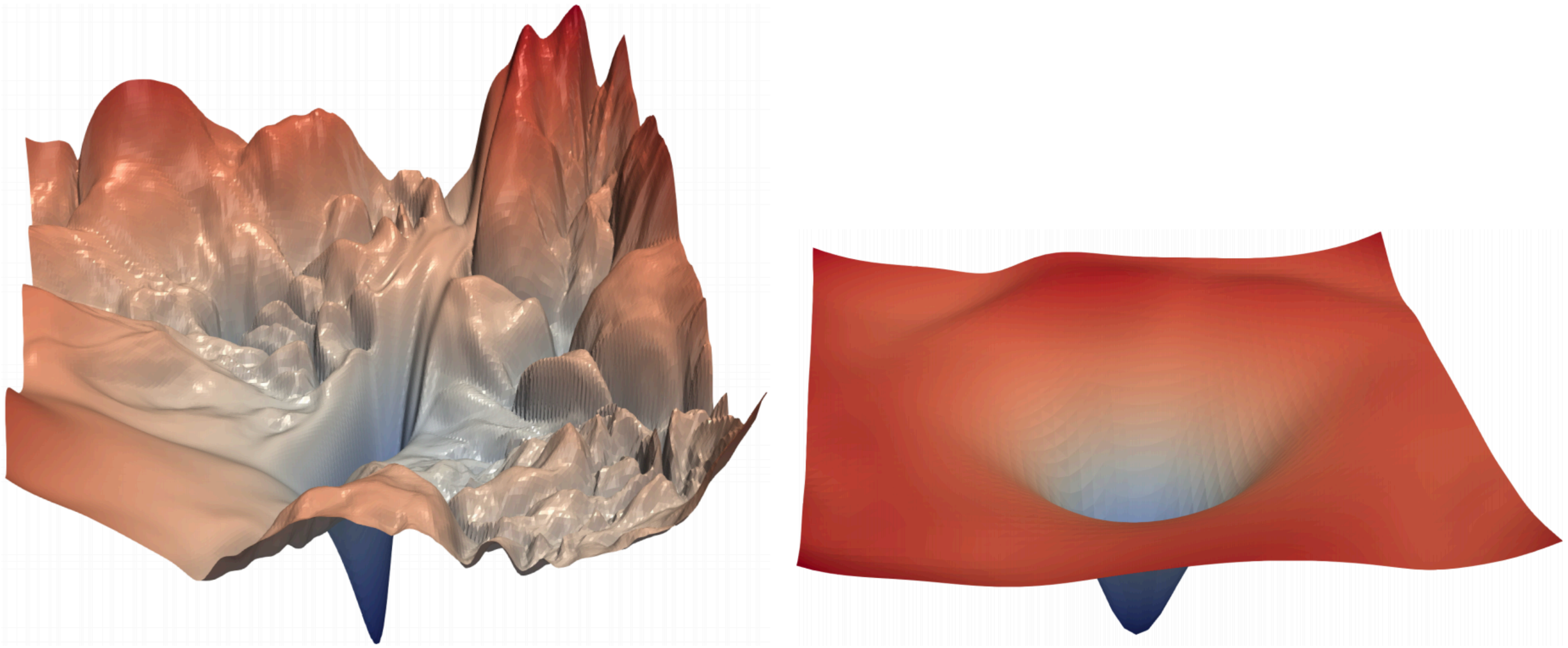
BNNs: A Loss Landscape Perspective



BNNs capture global geometries of the loss landscapes. x-axis indicates parameter values and y-axis plots the *negative* loss on train and test data.

Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan. "Deep ensembles: A loss landscape perspective." NeurIPS (2019).

Flat Loss Landscape Leads To Better Generalization



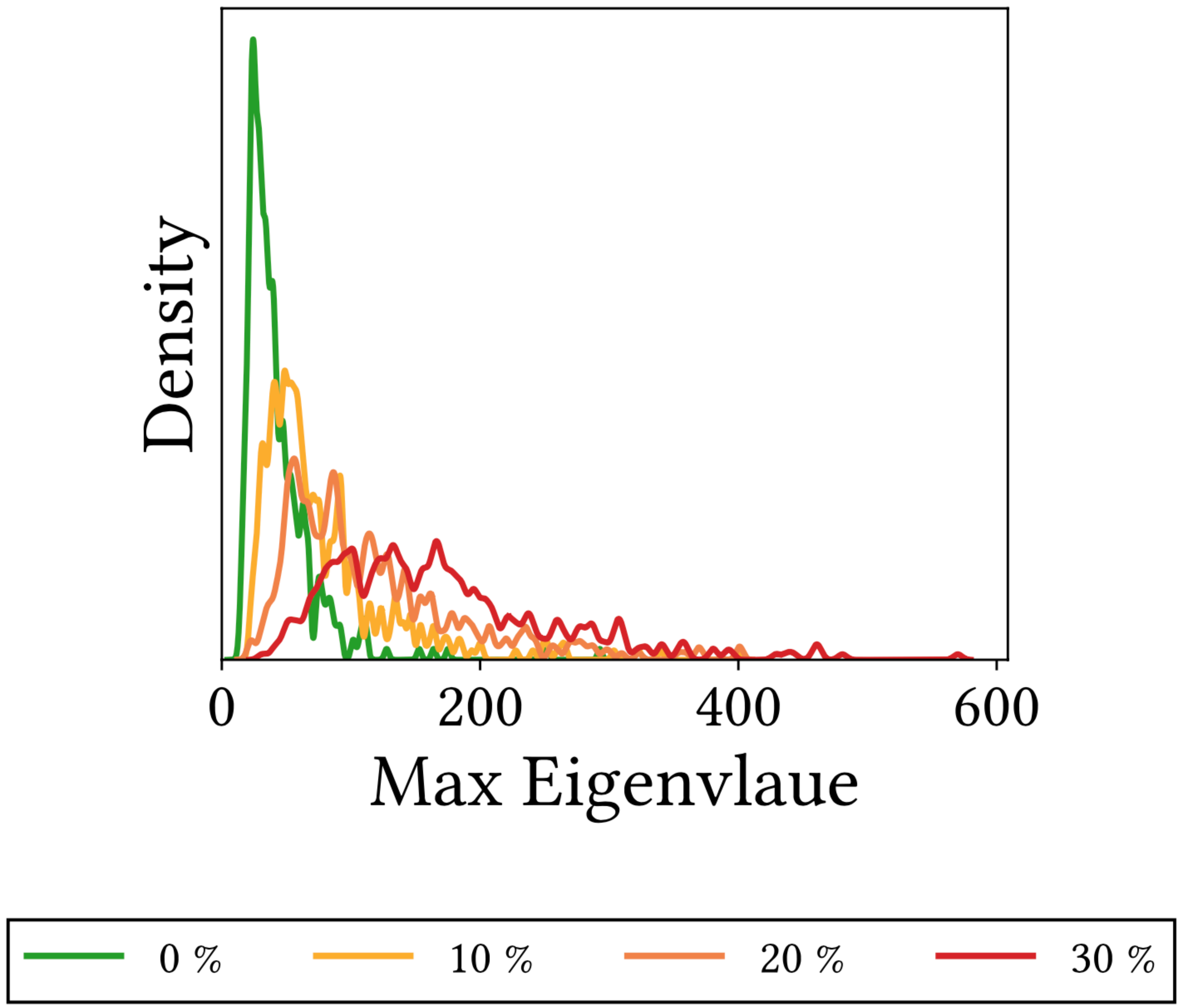
(a) without skip connections

(b) with skip connections

Flat loss landscapes help NNs learn better representations. We provide the loss surfaces of ResNet-56 with/without skip connections.

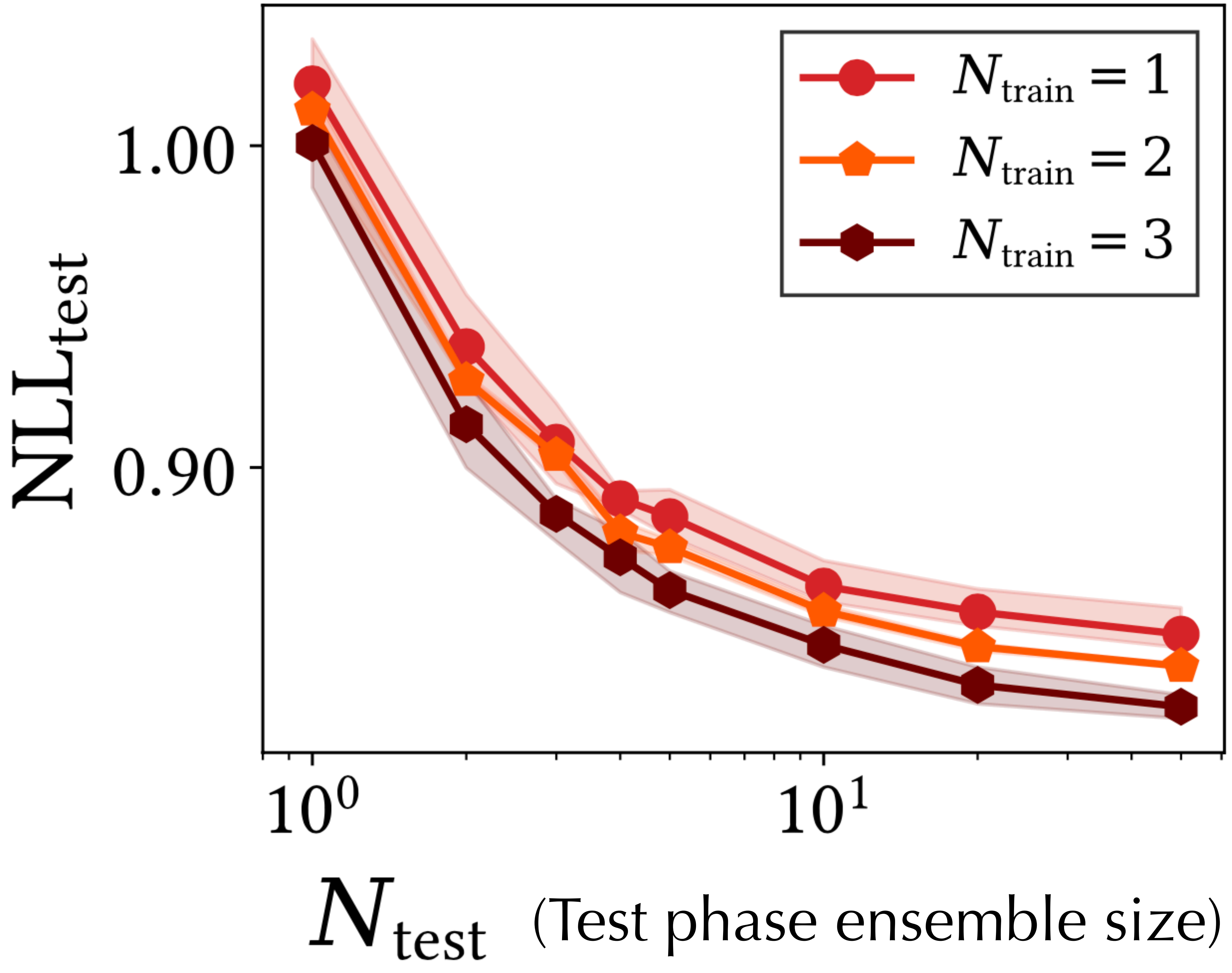
Li, Hao, et al. "Visualizing the Loss Landscape of Neural Nets." NeurIPS (2018).

Randomness of BNNs Disturbs Training



Randomness due to MC dropout sharpens the loss function. We provide Hessian max eigenvalue spectra of ResNet-18 on CIFAR-100.

Training Phase Ensemble Helps NN Optimization



Training phase ensemble helps NNs learn strong representations. Training phase ensemble improves the predictive performance on test dataset.

Advantages of BNNs

BNNs achieve better generalization:

- BNNs achieve high predictive performances.
- BNNs are well-calibrated.
- BNNs are robust against natural corruptions and adversarial attacks.

Disadvantages of BNNs

Bayesian NNs require a large number of predictions to produce reliable results leading to a significant increase in computational cost:

- In ***TEST*** phase
- In ***TRAINING*** phase

Appendix One

Temporal Smoothing

“Moving average is an ensemble!”

Park, Namuk, Taekyu Lee, and Songkuk Kim. "Vector Quantized Bayesian Neural Network Inference for Data Streams." AAAI (2021).

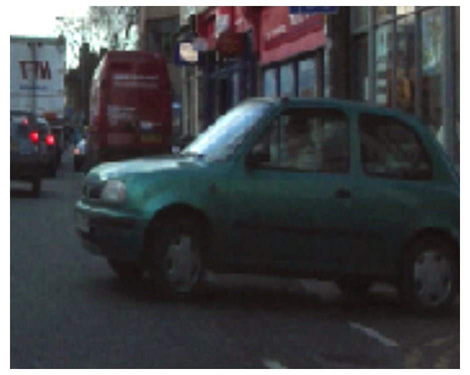
Temporal Smoothing: VQ-BNN for Data-Streams

t=0

t=-1

t=-2

t=-3

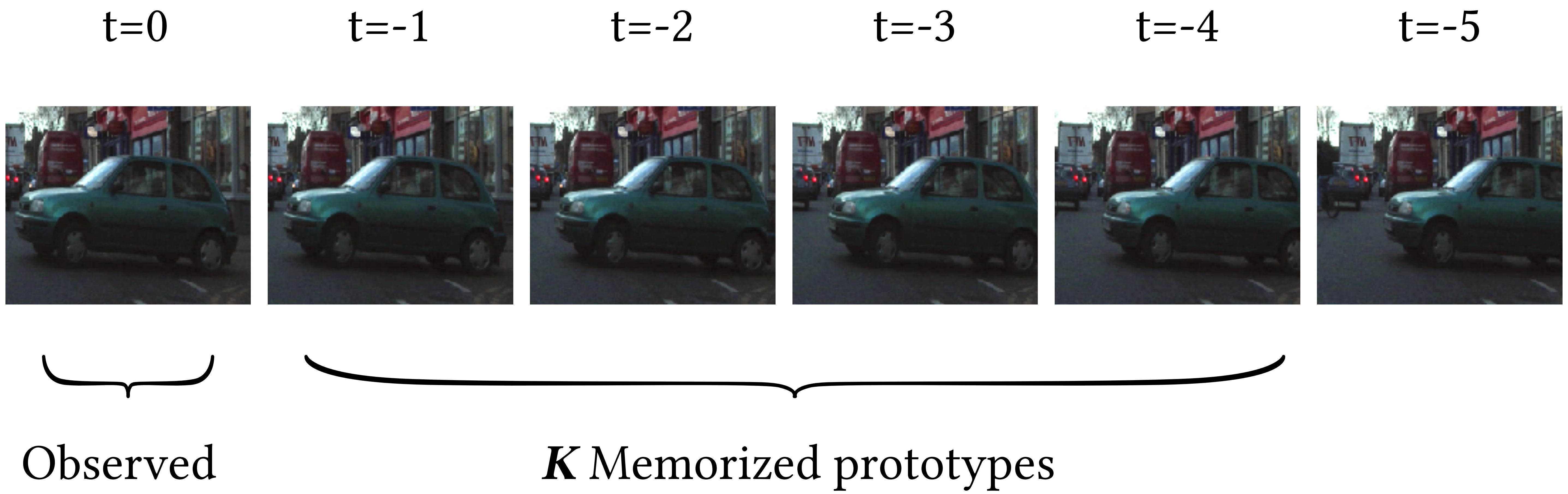


t=-4

t=-5

Most real-world data streams change continuously.

Temporal Smoothing: VQ-BNN for Data-Streams



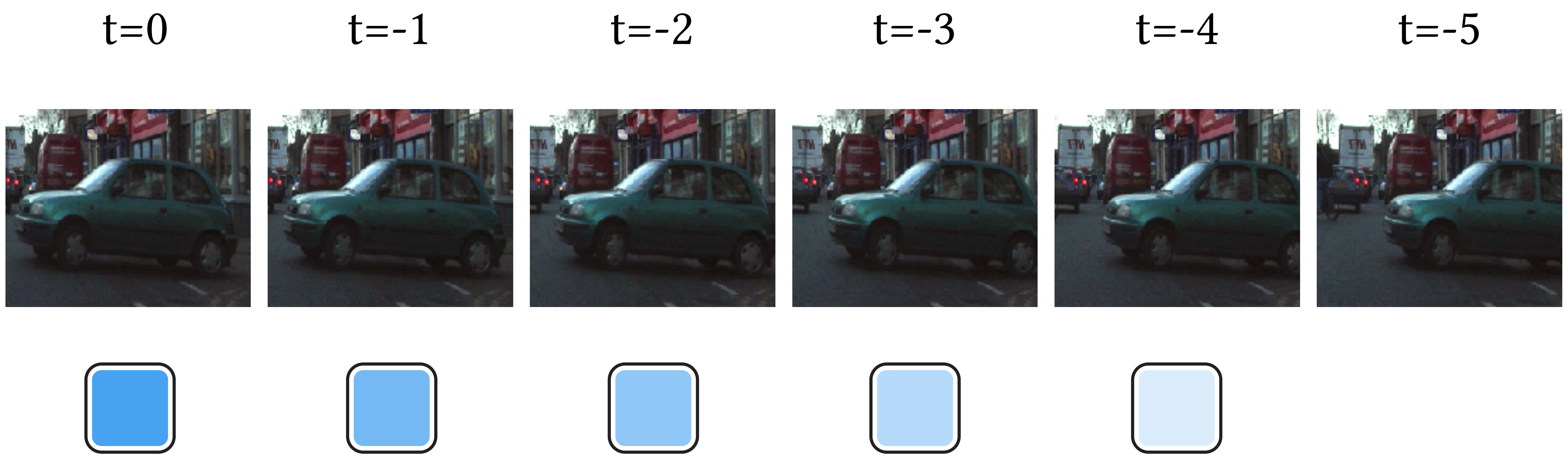
t=0 t=-1 t=-2 t=-3 t=-4 t=-5

Observed

K Memorized prototypes

We only take recent ***K*** frames as prototypes.

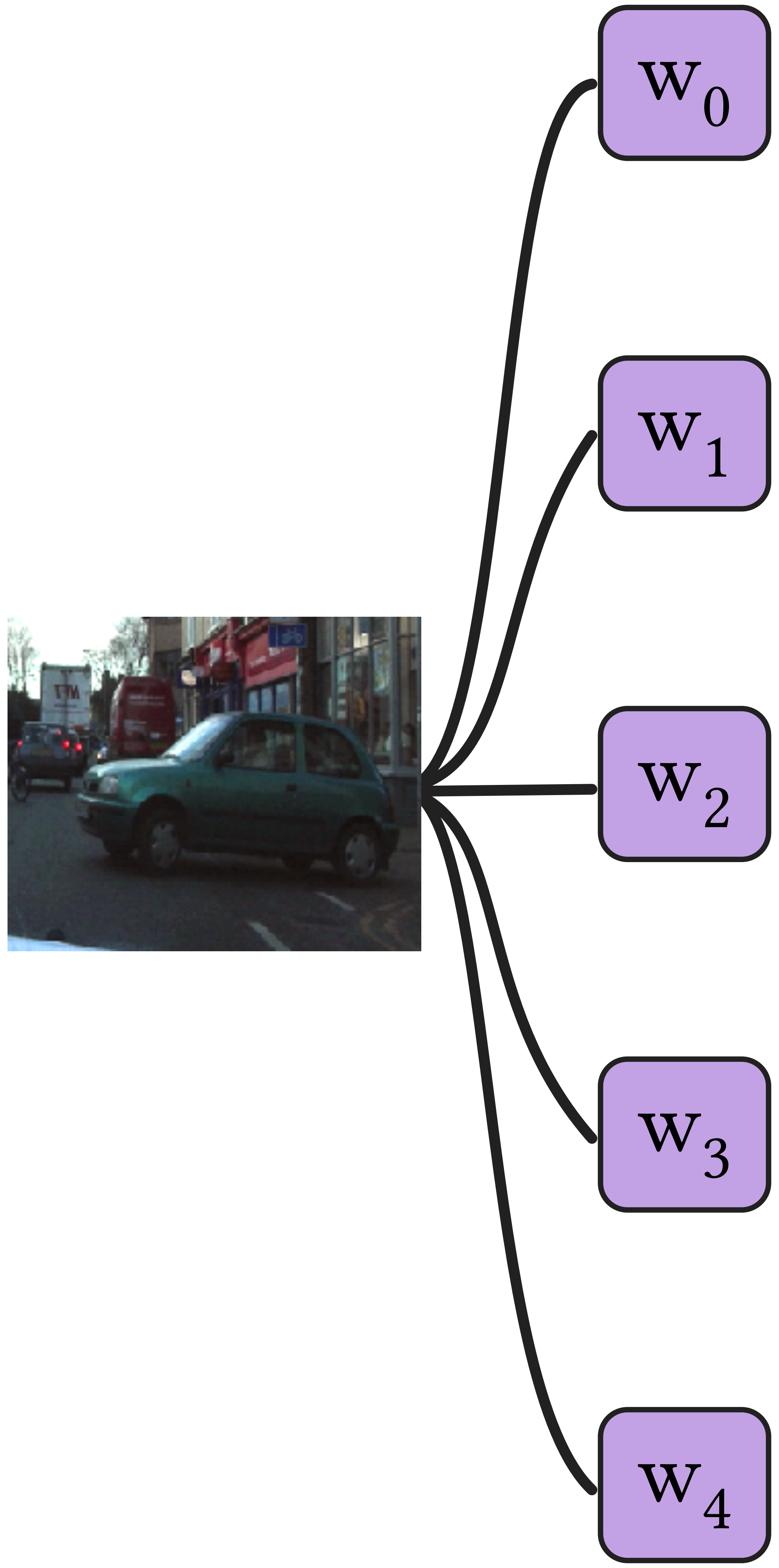
Temporal Smoothing: VQ-BNN for Data-Streams



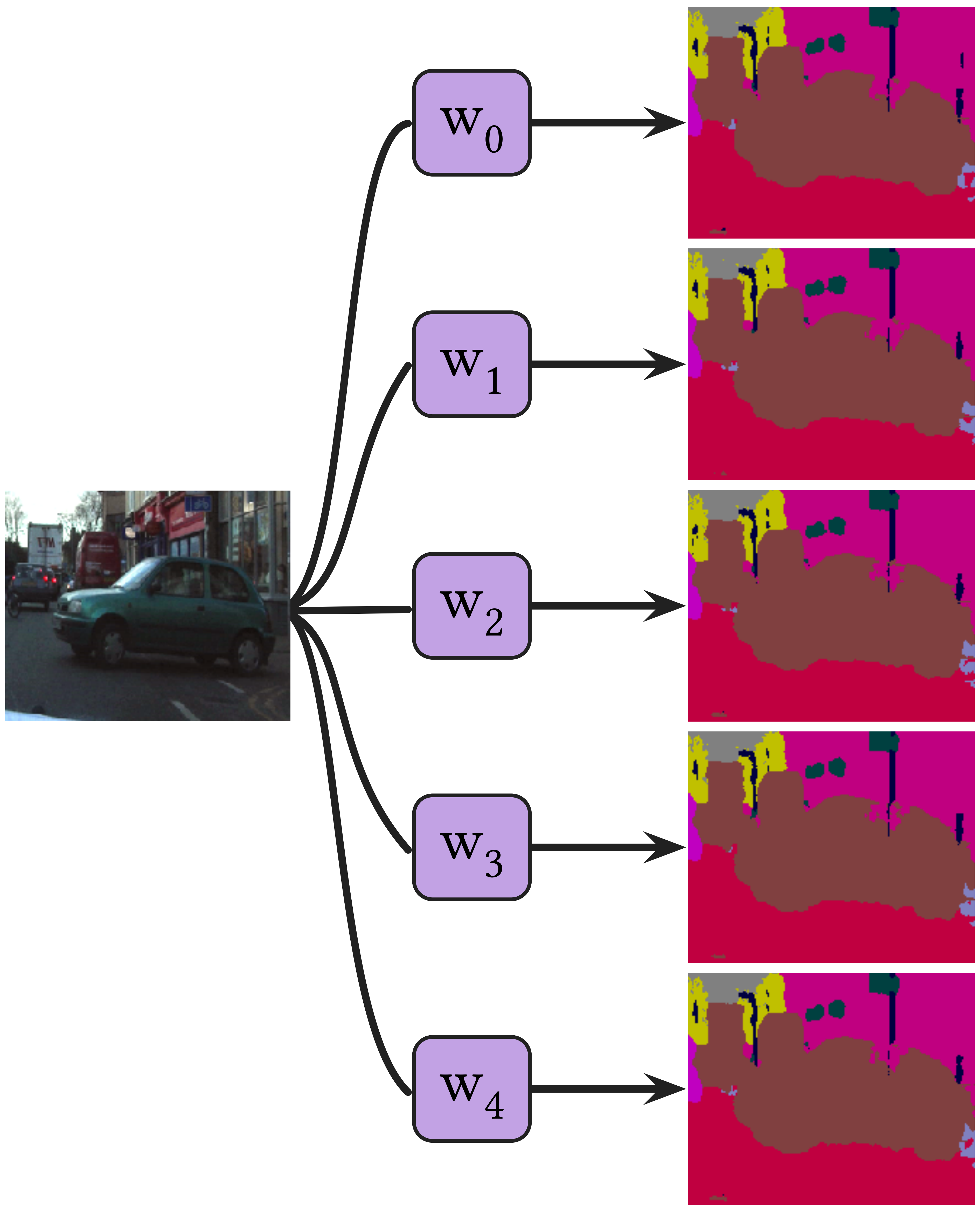
Importances (similarities) decrease exponentially.

Input NN weight Prediction Importance

Input NN weight Prediction Importance



Input NN weight Prediction Importance



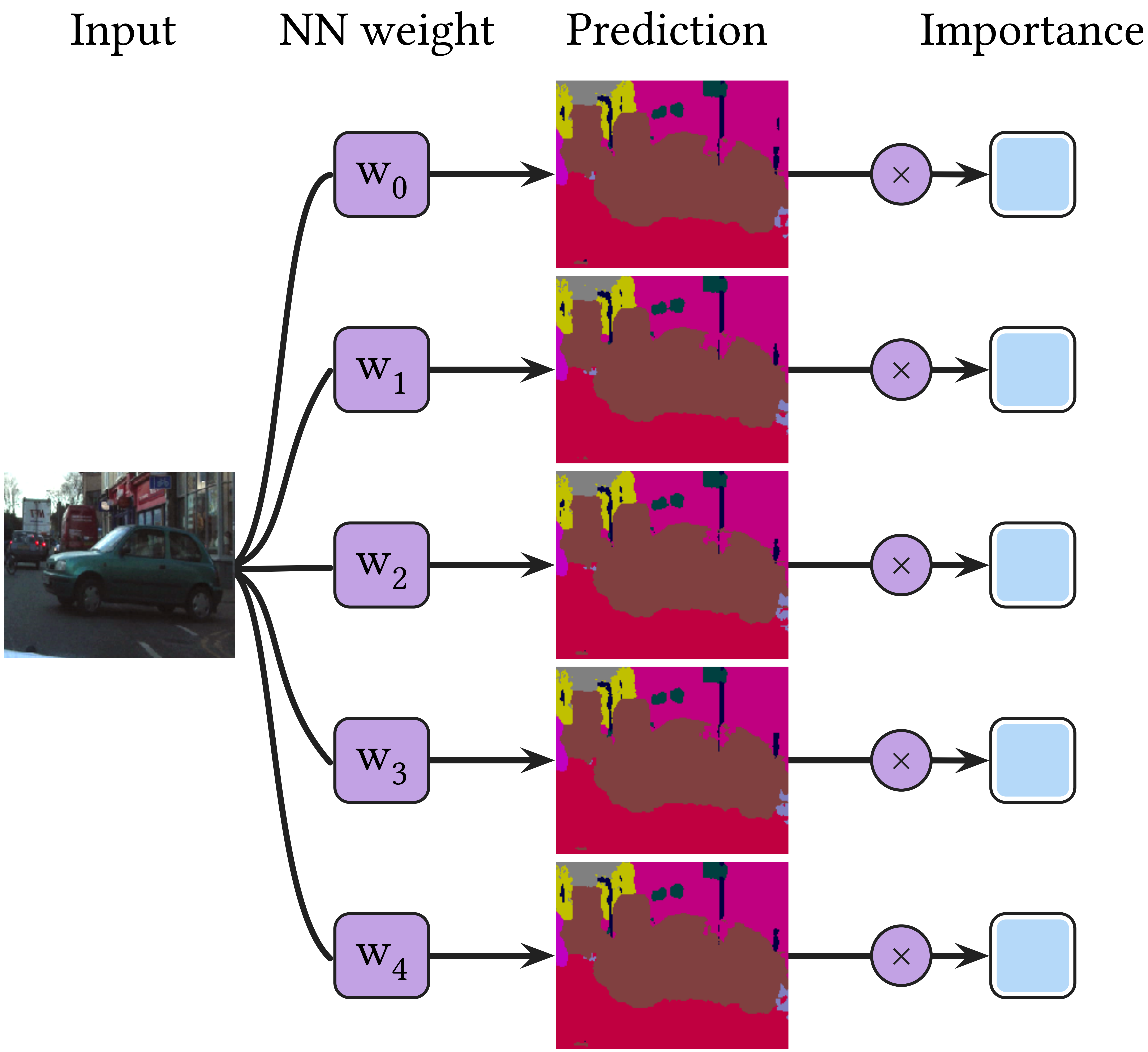
w_0

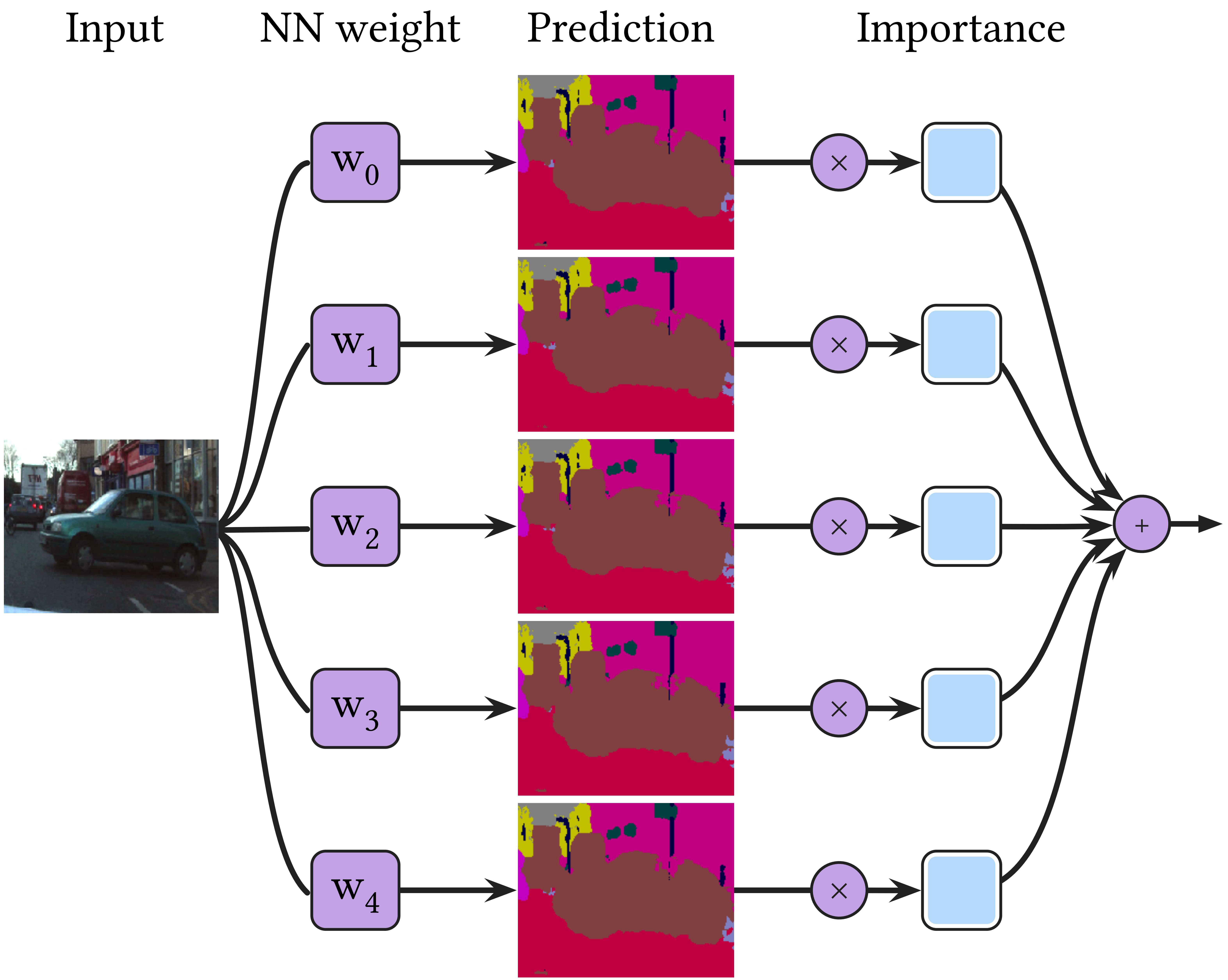
w_1

w_2

w_3

w_4


Input
NN weight
Prediction
Importance
w_0
w_1
w_2
w_3
w_4
×
×
×
×
×


Input
NN weight
Prediction
Importance
w_0
w_1
w_2
w_3
w_4
×
+

Input NN weight Prediction Importance

Observed

t=-4

Input NN weight Prediction Importance

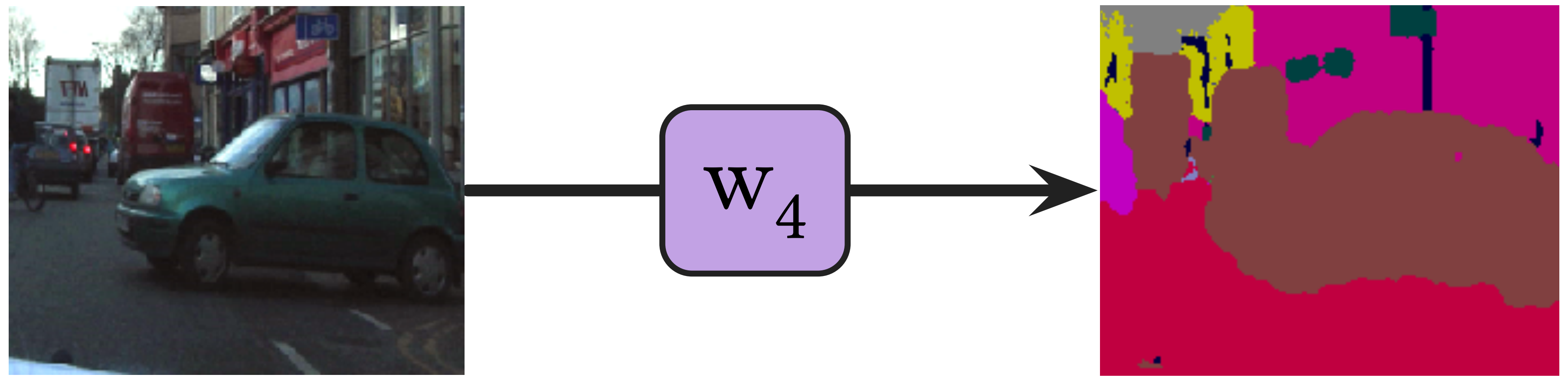
Observed

t=-4

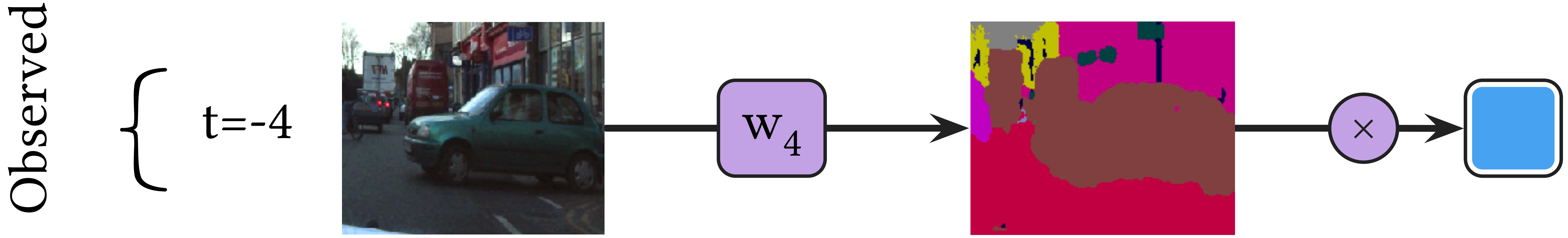
Input NN weight Prediction Importance

Observed

{ t=-4



Input NN weight Prediction Importance



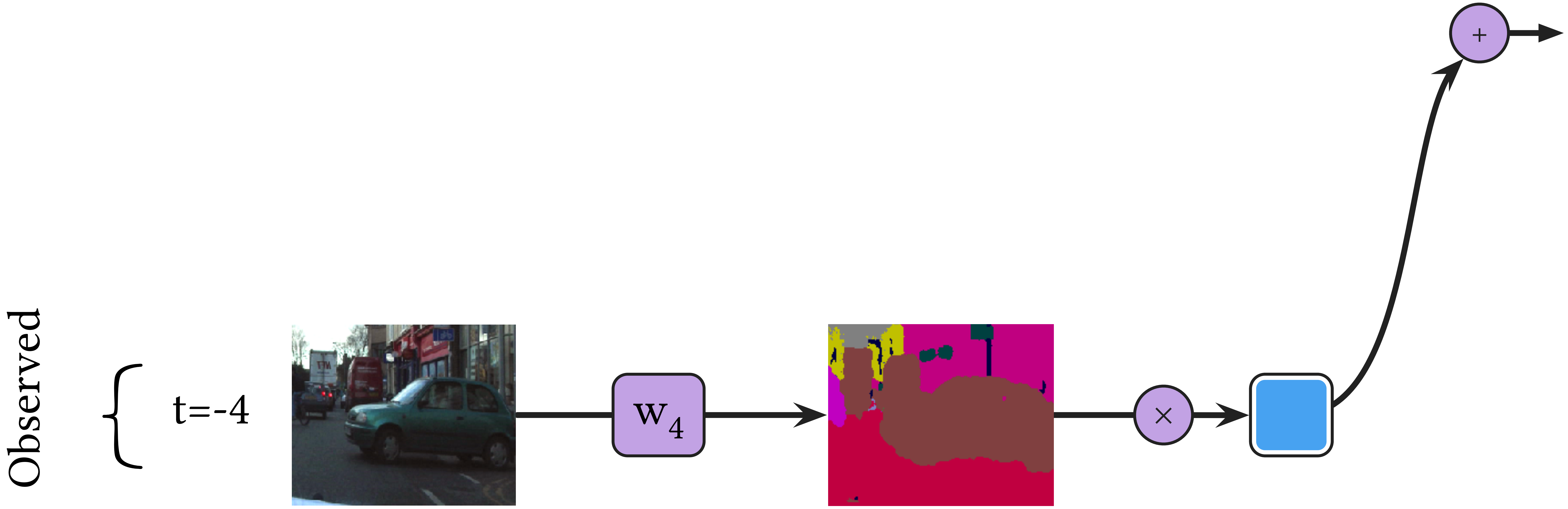
Observed

t=-4

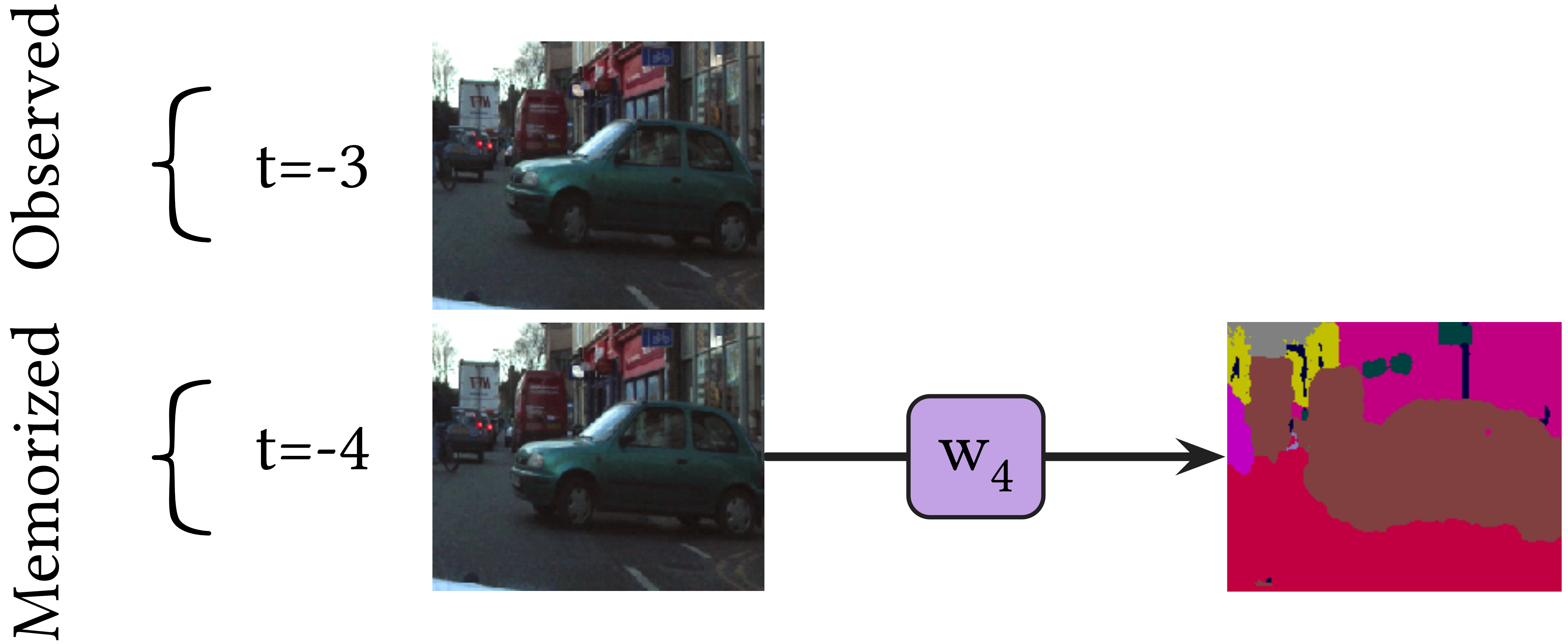
w_4

×

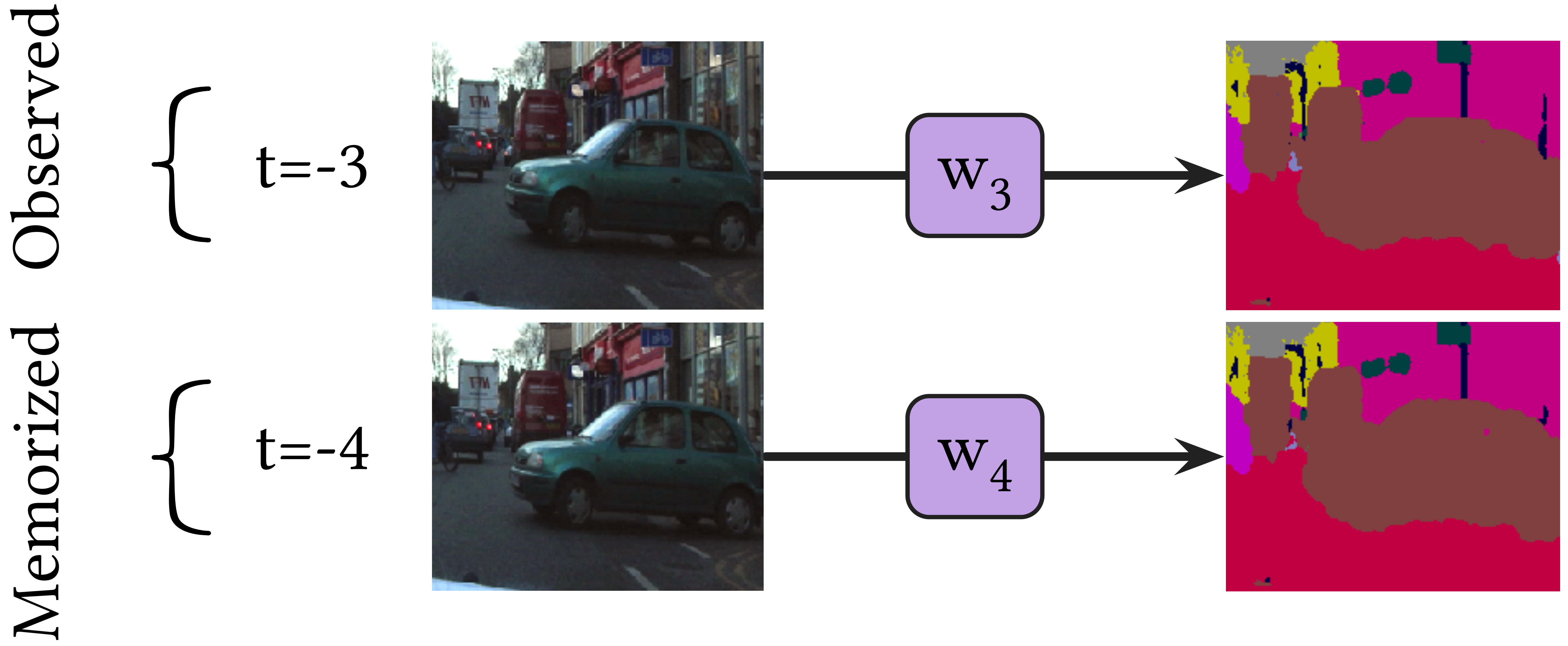
Input NN weight Prediction Importance



Input NN weight Prediction Importance



Input NN weight Prediction Importance



Observed

t=-3

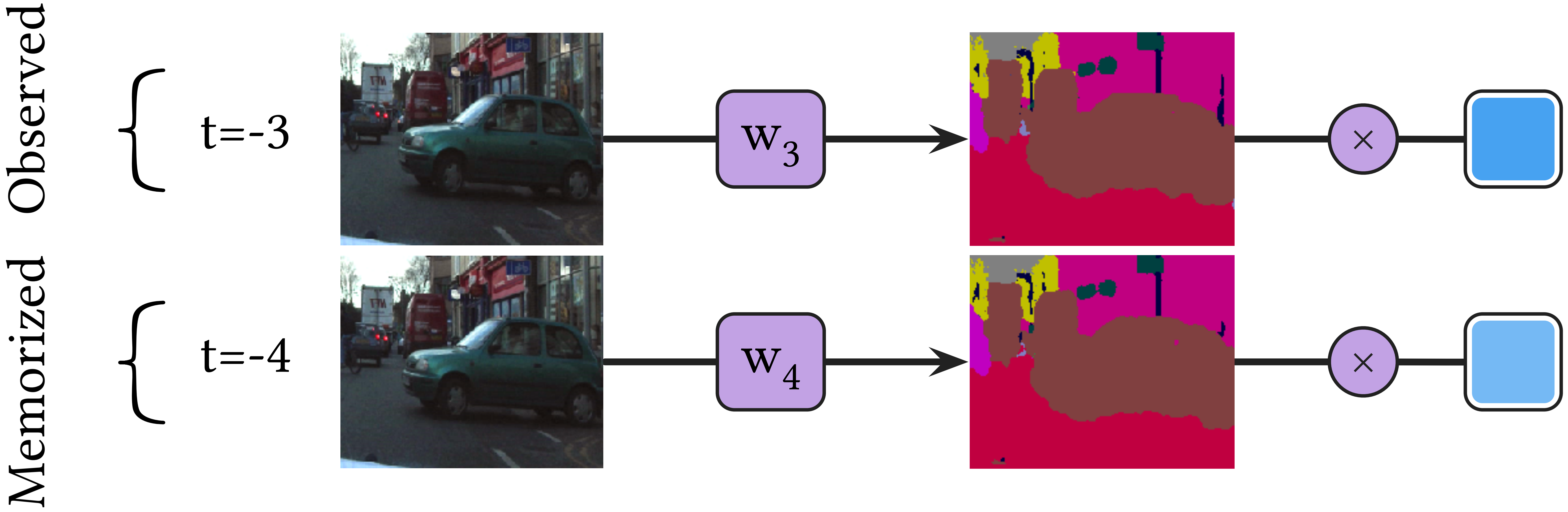
w_3

Memorized

t=-4

w_4

Input NN weight Prediction Importance



Observed

t=-3

w_3

×

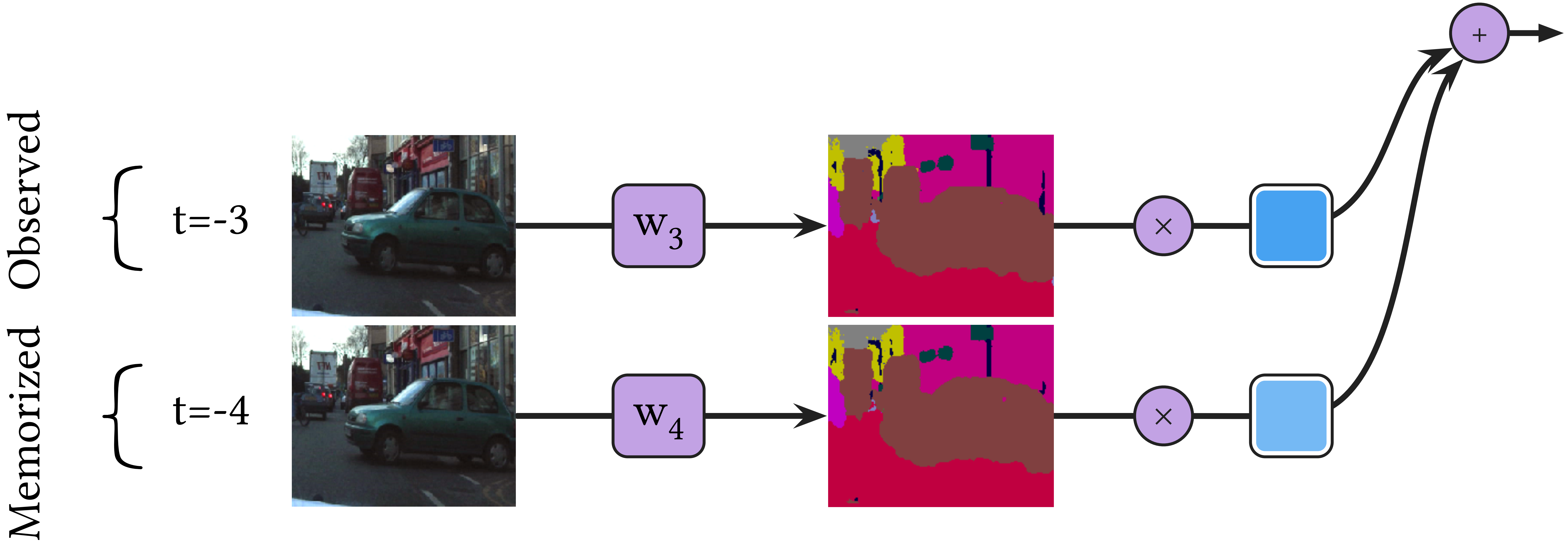
Memorized

t=-4

w_4

×

Input NN weight Prediction Importance



Observed

t=-3

w_3

×

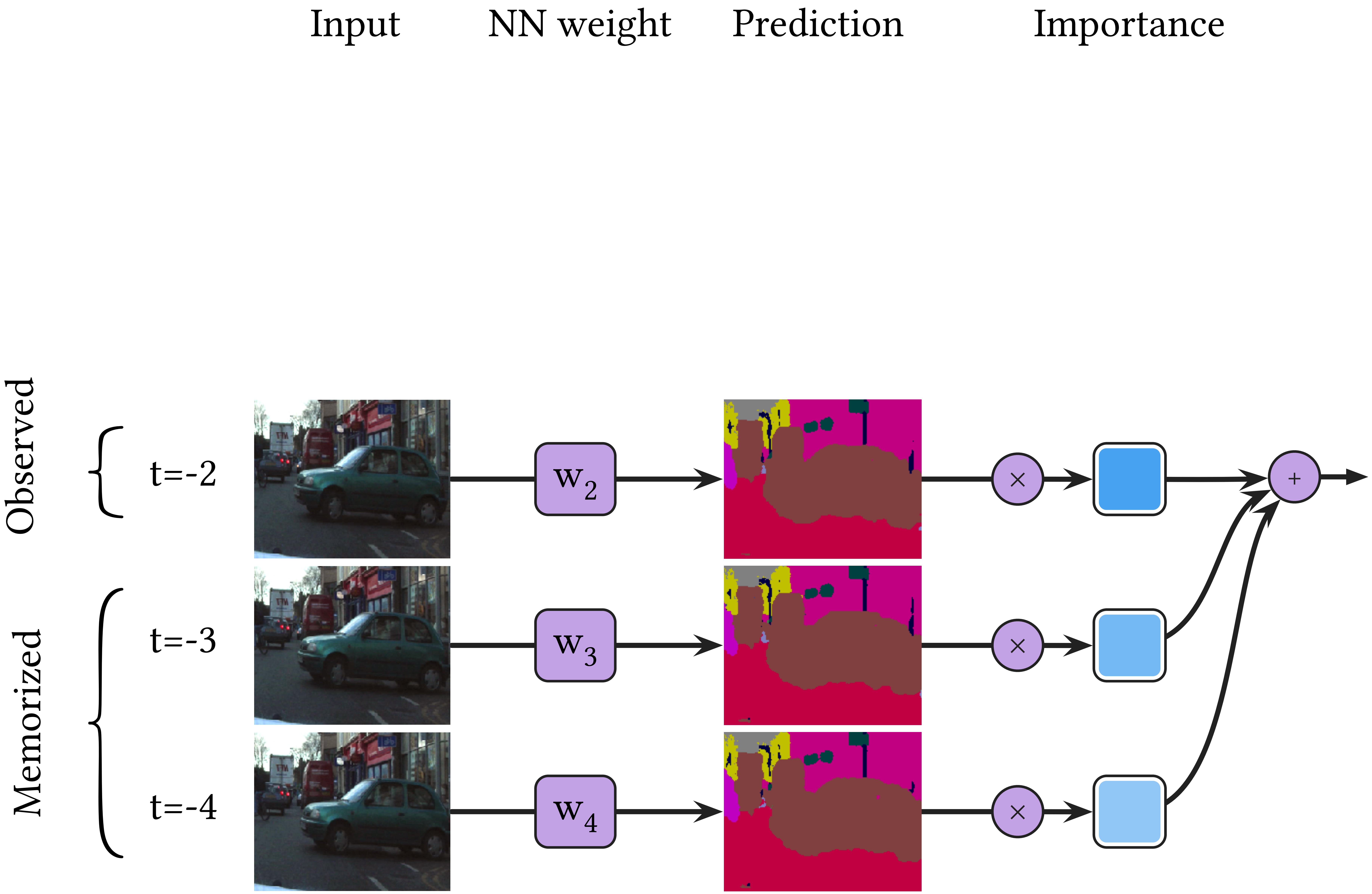
Memorized

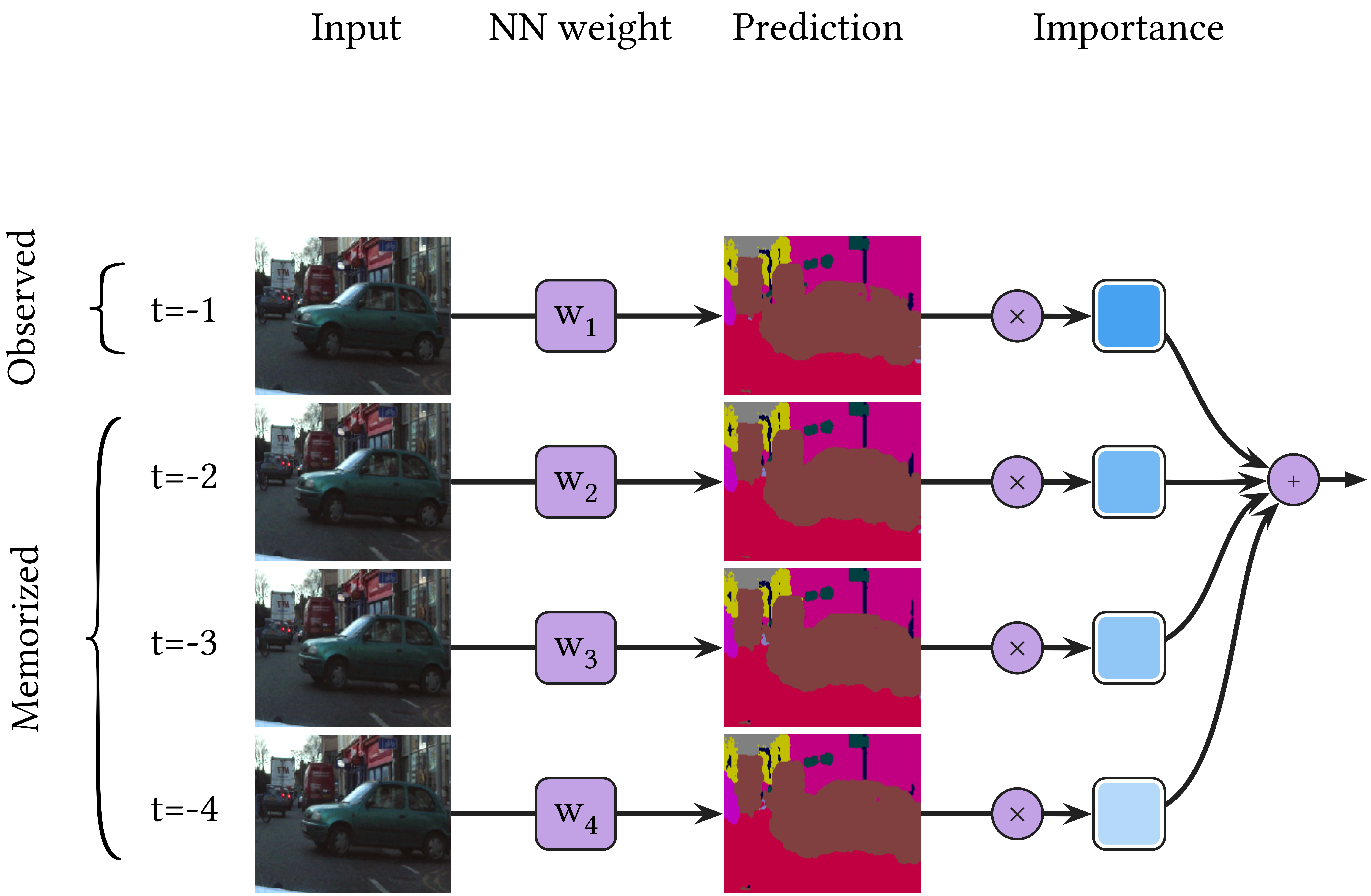
t=-4

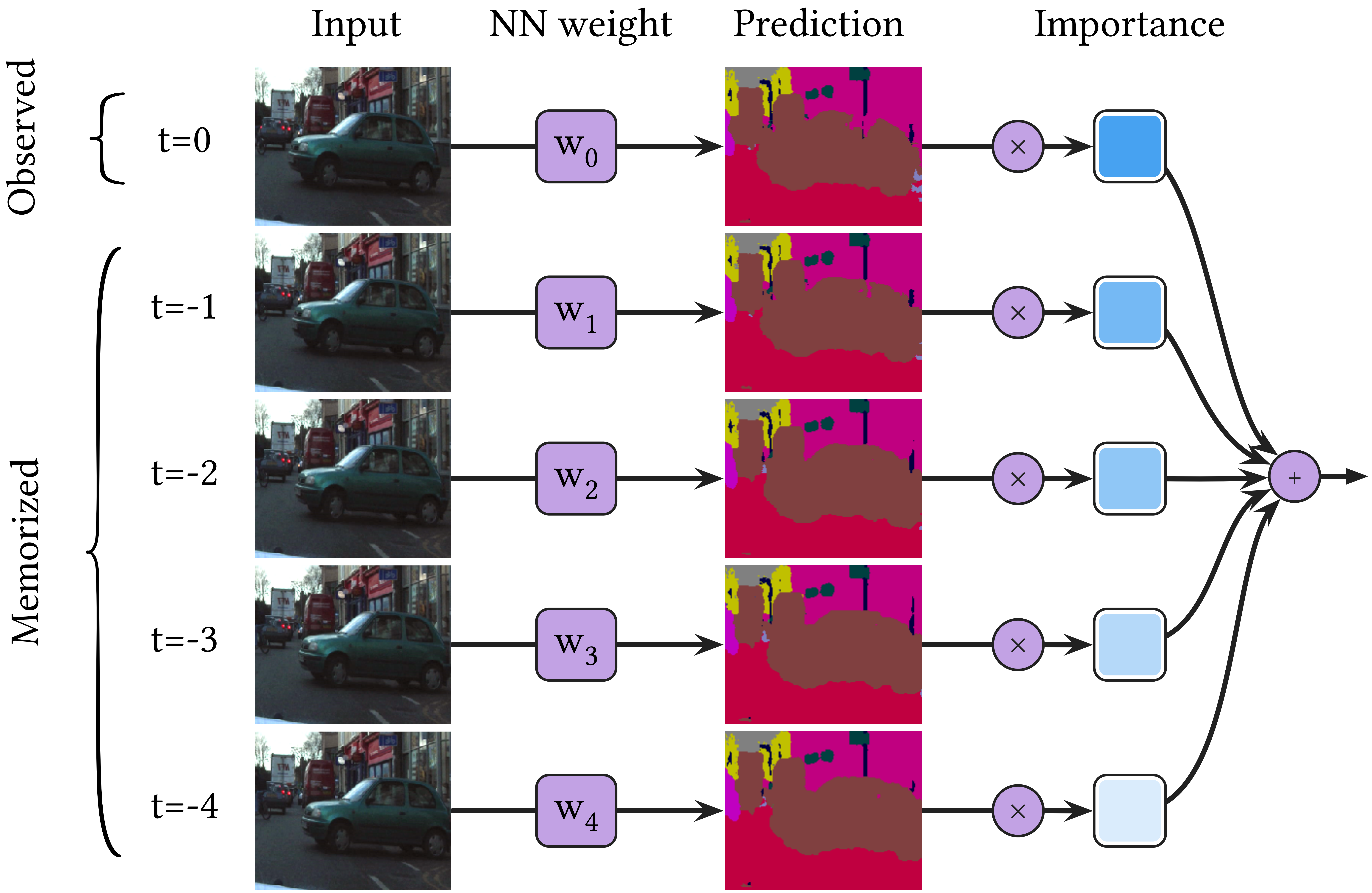
w_4

×

+


Input
NN weight
Prediction
Importance
Observed
Memorized
t=-2
t=-3
t=-4
w_2
w_3
w_4
×
×
×
+


Input
NN weight
Prediction
Importance
Observed
Memorized
t=-1
t=-2
t=-3
t=-4
w1
w2
w3
w4
×
+


Input
NN weight
Prediction
Importance
Observed
t=0
Memorized
t=-1
t=-2
t=-3
t=-4
w_0
w_1
w_2
w_3
w_4
×
+

Temporal Smoothing Significantly Improves the Performances

Method	Rel Thr (%, ↑)	NLL (↓)	Acc (%, ↑)	ECE (%, ↓)
Deterministic	**100**	0.314	91.1	4.31
MC dropout	2.99	0.276	91.8	3.71
MC dropout + Smoothing	92.7	**0.253**	**92.0**	**2.24**

Temporal smoothing improves both computational and predictive performance. Temporal smoothing—VQ-BNN for data streams—is as fast as deterministic NN and significantly faster than BNN. In addition, VQ-BNN predicts accurate results and reliable uncertainties. We report relative throughput (*Rel Thr*) and predictive performance of semantic segmentation on CamVid.

Temporal Smoothing Significantly Improves the Performances

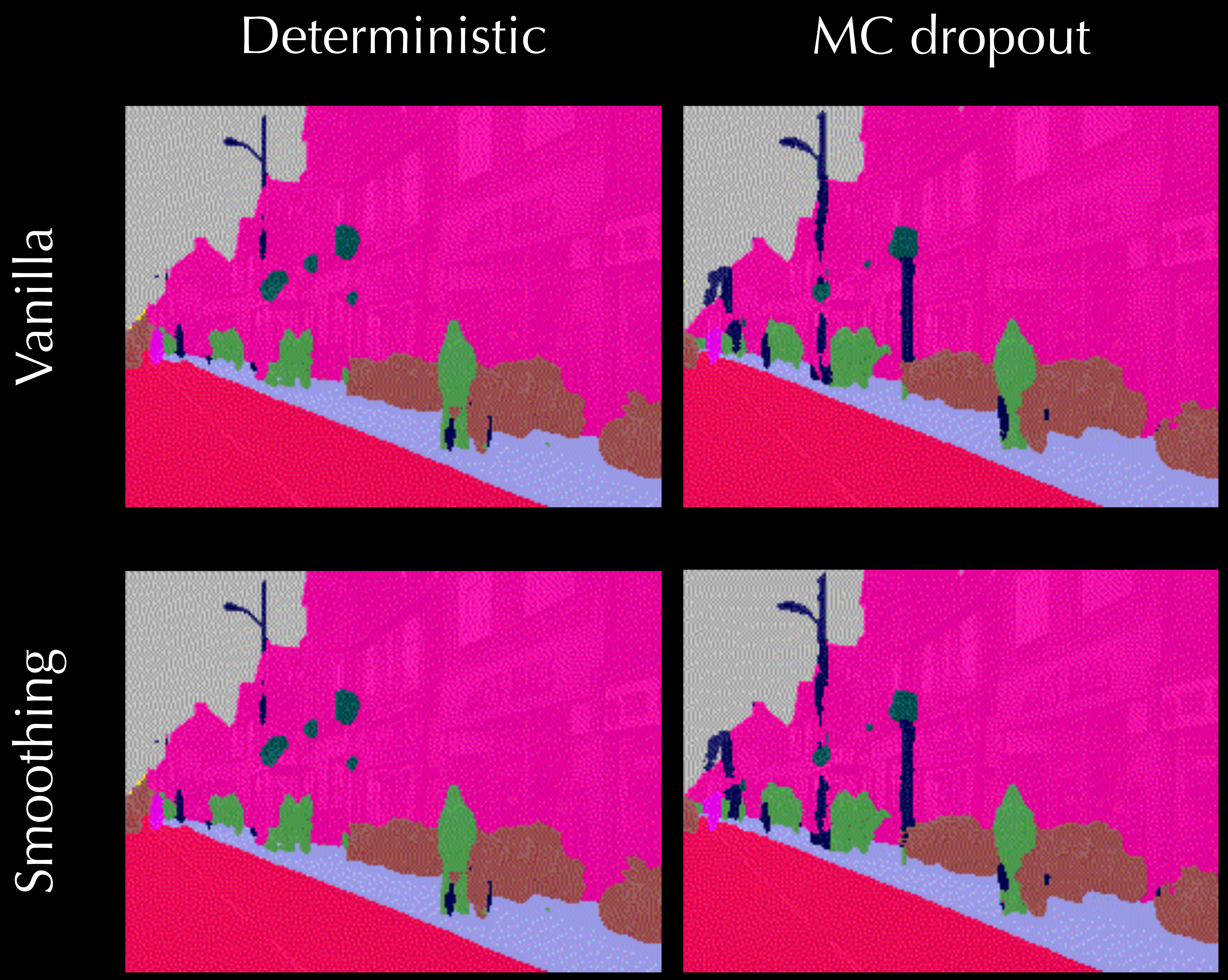
Method	Rel Thr	NLL	Acc	ECE
	(%, ↑)	(↓)	(%, ↑)	(%, ↓)
Deterministic	**100**	0.314	91.1	4.31
MC dropout	2.99	0.276	91.8	3.71
MC dropout + Smoothing	92.7	**0.253**	**92.0**	**2.24**

Temporal smoothing improves both computational and predictive performance. Temporal smoothing—VQ-BNN for data streams—is as fast as deterministic NN and significantly faster than BNN. In addition, VQ-BNN predicts accurate results and reliable uncertainties. We report relative throughput (*Rel Thr*) and predictive performance of semantic segmentation on CamVid.

Temporal Smoothing Significantly Improves the Performances

Method	Rel Thr (%, ↑)	NLL (↓)	Acc (%, ↑)	ECE (%, ↓)
Deterministic	**100**	0.314	91.1	4.31
MC dropout	2.99	0.276	91.8	3.71
MC dropout + Smoothing	92.7	**0.253**	**92.0**	**2.24**

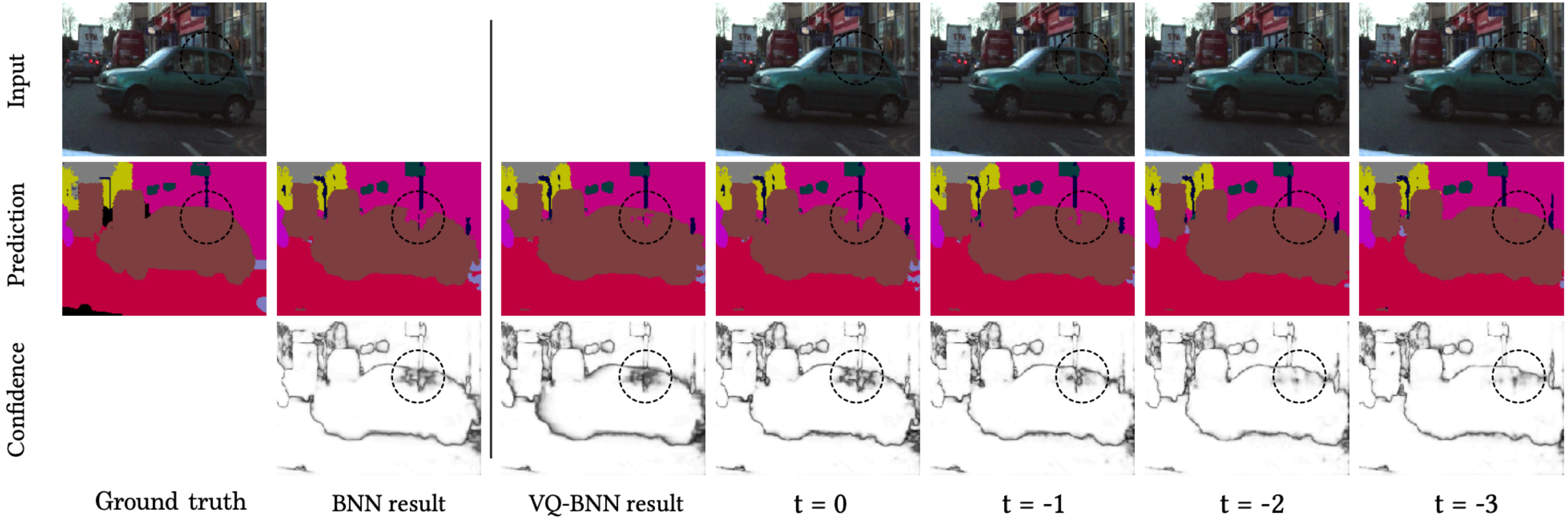
Temporal smoothing improves both computational and predictive performance. Temporal smoothing—VQ-BNN for data streams—is as fast as deterministic NN and significantly faster than BNN. In addition, VQ-BNN predicts accurate results and reliable uncertainties. We report relative throughput (*Rel Thr*) and predictive performance of semantic segmentation on CamVid.



Takeaway: The predictive results of Deterministic NN and BNN (MC dropout) are *noisy*, but temporal smoothings are *stabilized*.

This is an animation! Please visit: https://github.com/xxxnell/temporal-smoothing/

Past Predictions Compensate the Most Recent Prediction



Past predictions compensate the result. Even if VQ-BNN accidentally predicts a wrong result for the most recent frame, the past predictions compensate for this error. In this figure, the predictive uncertainty of VQ-BNN is mainly predictive result for the car than BNN does.

Appendix Two

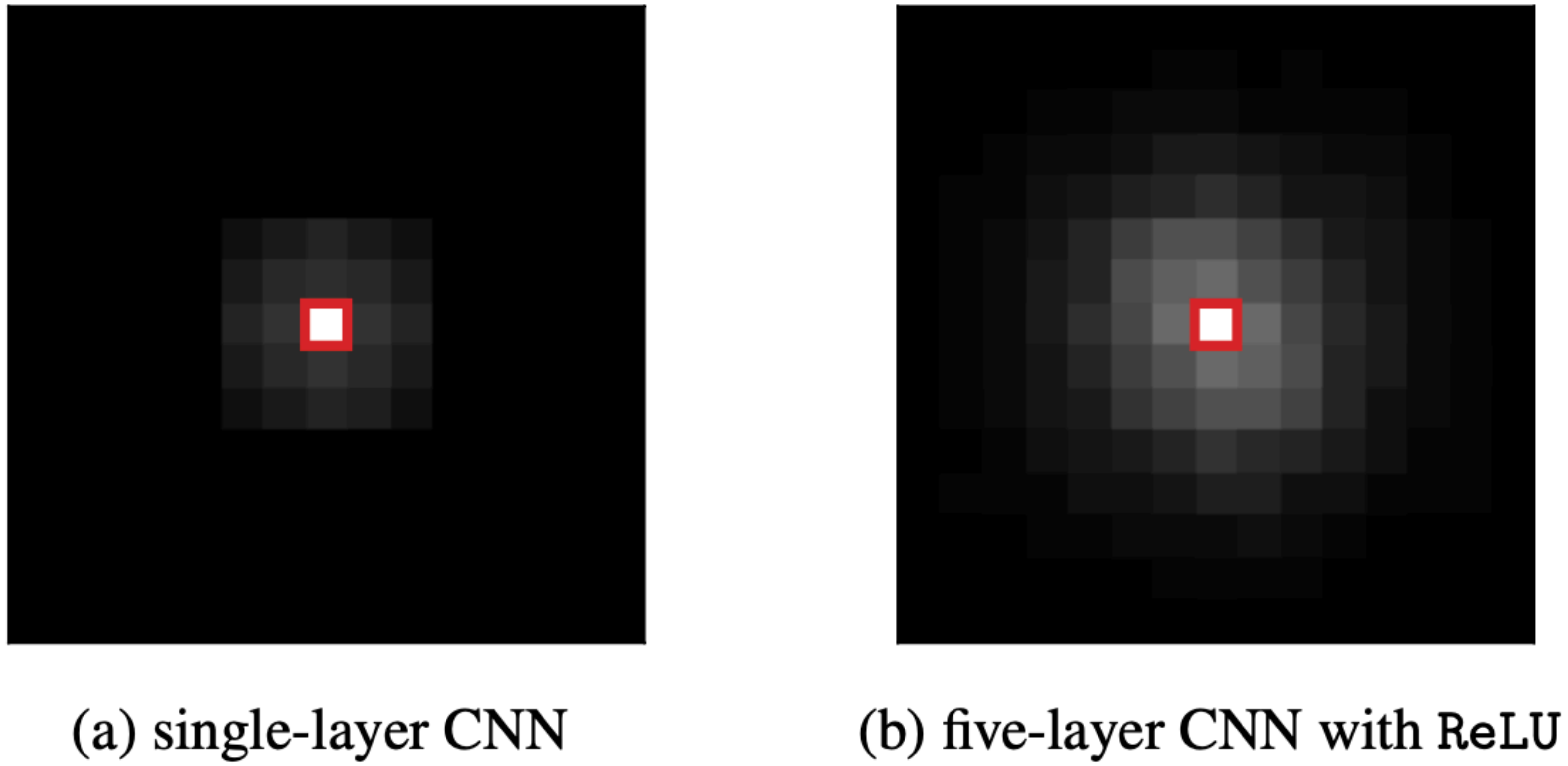
Spatial Smoothing

“Blur is an ensemble!”

Ensemble Averaging for Proximate Data Points

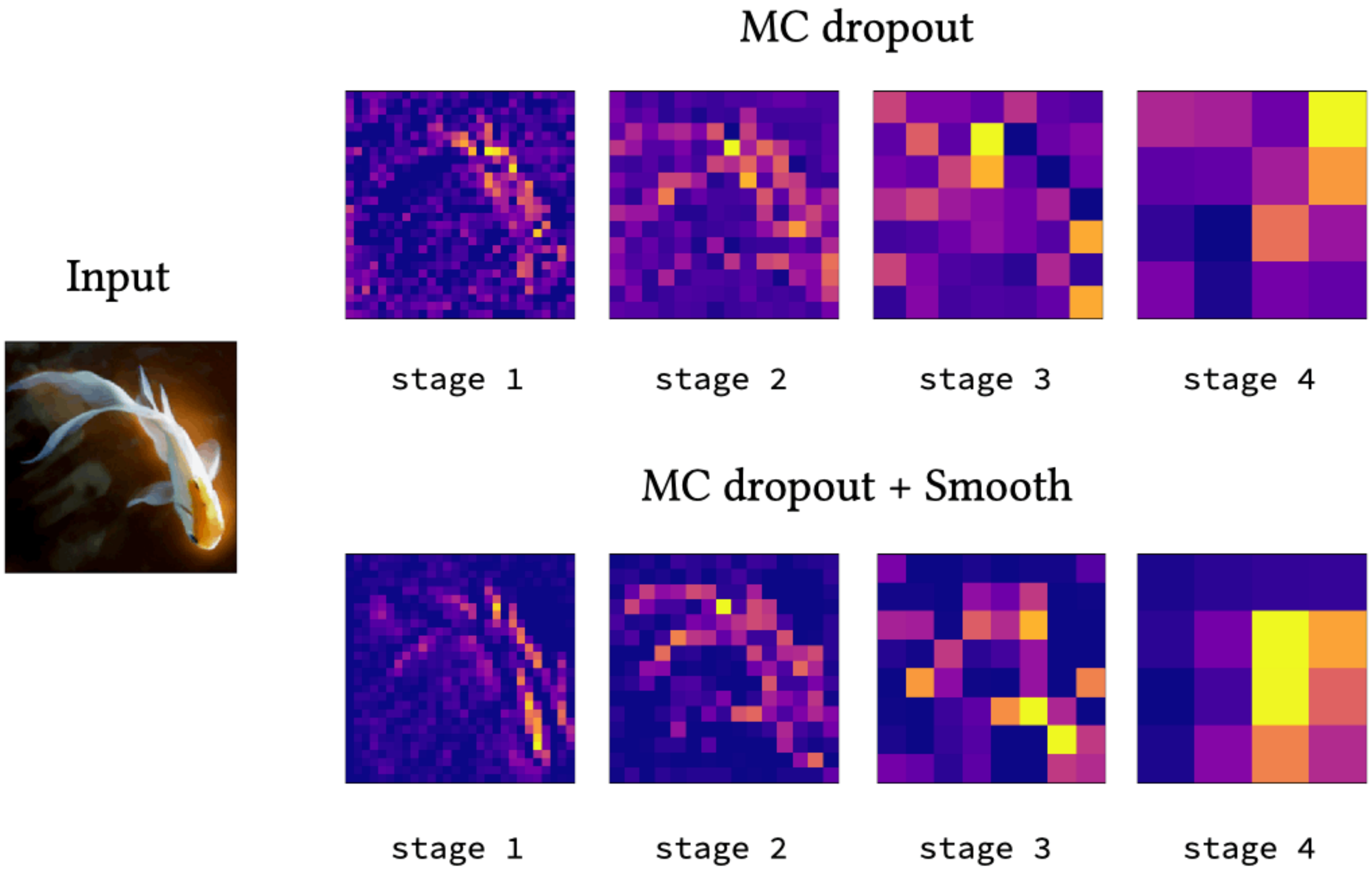
$$p(\boldsymbol{y}|\mathcal{S},\mathcal{D}) = \int \underbrace{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})}_{\text{NN predictions}}\overbrace{p(\boldsymbol{x}|\mathcal{S})}^{\textbf{\textit{Data uncertainty}}}\underbrace{p(\boldsymbol{w}|\mathcal{D})}_{\text{Model uncertainty}}d\boldsymbol{x}d\boldsymbol{w}$$

Spatial Consistency of Feature Maps in CNNs



Neighboring feature map points in CNNs are similar, even if input values are iid. We provide covariances of feature map points with respect to the center feature map (in the red square). Input values are Gaussian random noise. *Left*: A single convolutional layer correlates the target feature map with another feature map that is 3 pixels away, since the kernel size is 3×3. *Right*: A deep CNN more strongly correlates neighboring feature maps.

Spatial Smoothing Stabilizes Feature Maps

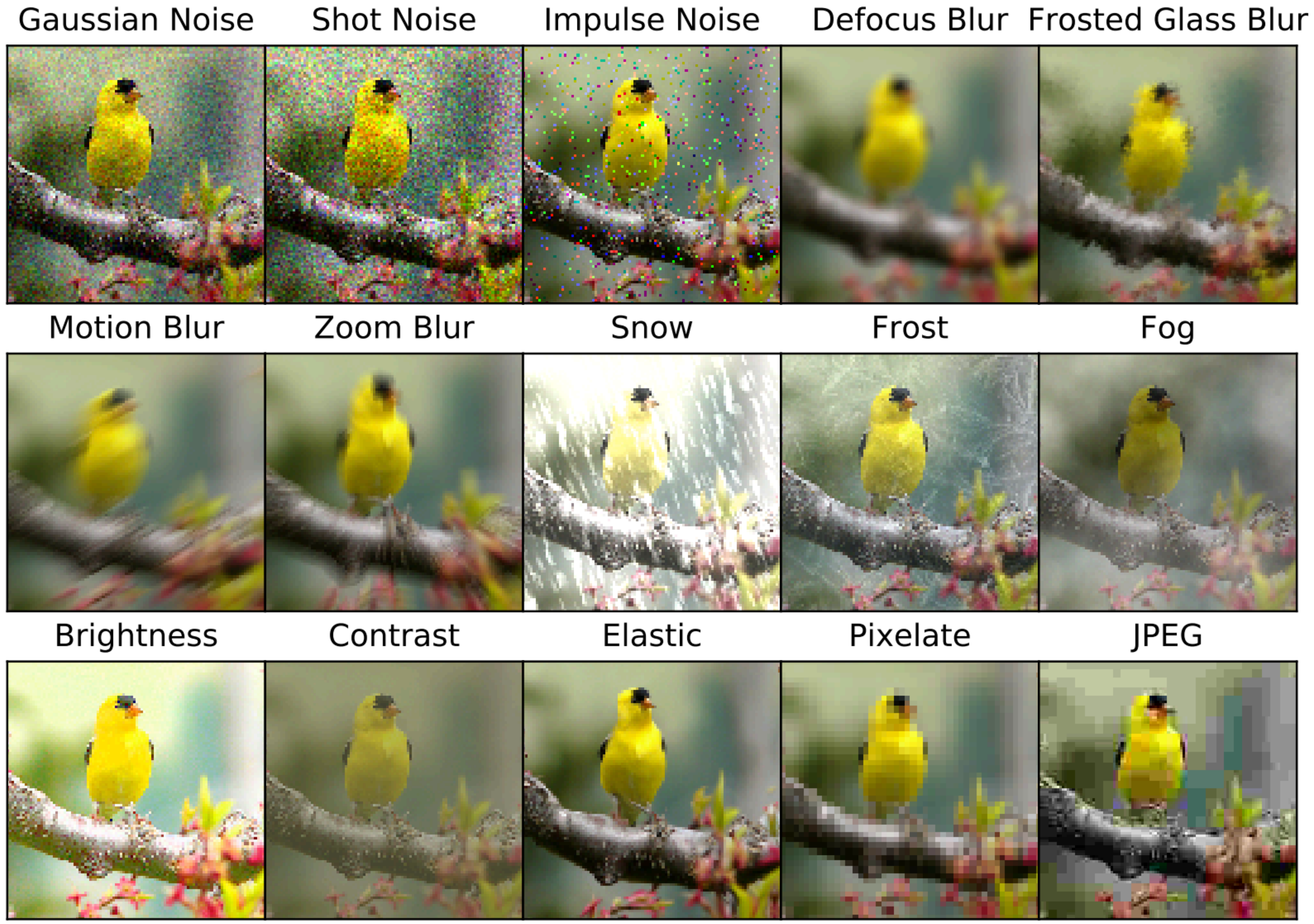


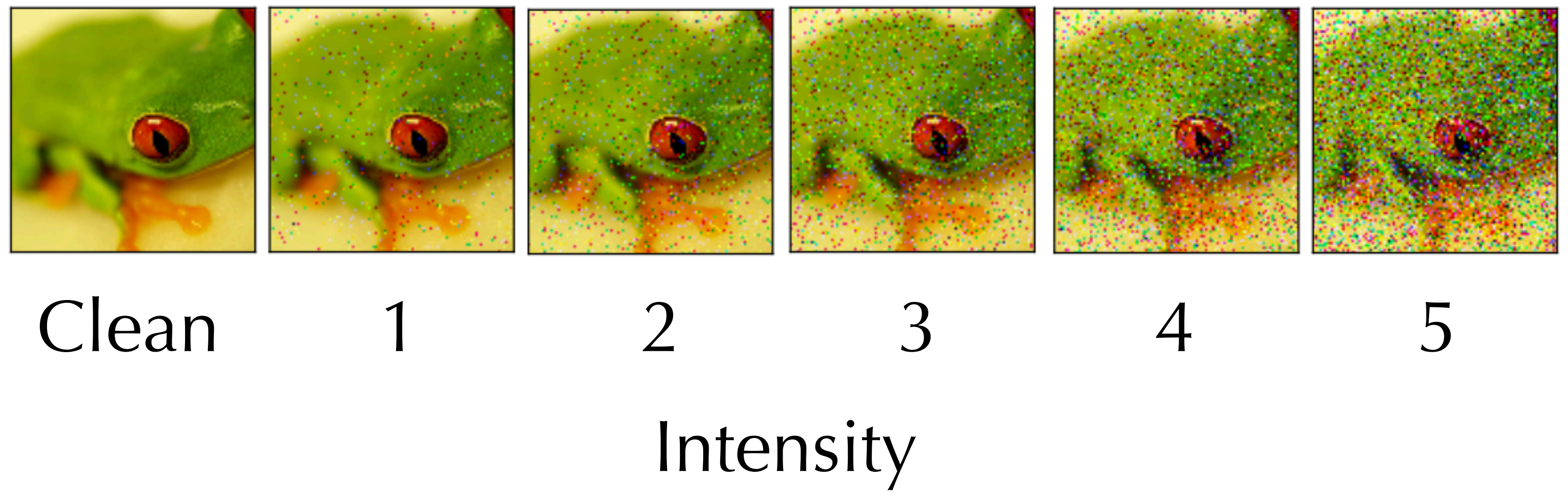
This is an animation! Please visit: https://github.com/xxxnell/spatial-smoothing/

Three Metrics for Uncertainty Estimation

- **Accuracy** (↑) on clean datasets and corrupted datasets.
- **Expected Calibration Error** (ECE, ↓) measures uncertainty only.
- **Negative Log-Likelihood** (NLL or Categorical Cross-entropy, ↓) measures both uncertainty and accuracy.

NNs Are Vulnerable

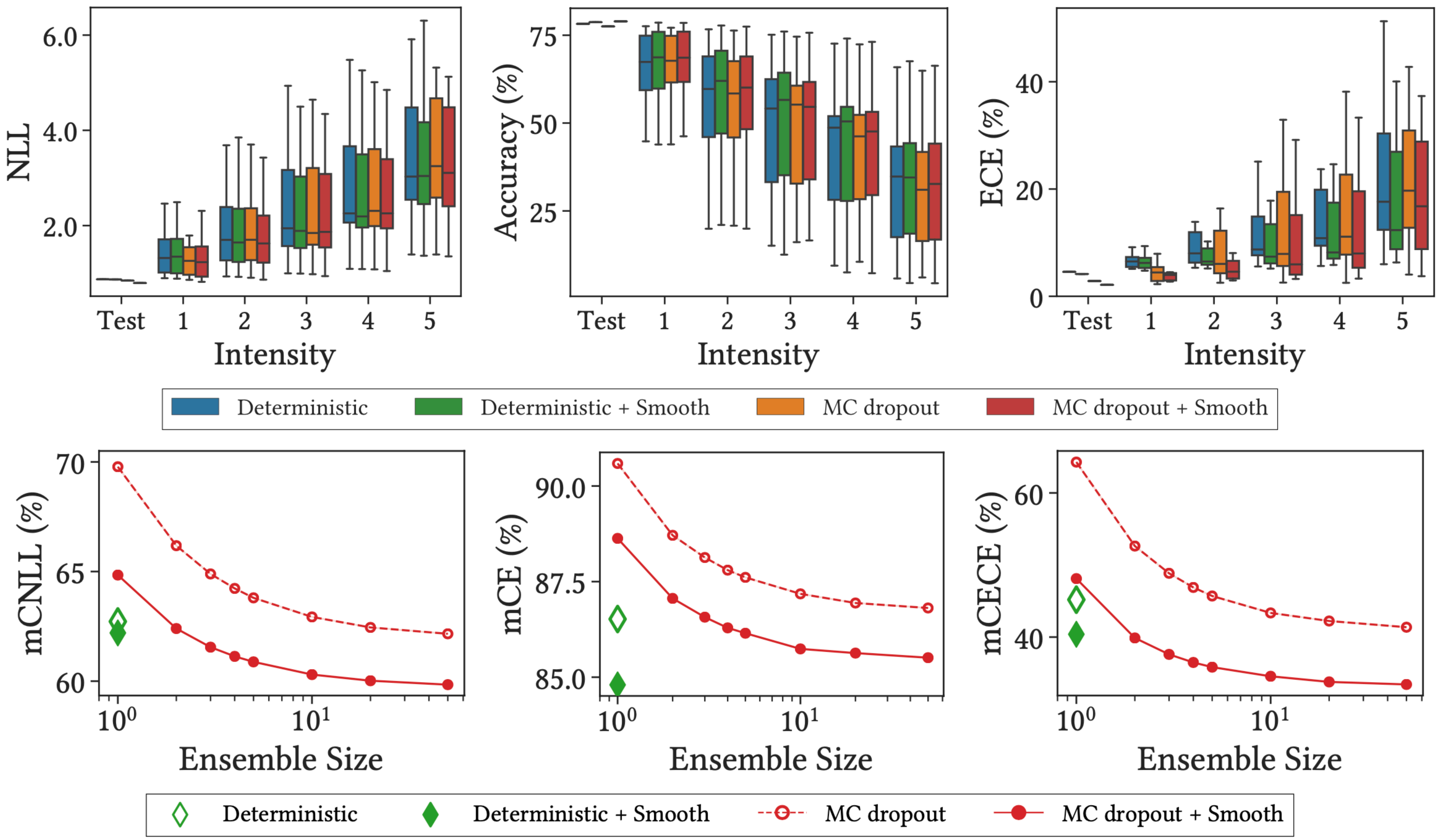




ImageNet-C dataset consists of 15 types of corruptions from noise, blur, weather, and digital categories. NNs are vulnerable against these corrupted datasets.

Hendrycks, Dan, and Thomas Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations." ICLR. 2018.

Spatial Smoothing Improves Robustness

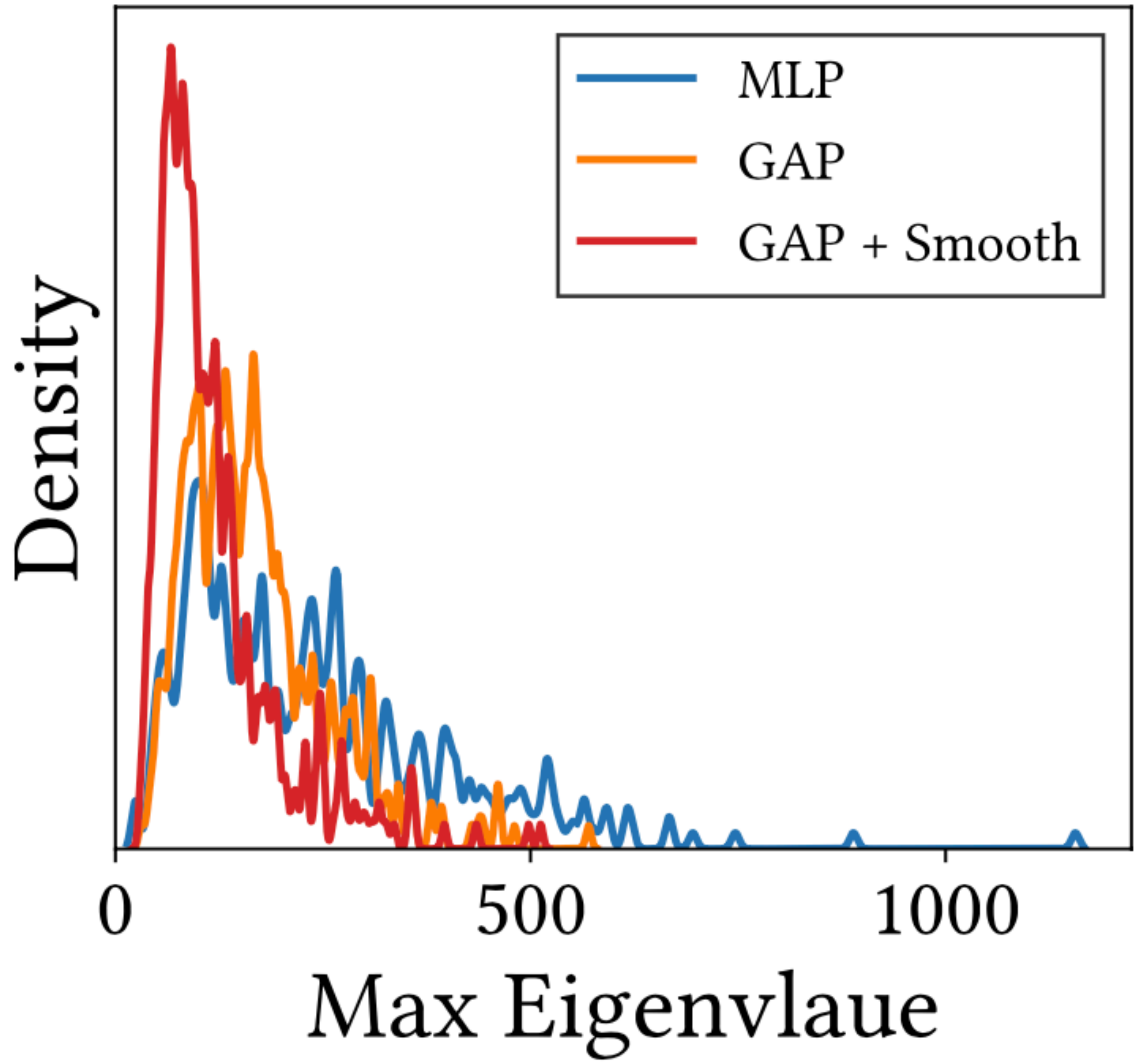


Spatial smoothing improves corruption robustness. We measure the predictive performance of ResNet-18 on CIFAR-100-C.

Three Key Properties of Ensembles

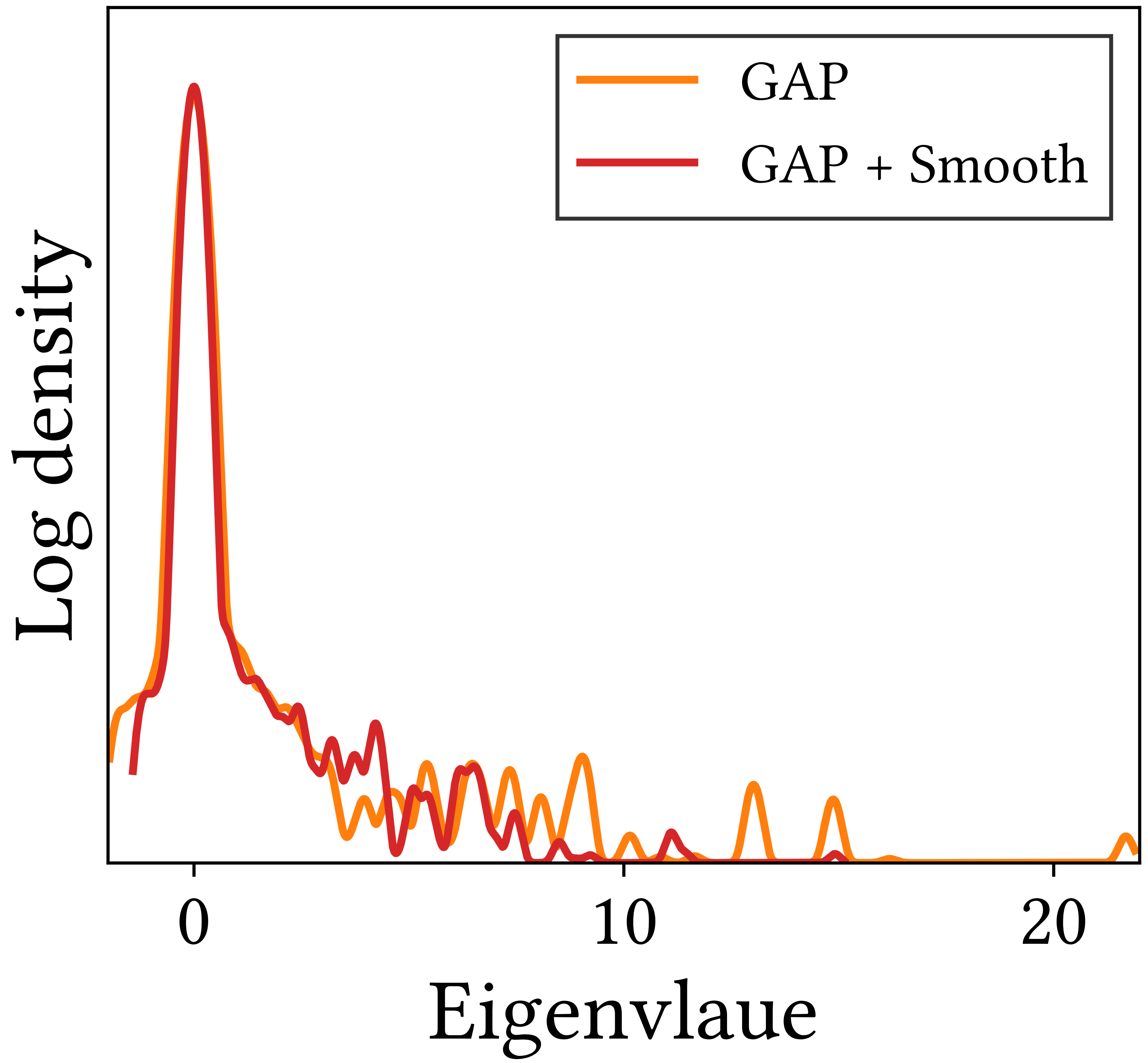
1. Ensembles flatten loss landscapes.
2. Ensembles filter out high-frequency signals.
3. Ensembles reduce feature map variances.

Spatial Smoothings Suppress Hessian Eigenvalue Outliers



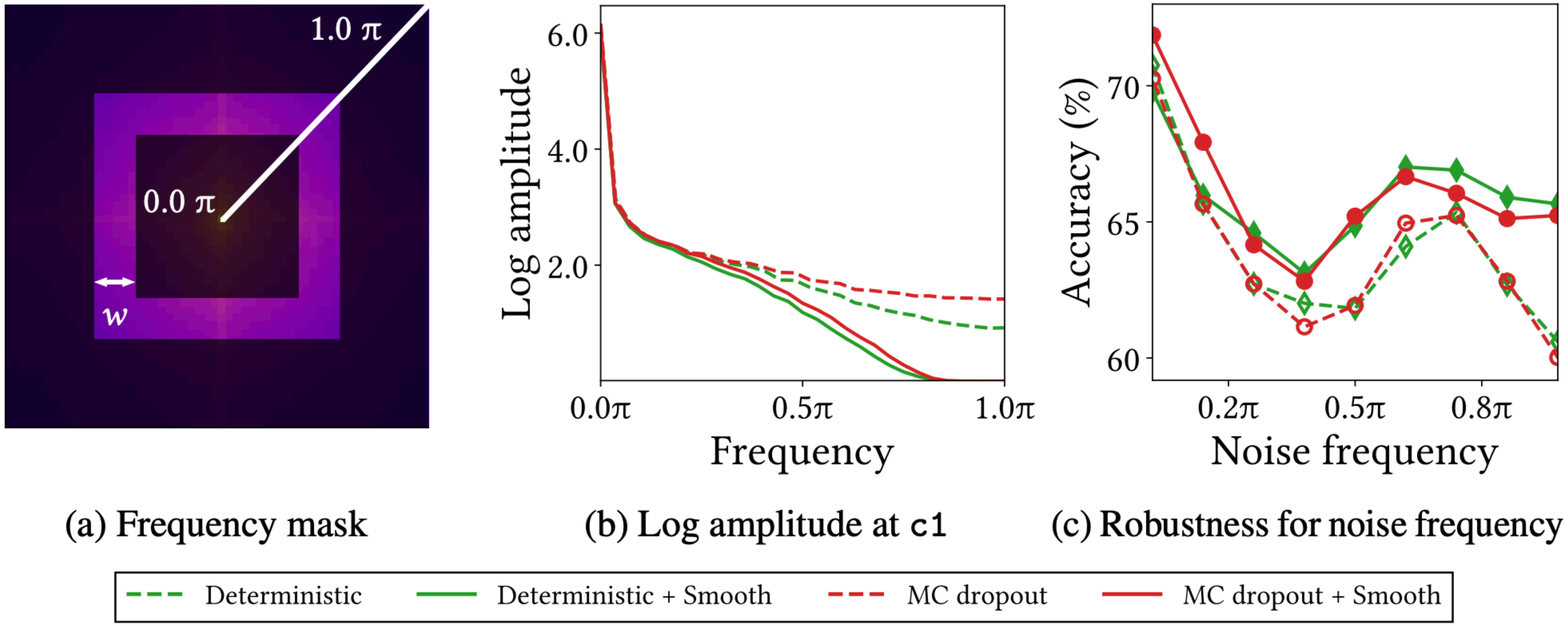
Both GAP and spatial smoothing suppress large Hessian eigenvalue outliers, i.e., they smoothen the loss function. The eigenvalue outliers disturb NN optimization.

Spatial Smoothings Suppress Hessian Eigenvalue Outliers



Both GAP and spatial smoothing suppress large Hessian eigenvalue outliers, i.e., they smoothen the loss function. The eigenvalue outliers disturb NN optimization.

Spatial Smoothing Makes NN Robust Against High-Freq Noise

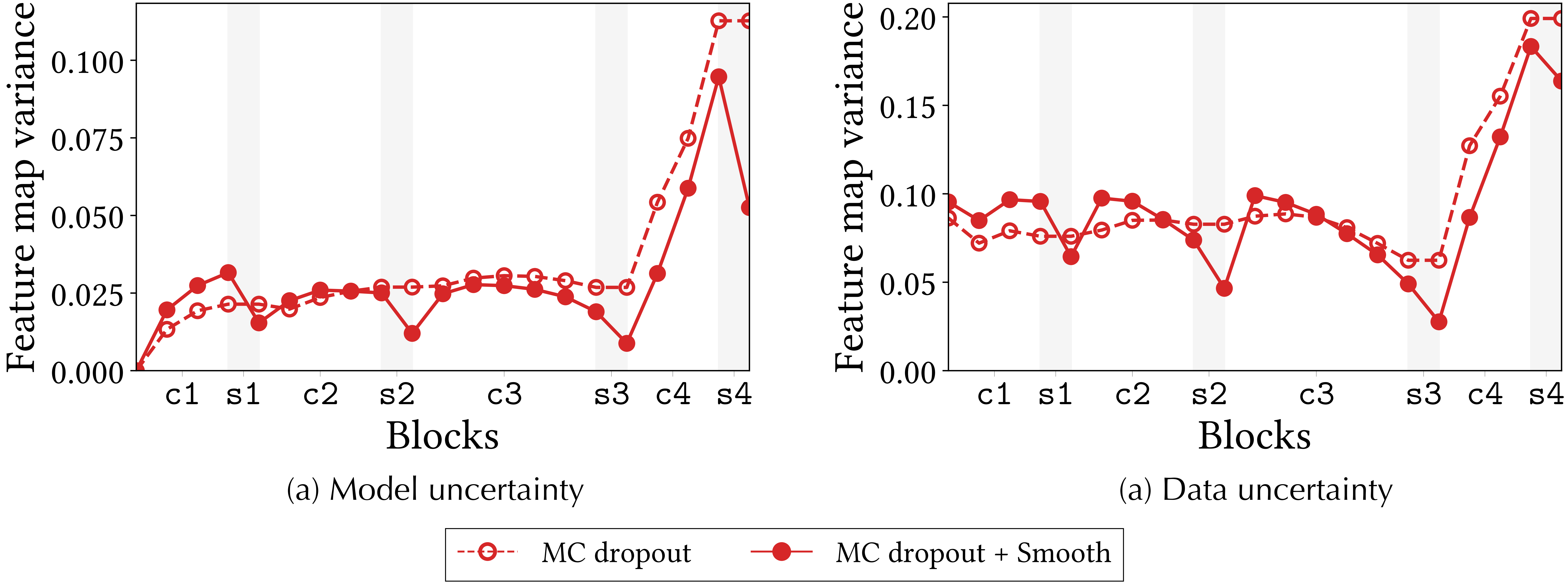


(a) Frequency mask (b) Log amplitude at `c1` (c) Robustness for noise frequency

MC dropout adds high-frequency noises, and spatial smoothing filters high-frequency signals.

Left: Frequency mask. *Middle*: Diagonal components of Fourier transformed feature maps at the end of the stage 1. *Right*: The accuracy against frequency-based random noise. ResNets are vulnerable to high-frequency noises. Spatial smoothing improves the robustness against high-frequency noises.

Spatial Smoothing Stabilize Feature Maps



(a) Model uncertainty

(a) Data uncertainty

Spatial smoothing layers reduce feature map variances, suggesting that they ensemble feature map points. “c1” to “c4” and “s1” to “s4” each stand for stages and spatial smoothing layers, respectively. Model uncertainty is represented by the average standard deviation of several feature maps obtained from multiple NN executions. Data uncertainty is represented by the standard deviation of feature map points obtained from one NN execution.

Formulation of Spatial Smoothing

Deterministic (vanilla) NN: $p(\boldsymbol{y}|\boldsymbol{x}_0, \mathcal{D}) = p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{w}_0)$

Bayesian NN:

$$p(\boldsymbol{y}|\boldsymbol{x}_0, \mathcal{D}) = \int p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{w})\, p(\boldsymbol{w}|\mathcal{D})\, d\boldsymbol{w}$$

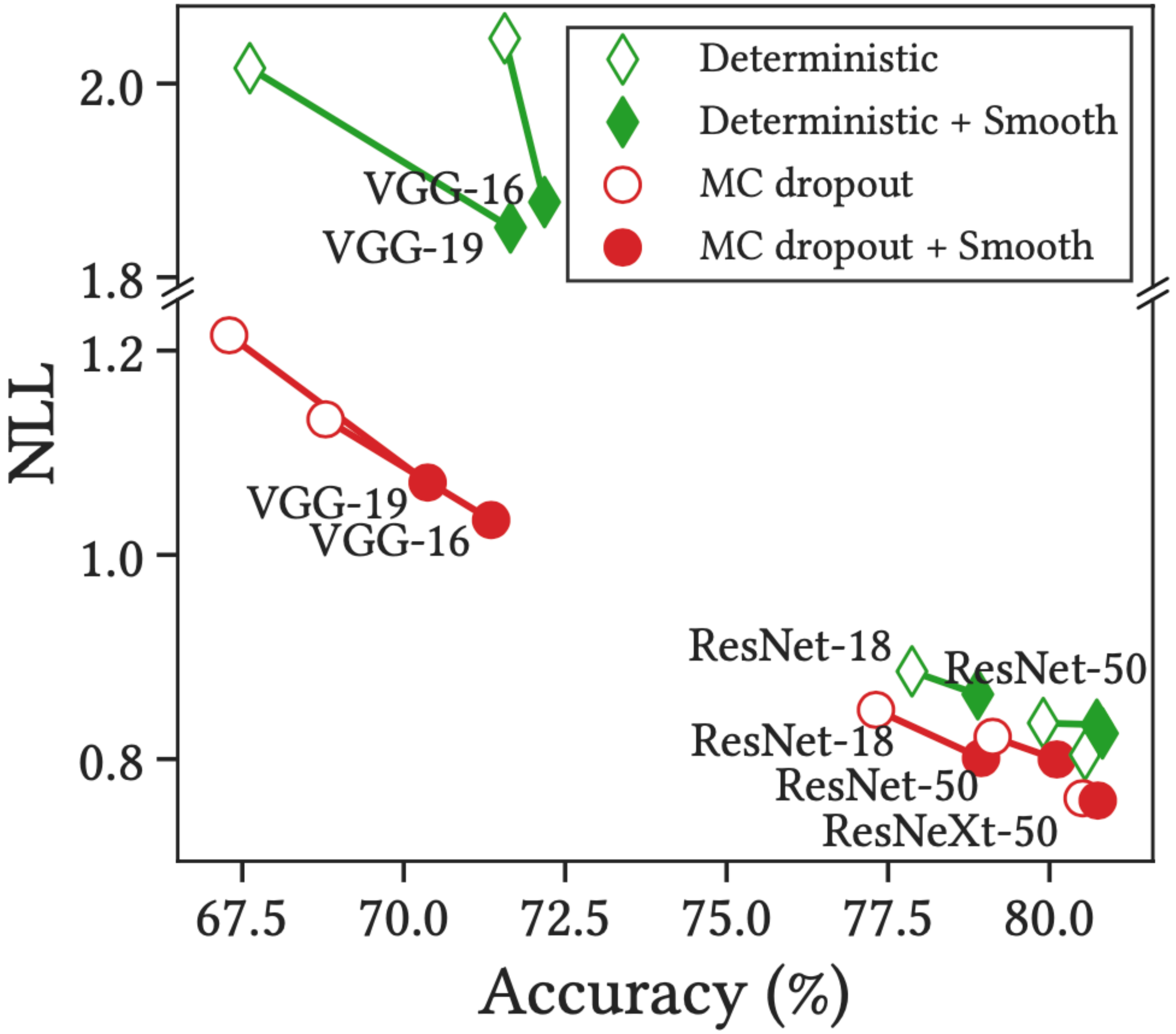
$$\simeq \sum_{i=0}^{N-1} \frac{1}{N}\, p(\boldsymbol{y}|\boldsymbol{x}_0, \boldsymbol{w}_i)$$

Data-complemented BNN:

$$p(\boldsymbol{y}|\tilde{\boldsymbol{x}}, \mathcal{D}) = \int p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})\, p(\boldsymbol{w}|\mathcal{D})\, p(\boldsymbol{x}|\tilde{\boldsymbol{x}})\, d\boldsymbol{w} d\boldsymbol{x}$$

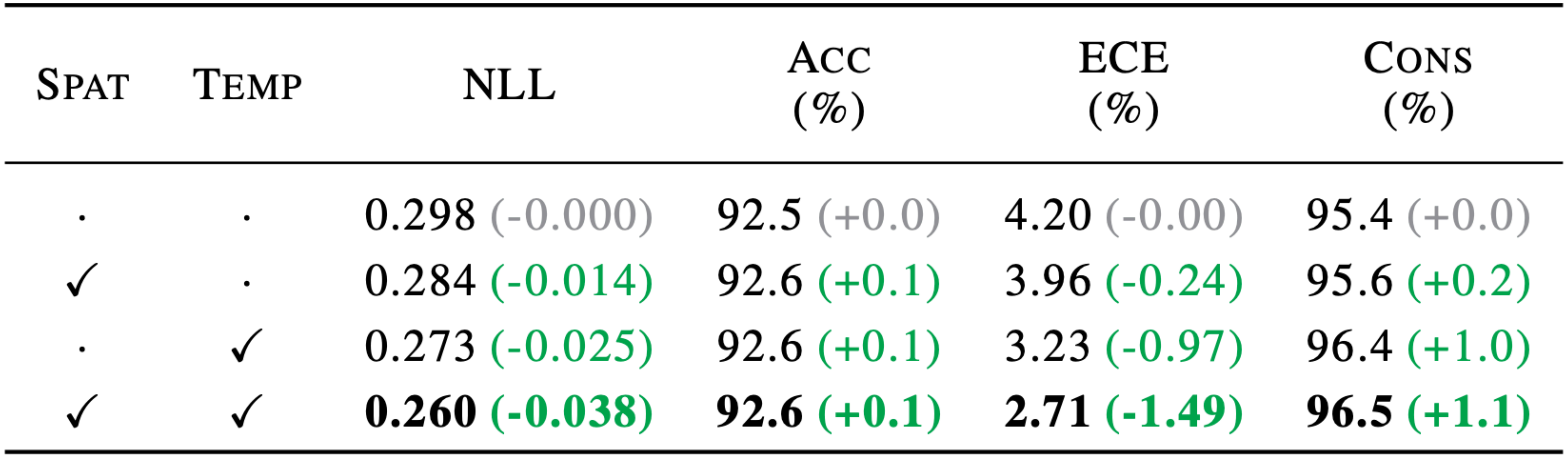
$$\simeq \sum_{i=0}^{N-1} \pi(\boldsymbol{x}_i|\boldsymbol{x}_0)\, p(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{w}_i)$$

Spatial Smoothing Improves Various Models



Spatial smoothing improves both accuracy and uncertainty (NLL) of various deterministic NN and BNN models. Smooth means spatial smoothing. Downward from left to the right (↘) means better accuracy and uncertainty.

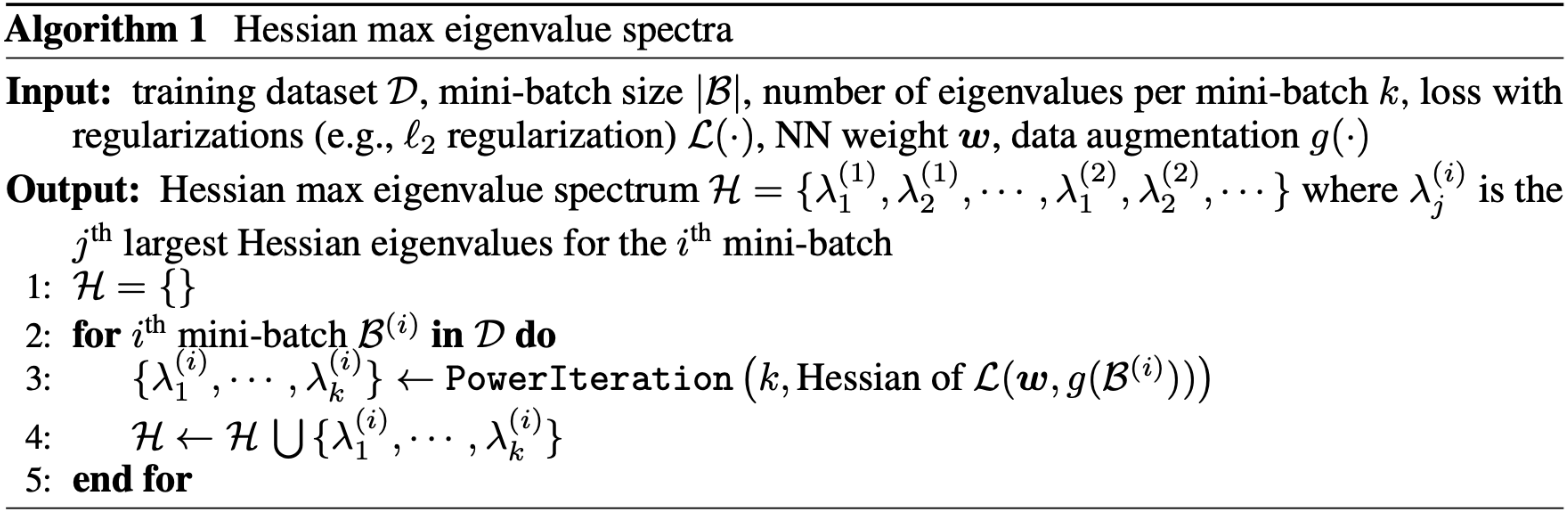
Semantic Segmentation



Spat	Temp	NLL	Acc (%)	ECE (%)	Cons (%)
·	·	0.298 (-0.000)	92.5 (+0.0)	4.20 (-0.00)	95.4 (+0.0)
✓	·	0.284 (-0.014)	92.6 (+0.1)	3.96 (-0.24)	95.6 (+0.2)
·	✓	0.273 (-0.025)	92.6 (+0.1)	3.23 (-0.97)	96.4 (+1.0)
✓	✓	**0.260 (-0.038)**	**92.6 (+0.1)**	**2.71 (-1.49)**	**96.5 (+1.1)**

Spatial smoothing and temporal smoothing are complementary. We provide predictive performance of MC dropout in semantic segmentation on CamVid for each method. “Spat” and “Temp” each stand for spatial smoothing and temporal smoothing. “Cons” stands for consistency. The numbers in brackets denote the performance improvements over the baseline.

Hessian Max Eigenvalue Spectra Algorithm



Algorithm 1 Hessian max eigenvalue spectra

Input: training dataset $\mathcal{D}$, mini-batch size $|\mathcal{B}|$, number of eigenvalues per mini-batch k, loss with regularizations (e.g., ℓ_2 regularization) $\mathcal{L}(\cdot)$, NN weight $\boldsymbol{w}$, data augmentation $g(\cdot)$

Output: Hessian max eigenvalue spectrum $\mathcal{H} = \{\lambda_1^{(1)}, \lambda_2^{(1)}, \cdots, \lambda_1^{(2)}, \lambda_2^{(2)}, \cdots\}$ where $\lambda_j^{(i)}$ is the j^{th} largest Hessian eigenvalues for the i^{th} mini-batch

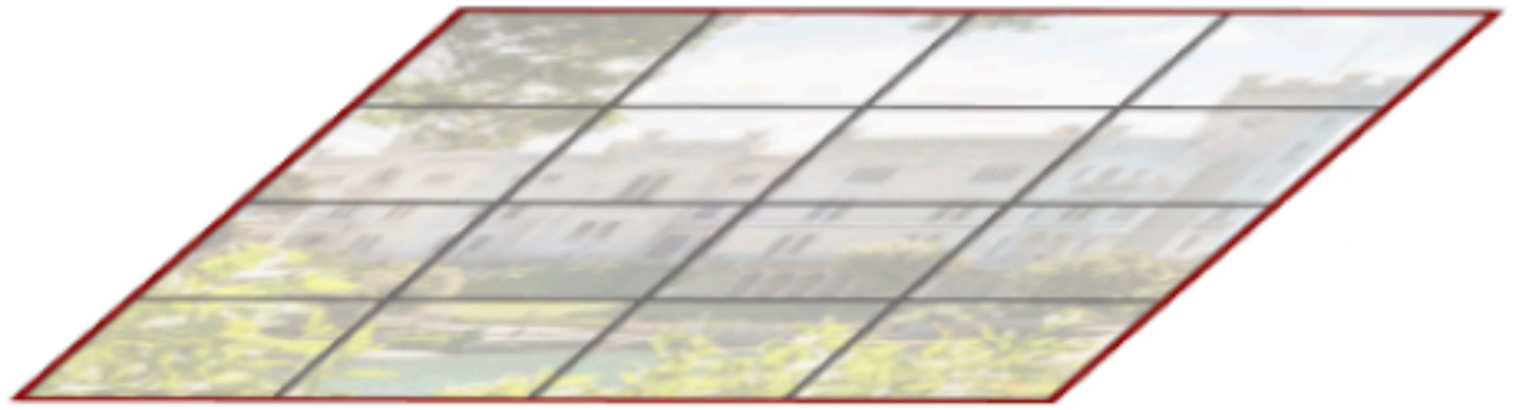
```
$\mathcal{H} = \{\}$
for $i^{\text{th}}$ mini-batch $\mathcal{B}^{(i)}$ in $\mathcal{D}$ do
    $\{\lambda_1^{(i)}, \cdots, \lambda_k^{(i)}\} \leftarrow \texttt{PowerIteration}\left(k, \text{Hessian of } \mathcal{L}(\boldsymbol{w}, g(\mathcal{B}^{(i)}))\right)$
    $\mathcal{H} \leftarrow \mathcal{H} \bigcup \{\lambda_1^{(i)}, \cdots, \lambda_k^{(i)}\}$
end for
```

Appendix Three

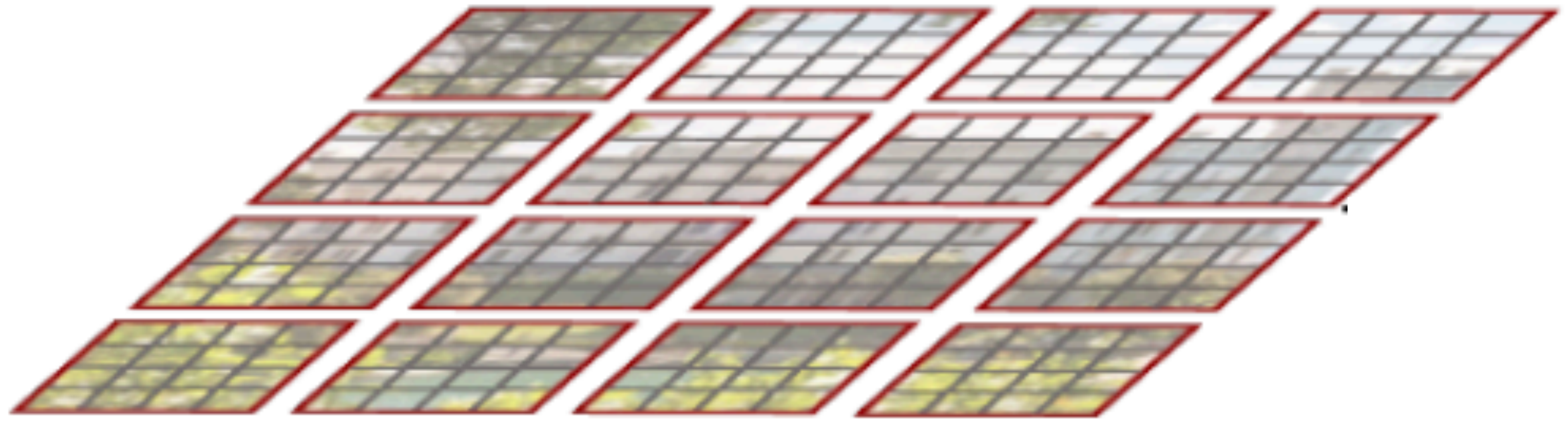
Vision Transformer

"ViT is also an ensemble!"

Locality for MSA Improves the Performance



Global MSA

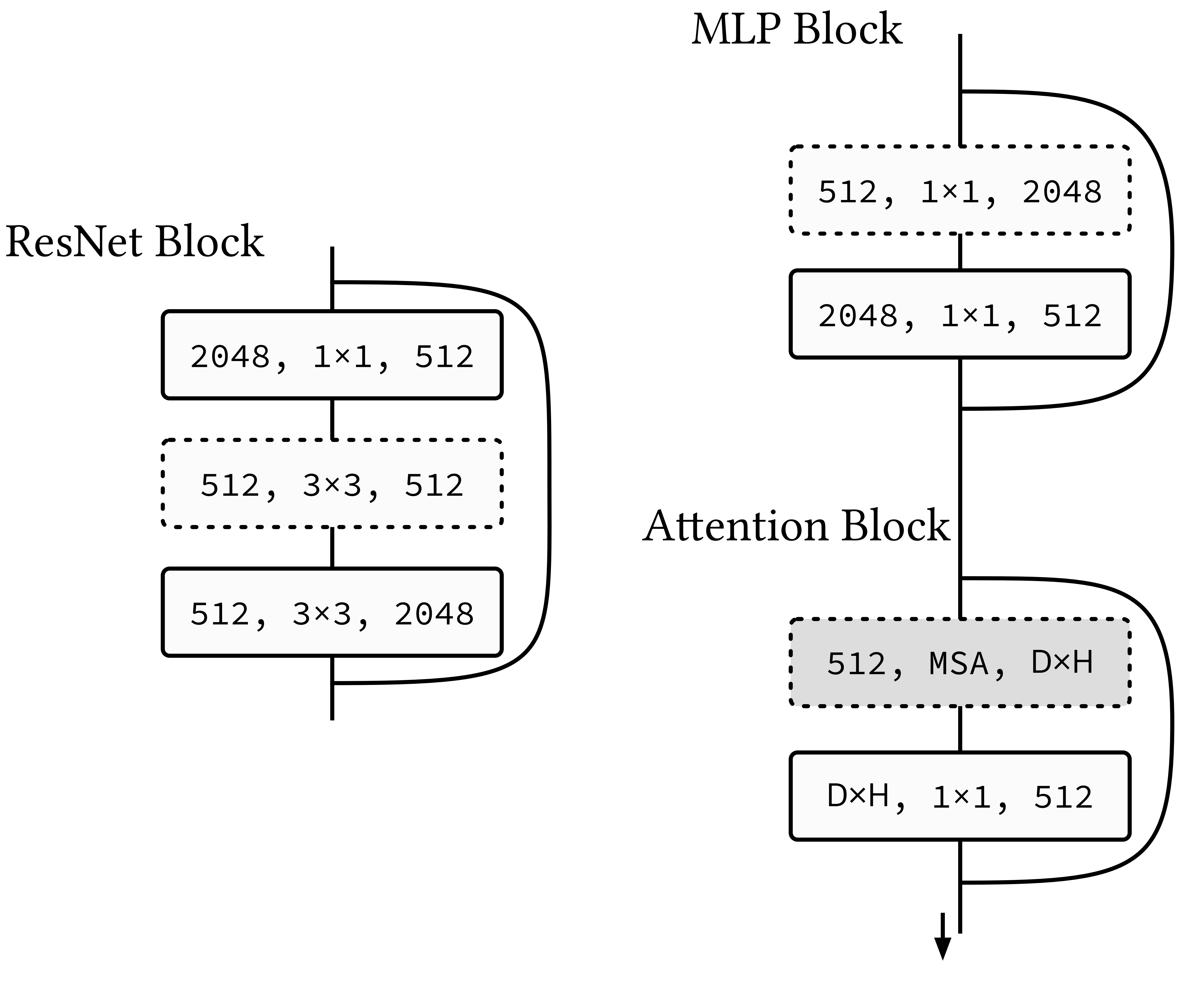


Local MSA

Local MSA restricts the scope of self-attention. Beyond our expectations, this constraint improves the computational & predictive performance.

Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." arXiv preprint arXiv:2103.14030 (2021).

Lesion Studies Remove One Unit From NNs



(a) ResNet

(b) Transformer

In lesion studies, one NN unit is removed from already trained ResNet and Swin during the testing phase. We remove one 3×3 Conv layer from the bottleneck block of ResNet, and one MSA or MLP block from ViTs.