Can Neural Nets Learn the Same Model Twice? Investigating Reproducibility and Double Descent from the Decision Boundary Perspective

Presented by

Maansi Reddy Jakkidi
Nasid Habib Barna

Introduction

Authors of the paper :

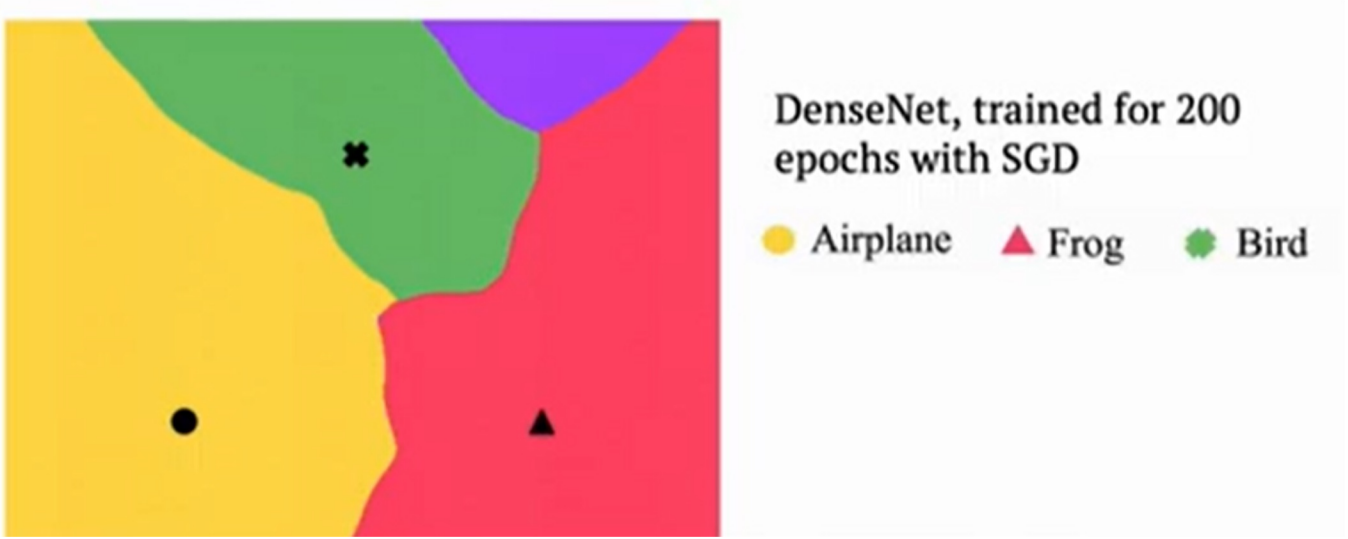
Gowthami Somepalli, Liam Fowl, Arpit Bansal, Ping Yeh-Chiang, Yehuda Dar, Richard Baraniuk, Micah Goldblum, Tom Goldstein

---

In this paper the authors discussed the methods for visualizing neural network decision boundaries and decision regions.

These visualizations were then used to investigate issues related to reproducibility and generalization in neural network training.

What is decision region?



Background

- Most of the current theories on training neural networks concentrate on understanding the geometry of loss landscapes.
- Meanwhile, considerably less is known about the geometry of class boundaries.
- The geometry of these regions depends strongly on the inductive bias of neural network models, which we do not currently have tools to rigorously analyze.
- The inductive bias of neural networks is impacted by the choice of architecture, which further complicates theoretical analysis

Questions that motivated this work

- Do neural networks learn the same model twice?
- Do different neural architectures have measurable differences in inductive bias?
- How are decision regions changing in double descent phenomenon in NNs?

Method

- In this work they used empirical tools to study
  - the geometry of class regions, and
  - how neural architecture impacts inductive bias.

- They do this using visualizations and quantitative metrics calculated using realistic models.

Plotting Decision Boundaries

The main goal while plotting the decision boundaries was to find a general-purpose visualization method that is simple, controllable, and captures important parts of decision space that lie near the data manifold.

Plotting Decision Boundaries

- On-manifold vs off-manifold behavior

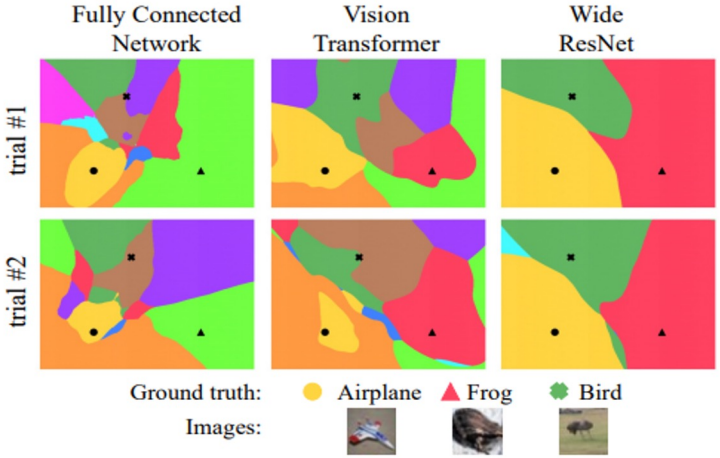


Figure 1. The class boundaries of three architectures, plotted on the plane spanning three randomly selected images. Each model is trained twice with random seeds. Decision boundaries are reproducible across runs, and there are consistent differences between the class regions created by different architectures.

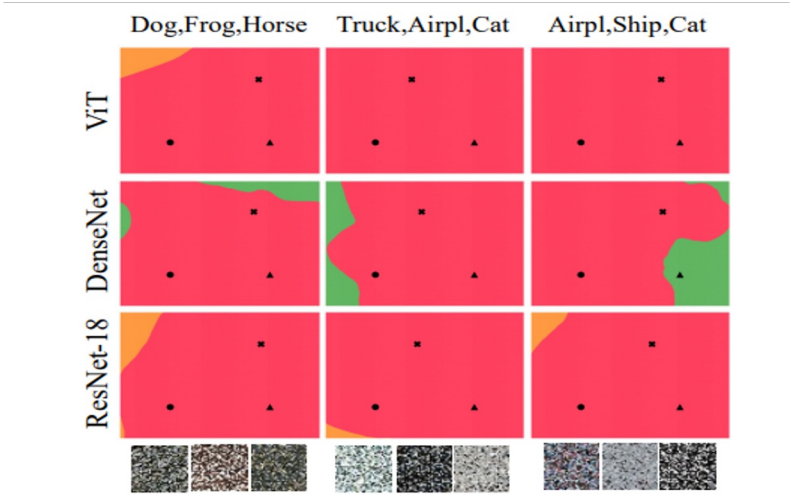


Figure 2. Off-manifold decision boundaries near "random" images created by shuffling pixels in CIFAR-10 images. Each column's title shows the labels of the unshuffled base images. Below each column we show the shuffled image triplet. Color-class mapping is as follow Red:Frog, Green:Bird, Orange:Automobile.

Plotting Decision Boundaries

- Capturing on-manifold behavior
    - The structure of image distributions is highly complex and difficult to model.
    - Observation: in addition to possessing structure near the data manifold, decision boundaries are also structured in the convex hull between pairs of data points
      (Ref: mixup: Beyond Empirical Risk Minimization)

Plotting Decision Boundaries

- Capturing on-manifold behavior
    - They plotted decision boundaries along the convex hull between data samples.
    - The inputs were sampled to the network with coordinates:

$$(x_1, x_2, x_3) \sim \mathcal{D}^3$$

$$\vec{v_1} = x_2 - x_1,\ \vec{v_2} = x_3 - x_1$$

$$\alpha \cdot \max(\vec{v_1} \cdot \vec{v_1}, |\text{proj}_{\vec{v_1}} \vec{v_2} \cdot \vec{v_1}|)\vec{v_1} + \beta(\vec{v_2} - \text{proj}_{\vec{v_1}} \vec{v_2})$$

$$-0.1 \leq \alpha, \beta \leq 1.1$$

Experimental Setup

Architectures used:

- Selected networks
  - a simple Fully Connected Network with 5 hidden layers and ReLU non-linearities
  - DenseNet-121
  - ResNet-18
  - WideResNet-28x10
  - WideResNet-28x20
  - WideResNet-28x30
  - ViT
  - MLPMixer
  - VGG-19

Experimental Setup

Architectures used:

- 100 epochs using SGD optimizer
- 3 multi-step learning rate drops
- Random Crop and Horizontal Flip data augmentations
- Selected learning rates using a grid search across { 0.001, 0.002, 0.005, 0.01, 0.02, 0.05} for each architecture and optimizer (Adam and SGD) combination, and training for 200 epochs.
- Mean test accuracy over 3 runs per model

Model Reproducibility and Inductive Bias

Networks have a strong tendency to converge on decision boundaries that generalize well.

Here they displayed the inductive bias phenomenon using decision boundary visualizations and discussed:

- Inductive bias depends on model class
- Quantitative analysis of decision regions
  - Reproducibility Score
  - Measuring architecture-dependent bias
- Does distillation preserve decision boundaries?
- The effect of the optimizer

Model Reproducibility and Inductive Bias

- Inductive bias depends on model class

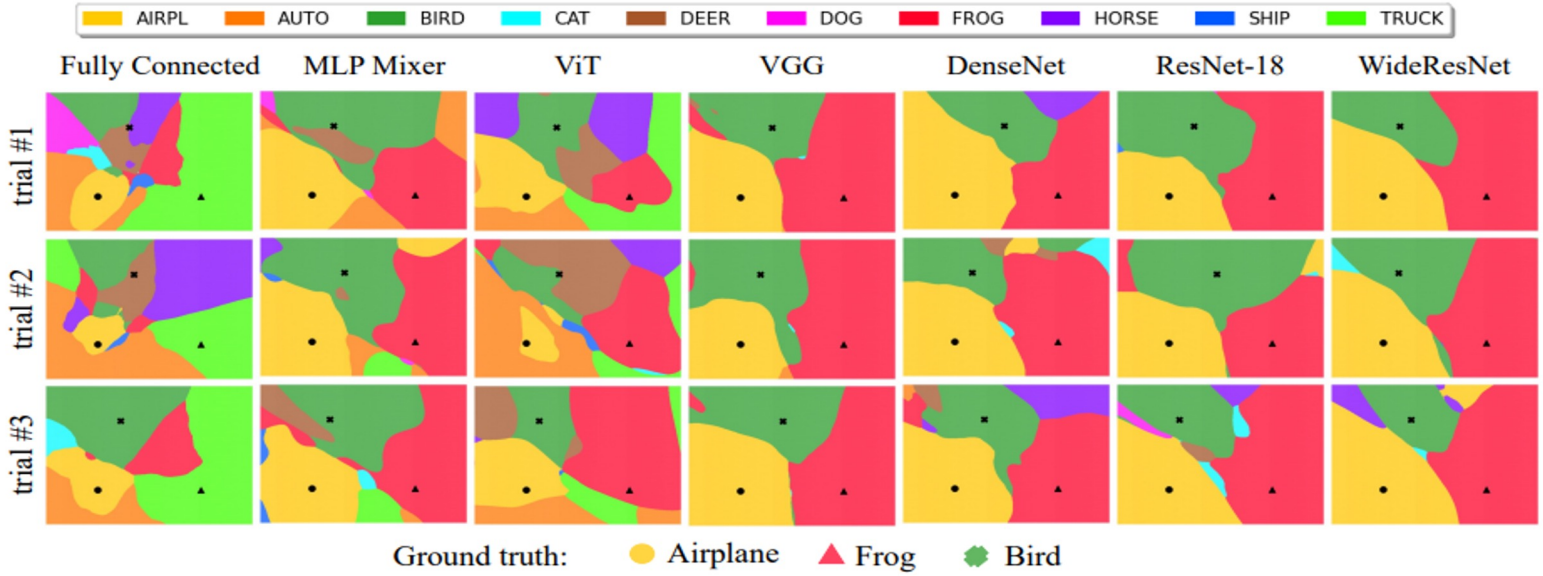


Figure 3. Decision regions through a triplet of images, for various architectures (columns) and initialization seeds (rows).

Model Reproducibility and Inductive Bias

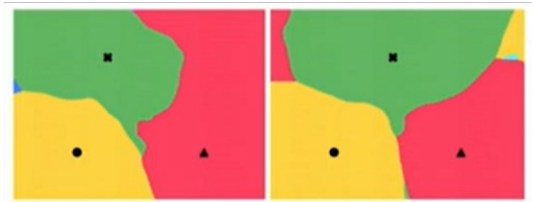
- Quantitative analysis of decision regions

Reproducibility is high within a model class, while differences in inductive bias result in low similarities across model families.

  - Reproducibility Score

$$R(\theta_1, \theta_2) = \mathbb{E}_{T_i \sim \mathcal{D}} \left[(|f(S_i, \theta_1) \cap f(S_i, \theta_2)|)/|S_i| \right]$$

T_i	Randomly chosen triplet
S_i	Decision region spanned by T_i
$f_{\theta_1}, f_{\theta_2}$	Same architecture, trained differently



ResNet-18, decision regions from 2 trials

Model Reproducibility and Inductive Bias

- Quantitative analysis of decision regions

Reproducibility is high within a model class, while differences in inductive bias result in low similarities across model families.

  - Measuring architecture-dependent bias

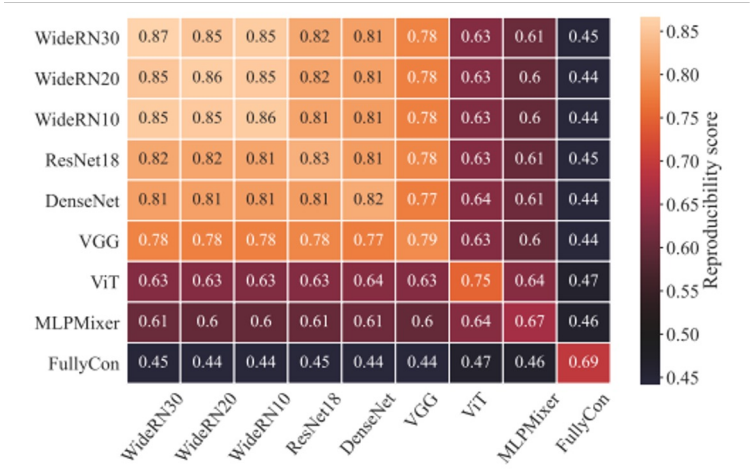


Figure 4. Reproducibility across several popular architectures.

Model Reproducibility and Inductive Bias

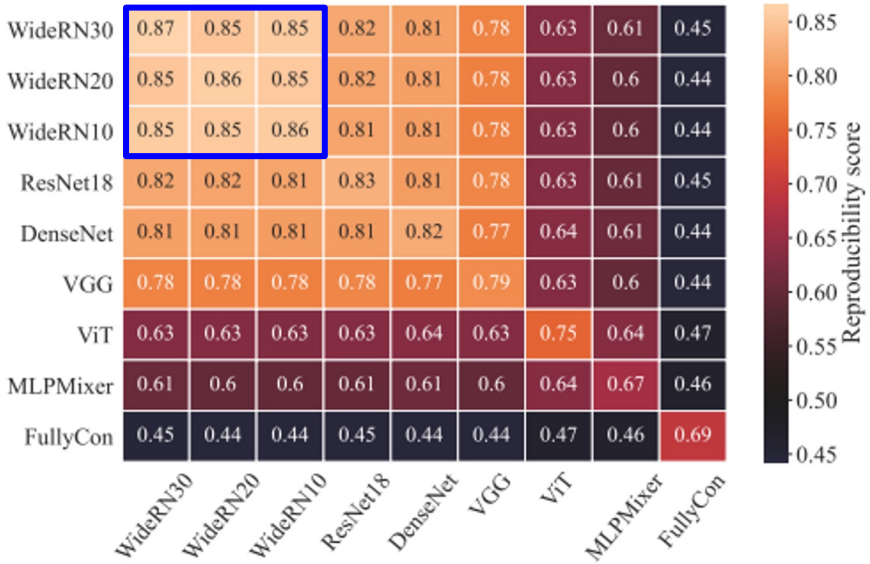


Figure 4. Reproducibility across several popular architectures.

Model Reproducibility and Inductive Bias

Figure 4. Reproducibility across several popular architectures.

Model Reproducibility and Inductive Bias

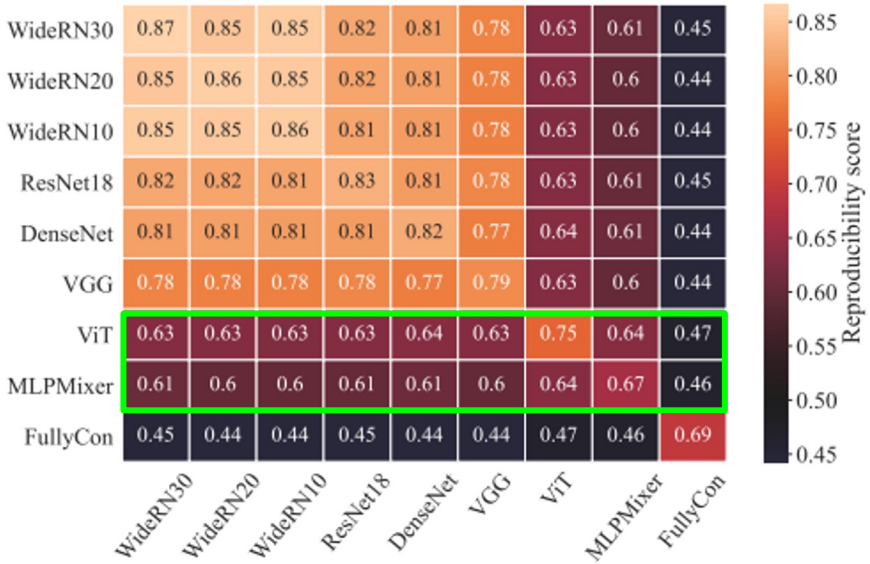


Figure 4. Reproducibility across several popular architectures.

Model Reproducibility and Inductive Bias

- Does distillation preserve decision boundaries?

Distilled students exhibit noticeably higher similarity to their teachers compared with their vanilla trained counterparts.

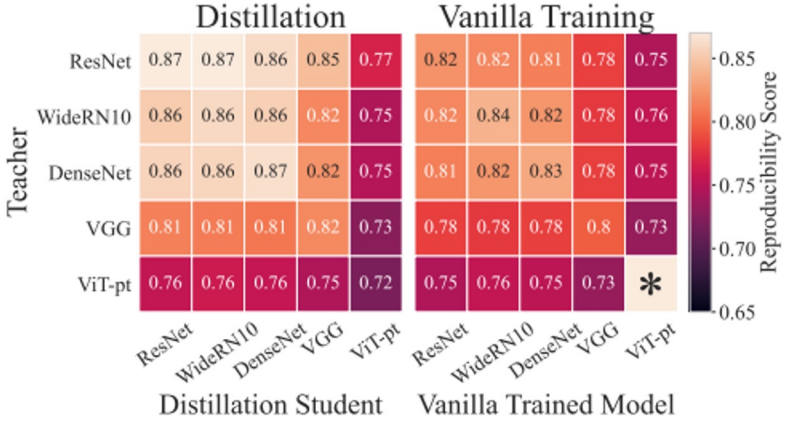


Figure 5. Differences in reproducibility comparing distilled model to vanilla trained model. *The reproducibility score is not applicable for this diagonal entry because we start from the same pre-trained model.

Model Reproducibility and Inductive Bias

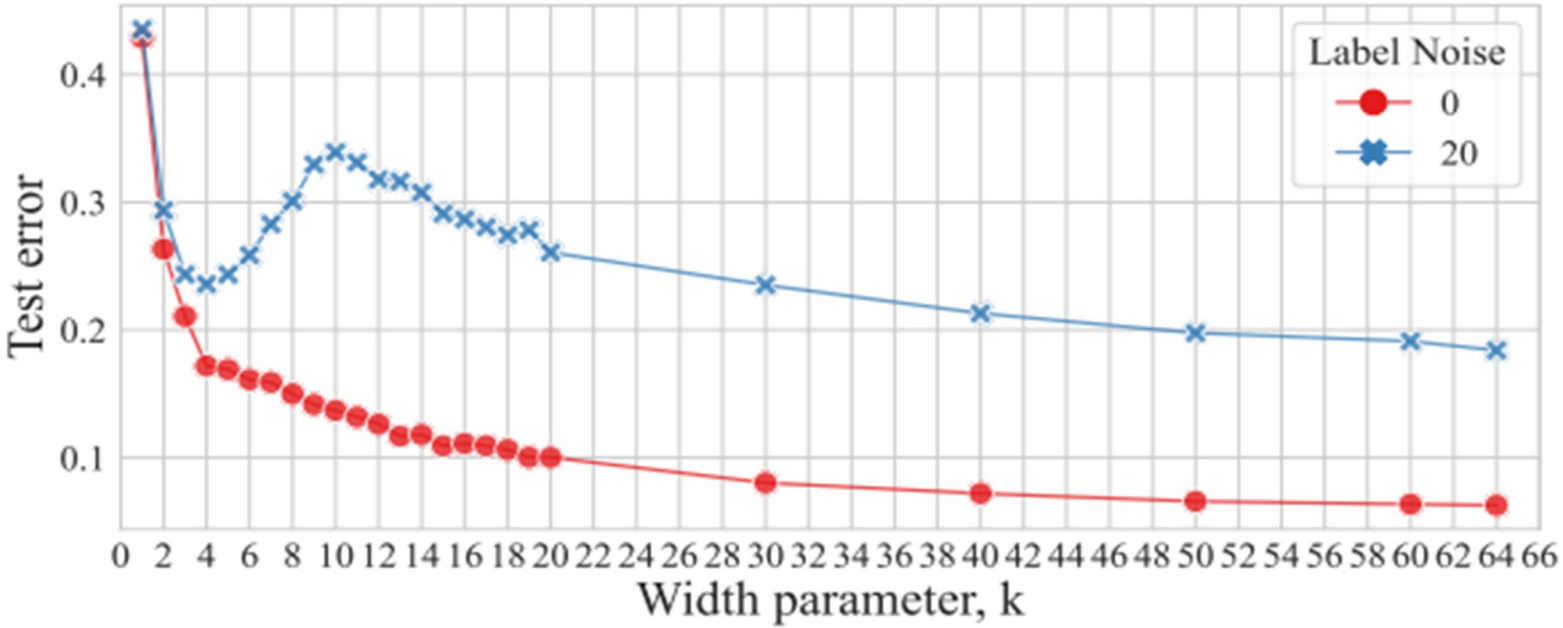
- The effect of the optimizer

Reproducibility			
	Adam	SGD	SGD + SAM
ResNet-18	79.81%	83.74%	**87.22%**
VGG	81.19%	80.92%	**84.21%**
MLPMixer	67.80%	66.51%	**68.06%**
VIT	69.55%	75.13%	**75.19%**

Test Accuracy			
	Adam	SGD	SGD + SAM
ResNet-18	93.04	95.30	**95.68**
VGG	92.87	93.13	**93.90**
MLPMixer	**82.22**	82.04	82.18
VIT	70.89	**75.49**	74.72

Table 1. Reproducibility of different models when using different optimizers. SGD produces more reproducible decision boundaries relative to Adam, and SGD+SAM almost always consistently increase reproducibility of the model relative to SGD.

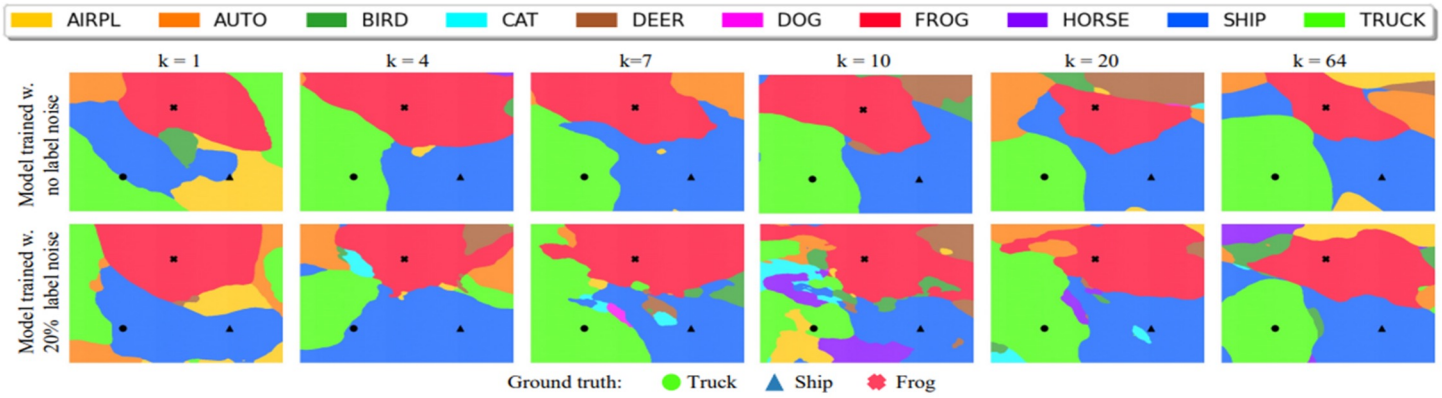
DOUBLE DESCENT



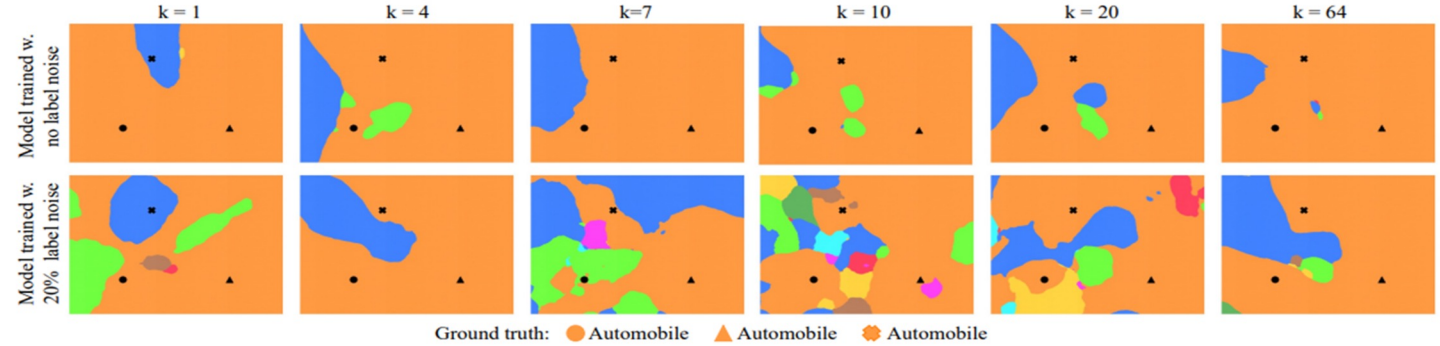
Experimental Setup

- ResNet18s by scaling the width (number of filters) of convolutional layers.
- used layer widths [k, 2k, 4k, 8k] for varying k.
- The standard ResNet18 corresponds to k = 64
- Trained with cross-entropy loss, and the optimizer Adam with learning-rate 0.0001 for 4000 epochs
- Label noise(20%)

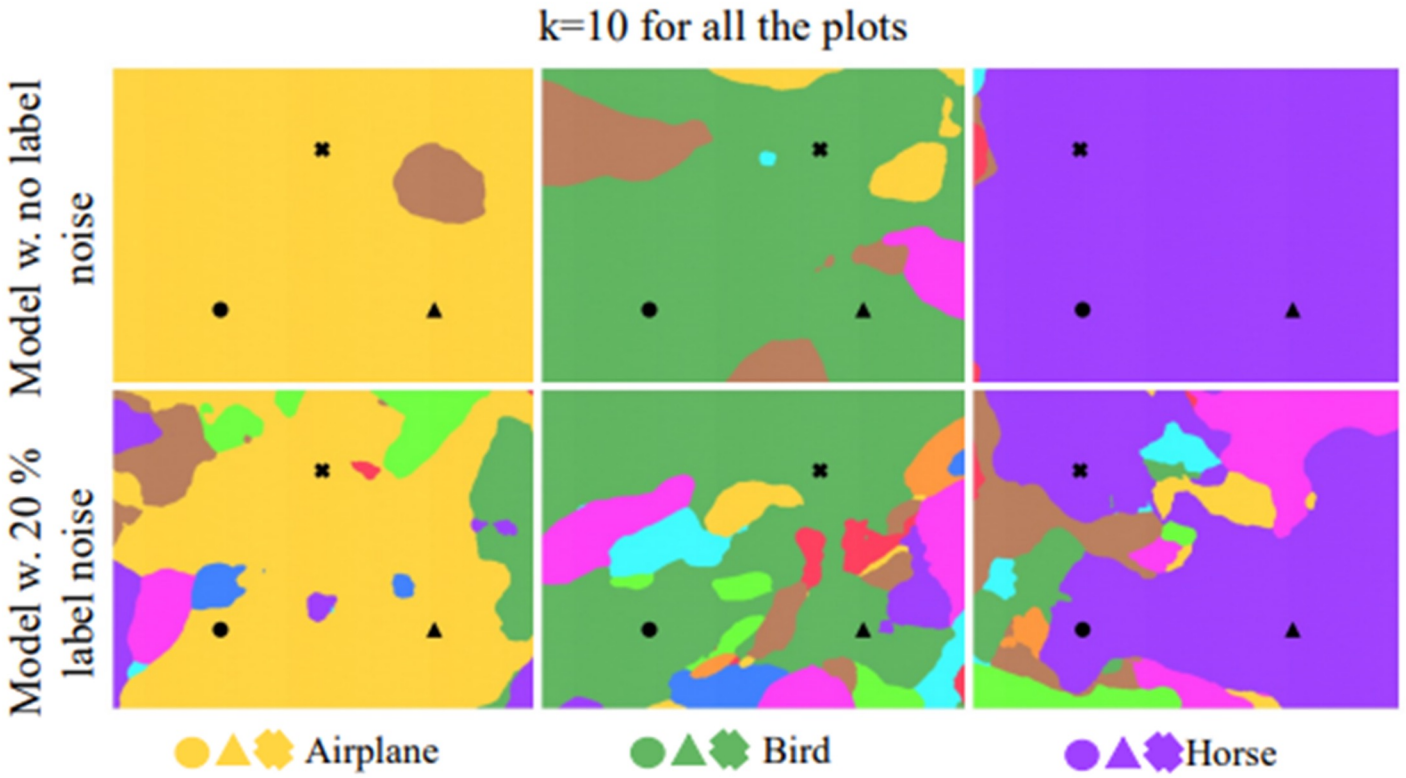
How do decision boundaries change as we cross the interpolation threshold?



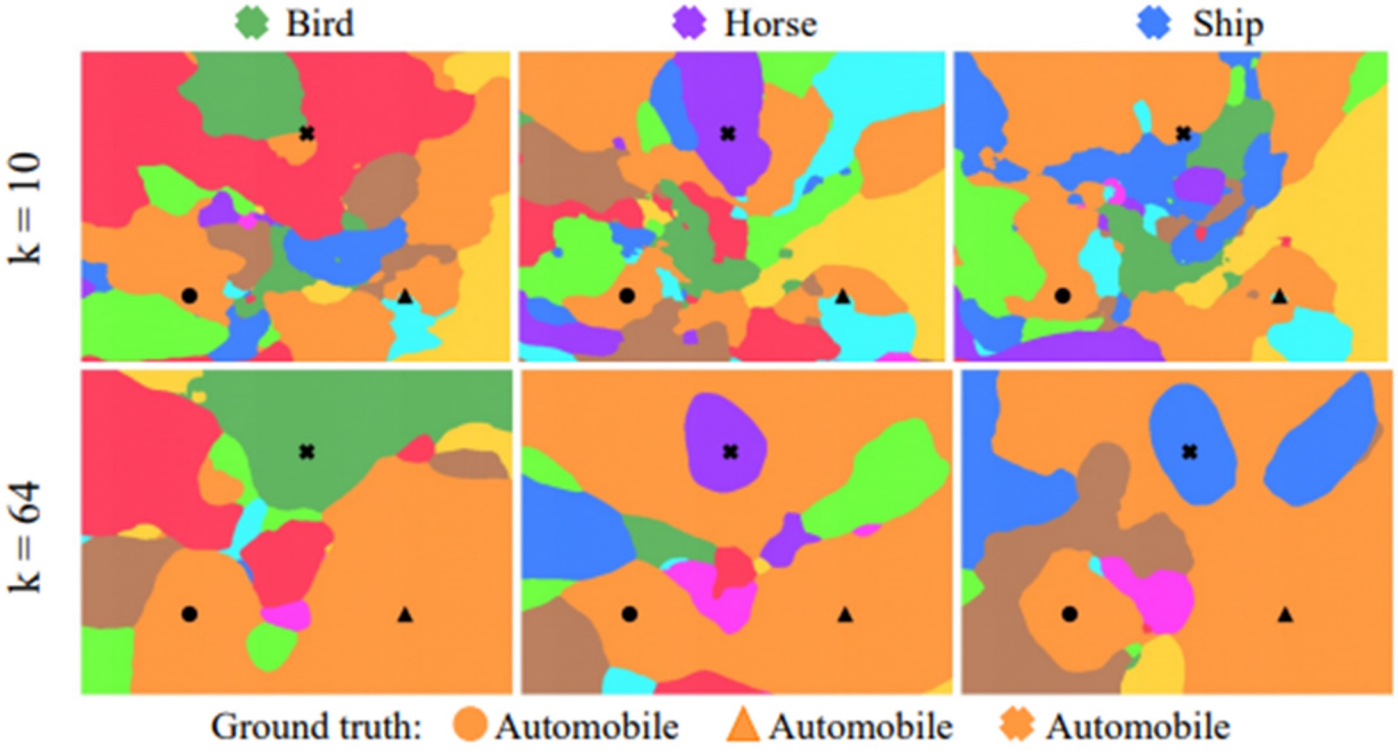
(a) All the points in the triple are from different classes, and are correctly labeled in the train set (even in the label noise case).



(b) All points in the triple are from the same class, Automobile, and are correctly labeled in the train set (even in the label noise case).


k=10 for all the plots
Model w. no label noise
Model w. 20 % label noise
Airplane
Bird
Horse

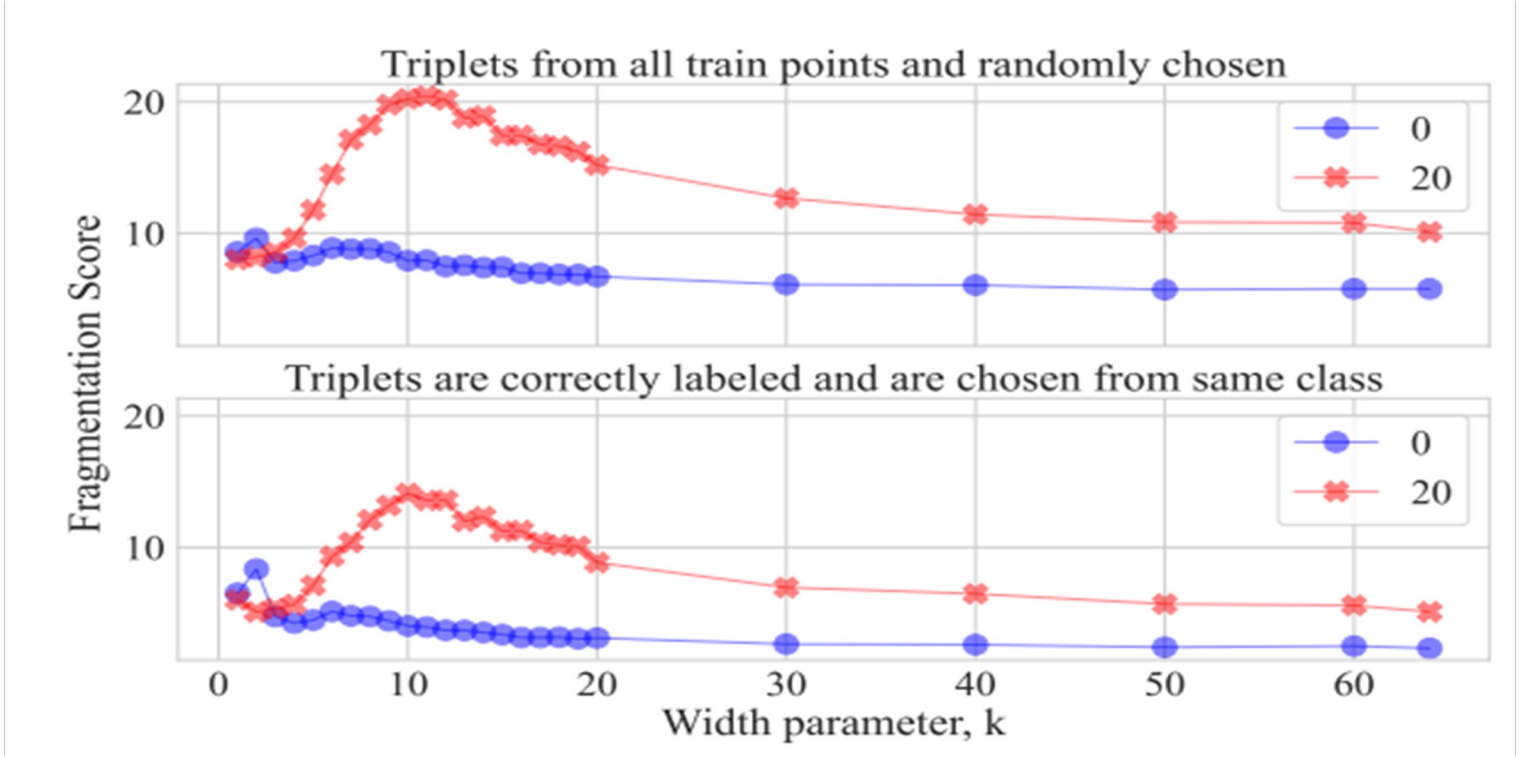
Decision boundaries around mislabeled images



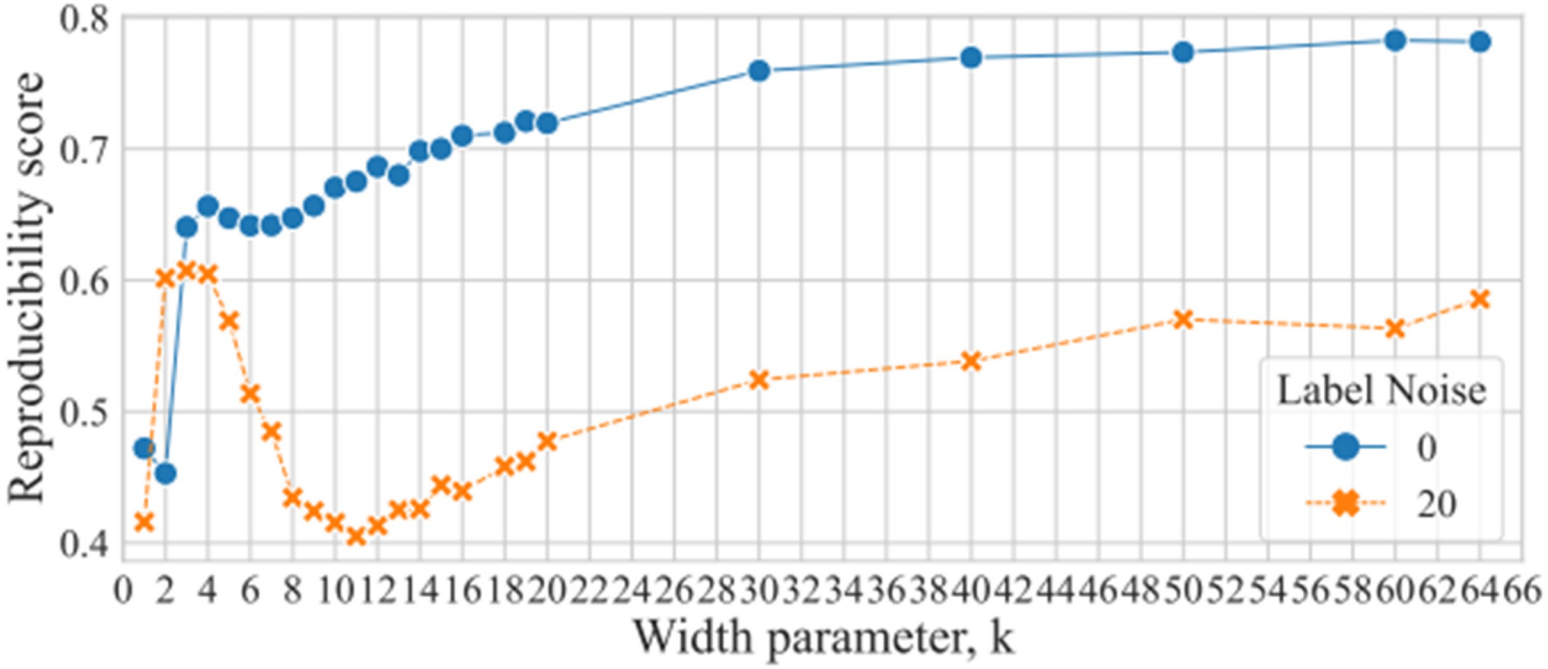
Quantifying fragmentation

- corresponding to a single predicted class label for the model with parameters θ.,
- The fragmentation score F(θ, Ti) of model θ within the decision region defined by Ti is the number of path-connected regions.
- The overall fragmentation score for a model is F(θ) = F(θ, Ti).

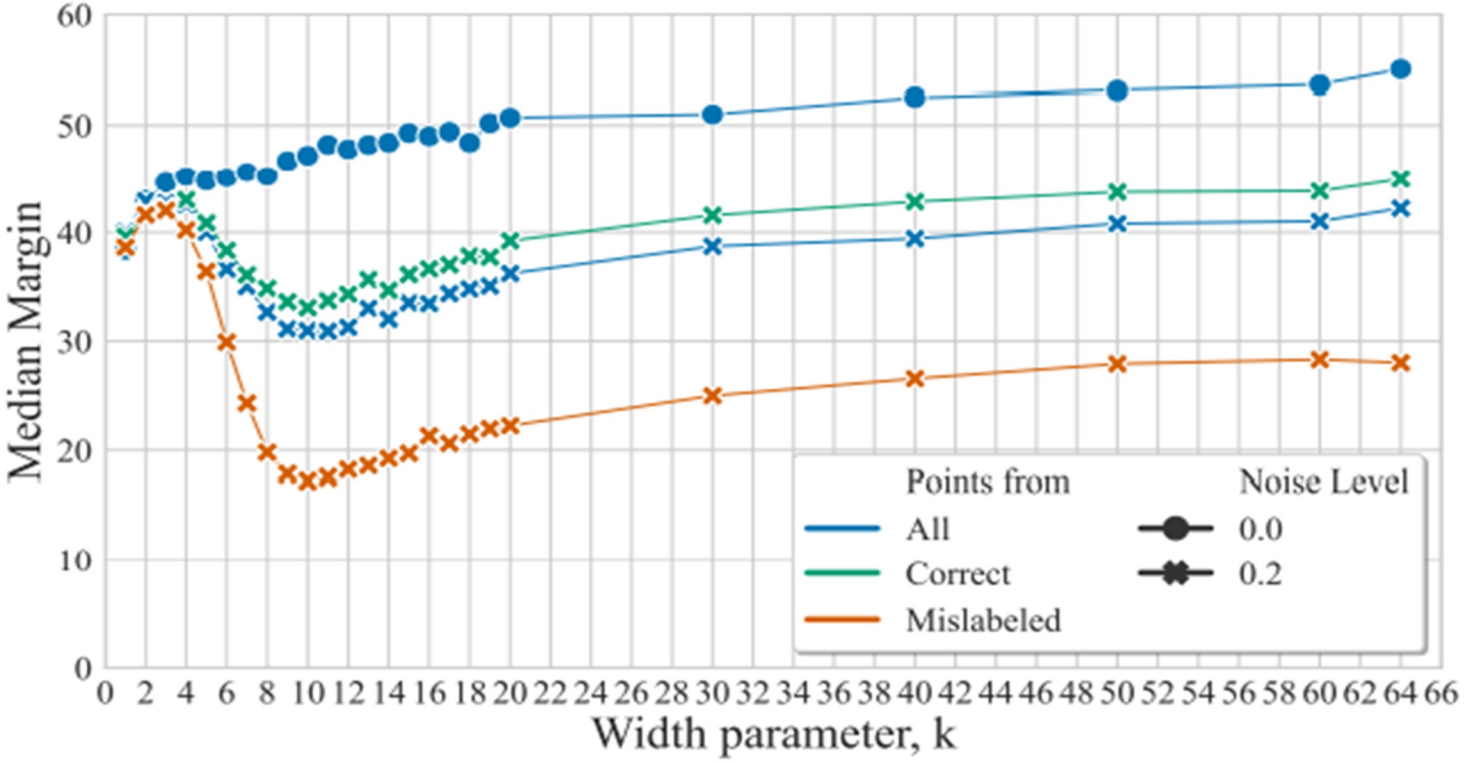
Fragmentation scores as a function of model width



Quantifying class region stability



Why does label noise amplify double descent?



Conclusion

- The authors use a decision boundary perspective to examine the relationship between model complexity, generalization error, and reproducibility.
- Overall, the paper provides insights into the behavior of neural networks at different levels of model complexity
- highlights the importance of considering the decision boundary perspective in understanding the generalization properties of neural networks.

THANK YOU