南京航空航天大學
Nanjing University of Aeronautics and Astronautics

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

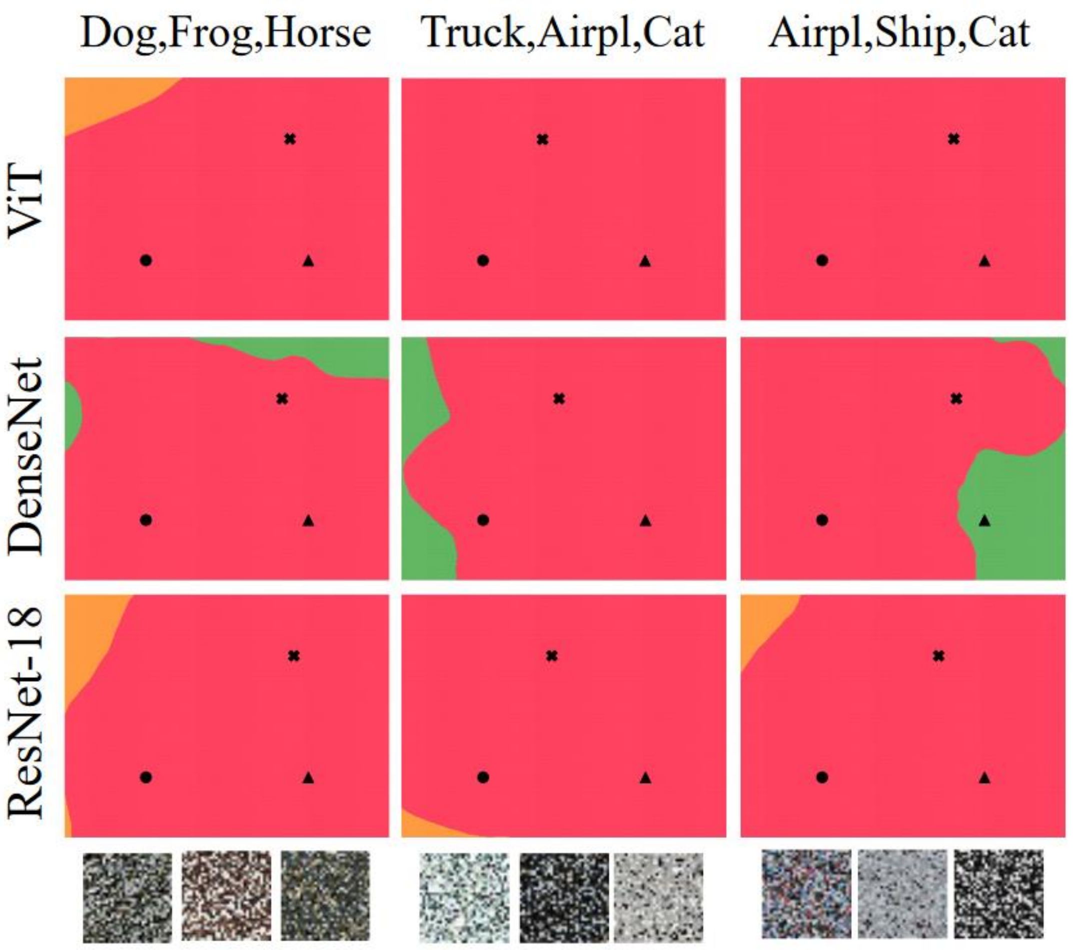
Can Neural Nets Learn the Same Model Twice? Investigating Reproducibility and Double Descent from the Decision Boundary Perspective

Gowthami Somepalli[1], Liam Fowl[1], Arpit Bansal[1], Ping Yeh-Chiang[1], Yehuda Dar[2], Richard Baraniuk[2], Micah Goldblum[3], Tom Goldstein[1]

[1]University of Maryland, College Park [2]Rice University [3]New York University

CVPR 2022

Preliminary



Lemma 2.1 Let $f : [0,1]^n \rightarrow [0,1]$ be a neural network satisfying $|f(x) - f(y)| \leq \frac{L}{\sqrt{n}} \|x - y\|$. Let $\bar{f}$ denote the median value of f on the unit hypercube. Then, for an image $x \in [0,1]^n$ of uniform random pixels, we have $|f(x) - \bar{f}| \leq t$ with probability at least

$$1 - \frac{Le^{-2\pi n t^2/L^2}}{\pi t \sqrt{n}}.$$

Decision Boundary

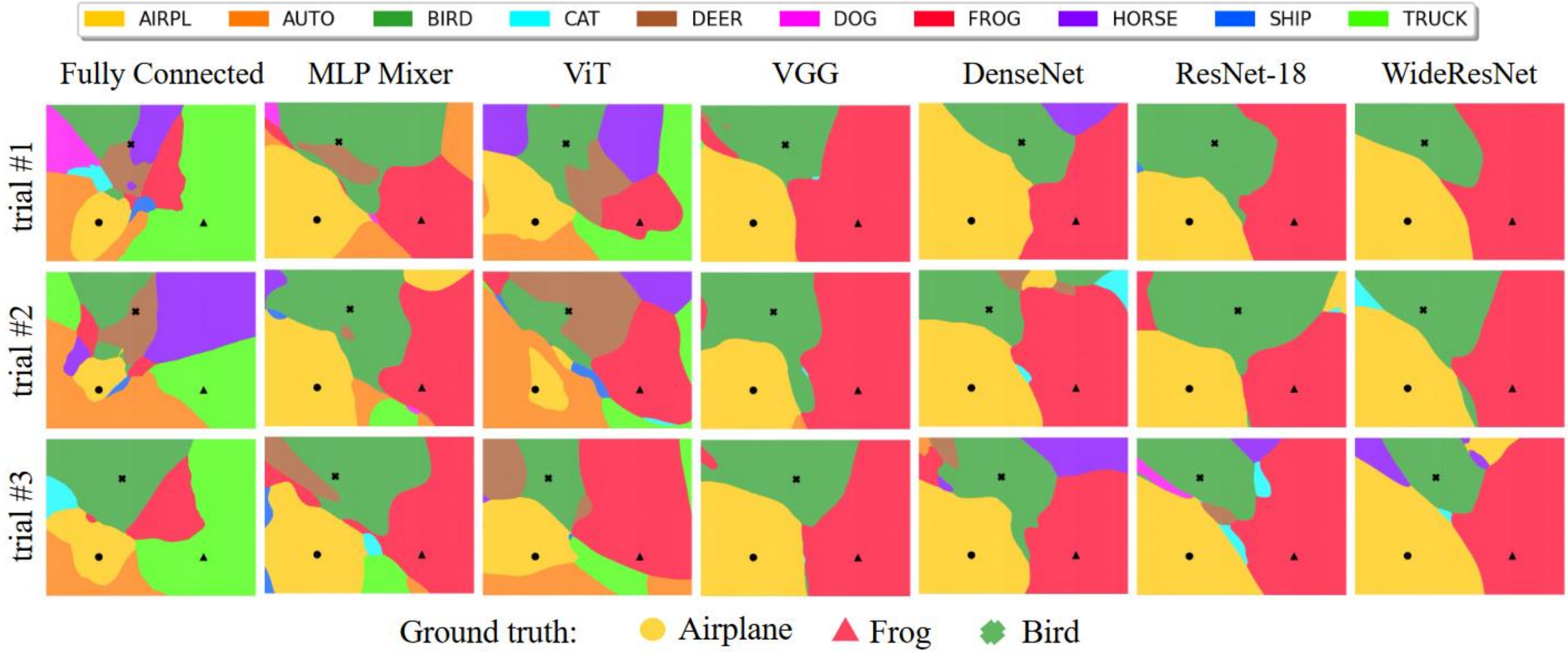
We take a page from the mixup playbook and plot decision boundaries along the convex hull between data samples. We first sample a triplet $(x_1, x_2, x_3) \sim D^3$ of $i.i.d.$ images from the distribution D. Then, we construct the plane spanned by the vectors $\overrightarrow{v_1} = x_2 - x_1$, $\overrightarrow{v_2} = x_3 - x_1$ and plot the decision boundaries in this plane. To be precise, we sample inputs to the network with coordinates

$$\alpha \cdot max(\overrightarrow{v_1} \cdot \overrightarrow{v_2}, |proj_{\overrightarrow{v_1}} \overrightarrow{v_1} \cdot \overrightarrow{v_2}|)\overrightarrow{v_1} + \beta(\overrightarrow{v_2} - proj_{\overrightarrow{v_1}} \overrightarrow{v_2})$$

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

Decision Boundary

南京航空航天大学
Nanjing University of Aeronautics and Astronautics



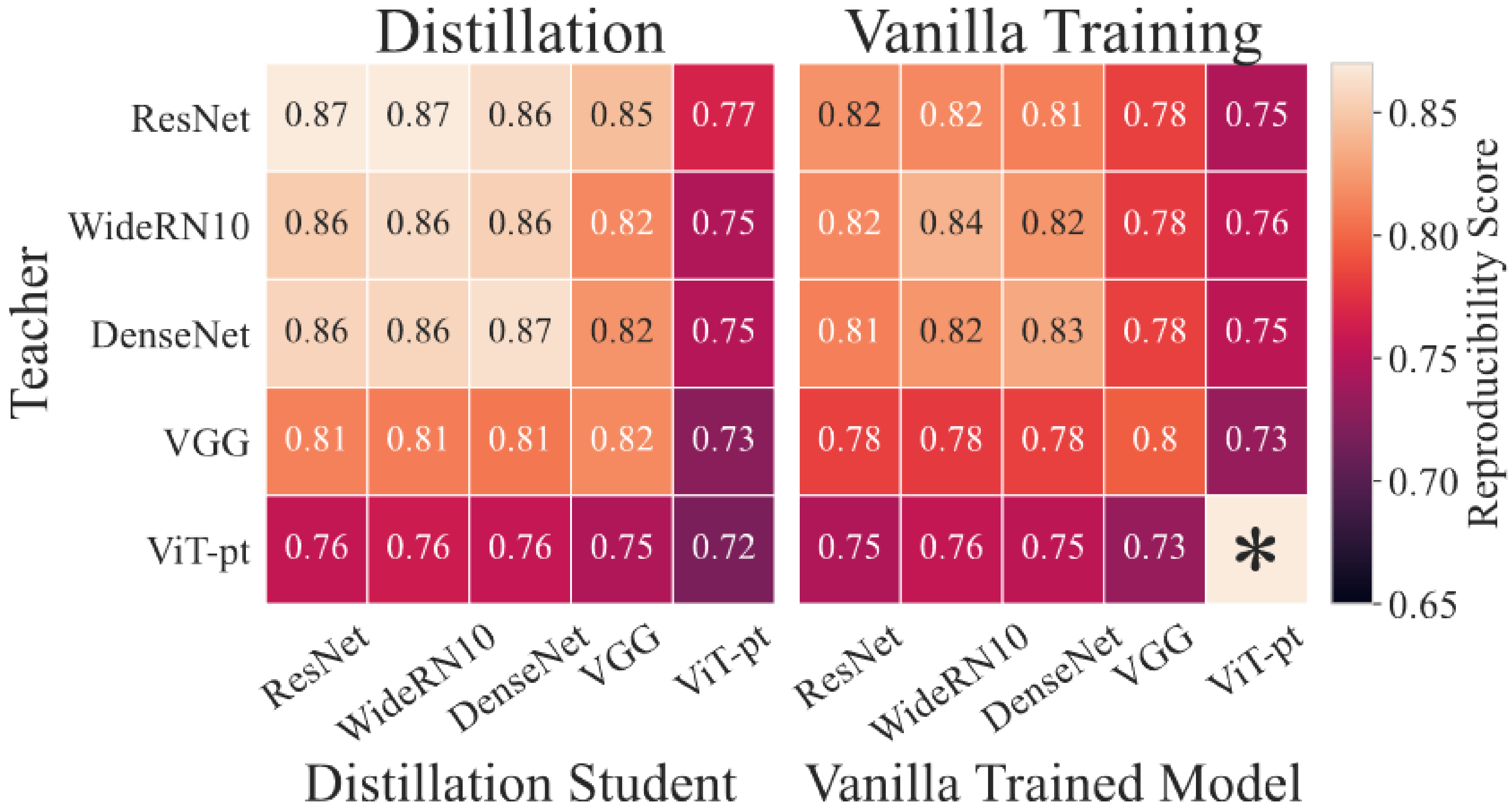
Reproducibility Score

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

	WideRN30	WideRN20	WideRN10	ResNet18	DenseNet	VGG	ViT	MLPMixer	FullyCon
WideRN30	0.87	0.85	0.85	0.82	0.81	0.78	0.63	0.61	0.45
WideRN20	0.85	0.86	0.85	0.82	0.81	0.78	0.63	0.6	0.44
WideRN10	0.85	0.85	0.86	0.81	0.81	0.78	0.63	0.6	0.44
ResNet18	0.82	0.82	0.81	0.83	0.81	0.78	0.63	0.61	0.45
DenseNet	0.81	0.81	0.81	0.81	0.82	0.77	0.64	0.61	0.44
VGG	0.78	0.78	0.78	0.78	0.77	0.79	0.63	0.6	0.44
ViT	0.63	0.63	0.63	0.63	0.64	0.63	0.75	0.64	0.47
MLPMixer	0.61	0.6	0.6	0.61	0.61	0.6	0.64	0.67	0.46
FullyCon	0.45	0.44	0.44	0.45	0.44	0.44	0.47	0.46	0.69

Reproducibility score

Reproducibility Score



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

Double Descent

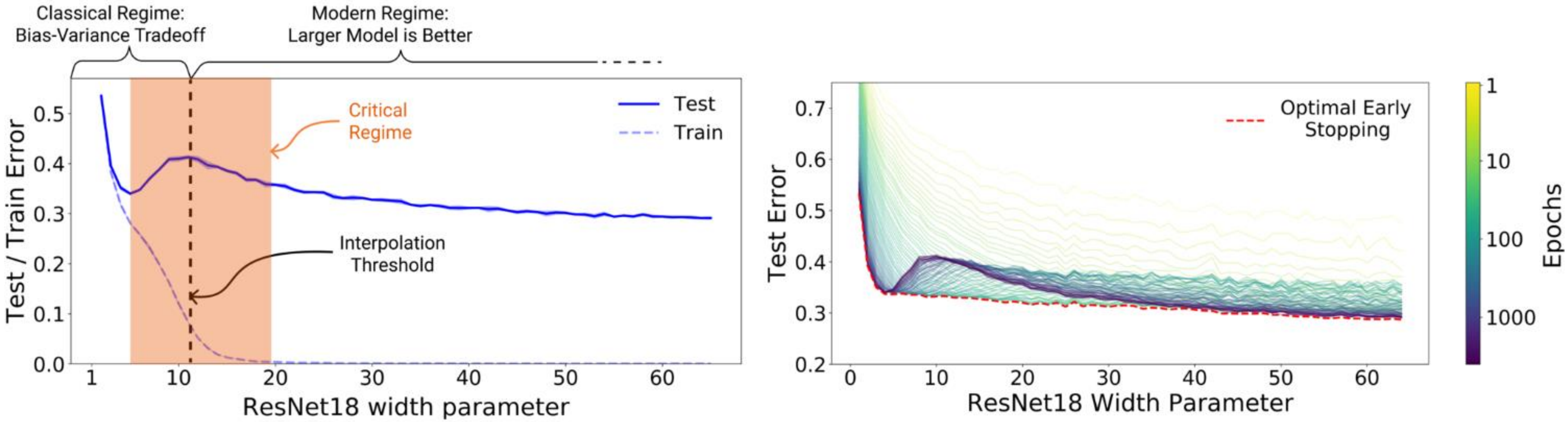
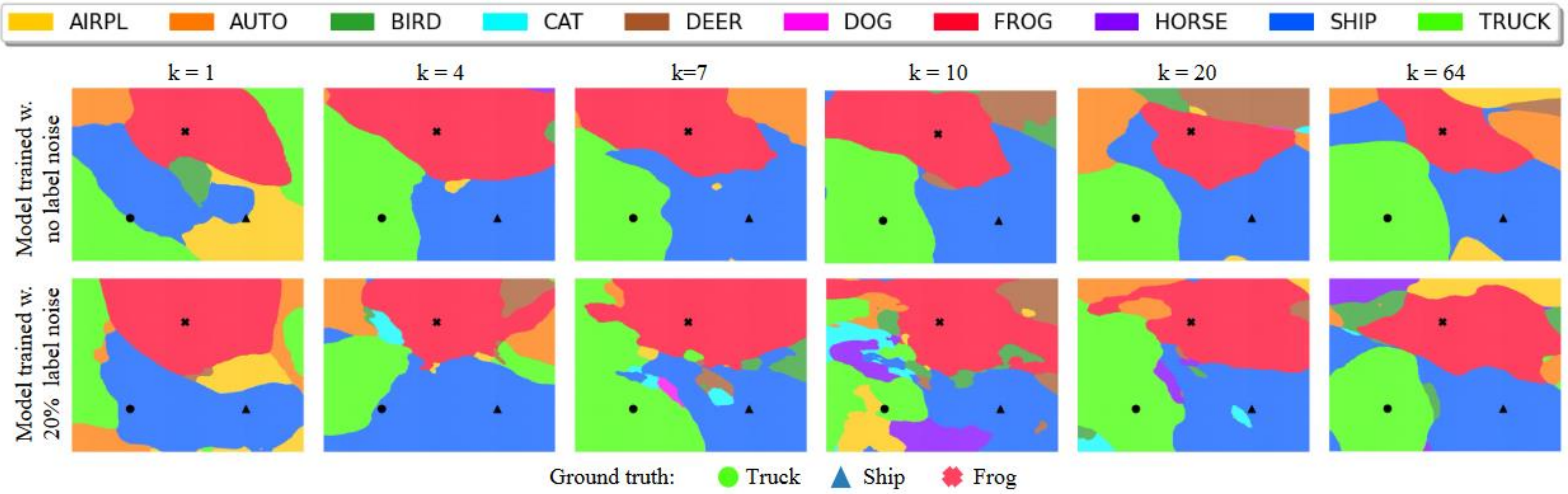
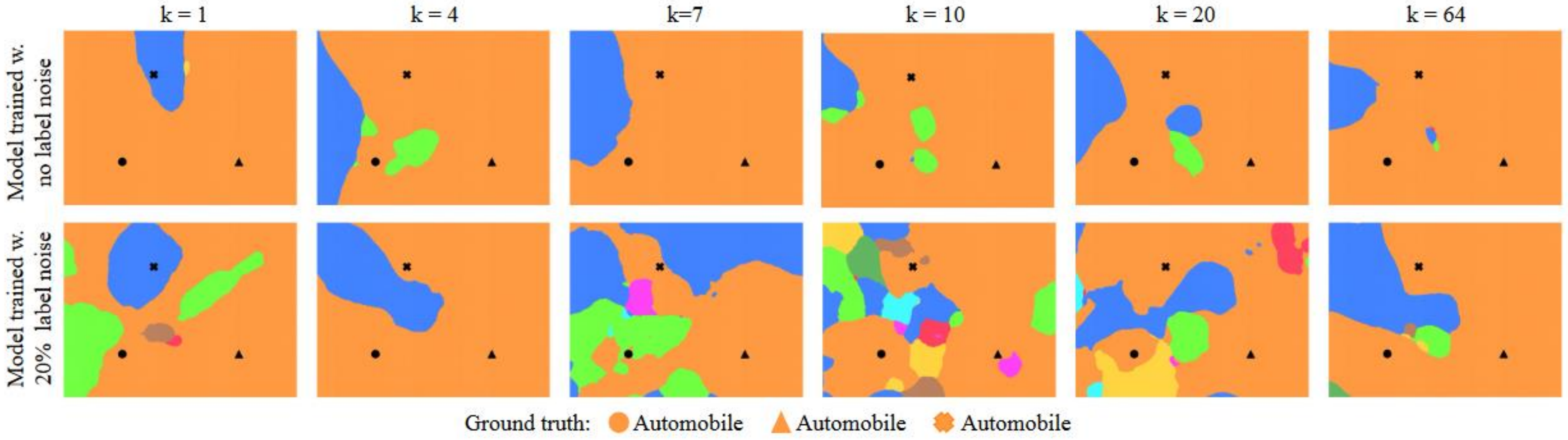


Figure 1: **Left:** Train and test error as a function of model size, for ResNet18s of varying width on CIFAR-10 with 15% label noise. **Right:** Test error, shown for varying train epochs. All models trained using Adam for 4K epochs. The largest model (width 64) corresponds to standard ResNet18.

Double Descent



(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).

Double Descent

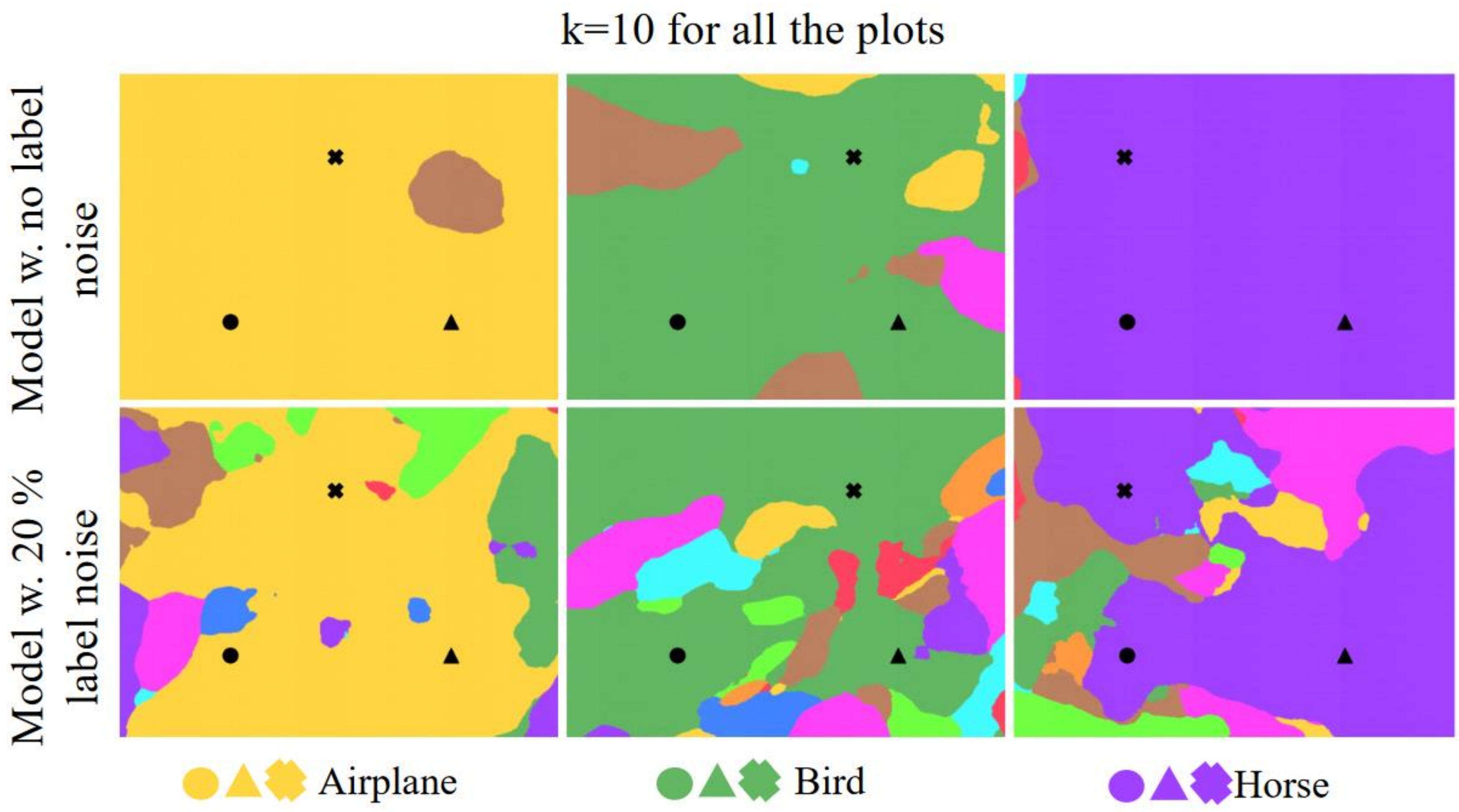


Figure 8. Decision boundaries of 3 correctly labeled points at $k = 10$ on models with and without label noise.

Double Descent

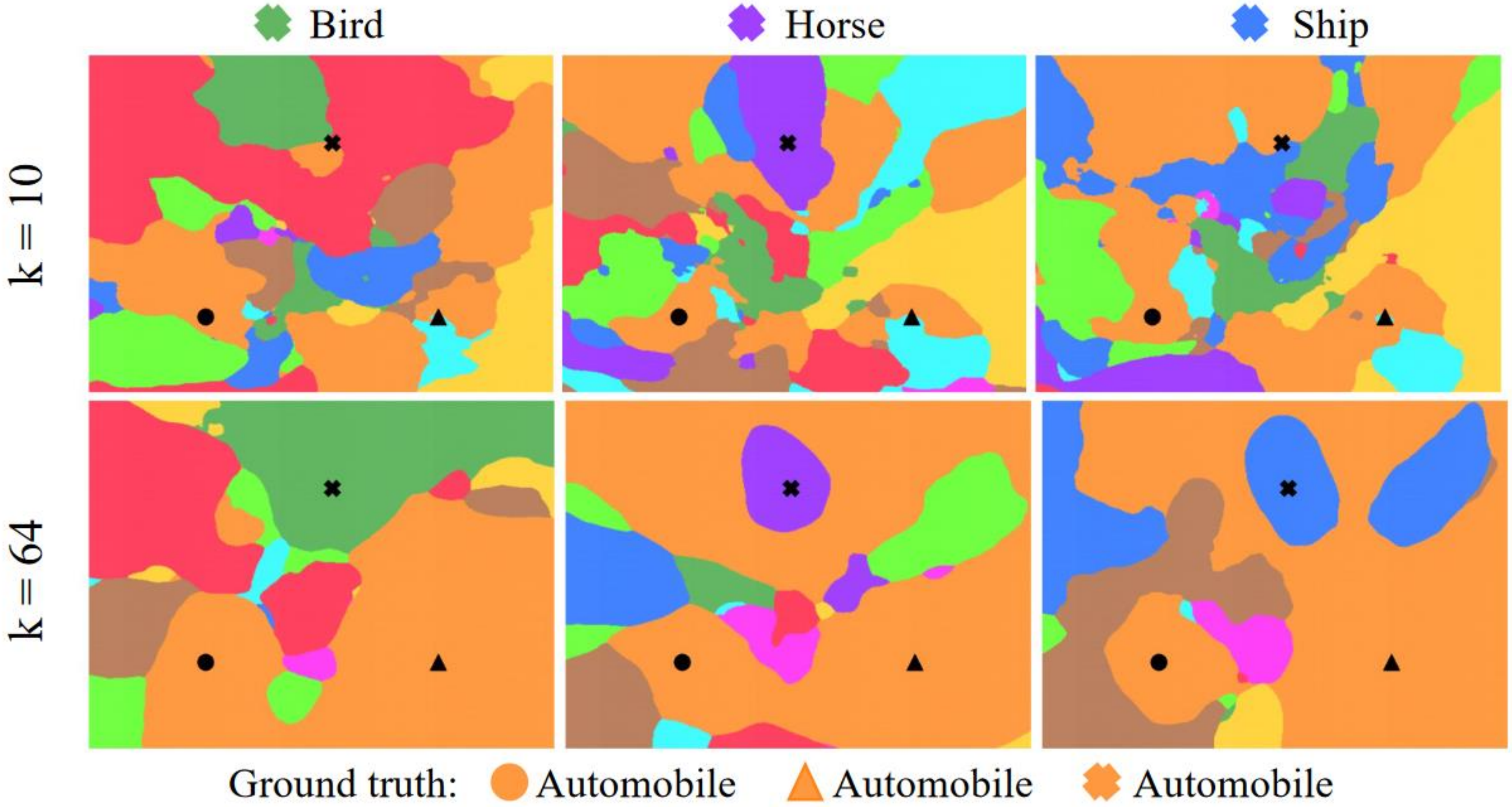


Figure 9. Decision boundaries with 1 mislabeled automobile and 2 correctly labeled automobiles. Each column represents a different image triplet. The mislabeled point is marked by **x**.

Fragmentation Score

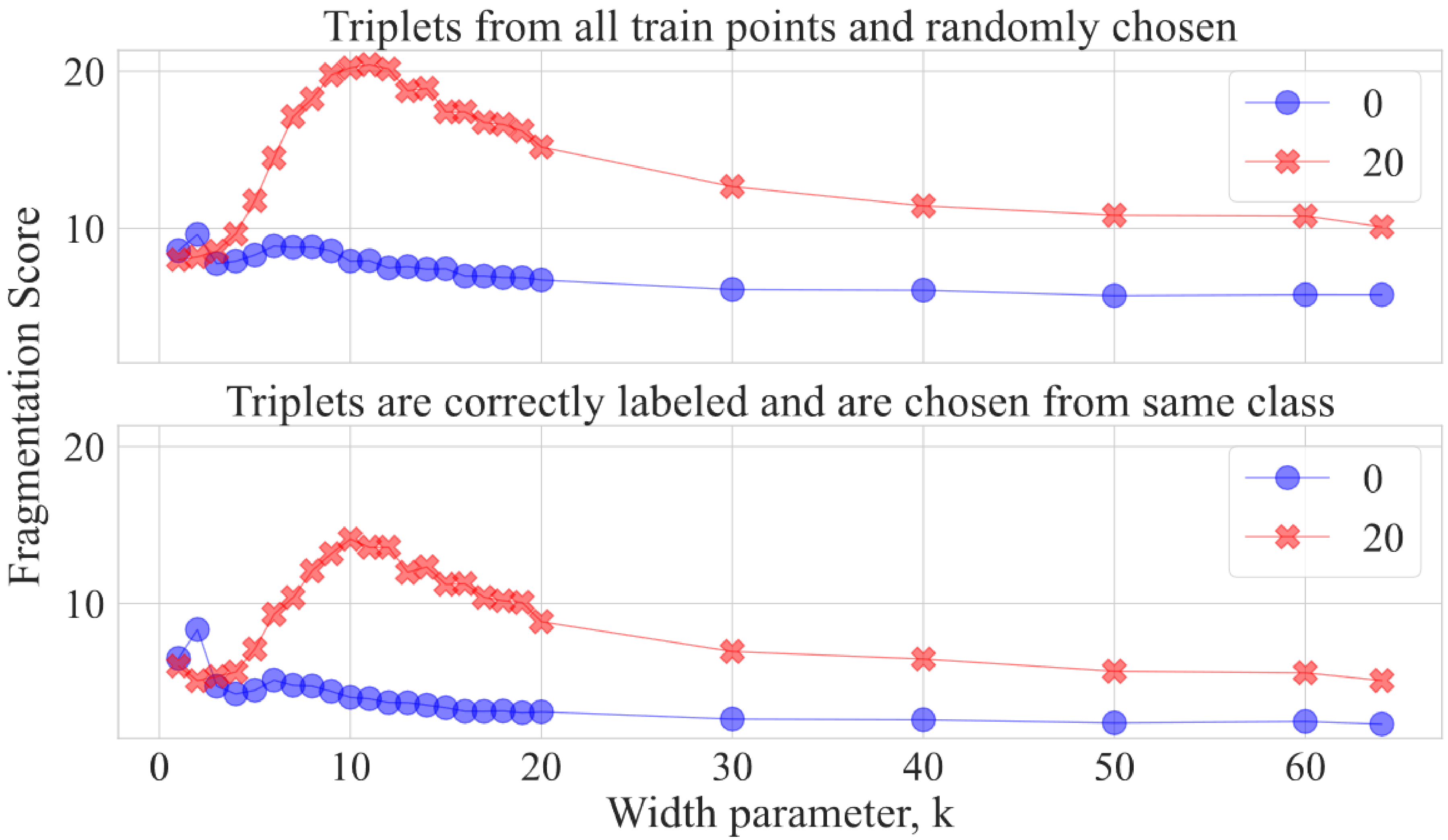


Figure 10. Fragmentation scores as a function of model width for models trained with and without label noise.

Reproducibility Score

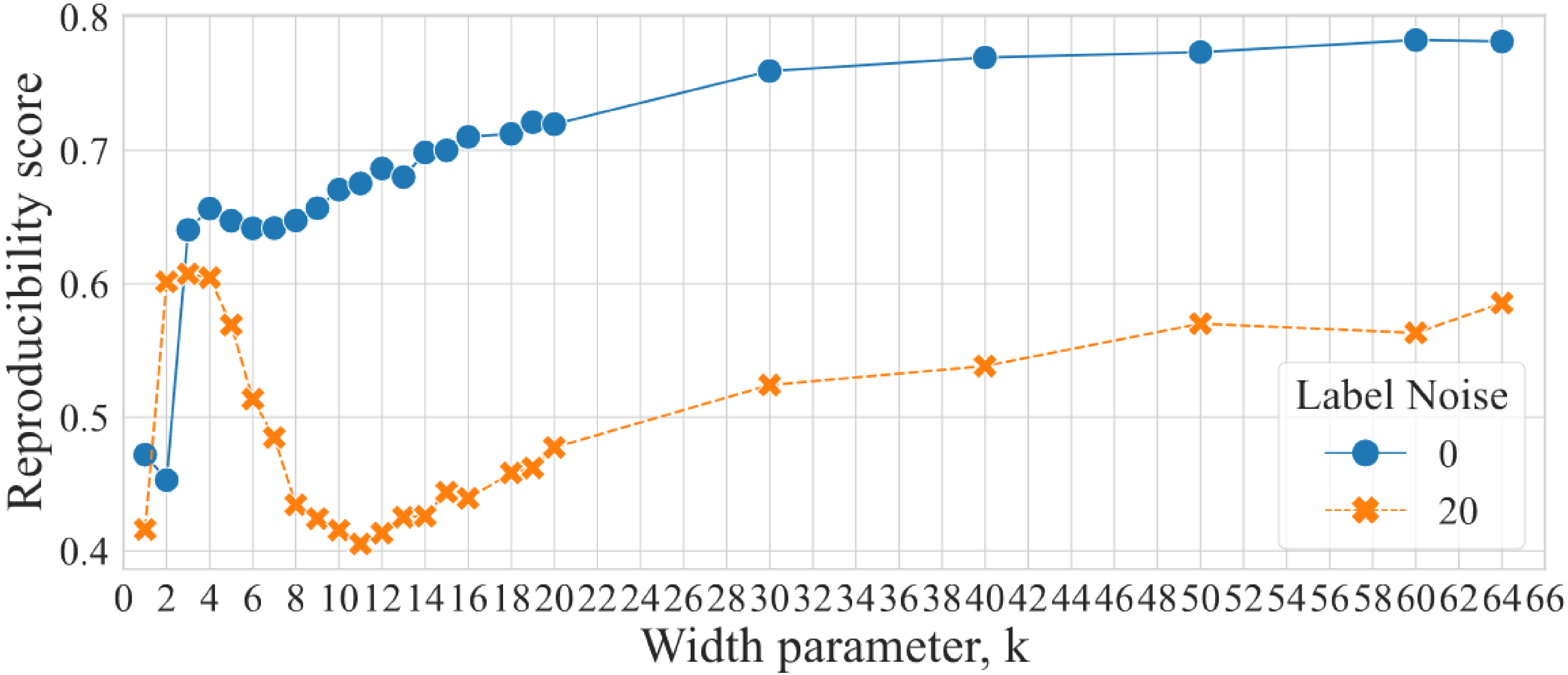


Figure 11. Reproducibility with respect to random initialization for models of different widths.

Median Margins

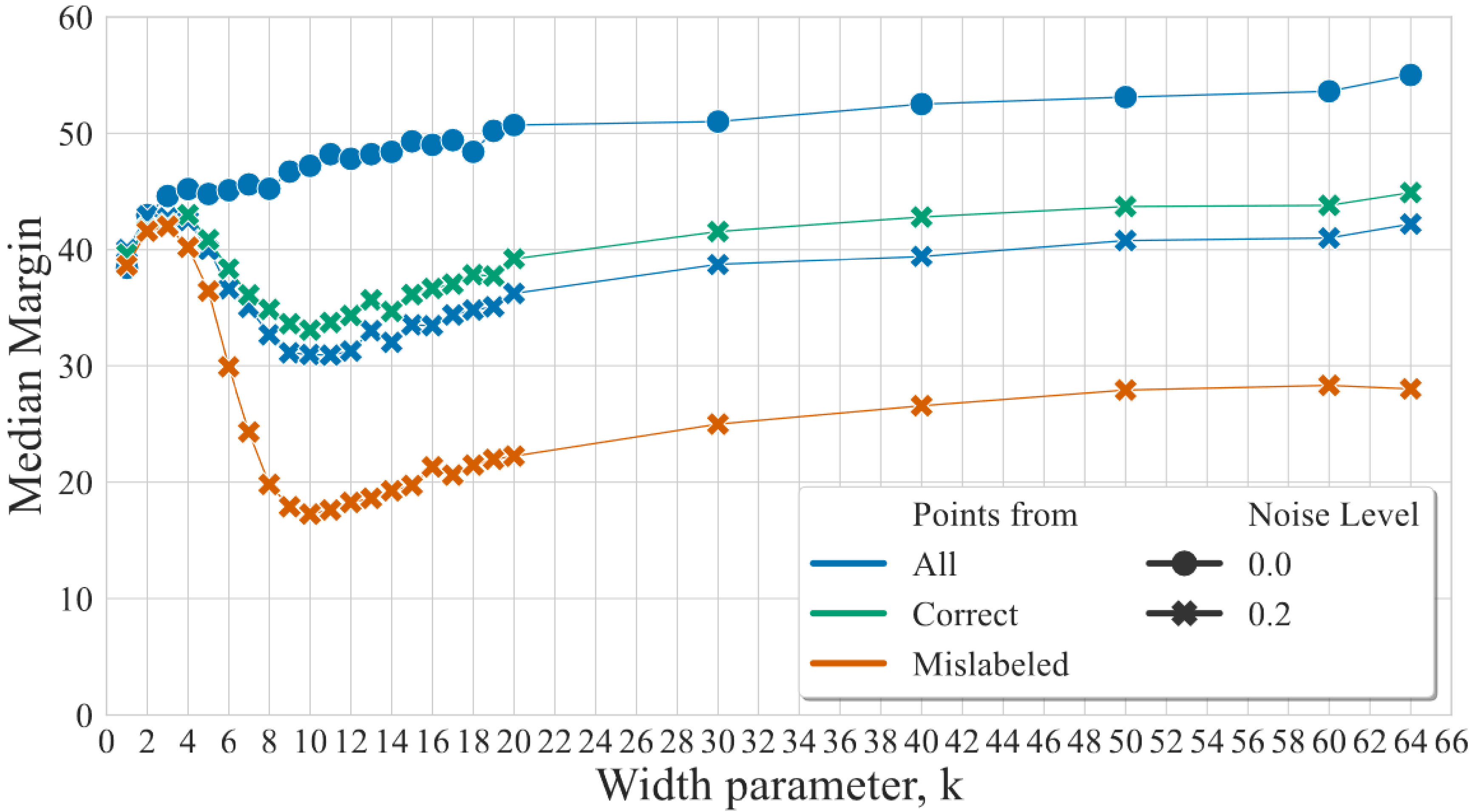


Figure 12. Median Margins - models with and without label noise. Y-axis reflects the average perturbation size needed to reach decision boundary in a random direction.