Impact of Label Noise on Learning Complex Features

Rahul Vashisht[1]* **P. Krishna Kumar** [1]* **Harsha Vardhan Govind**[2]†
Harish G. Ramaswamy[1]

[1] Indian Institute of Technology Madras [2] IIITDM Kancheepuram
{rahul,pkrishna,hariguru}@cse.iitm.ac.in, cs21b1052@iiitdm.ac.in

Abstract

Neural networks trained with stochastic gradient descent exhibit an inductive bias towards simpler decision boundaries, typically converging to a narrow family of functions, and often fail to capture more complex features. This phenomenon raises concerns about the capacity of deep models to adequately learn and represent real-world datasets. Traditional approaches such as explicit regularization, data augmentation, architectural modifications, etc., have largely proven ineffective in encouraging the models to learn diverse features. In this work, we investigate the impact of pre-training models with noisy labels on the dynamics of SGD across various architectures and datasets. We show that pretraining promotes learning complex functions and diverse features in the presence of noise. Our experiments demonstrate that pre-training with noisy labels encourages gradient descent to find alternate minima that do not solely depend upon simple features, rather learns more complex and broader set of features, without hurting performance.

1 Introduction

Overparameterized models trained using stochastic gradient descent tend to focus on only a small fraction of the available features. Such behavior reduces the diversity of features that contribute to the classification of data. This phenomenon has been discussed in the context of *simplicity bias* [20, 17], where neural models solely rely on simpler, easy-to-learn, features. It is also studied in relation to the intrinsic regularization properties of SGD that inherently favors lower-complexity models [10]. Previous works have shown that models trained using SGD learn linear functions first, and as training progresses they learn functions of increasing complexity [10]. This preference causes many potentially useful features to remain unlearned and underutilized, resulting in models that have inferior discriminative quality and rely on features that are coincidental rather than causal [2]. Such models when faced with distributional shifts or adversarially perturbed data, are unable to generalize to novel or altered environments.

Recent works have attributed this biased behavior to the intrinsic regularization properties of SGD that naturally favors low-complexity solutions [10]. Models that exhibit poor generalization for minority groups are shown to dedicate excess parameters to memorizing a small number of data points [19]. Complex features are often overshadowed by the amplification and replication of simpler features [1]. In real-world datasets, simplicity bias in neural networks is often defined as the propensity to learn low-dimensional projections of inputs [14], where features such as shape and color (e.g., in MNIST or CIFAR) define decision boundaries of varying complexity. Standard approaches like ensembling and adversarial training fail to effectively address the limitations imposed by this bias [20]. There is an increasing interest in understanding the factors that contribute to SGD's bias toward simpler feature sets, *shortcut feature learning*, as well as strategies to mitigate or exploit this tendency to improve model's performance. These include approaches that aim to reduce feature correlation [1, 24, 14, 23],

*Equal contribution with random order.
†Work done while at RBCDSAI, IIT Madras.

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

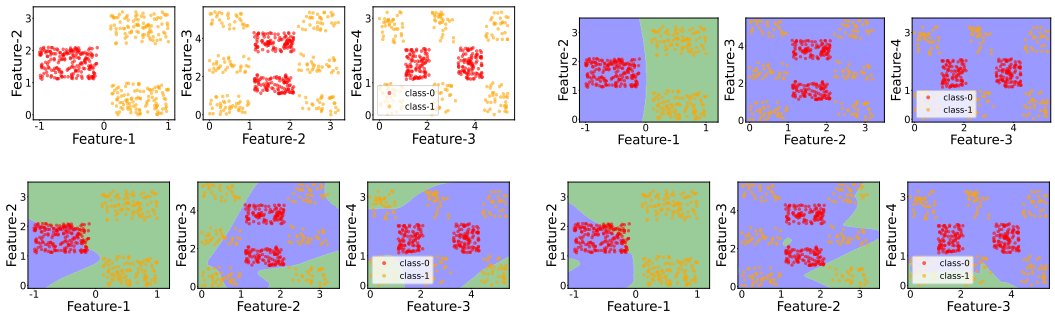


Figure 1: *(Top-left)* First 4 dimensions of Slab-data. *(Top-right)* Decision boundary learned from regular training. *(Botton row)* Two different decision boundaries learned after noisy pre-training. All models achieve 100% training and test accuracy.

and use empirical risk minimization to learn more complex functions [9]. More advanced methods that regularize the conditional mutual information of simpler models compel them to utilize a broader range of features, which has been shown to enhance out-of-distribution generalization [24]. SGD can also learn poorly generalizable functions when the models parameters are initialized by training on random labels [13]. Similarly, the features selected across data points are shown to be strongly correlated with neural feature matrix (NFM) [18, 3]. In the context of convolutional neural networks (CNNs), filter covariance is shown to closely mimic the average gradient outer product concerning input [3]. Despite these efforts, this bias remains a critical area of concern, necessitating further exploration to develop more robust and OOD generalizable neural models.

Pretraining with noisy labels is known to alter the optimization trajectory and often change the local minima to which SGD converges [4]. While adding noise is shown to help models generalize better, it has a tendency to overfit in overparameterized models [5]. The effects of adding noise is shown to be equivalent of employing a regularized loss function that depends on factors such as noise strength, batch size, etc. [4, 8] Noisy labels have also been proposed for robust-loss-functions but they are often insufficient to learn accurate models [26]. However, all these works do not explore the qualitative nature of diverse features that models learn.

In this study, we investigate the learned decision-boundaries through the lens of feature diversity. We particularly focus on the role of parameter initialization in shaping the set of features utilized for classification. Our empirical analysis shows that:

- A simple pre-training phase by optimizing log-loss over perturbed *noisy-labels* can be utilized to learn a more diverse family of functions by neural models. This is in contrast to the usual convergence of models to a similar family of decision functions.
- Neural models are capable of leveraging a broader array of features that are likely to exhibit greater generalization when faced with distribution shifts.

2 Effect of Noisy Pre-training on Learned Decision Functions

We study a two stage training procedure: First, we pre-train an overparameterized model on corrupted labels (by randomly flipping the labels for a fraction of data points) called as **noisy pre-training**. This pre-trained model is then optimised to minimize the training loss, which naturally leads to a high training accuracy over the noisy-labels. Second, we utilize the pretrained model to once again train over the original unmodified labels. In subsequent sections, we compare this two-stage method with standard training without explicit regularization.

Consider a d-dimensional synthetic Multi-slab dataset $\mathcal{D}$, whose first coordinate is a linear block with instances for class-0 sampled from Uniform distribution in $[-1, 0]$ and class-1 sampled from Uniform distribution in $[0, 1]$. The remaining $d-1$ coordinates have samples from two classes distributed in k-well separated alternating regions [3] [20]. We consider a a 4-dimensional multi-slab data with increasing complexity. Figure 1 illustrates the 4-dimensional multi-slab data. Here, an overparameterized ReLU network can potentially use any feature to determine the class label. Note

[3]To eliminate any special association related to the ordering of features, we apply a random orthogonal projection to the dataset, and use the projected features instead of the original feature set.

that, Feature-1 requires a simple linear decision function, whereas Feature-2 requires a $3-$piecewise linear function. Standard training leads to decision functions that solely depend on feature-1 and thus learn a linear function. While this is perfectly good for linearly separable data, this tendency persists even in cases with slight feature overlap in classes along Feature-1, diminishing the model's generalizability. This observation aligns with the well-documented phenomenon of *shortcut learning* [7, 15, 17] and the behavior of neural networks as max-margin classifiers [21, 6].

In case of noisy pretraining, this sole dependence on Feature-1 is lost, and dependence on other feature increases as shown in the Figure 1. Randomized shuffle accuracy help understand the dependence on other features. Table 1 shows the accuracy for randomly shuffling different features. Noisy pre-training increases the shuffle accuracy for feature-1 from 50.3% to 64.2%, thus showing the increased dependence of the decision functions on other features. This is also evident from the reduced shuffle accuracy for feature-2 and feature-3 under noisy pre-training setting. The key observations are that the decision boundary is no longer solely reliant on the first dimension, but now incorporates information from multiple dimensions, and different noisy-datasets, $\mathcal{D}$ and $\mathcal{D}'$, result in distinct decision boundaries. This is unlike earlier instances where the model consistently converged to the same decision boundary.

Significance of this observation: The benefits of moving beyond the simpler models typically learned by SGD-based optimizers are evident for improved *out-of-distribution* generalization. Rather than basing classification decisions on a single feature, the model begins to leverage a broader range of features. Additionally, this result provides evidence that there are multiple local minima with similar levels of accuracy across $\mathcal{D}$, and that SGD is capable of converging to these diverse solutions, rather than being confined to a single-simple decision boundary.

3 Experiments

In this section, we empirically demonstrate the impact of initializing parameters from noisy pretraining over Dominoes [20] and Waterbirds data [16]. See Appendix A.1 for more datasets.

Datasets: *MNIST-fMNIST* dataset from Dominoes consists of collated images, where $\{0, 1\}$ MNIST digits are vertically stacked with {*shirt, frock*} Fashion-MNIST apparel. Here, MNIST block is the easier feature to learn, whereas fMNIST is complex. WaterBirds dataset comprises of birds natively found on land/water, with different backgrounds of land or water. An image of water-bird with water in the background (similarly land-bird on land) is called *in-group* partition, whereas mismatch of waterbird-on-land or landbird-on-water is referred to as *out-group* partition. Details in Appendix A.1.

Measuring feature dependence: To identify the feature(s) that the learned model is utilizing for classification decision, we use: **(1).** *Randomized shuffle accuracy*: to selectively shuffle one among MNIST (top) or fMNIST (bottom) part of the test images, and reevaluate the model. If its performance is drastically reduced by shuffling a feature, it implies the dependence of model's decision on the corresponding feature. **(2).** *Visualizing Gram matrix*: We plot diagonal entries of Gram matrix $W_1^T W_1$ to observe changes in the first layer's parameter W_1. The brighter parts of $W_1^T W_1$ denote greater dependence on those corresponding pixels in the input image. Further, Appendix A.6 shows eigenvector based visualization for Gram matrix.

Varying correlation between features & class labels: We control the predictive powers of features by perturbing their correlation with true labels. For example: In Dominoes dataset $\mathcal{D}$, we create partially-correlated data $\mathcal{D}'$ by collating images such that top MNIST-block is only 95% correlated with classification labels (not 100% accurately predictive). This implies that if predictions were solely on the basis of top-block, then the bayes-error will be 5%. The rationale of adding adversarial correlation is to push the models to those regimes where no simple solution achieves perfect accuracy, thus getting closer to real world scenarios.

Parameter Initialization Regime	Feature Shuffle Accuracy		
	Feature-1	Feature-2	Feature-3
Standard data-agnostic (like Xavier-glorot)	50.3 ± 1.78	100 ± 0.0	100 ± 0.0
Pretraining model with 10% noisy labels	64.2 ± 6.8	91.6 ± 4.3	97.1 ± 2.2

Table 1: **Multi-Slab Data:** Randomized feature shuffle accuracies, averaged over 10 runs. All models are trained to achieve 100% training and test accuracy.

Data	Standard Training			Noisy Pre-training		
	Train Acc.	MNIST Rnd.	F-MNIST Rnd.	Train Acc.	MNIST Rnd.	F-MNIST Rnd.
$\mathcal{D}$	99.9 ± 0.0	52.5 ± 0.33	98.3 ± 0.05	99.7 ± 0.07	53.6 ± 1.56	88.6 ± 0.76
$\mathcal{D}'$	98.5 ± 0.07	93.1 ± 0.33	56.5 ± 0.42	99.9 ± 0.06	81.2 ± 1.02	57.2 ± 1.50

Table 2: **Dominoes MNIST-FMNIST** (Rnd. short for Randomized) : Randomized shuffle accuracies, averaged over 3 runs, with 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of MNIST with the true labels. Each for 3 and 4 layer fully connected networks. Noisy pre-training uses 10% corrupt labels.

Noisy Pre-training Helps Learn Complex Features: We observe that neural models usually learn simple decision functions that are restricted to *easy-to-learn* features. Table1 shows that randomizing feature-1 reduces the accuracy to 50%, proving absolute reliance on this feature only. Similarly, Table2 shows randomizing top MNIST block reduces model accuracy to 52.5% and Table3 shows better performance for in-group images. Conversely, the model does not deteriorate performance upon randomizing other complex features in all datasets. However, when we initialize parameters from noisy pretrained models, we see that not only over reliance of easy-features is reduced, the model's ignorance to other features is reduced. Table1 and 2 evince increase in randomized shuffling accuracy and Table3 shows better out-group performance.

Noisy Pre-training Helps Learn Diverse Functions: With different label noise, we observe that the decision boundaries learned are varying across each feature. Figure1 bottom-row shows two different decision boundaries for varying noise during pretraining. Diverse models are helpful because they tend to focus on different aspects of of high dimensional data. In practical scenarios, we can make much more robust predictions by bagging these multiple decision boundaries to give aggregated class prediction.

Impact of Label Smoothing: Please refer Appendix A.4.

With noisy-pretraining and label smoothing, the models learn to focus on multiple features in Multi-slab and Dominoes data, and learn to classify images based on birds, rather than their backgrounds in case of WaterBirds. Figure 2 and 3 shows increase in the brightness of pixels in the Gram-matrix that confirms learning of diverse/complex features.
Why are diverse models better? Model diversity is evident in slab-data from Figure9. It is interesting to note that each noisy pretraining leads to a different minina, and different family of function. The predictions from diverse models, focusing on diverse features, can be ensembled to have more robust predictions. Similar trends are also evident from high standard-deviation values in accuracy of models with noisy pretraining in Table2 and 3. The theoretical validation of such ensemble's superiority is for further exploration.

Data	Standard Training			Noisy Pre-training		
	Train Acc.	In-group	Out-group	Train Acc.	In-group	Out-group
$\mathcal{D}$	100 ± 0.0	85.2 ± 0.43	38.5 ± 0.88	99.5 ± 0.34	78.1 ± 1.02	44.1 ± 1.60
$\mathcal{D}'$	100 ± 0.0	84.1 ± 0.48	44.4 ± 0.67	99.5 ± 0.73	77.8 ± 1.15	46.9 ± 0.92

Table 3: **WaterBirds Dataset**: Test In-group and Out-group accuracies averaged over 10 runs on waterbirds dataset, with 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of background with the true labels. For noisy pre-training, we corrupt 10% of labels.

4 Discussion and Conclusion

Overparameterized neural networks tend to learn decision functions based on a limited and simpler set of features if they exist. Our experiments demonstrate that initializing the model parameters by first training over noisy-labels can enable the model to overcome this constraint, and allow for the learning of more diverse range of features. One possible explanation for this phenomenon is that the neural model reaches a minima that is characterized by a complex decision boundary to fit the noisy labels. The model, however, remains trapped in this minima and is unable to revert to a simpler decision boundary despite the intrinsic regularization properties of SGD. This observation challenges the commonly accepted opinion that SGD provides effective implicit regularization.

Our experiments motivate further investigation into the nature of the role of initialization in guiding SGD optimizers. Note that all the models, with or without noisy label pretraining, have similar (high) accuracy on unseen data. Since each local minima that is reached during the noisy pre-training leads to a distinct family of functions, it hints at interesting future works about characterizing loss-landscape of ReLU networks. This is in contrast to the standard data-agnostic initializations, such as Xavier-Glorot, that converge to similar decision boundaries. Our work connects feature subset selection with the learning of diverse families of functions. Future works will explore any theoretical relationships between these concepts and to quantify their interdependency.

Our observations challenge the prevalent view in the literature that neural networks are extremely biased towards learning only simple features if they exist. We demonstrate that models can be guided to learn more complex features through pre-training on noisy labels. As a corollary, we show that the phenomenon of *shortcut learning* is not as extreme as previously thought and can be mitigated to some extent. Our experiments further reveal that neural networks are capable of learning multiple, higher-complexity features, and diverse families of functions.

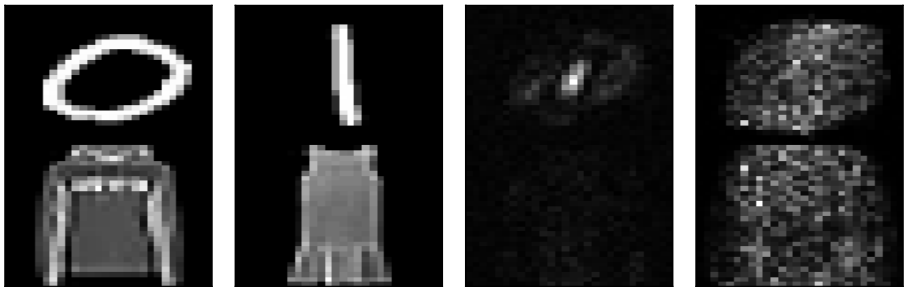


Figure 2: MNIST-FMNIST dominoes data samples with visualizing first layer neural feature matrix $W_1^\top W_1$, for **(Left)** Standard training, and **(Right)** Random pre-training respectively.

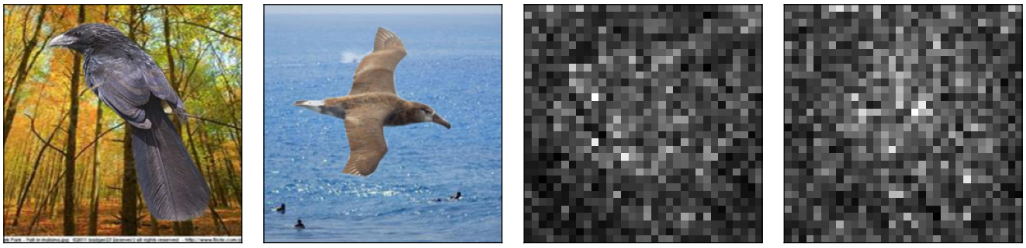


Figure 3: Samples from Waterbirds dataset. We visualize the diagonal of first layer neural feature matrix $W_1^\top W_1$ for **(Left)** standard training and **(Right)** random pretraining.

References

[1] Sravanti Addepalli, Anshul Nasery, Venkatesh Babu Radhakrishnan, Praneeth Netrapalli, and Prateek Jain. Feature reconstruction from outputs can mitigate simplicity bias in neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. 2019. cite arxiv:1907.02893.

[3] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Parthe Pandit, and Misha Belkin. Mechanism of feature learning in convolutional neural networks. *ArXiv*, abs/2309.00570, 2023.

[4] Alex Damian, Tengyu Ma, and Jason Lee. *Label noise SGD provably prefers flat global minimizers*. Curran Associates Inc., Red Hook, NY, USA, 2024.

[5] Spencer Frei, Yuan Cao, and Quanquan Gu. Provable generalization of sgd-trained neural networks of any width in the presence of adversarial label noise. In *International Conference on Machine Learning*, pages 3427–3438. PMLR, 2021.

[6] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[7] Katherine L. Hermann, Hossein Mobahi, Thomas Fel, and Michael C. Mozer. On the foundations of shortcut learning, 2024.

[8] JE Huh and P Rebeschini. Generalization bounds for label noise stochastic gradient descent. volume 238 of *Proceedings of Machine Learning Research*, pages 1360–1368. PMLR, 2024.

[9] Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew Gordon Wilson. On feature learning in the presence of spurious correlations. NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[10] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin L. Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems 32, NeurIPS 2019, Dec 8-14, 2019, Vancouver, BC, Canada*, pages 3491–3501, 2019.

[11] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[12] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. 2005.

[13] Shengchao Liu, Dimitris S. Papailiopoulos, and Dimitris Achlioptas. Bad global minima exist and SGD can reach them. In *Advances in Neural Information Processing Systems 33, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[14] Depen Morwani, Jatin Batra, Prateek Jain, and Praneeth Netrapalli. Simplicity bias in 1-hidden layer neural networks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[15] Nihal Murali, Aahlad Puli, Ke Yu, Rajesh Ranganath, and Kayhan Batmanghelich. Beyond distribution shift: Spurious features through the lens of training dynamics, 2023.

[16] Suzanne Petryk, Lisa Dunlap, Keyan Nasseri, Joseph E. Gonzalez, Trevor Darrell, and Anna Rohrbach. On guiding visual attention with language specification. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18071–18081, 2022.

[17] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: a learning proclivity in neural networks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc.

[18] Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features. 2022.

[19] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[20] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems NeurIPS 2020, Dec 6-12, 2020, virtual*, 2020.

[21] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, 19(1):2822–2878, jan 2018.

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[23] Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16761–16772, June 2022.

[24] Bhavya Vasudeva, Kameron Shahabi, and Vatsal Sharan. Mitigating simplicity bias in deep learning for improved ood generalization and robustness, 2023.

[25] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[26] Deng-Bao Wang, Yong Wen, Lujia Pan, and Min-Ling Zhang. Learning from noisy labels with complementary loss functions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10111–10119, 2021.

[27] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[28] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.

A Appendix

A.1 Dataset Description:

Slab Data: Figures 4 and 5 shows 4-dimensional dataset and noisy 4-dimensional multi-slab dataset.

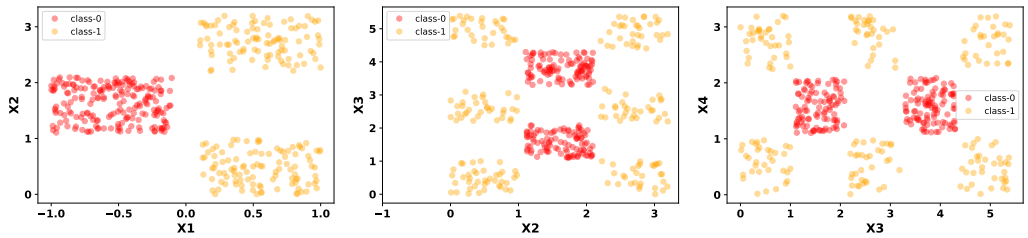


Figure 4: Noise augmented multi-slab dataset

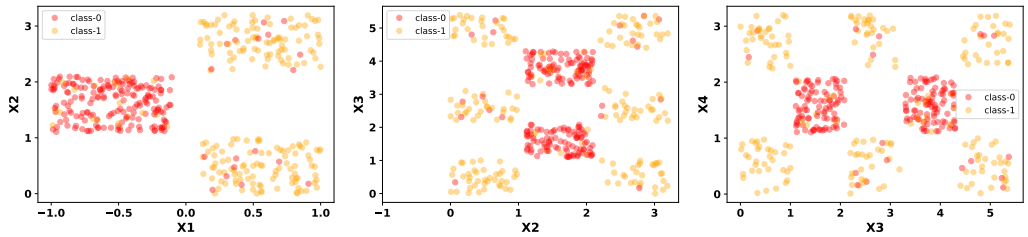


Figure 5: Noise augmented multi-slab dataset

Dominoes Data: We consider Dominoes binary classification datasets consisting of 3 independent datasets, where the top half of the image contains MNIST digits [12] from classes 0, 1, and the bottom half contains MNIST images from classes 7, 9 (MNIST-MNIST), Fashion-MNIST [27] images from classes coat, dress (MNIST-Fashion) or CIFAR-10 [11] images from classes car, truck (MNIST-CIFAR). In all Dominoes datasets, the top half of the image (MNIST 0-1 images) presents a easy to learn feature; the bottom half of the image presents a harder-to-learn feature. The images are made into gray-scale and scaled to appropriate size so that they can be collated top-bottom style. Figure 6 shows the examples of dominoes dataset. In case of 100% correlation Note that each block, top and bottom, is fully predictive of the class label in case of 100% correlation, while only harder to learn feature is fully predictive of class label in case of 95% correlation.

Waterbirds Dataset: The waterbirds dataset is constructed by cropping birds from Caltech-UCSD Birds-200-2011 (CUB) dataset [25] and taking backgrounds from the Places dataset [28] and placing the birds from CUB dataset to backgrounds from Places dataset. Figure 7 shows sample instances from waterbirds dataset. Table 4 describes the proportions of different groups in the waterbirds dataset. The waterbirds on land and landbirds on water are known as out-group, while landbird on land and waterbirds on water are known as the in-group.

Figure 6: Samples from Dominoes Dataset, MNIST-FMNIST, MNIST-MNIST, and MNIST-CIFAR

Figure 7: Samples from Waterbirds dataset

Data Split	Landbirds, Land	Landbirds, Water	Waterbirds, land	Waterbirds, Water
Train $\mathcal{D}$	3694 (77.04%)	0	0	1101 (22.96%)
Train $\mathcal{D}'$	3498 (72%)	184 (3.8%)	56 (1.2%)	1057 (22%)
Validation	467 (38.95%)	466 (39.95%)	133 (11.1%)	133(11.1%)
Test	2255 (38.92%)	2255(38.92%)	642 (11.1%)	642 (11.1%)

Table 4: WaterBirds data description: $\mathcal{D}$ and $\mathcal{D}'$ with 100% and 95% correlation respectively [16]

A.2 Network Architectures

Slab Dataset: We use 2-layer multi-layer perceptron network with ReLU activation, the first layer contains 100 hidden unit and second layer contains 200 hidden units. We use SGD with 0.1 learning rate as optimizer.

Dominoes Dataset: For all dominoes dataset, "MNIST-FMNIST", "MNIST-MNIST", and "MNIST-CIFAR", datasets, we use 4-layer fully connected networks with 10 hidden units and ReLU activation. We use SGD optimizer with learning rate 0.01.

WaterBirds Dataset: We use fully connected network, with 4 layers and ReLU activation having 20 hidden units. We use SGD optimizer with learning rate 0.05 .

A.3 Additional Results:

WaterBird Dataset	Standard Training			Noisy Pre-training		
	Train Acc.	In-group	Out-group	Train Acc.	In-group	Out-group
$\mathcal{D}$	100 ± 0.0	84.48 ± 0.59	43.64 ± 3.56	99.89 ± 0.07	77.77 ± 0.88	49.89 ± 2.17
$\mathcal{D}'$	100 ± 0.0	83.48 ± 0.53	49.44 ± 3.80	99.68 ± 0.08	78.41 ± 0.56	51.46 ± 1.48

Table 5: **WaterBirds Dataset**: Test In-group and Out-group accuracies averaged over 10 runs on waterbirds dataset, with 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of background with the true labels. For noisy pre-training, we corrupt 10% of labels. Label smoothing (LS) value is 0.1

MNIST -MNIST	Hidden Dim	Standard Training			Noisy Pre-training		
		Train Acc.	Top Shuffle	Bottom Shuffle	Train Acc.	Top Shuffle	Bottom Shuffle
$\mathcal{D}$	10	100	50.89	99.75	99.82	53.20	98.31
	50	100	50.15	99.71	100	49.94	99.50
$\mathcal{D}'$	10	98.81	88.98	59.81	99.82	90.81	56.99
	50	99.20	90.74	58.19	99.92	92.63	57.76

Table 6: **Dominoes MNIST-MNIST**: Randomized shuffle accuracies, with 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of MNIST with the true labels. **Top**:MNIST, **Bottom**:MNIST. Each for 3 and 4 layer fully connected networks. Noisy pre-training uses 10% corrupt labels.

MNIST -CIFAR	Standard Training			Noisy Pre-training		
	Train Acc.	MNIST Shuffle	CIFAR Shuffle	Train Acc.	MNIST Shuffle	CIFAR Shuffle
$\mathcal{D}$	100	50.88	99.79	99.66	51.44	99.58
$\mathcal{D}'$	100	54.62	87.14	96.22	53.48	92.30

Table 7: **Dominoes MNIST-CIFAR**: Randomized shuffle accuracies, with 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of MNIST with the true labels. **Top**:MNIST, **Bottom**:CIFAR. Noisy pre-training uses 10% corrupt labels.

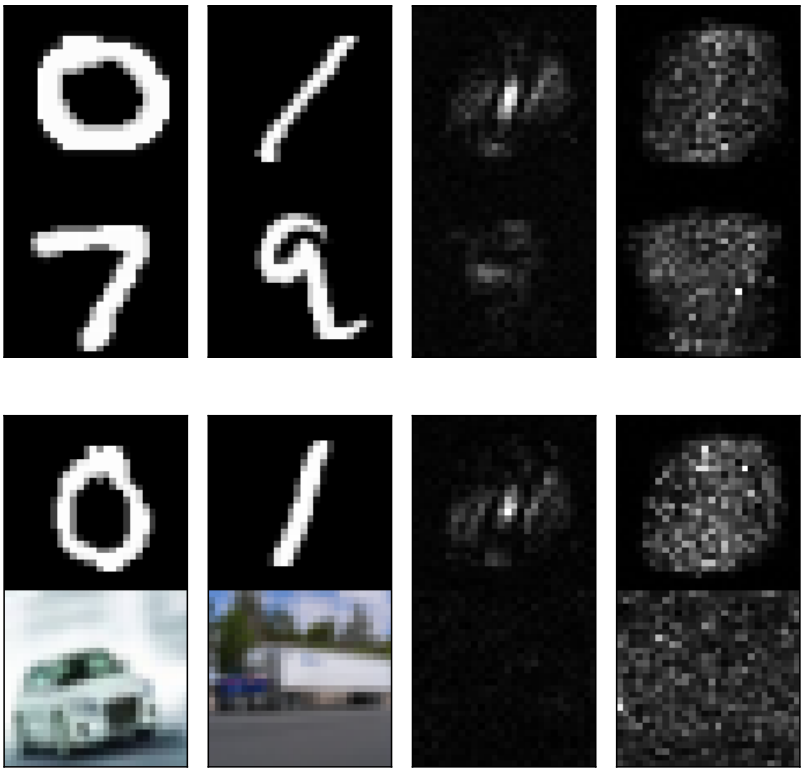


Figure 8: Visualizing the diagonal of first layer neural feature matrix $W_1^\top W_1$, where W_1 is parameters of first layer network of the learned network, **(Top)** MNIST-MNIST, and **(Bottom)** MNIST-CIFAR Dominoes data samples with standard training and noisy pre-training respectively.

A.3.1 4D Slab Data Decision Boundary plots for Different Random Seeds

In this section, we show that under standard training, the model converges to a similar family of decision functions, which are easier to learn. This convergence does not happen under pretrained noise based training, which allows model to learn diverse set of functions with reliance on harder to learn features. Figure 9 and 10 demonstrates this finding using decision boundary plots for 4d slab data described in Section 2.

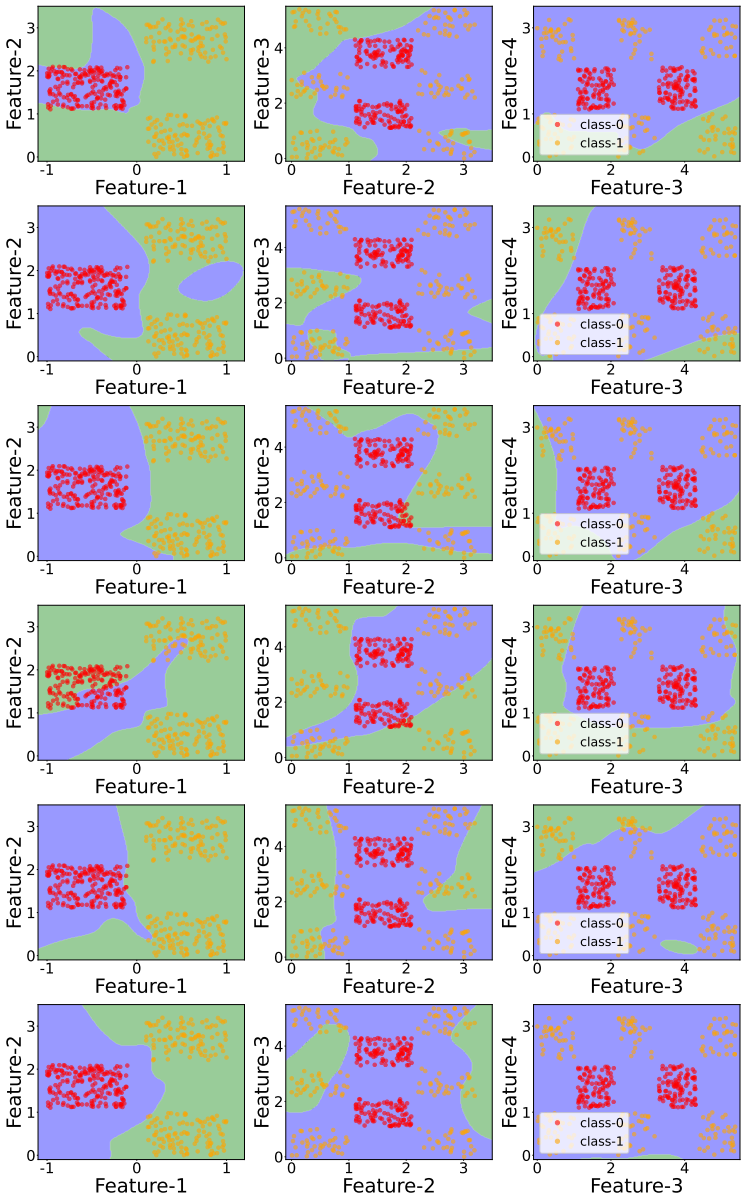


Figure 9: **Diverse Decision boundary** plots for different random initializations, Each subfigure shows decision boundary across 4 dimension for pre-training with noisy random labels.

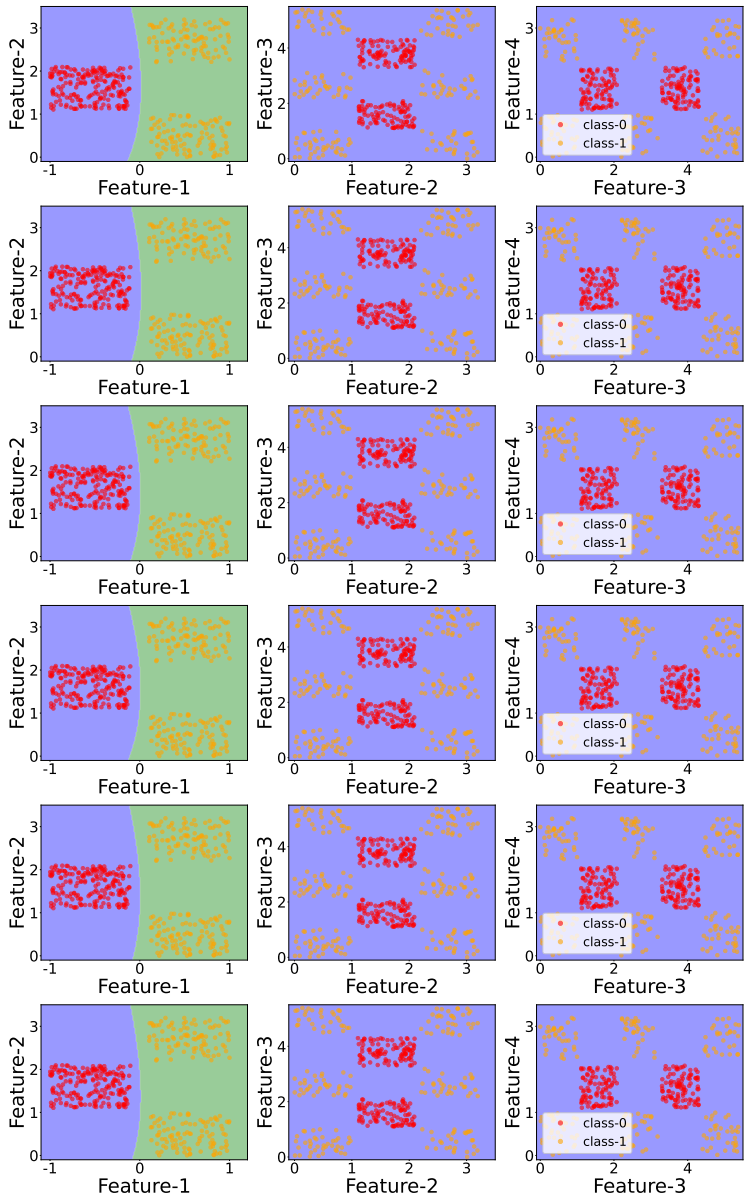


Figure 10: **Similar Decision boundary** plots for different random initializations, Each subfigure shows decision boundary across 4 dimension for training with correct label (without any pretraining).

A.4 Impact of Label Smoothing

Label smoothing [22] is a regularization technique, which replaces the one-hot ground truth vectors, with mixture of ground truth vectors and uniform distribution. In case of noisy pretraining, flip the labels of a fraction of data samples, whereas in case of label smoothing, the ground truth labels are mixture of one hot vector and uniform distribution, which act as a addition of noise to labels, thus showing equivalent effect.

Data	LS coeff.	Standard Training			Noisy Pre-training		
		Train Acc.	In-group	Out-group	Train Acc.	In-group	Out-group
$\mathcal{D}$	0.0	100 ± 0.0	85.2 ± 0.43	38.5 ± 0.88	99.5 ± 0.34	78.1 ± 1.02	44.1 ± 1.60
	0.2	100 ± 0.0	84.4 ± 0.42	43.5 ± 3.62	99.8 ± 0.11	77.3 ± 0.75	51.1 ± 2.17
$\mathcal{D}'$	0.0	100 ± 0.0	84.1 ± 0.48	44.4 ± 0.67	99.5 ± 0.73	77.8 ± 1.15	46.9 ± 0.92
	0.2	100 ± 0.0	83.4 ± 0.49	49.1 ± 3.63	99.5 ± 0.11	78.5 ± 0.28	52.2 ± 1.29

Table 8: **WaterBirds Dataset**: Test In-group and Out-group accuracies averaged over 10 runs on waterbirds dataset, with 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of background with the true labels. For noisy pre-training, we corrupt 10% of labels. Label smoothing (LS) varies between 0.0 and 0.2

Parameter Initialization Regime	Feature Shuffle Accuracy		
	Feature-1	Feature-2	Feature-3
Label Smoothing (0.2) for feature confidence	75.0 ± 4.4	92.9 ± 3.9	91.5 ± 3.4

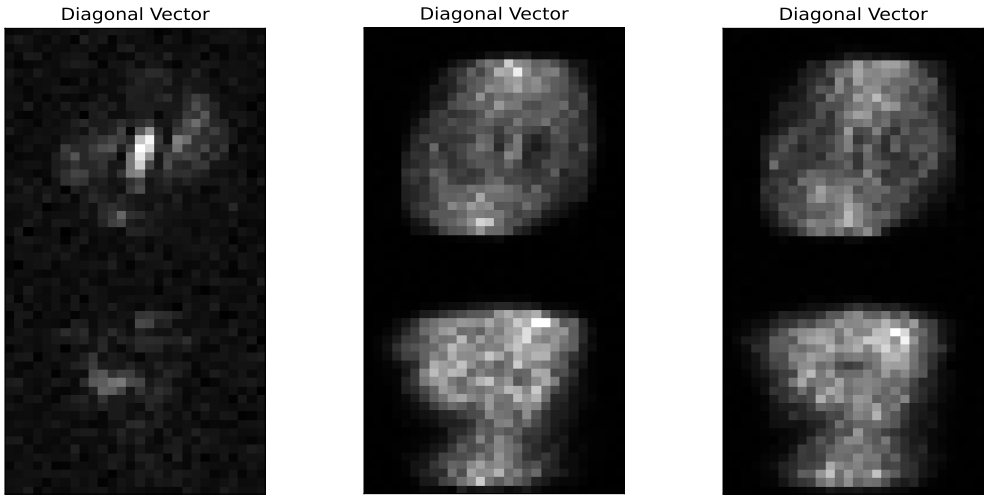
Table 9: **Multi-Slab Data:** Randomized feature shuffle accuracies, averaged over 10 runs. All models are trained to achieve 100% training and test accuracy.

A.5 Varying Model Architecture

Data	Model Layer	Standard Training			Noisy Pre-training		
		Train Acc.	Top Rnd.	Bottom Rnd.	Train Acc.	Top Rnd.	Bottom Rnd.
$\mathcal{D}$	3	99.9 ± 0.0	52.5 ± 0.33	98.3 ± 0.05	99.7 ± 0.07	53.6 ± 1.56	88.6 ± 0.76
	4	100 ± 0.0	52.6 ± 0.24	98.1 ± 0.05	99.8 ± 0.03	55.6 ± 1.50	87.6 ± 0.81
$\mathcal{D}'$	3	98.5 ± 0.07	93.1 ± 0.33	56.5 ± 0.42	99.9 ± 0.06	81.2 ± 1.02	57.2 ± 1.50
	4	100 ± 0.0	92.5 ± 0.17	56.8 ± 0.08	100 ± 0.0	84.4 ± 0.74	56.9 ± 1.30

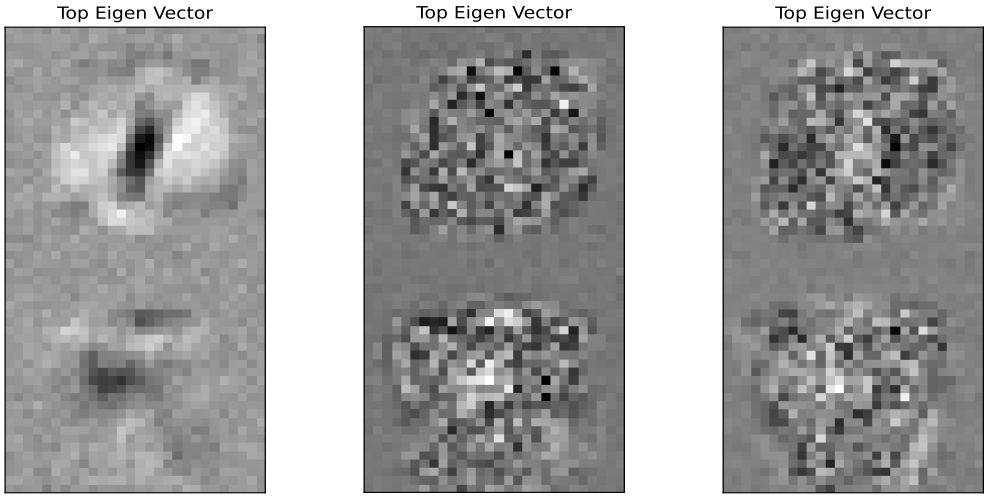
Table 10: **Dominoes MNIST-FMNIST**: Randomized shuffle accuracies, averaged over 3 runs, with 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of MNIST with the true labels. MNIST:FMNIST. Each for 3 and 4 layer fully connected networks. Noisy pre-training uses 10% corrupt labels.

A.6 Visualizing Feature Importance Using Top Eigenvectors



(a) $\mathcal{D}$ standard training) (b) $\mathcal{D}'$ noisy pre-training (c) $\mathcal{D}$ noisy pre-training

Figure 11: Visualizations for diagonal of the first layer neural feature matrix $W_1^\top W_1$. 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of instances with the true labels



(a) $\mathcal{D}$ standard training (b) $\mathcal{D}'$ noisy pre-training (c) $\mathcal{D}$ noisy pre-training

Figure 12: Visualizations for top eigenvector of the first layer neural feature matrix $W_1^\top W_1$. 100% ($\mathcal{D}$) and 95% ($\mathcal{D}'$) correlation of instances with the true labels