Unraveling the Complexities of Simplicity Bias: Mitigating and Amplifying Factors

Xuchen Gong*, Tianwen Fu* (equal contribution)

Abstract

Simplicity bias (SB) is an implicit bias of deep neural networks (NN) optimized by SGD: models prefer simple but noisy features to complex yet predictive ones for prediction, which is found to harm generalization.

We present three intriguing factors that mitigate/amplify simplicity bias. In summary,

- Many traditional insights such as increasing training size and increasing informative feature dimensions are **not effective;**
- Rather, balancing the modes of our data distribution, distorting the simplistic features, and searching for a good initialization are **effective.**

This work calls for future investigations on a more thorough understanding of simplicity bias and its interplay with the related fields.

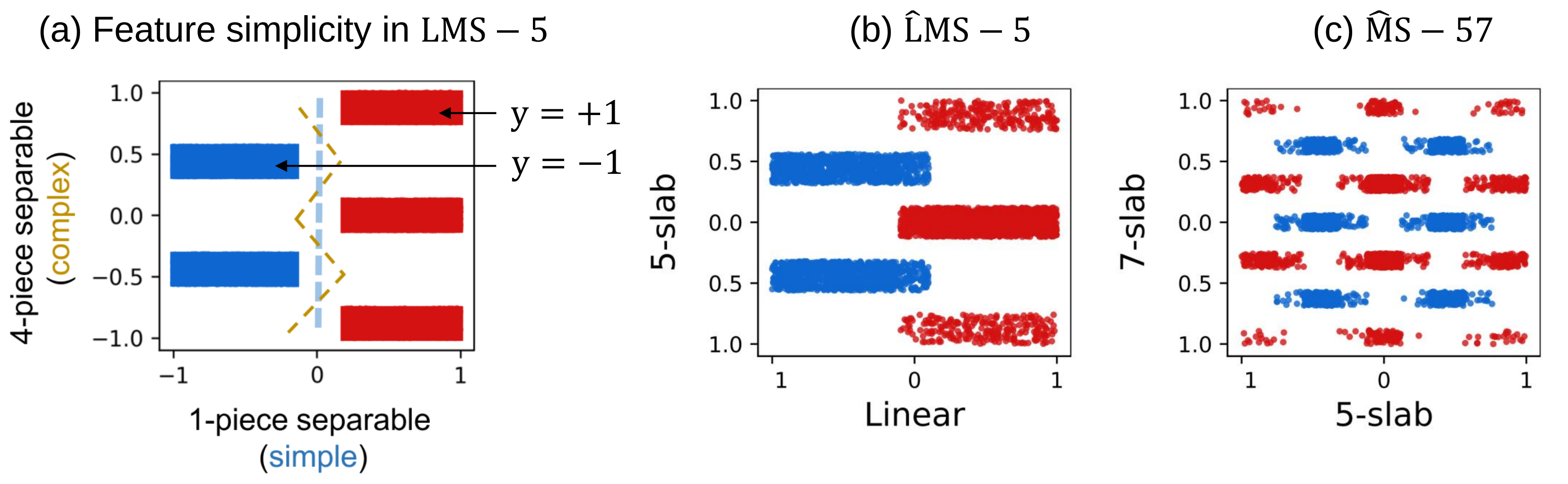
Preliminaries and Setup

Our synthetic dataset, evaluation metrics, and model all follow prior work on SB.

Dataset. X is d-dimensional feature vector, Y is binary $\in \{+1, -1\}$. Among the d features, one feature is designed to be "simple" and the remaining are engineered to be "complex" in the sense that **a complex feature requires a more sophisticated discriminatory boundary to separate the data**. Our dataset contains slab groups of data, so a boundary's sophistication is measured by its number of linear pieces.

LM-5: one feature dimension forms a 5-slab and the remaining dimensions form 2-slab
MS-57: one dimension forms a 7-slab and the remaining form 5-slab

Then we **make the simple features non-predictive** by having 10% of samples sampled from margins; we denote such a noisy dataset by \hat.



(a) Feature simplicity in $\mathrm{LMS}-5$ (b) $\widehat{\mathrm{LMS}}-5$ (c) $\widehat{\mathrm{MS}}-57$

Figure 1: Illustration of feature simplicity and two dimensions of the generated noisy dataset. In each figure, one dot represents one data sample and x- and y-axis each visualizes one feature dimension.

Metrics. Apart from training and validation accuracy, we also report

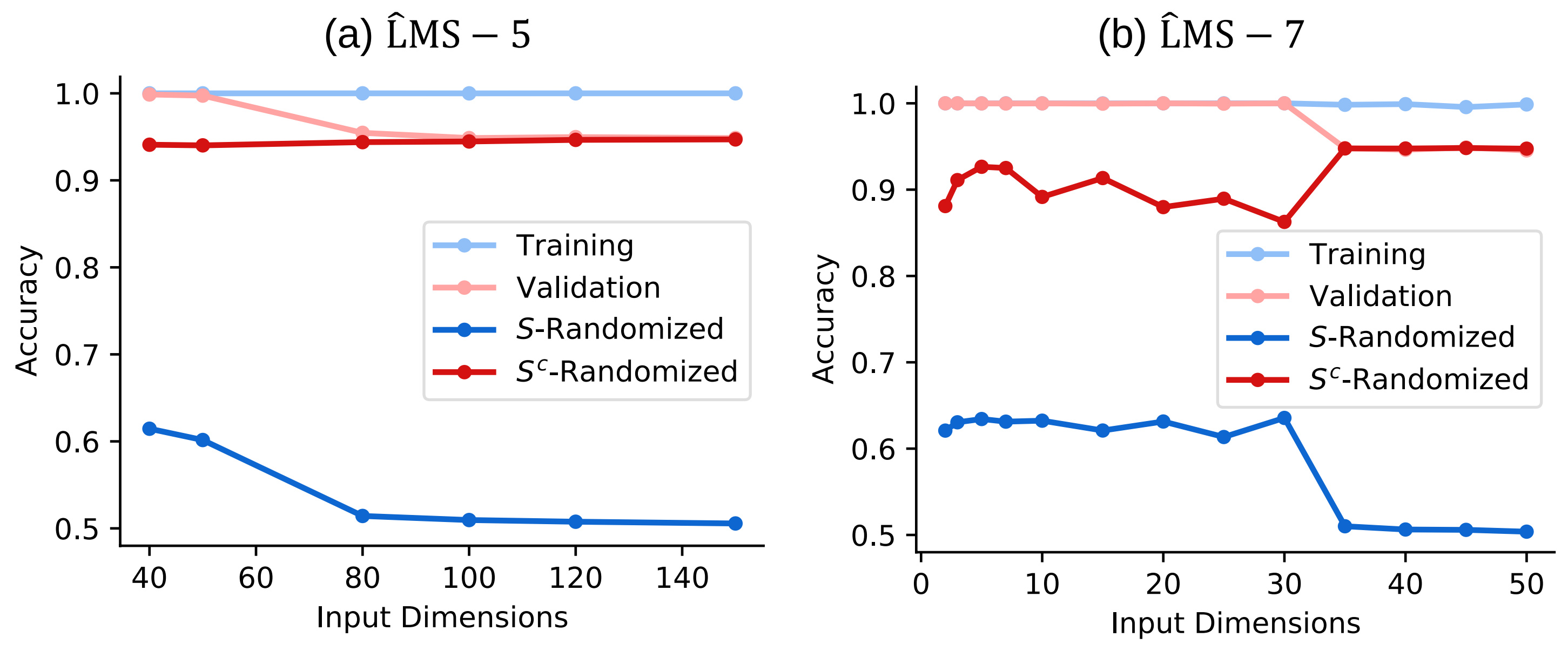
- S**-Randomized accuracy**: validation classification accuracy with the simple feature values randomly shuffled
- S^c**-Randomized accuracy**: validation classification accuracy with the complex feature values randomly shuffled

low S-Randomized accuracy ⟺ model heavily relies on simple features for prediction

Model. We use the model, a fully connected NN with ReLU activations and one 300-dimention hidden layer, as used in prior work. It is trained with SGD with LR = 0.3, weight decay = 5×10^{-4}, and no momentum.

① More predictive features are more of a hindrance

When $d = 50$, each sample has one noisy simple feature and 49 predictive complex features. Intuitively, providing more informative features (i.e., increasing d) should reduce the model's reliance on simple features and thus mitigate SB.



(a) $\widehat{\mathrm{LMS}}-5$ (b) $\widehat{\mathrm{LMS}}-7$

Figure 2: Accuracies vs. number of predictive features on $\widehat{LMS}$ datasets.

However, as input dimension d increases, models of the same complexity tend to focus more exclusively on the simple but noisy feature.

② Number of training samples matters less than balanced modes in data distribution

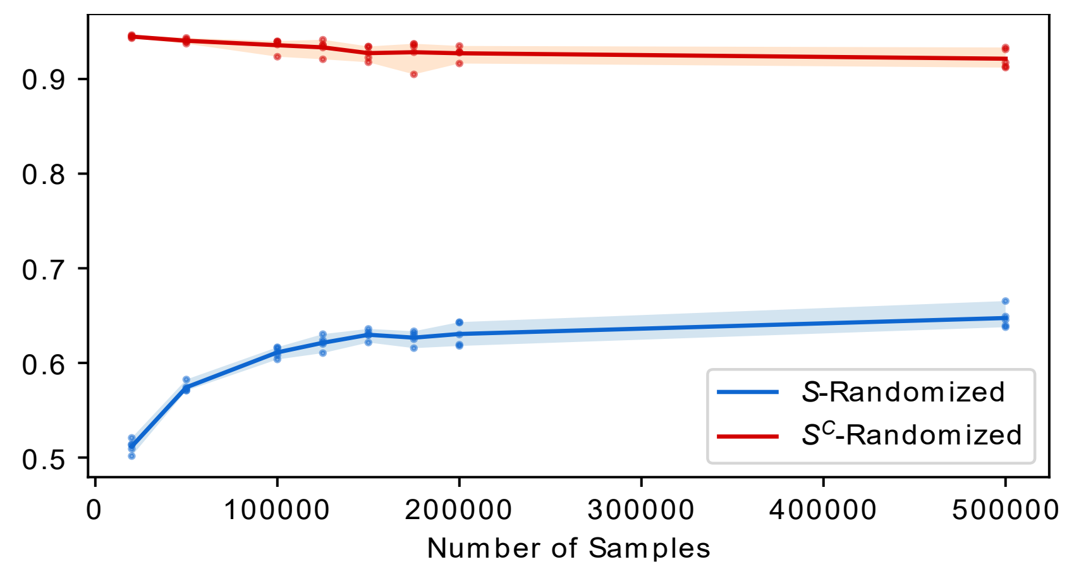
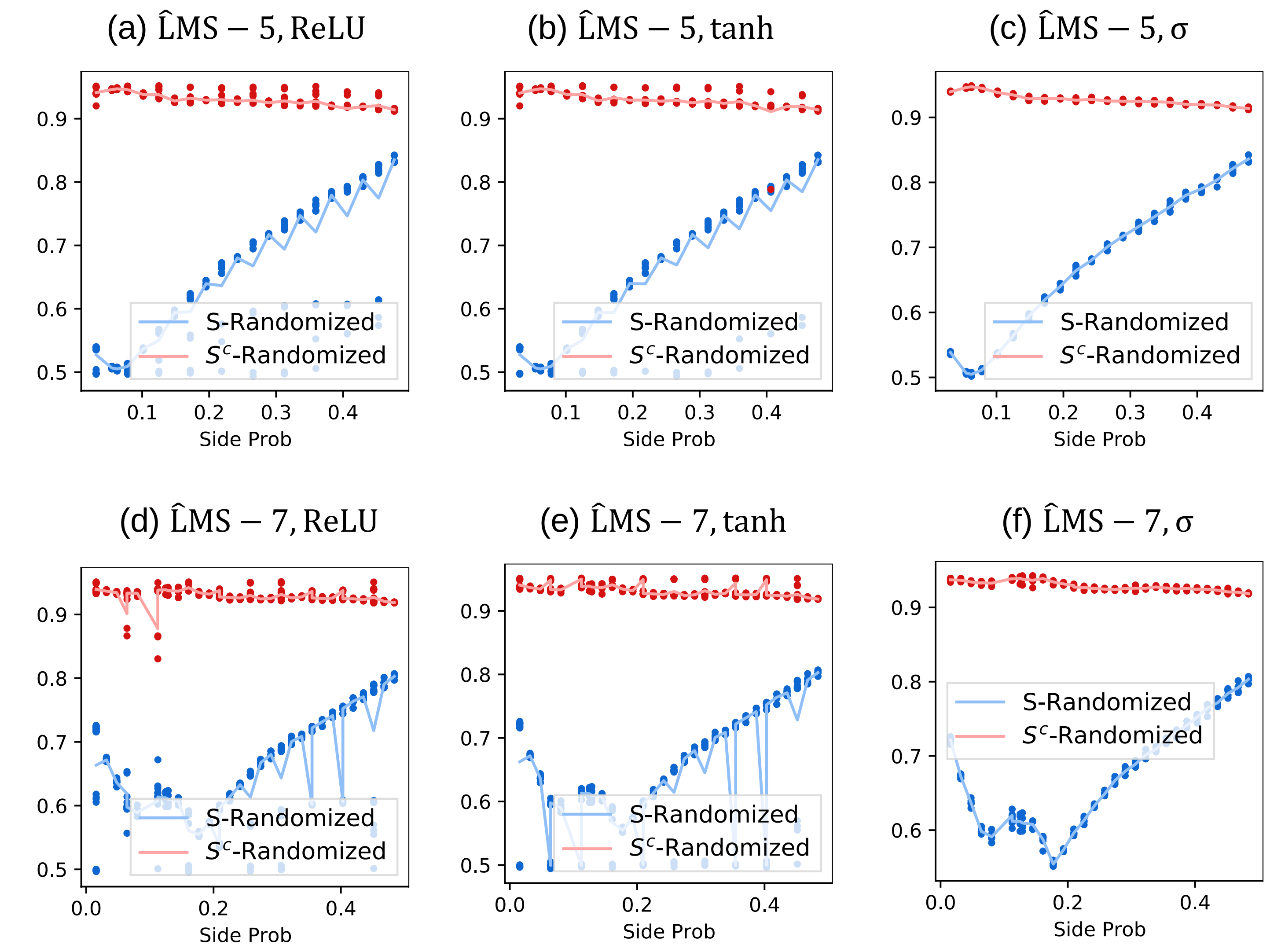


Figure 3: Varying number of samples. Though it is a consensus that increasing training size should mitigate SB given that the model has enough capacity., our results show SB stays invariant to the number of training samples.

Also, we denote the **probability of sampling from the slabs on the two farthest sides** as $\mathcal{P}^+$ and evaluate the performance under different values of $\mathcal{P}^+$.



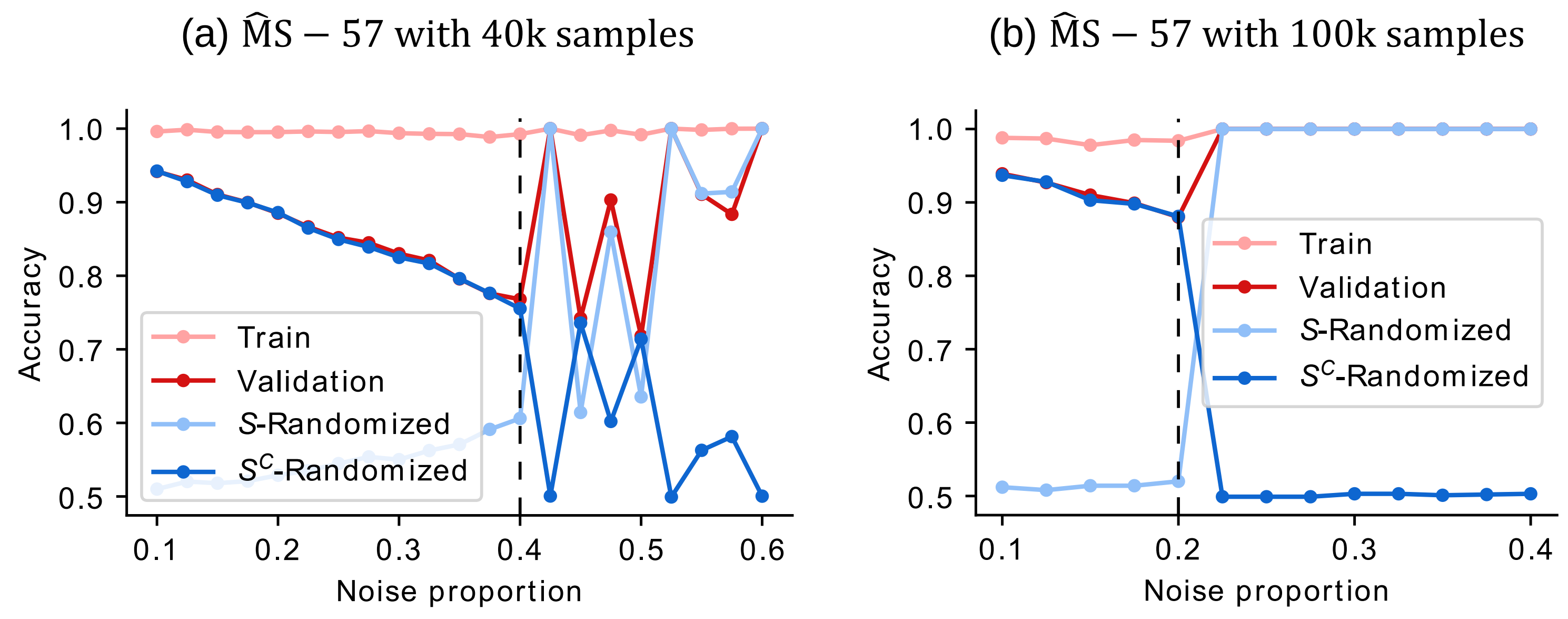
(a) $\widehat{\mathrm{LMS}}-5, \mathrm{ReLU}$ (b) $\widehat{\mathrm{LMS}}-5, \tanh$ (c) $\widehat{\mathrm{LMS}}-5, \sigma$
(d) $\widehat{\mathrm{LMS}}-7, \mathrm{ReLU}$ (e) $\widehat{\mathrm{LMS}}-7, \tanh$ (f) $\widehat{\mathrm{LMS}}-7, \sigma$

Figure 4: Varying side proportion. It is observed that reliance on simple features can be largely reduced by increasing $\mathcal{P}^+$, i.e., balancing the distribution modes, regardless the activation functions used.

③ Corrupting simple features makes them less simple

A neural network prefers noisy simple features to predictive complex features, while how noisy is too noisy?

We investigate how the noisiness of simple features affects a model's preferred features for prediction by varying the noise proportion (i.e., the proportion of the data points that are noisy) in the range $[0.1, 0.4]$.



(a) $\widehat{\mathrm{MS}}-57$ with 40k samples (b) $\widehat{\mathrm{MS}}-57$ with 100k samples

Figure 5: Accuracies vs. noise proportion on $\widehat{\mathrm{MS}}-57$ of different sizes. As noise proportion increases, S-randomized accuracy increases, i.e., the model's predictions rely less and less on the simple features.

- A model has SB when little data is noisy; a large noise proportion gives no SB.
- When simple features become noisy enough, there exists an "**inflection point**," after which the model occasionally (left) or entirely (right) resorts to complex features for prediction and achieves a smaller generalization gap.
- Notably, the model's preference for features does not vary smoothly w.r.t. noisiness of the simple features but **shifts drastically** beyond a certain point.

⇒ NNs might have **a more complicated perception of "feature simplicity"** than our notion of "simplicity" (complexity of the optimal decision boundary) as defined above.

Conclusion of Insights

- More complex but predictive features do not induce the model to utilize them;
- A balanced dataset can be more useful than a large long-tail dataset;
- The dramatic change in the feature preferences of models when the simple features are more corrupted may imply the effectiveness of deliberate distortion of simplistic features to boost model performance.

Carnegie Mellon University, School of Computer Science
{xuchengo, tianwenf}@cs.cmu.edu