Exploring CNN Training Dynamics and Double Descent Phenomenon under Distribution Shifts

Bin Li [1] **Enze Chen** [1]

Abstract

In the field of machine learning, double descent is used to describe a phenomenon when model performance first gets worse and then gets better as we increase model complexity. Reconciling this phenomenon with the classical framework of bias-variance trade-off, which predicts a U-shape curve of performance as model complexity increases, is an outstanding problem in machine learning. The phenomenon of double descent has attracted much attention, yet it has not been verified whether this phenomenon also shows on datasets where the test data are generated from distributions that are shifted from that of training data. In this project, we explored the training dynamics of CNN-style models on CAMELYON17-WILDS, a distribution-shift dataset for binary classification task. We found no evidence of double descent when training CNN-style model on CAMELYON17-WILDS. Yet we did find that certain CNN architecture is much better suited for CAMELYON17-WILDS than its default baseline under over-parameterized scheme. We suggested to change the baseline to our proposed model, which will lead to a more realistic assessment of improvement of the benchmark for future work.

1. Introduction

The bias-variance trade-off, a long-held principle in machine learning practice, has been challenged by a growing number of studies that focus on a phenomenon that contradicts the classical view—the phenomenon of so-call "double descent." Double descent is used to describe when the test error of an over-parameterized model (which we will define later) first decreases, then increases, and finally continues to decrease. The studies on double descent reveal that a complex model is not necessarily more prone to overfit than a simple model, and this is especially true for many over-parameterized models such as boosting methods and neural networks. Interestingly, some works in fair and robust algorithms also lend support to this view. For example, Kleinberg & Mullainathan (2019) have shown that, under certain assumptions, simple models are necessarily worse than complex models in the sense that the former is strictly improvable; that there exists a more complex prediction function that is both strictly more efficient and also strictly more equitable. These findings present challenges to some deep-seated beliefs in the classical statistics and to the principle of Occam's razor, which has been the foundation of many scientific disciplines.

Among different theories that attempt to make sense of double descent, one hypothesizes that this phenomenon is linked to the inductive biases induced by training algorithms which impact model generalizability (Belkin et al., 2018). In the mean time, a parallel set of studies on model generalizability has been focusing on improving model performance under distribution shifts (Koh et al., 2021). In this setup, models are fitted and evaluated on training and test sets that are carefully designed so that they are generated respectively by two overlapping but shifted distributions.

This project is motivated by the connection between these two seemingly parallel yet related bodies of work, and aims to shed some light on the problem of domain generalizability by examining the training dynamics of over-parameterized models. To the best of our knowledge, this project is among the first group of work that examines the training dynamics of CNN-style models on a distribution-shift dataset with a particular focus on the over-parameterization scheme and double descent phenomenon.

In this project, we first replicated the phenomenon of double descent with a simple CNN on CIFAR-10 dataset and then explored the training dynamics of CNN-style over-parameterized models on CAMELYON17-WILDS, a distribution-shift dataset for binary classification task. Our experiments did not show any double descent phenomenon when training CNN models on CAMELYON17-WILDS dataset. However, we found that complex models are not more prone to overfit and can even lead to better gen-

[1]Department of Computer Science, Columbia University, New York, NY. Correspondence to: Bin Li <bl2899@columbia.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

eralization on this distribution shift dataset. Finally, we discovered that a simple 6-layer CNN is better suited on CAMELYON17-WILDS than DenseNet121, with the former showing more well-behaved training dynamics and resulting in lower test error and better generalization.

2. Related Works

2.1. Double Descent, Over-parameterization, and Generalization

The term *over-parameterization* is often used to describe the situation when a model is excessively complex with respect to the size of dataset and results in perfectly fitting the training data.

The classical theory of bias-variance trade-off suggests that the optimal generalization performance of a model should occur at an intermediate model complexity, since simpler models usually exhibit high bias and more complex models shows high variance of the predictive function. However, modern machine learning models do not always exhibit such a simple trade-off, with large deep learning models consistently attain low bias and variance in the heavily over-parameterized regime.

This pecularity of over-parameterized models can manifest in the form of double descent, a phenomenon where model performance first gets worse and then gets better as we increase model complexity.

The discussion of double descent dates back at least to 1989 (Vallet et al., 1989), and has since attracted steady but modest attention in the research community. While previous effort on understanding this phenomenon has been focusing on classical machine learning methods such as Boosting (Wyner et al., 2017), recently Nakkiran et al. (2021) have shown that double descent can also exhibit on various kinds of deep neural network ranging from convolutional neural networks to transformer networks (Vaswani et al., 2017).

Even though much of the mechanism behind this phenomenon is still shrouded in mystery, many theories have been proposed to reconcile the seeming contradiction between double descent and bias-variance trade-off. For example, Belkin et al. (2019) suggest that this phenomenon is the result of a mismatch between model complexity measure and model inductive bias, with model complexity not properly reflecting the usefulness of model inductive bias in the over-parameterized regime. They argue that many seemingly high complexity models actually lead to inductive bias that is more suited for generalizing over some particular type of dataset.

2.2. Robustness and Distribution Shifts

In the meantime, another branch of research tackle the model generalization and robustness issues from the perspective of distribution (or domain) shifts, where test data might be drawn from a distribution not exactly similar to that of training data. This research direction, the so-called Domain Generalization (DG) problem, is motivated by the modeling needs in real-world scenarios, where generalization over several domains is an important desideratum.

Current methodologies for DG can be grouped under the following three major branches.

1. Data manipulation: Methods of this branch manipulate inputs to assist in learning general representations. Two major techniques subsume under this branch are:

   a). Data augmentation (e.g., Sohn et al. (2020)), which augments, distorts, and/or transforms input data;

   b). Data generation (e.g., Robey et al. (2021)), which generates additional samples to facilitate learning.

2. Representation learning: Methods of this branch include some of the most popular techniques in domain generalization. There are two main type:

   a). Domain-invariant learning, which explicitly learn a domain-invariant representation through optimization, adversarial learning, and other methods (e.g., Arjovsky et al. (2019));

   b). Feature disentanglement, which seeks to separate feature space into domain-invariant and domain-dependent parts (e.g., Peng et al. (2020)).

3. Learning strategy: Methods of this branch invent novel general learning algorithms that aim to promote model generalization. Many categories of methods can be grouped under this branch, include:

   a). Ensemble learning, which aggregate several models to reduce prediction variance and improve domain robustness;

   b). Meta-learning, which leverages second-order gradient information to create better initialization condition that can be robust to domain shift;

   c). Gradient operation, which learn generalized representations by directly operating on gradients;

   d). Distributionally robust optimization, which optimizes to improve the worst-case domain performance when training (e.g. Hu et al. (2018));

   e). Self-supervised learning, which pre-trained models on pretext tasks to learn rich and robust representations.

Methods from these three branches attempt to solve the generalization problem through a myriad of conceptually

or practically different angles. They can be complementary to each other and mix-and-match to create a more robust learning algorithm.

3. Methods and Experiments

For our first experiment, we first attempt to replicate the double descent phenomenon shown in Nakkiran et al. (2021). To ground this phenomenon into some concrete concepts, we begin by providing some formal descriptions of the problem here.

A *training procedure* T is a procedure that training classifier $T(S)$ to map the $(data, label)$ set $S = (x_1, y_1), ..., (x_n, y_n)$. The *effective model complexity* of T with respect to distribution D is the maximum number of samples n satisfying the requirement that training error is approximately 0.

With these definitions, the double descent phenomenon can be described as follows: For any natural training dataset D, with a small $\epsilon > 0$ and a neural network model training procedure T, if we are considering making prediction labels based on the *effective model complexity* n from D, then we can observe 3 following results when we choose different *effective model complexity*:

Under-parameterized regime: If *effective model complexity* is sufficiently smaller than n, increasing *effective model complexity* will decrease the test error.

Over-parameterized regime: If *effective model complexity* is sufficiently larger than n, increasing *effective model complexity* will decrease the test error.

Critically parameterized regime: If *effective model complexity* $\approx n$, increasing *effective model complexity* might increase or decrease the test error.

To try replicating the phenomenon, it is important to provide a definition of *effective model complexity*. Here to simplify the experiment, we set separate parameter (which is highly related to effective model complexity) when using different model instead of uniform formula of effective model complexity.

After replicating the results from Nakkiran et al. (2021), we apply the same experiment paradigm on CAMELYON17-WILDS dataset with different CNN models and varying conditions such as the presence of label noise, data augmentation, and novel training algorithm.

3.1. Model Setup and Data Processing

We use 2 models for this project. A simple 6-layer CNN and DenseNet121 (Huang et al., 2017), a far more complicated 5-layer network structure containing multiple dense blocks and transition layers. The 6-layer CNN contains 5 $[3 \times 3]$ convolutional layers of width $[k, 2k, 4k, 8k, 16k]$. Different k will directly vary *effective model complexity*. For DenseNet121, we use different growth rate to generate models of different *effective model complexity* (default growth rate is 32).

We use a single Nvidia RTX 3090 and 8 Nvidia Tesla T4 for our experiments. But this amount of computing resource is still limiting for the scope of our project. As a result, we only run one trial of training for model of any complexity for a given experiment and thus cannot provide the confidence interval for our reported statistics. To improve the signal-to-noise ratio when plotting our experiment data, we pass each plotted signal through a one dimensional Gaussian filter with $\sigma = 0.5$.

3.2. Datasets

3.2.1. CIFAR-10

The CIFAR-10 dataset Krizhevsky et al. (2009) is a image classification dataset with 60000 32×32 colored images belonging to 10 classes. Each class has 6000 images. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 1000 images.

3.2.2. CAMELYON17-WILDS

CAMELYON17 (Bandi et al., 2018) is a binary classification task, which means the answer is only yes or no. It aims to predict if a given region of tissue contains any tumor tissue. A single input of dataset is a 96×96 image, and its label is a binary indicator of whether the central region contains any tumor tissue.

Koh et al. (2021) have created a distribution-shifted version of this dataset called CAMELYON17-WILDS. The domain of CAMELYON17-WILDS is hospital, and the model is tested to see whether it can generalize from samples obtained from one hospital to those of another.

3.3. Experiment Setup

3.3.1. CNN Setup

For simple CNN model, we use model width (k) from 1 to 64 to cover all the 3 circumstances comparing with *effective model complexity* n. Number of max epochs is 1000 for CIFAR-10 and 100 for CAMELYON17-WILDS, batch size is 256 and the model is optimized with cross-entropy loss by Adam optimizer using a small learning rate of 0.0001.

3.3.2. DenseNet121

For DenseNet121, we use growth rate (k) from 1 to 32 to sweep all the 3 circumstances comparing with *effec-*

tive model complexity n. Number of max epochs is 100, batch size is 256 and the model is optimized with cross-entropy loss by Adam optimizer using a small learning rate of 0.0001.

3.3.3. Label Noise

Previous studies have found adding label noise can induce double descent in training dynamics (Nakkiran et al., 2021). To test whether this is still true in distribution shift dataset, we add label noise with probability p to the training data. Practically this is done by randomly sampling p point from all training data, and change them to a uniformly random wrong label.

4. Results and Discussion

4.1. Replication of Double Descent in CIFAR-10

Following the setup of Nakkiran et al. (2021), the double descent phenomenon can be clearly seen in the training dynamics illustrated in Figures 1 (left) and 2. Unlike Nakkiran et al. (2021), we perform only one trial for each data point instead of five due to constraint on computation resource. Still, we observe the phenomenon despite the presence of significantly more noise in the training statistics. Epoch-wised statistics in Figures 1 (right) also confirm the existence of double descent, with the intermediate-sized model exhibits traditional U-shape curve as demonstrated in Nakkiran et al. (2021). The effect of double descent is robust for simple CNN model on CIFAR-10 dataset when label noise is presented.

4.2. DenseNet121 on CAMELYON17-WILDS Without Label Noise and Data Augmentation

We first conducted an experiment using DenseNet121 (Huang et al., 2017) in accordance with the baseline used in CAMELYON17-WILDS benchmark (Koh et al., 2021), without label noise nor data augmentation. Each model is trained for 30 epochs to guarantee that the model is overfitting and interpolating all training data.

As seen in Figure 3, which shows worst group error curve (average error curve can be found in Appendix Figure 10 and shows similar trend), DenseNet121 can obtain near-perfect accuracy on the training set after one training epoch, likely due to the lack of sophistication of the prediction task (binary classification) and the lack of regularization in training scheme. Test error increases with model complexity as predicted by the classical statistics, indicating the presence of overfitting and reduced generalization. Interesting, the model error in out-of-distribution validation set diverges from that of the test set, even though both are constituted by our-of-distribution samples. Double descent is not observed in this experiment (see Appendix Figure 11).

4.3. DenseNet121 on CAMELYON17-WILDS With Label Noise

In this experiment, we add 20% label noise to the training data. In practice, this is achieved by flipping the labels of 20% of the randomly chosen training data. We extend the training time from 30 to 100 epochs in order to guarantee the model reaching the interpolation threshold. No data augmentation is used for this experiment.

We found that adding 20% label noise to the training data effectively stop the model from generalizing to the evaluation sets, whether it is in- or out-of-distribution. As shown in Figure 4, the training error decreases as model complexity increases, but in the meantime, all evaluation errors (ID/OOD validation and test error) increase and eventually cap at 0.5 when model complexity reaches the interpolation threshold (at around the complexity of 10 in the figure).

Our findings regarding the effect of label noise differ from the account of Nakkiran et al. (2021), in which the authors report no severe adverse effect on evaluation statistics when adding label noise to the CIFAR-10 dataset. We hypothesize that this divergence is due to the difference in the nature of the prediction tasks of CAMELYON17-WILDS and CIFAR-10; the former is binary classification while the latter multi-class classification. For binary classification, a small amount of label noise might be more than enough to irreversibly corrupt the quality of training data and lead to ineffective generalization.

4.4. DenseNet121 on CAMELYON17-WILDS With Data Augmentation

In this experiment, we apply special data augmentation called RandAugment (Sohn et al., 2020) as implemented by Zhang et al. (2021), matching the conditions of baseline model used in Sagawa et al. (2022).

We also test two training algorithms commonly used on datasets with distribution shifts. One is Empirical Risk Minimization (ERM) algotithm, which minimizes averaged labeled loss. The other one is a newly invented domain invariant learning method called Group DRO (Hu et al., 2018), which minimizes the worst-case domain loss to encourage the model to be domain-invariant.

4.4.1. ERM Algorithm

With data augmentation and ERM, the complexity-performance curve of over-parameterized models seems to defy what is predicted by the classical theory of statistics. The test error, as illustrated in Figure 5 and Figure 6 (left), first goes up then goes down as model complexity increases.

Figure 1. Left: Test error of CNNs with varying width (complexity) parameters in CIFAR-10 dataset with 20% label noise. Right: CNNs of three different sizes and their epoch-wised training dynamics.

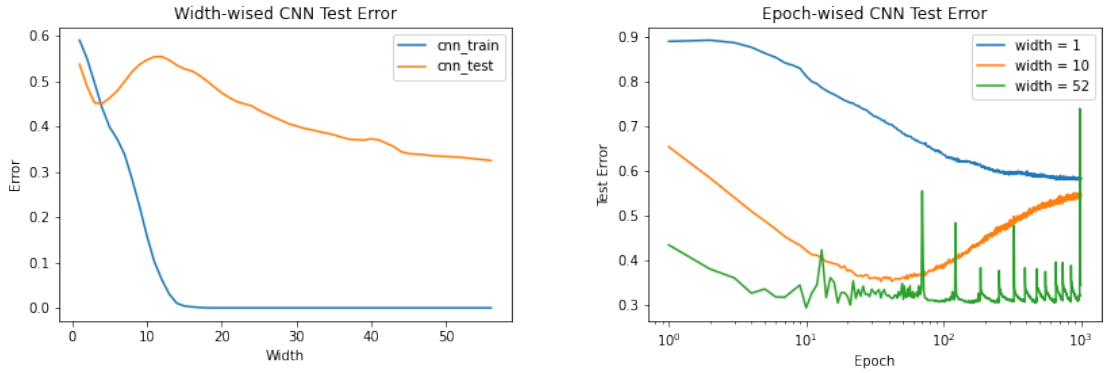
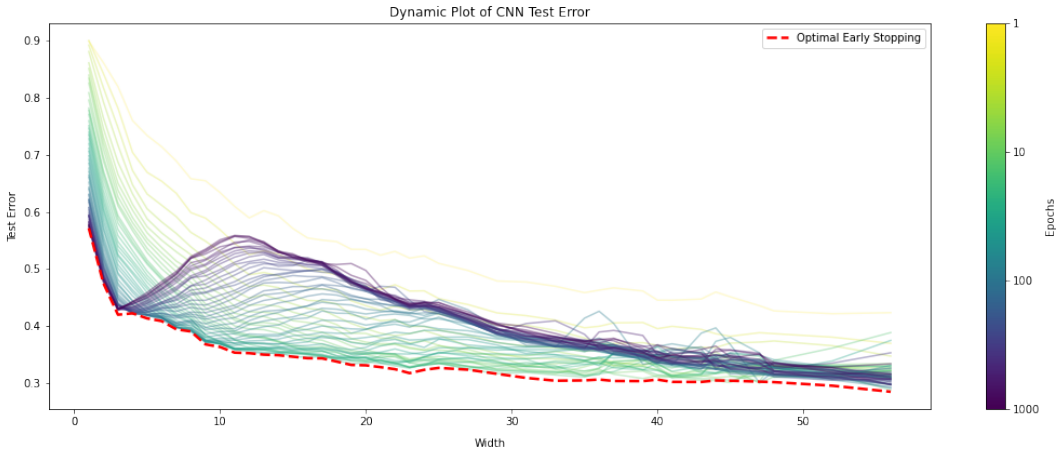


Figure 2. Training dynamics of CNNs with varying width (complexity) parameters in CIFAR-10 dataset with 20% label noise.



4.4.2. Group DRO: a Distributionally Robust Algorithm

Similar to the case of ERM, over-parameterized models optimized using Group DRO algorithm also defy the classical statistics and show an inverse U-shape complexity-performance curve in Figure 7 and Figure 6 (right). Our experiment also corroborates the findings of Koh et al. (2021), which show that Group DRO algorithm offers no performance boost on CAMELYON17-WILDS dataset compared to ERM.

4.4.3. Model Selection with Inverse U-shape Complexity-Performance Curve

The theory of bias-variance trade off predicts that complexity-performance curve is roughly U-shape, which leads to a strategy of model selection that is analogous to the Goldilock Principle; that is, the best model should be of a proper intermediate size—not too simple, which can lead to high bias, and not too complex, which can lead to high variance.

In this experiment, however, we demonstrate that model fitting, likely when coupled with some regularization techniques like data augmentation, can follows Inverse Goldilock Principle in the over-parameterized scheme. Namely, selecting the most simple model that has the best cross-validated evaluative score, or selecting the most complex model computationally affordable, might be a better strategy. Figure 5 (right) and Figure 7 (right) both show that the intermediate models (model complexity at 5 and 10 respectively) perform worse than the simple and complex ones as models enter the over-parameterized regime via extended epochs of training.

It has been commonly observed that an intermediate-sized deep learning model is not always more desirable than a complex one, especially in the the heavily over-parameterized regime (Adlam & Pennington, 2020). We hypothesize that an intermediate-sized model is more prone to *overthink* compared to a simple model, as the latter likely

Figure 3. Worst group errors of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *without label noise or data augmentation*.

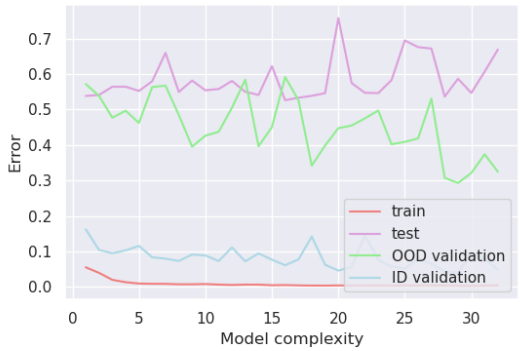
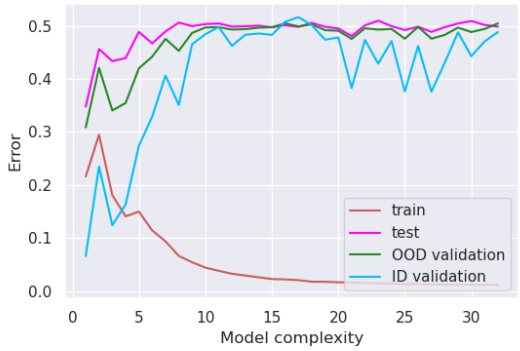


Figure 4. Average errors of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *with label noise*.



induces an a priori straightforward, low-variance solution, according to the framework of bias-variance trade-off. An intermediate-sized model can also *underthink* compared to a more complex model, as the latter is likely more flexible and allow the optimization strategy to find more useful inductive bias in the over-parameterized scheme. This "underthinking" account might be linked to the hypothesis proffered by Belkin et al. (2019) that attempts to reconcile double descent phenomenon in the framework of bias-variance trade-off.

4.5. Simple CNN on CAMELYON17-WILDS With Data Augmentation

In this experiment we replace DenseNet121 with a plain and simple 6-layer CNN to test the impact of model architecture on learning useful inductive bias under distribution shifts and over-parameterization.

As shown in Figure 8, the average errors for all splits of CAMELYON17-WILDS dataset continue to descend as model complexity increases. The trend of descending seems to be able to sustain beyond the model interpolation point at around $complexity = 20$, where the train error get extremely close to zero.

The CNN model also achieves much better performance compared to DenseNet121 in any experiment setting, with CNN consistently reaching lower test and OOD validation error than DensenNet121. Training dynamics in Figure 9 not only show the same descending trend, but also suggest that using OOD validation error to stop the CNN model training early (represented as the red dotted line in the figure) is an effective strategy to obtain the optimal test error for model at almost any level of complexity. In general, it seems that the inductive bias introduced by a simple and over-parameterized CNN model generalize better on and thus much better suited for CAMELYON17-WILDS dataset than that of DenseNet121.

Our findings suggest that DenseNet121 might not be the most optimal choice of baseline model for CAMELYON17-WILDS. As a result, many proposed improvements inadvertently present an exaggerated assessment of progress, when the actual improvement is much modest if we replace the baseline model with a simple 6-layer CNN. For example, Gao et al. (2022) claim to have achieved a massive 14.4% raw improvement ($77.7\% \rightarrow 92.1\%$) over the state of the art at the time. But as we demonstrated in Table 1, our model bridges this performance discrepancy with less parameters, significantly narrowing the gap to 2.6% between our proposed baseline and the state-of-the-art.

5. Conclusion

In summary, our contributions to the current work on double descent, generalization, and domain shifts are threefold.

Firstly, we replicated the phenomenon of double descent with a simple CNN trained and tested on CIFAR-10, suggesting that the phenomenon is robust and easy to replicate at the presence of label noise on this dataset.

Secondly, we explored the training dynamics of two CNN-style models on CAMELYON17-WILDS, a distribution-shift dataset for binary classification task. In our experiments, we did not observe any double descent phenomenon, suggesting that double descent might be hard to detect and easy to miss on this dataset.

Lastly, we also discovered that a simple, over-parameterized CNN has much better performance than DenseNet121 for CAMELYON17-WILDS dataset. We suggest to modify the current benchmark design for CAMELYON17-WILDS by adding simple CNN as one of the baseline models. In our experiment, this proposed change will greatly reduce the jump of improvement claimed by many recent works and result in a more realistic assessment of progress.

Figure 5. Left: Average errors of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *with data augmentation and ERM optimization*. Right: DenseNet121 of three different complexities and their epoch-wised training dynamics on the same setup.

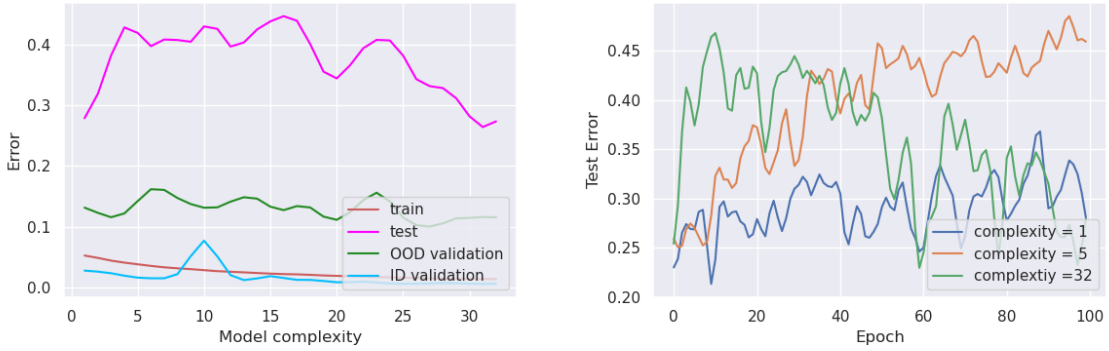
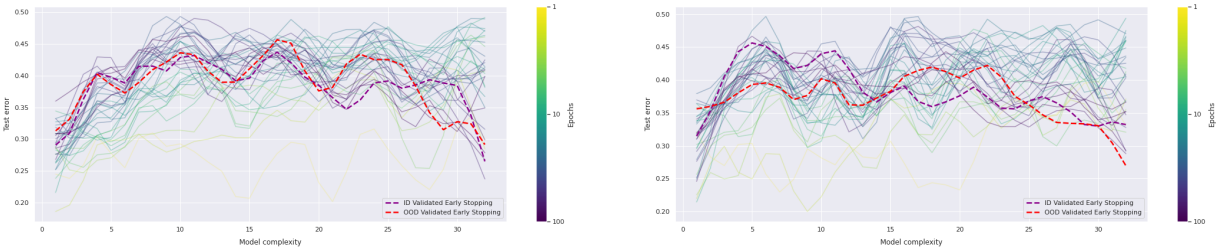


Figure 6. Left: Training dynamics of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *with data augmentation and ERM optimization*. Right: the same setup but with *Group DRO optimization*



References

Adlam, B. and Pennington, J. Understanding double descent requires a fine-grained bias-variance decomposition. *Advances in neural information processing systems*, 33: 11022–11032, 2020.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Bandi, P., Geessink, O., Manson, Q., Van Dijk, M., Balkenhol, M., Hermsen, M., Bejnordi, B. E., Lee, B., Paeng, K., Zhong, A., et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018.

Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pp. 541–549. PMLR, 2018.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Gao, I., Sagawa, S., Koh, P. W., Hashimoto, T., and Liang, P. Out-of-distribution robustness via targeted augmentations. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Hu, W., Niu, G., Sato, I., and Sugiyama, M. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pp. 2029–2037. PMLR, 2018.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Kleinberg, J. and Mullainathan, S. Simplicity creates inequity: implications for fairness, stereotypes, and interpretability. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 807–808, 2019.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

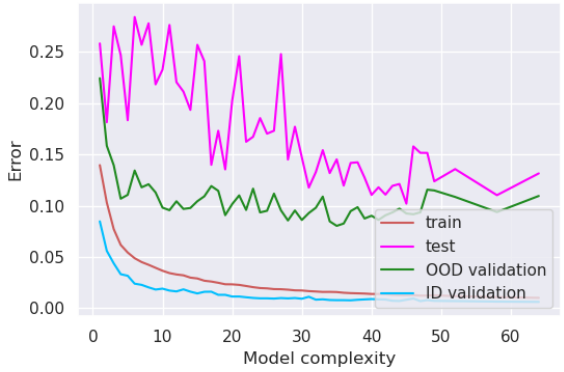
Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Figure 7. Left: Average errors of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *with data augmentation and Group DRO optimization*. Right: DenseNet121 of three different complexities and their epoch-wised training dynamics for the same setup.

Table 1. Comparison of model performance on CAMELYON17-WILDS. * indicates results are obtained from (Koh et al., 2021) or (Sagawa et al., 2022)

Algorithm	Model	Params	Validation Accuracy	Test Accuracy
ERM w/ targeted aug*	DenseNet121	8M	92.7	92.1
ERM w/ data aug*	DenseNet121	8M	90.6	82.0
Group DRO*	DenseNet121	8M	85.5	68.4
ERM w/ data aug	6-layer CNN (Ours)	6.3M	91.8	89.5

Figure 8. Average errors of *6-layer CNN* with varying complexities on CAMELYON17-WILDS dataset with data augmentation and ERM optimization.



Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Peng, X., Li, Y., and Saenko, K. Domain2vec: Domain embedding for unsupervised domain adaptation. In *European conference on computer vision*, pp. 756–774. Springer, 2020.

Robey, A., Pappas, G. J., and Hassani, H. Model-based domain generalization. *Advances in Neural Information Processing Systems*, 34:20210–20229, 2021.

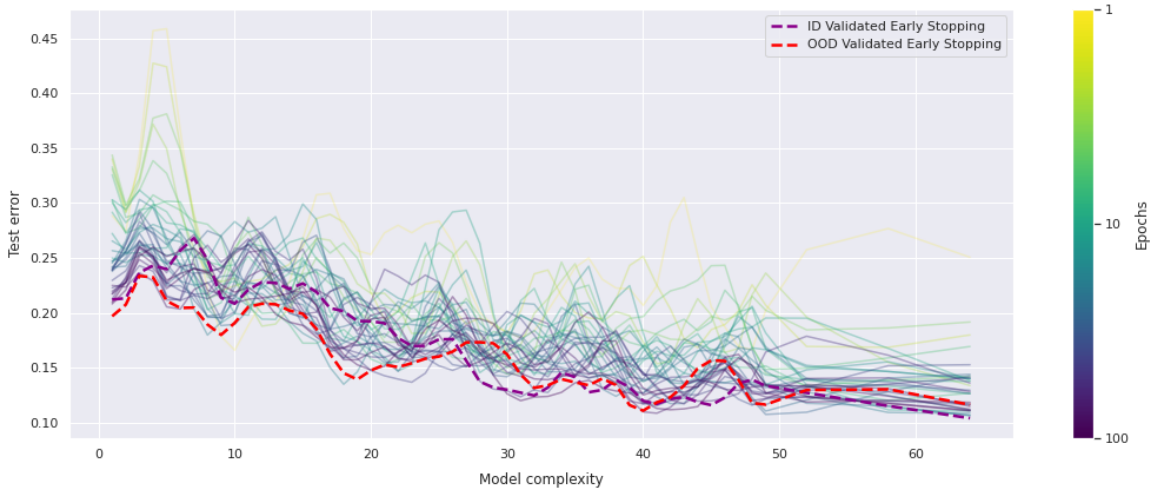
Sagawa, S., Koh, P. W., Lee, T., Gao, I., Xie, S. M., Shen, K., Kumar, A., Hu, W., Yasunaga, M., Marklund, H., Beery, S., David, E., Stavness, I., Guo, W., Leskovec, J., Saenko, K., Hashimoto, T., Levine, S., Finn, C., and Liang, P. Extending the wilds benchmark for unsupervised adaptation. In *International Conference on Learning Representations (ICLR)*, 2022.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Vallet, F., Cailton, J.-G., and Refregier, P. Linear and nonlinear extension of the pseudo-inverse solution for learning boolean functions. *EPL (Europhysics Letters)*, 9(4):315, 1989.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. At-

Figure 9. Training dynamics of *6-layer CNN* with varying complexities on CAMELYON17-WILDS dataset with data augmentation and ERM optimization.



tention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wyner, A. J., Olson, M., Bleich, J., and Mease, D. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18(1):1558–1590, 2017.

Zhang, Y., Zhang, H., Deng, B., Li, S., Jia, K., and Zhang, L. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*, 2021.

A. Supplementary Figures

Figure 10. Left: Average errors of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *without label noise or data augmentation*. Right: Epoch-wise training statistics of DenseNet121 in three different complexities on the same setup.

Figure 11. Dynamic Plot of of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *without label noise or data augmentation*.

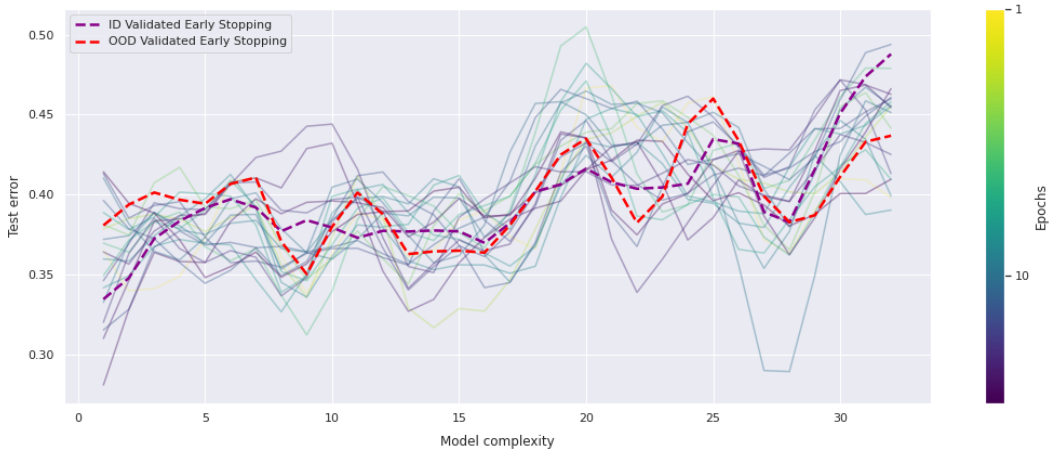


Figure 12. Left: Worst group errors of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *with label noise*. Right: Epoch-wise training statistics of DenseNet121 in three different complexities on the same setup.

Figure 13. Dynamic Plot of of DenseNet121 with varying complexities on CAMELYON17-WILDS dataset *with label noise*.

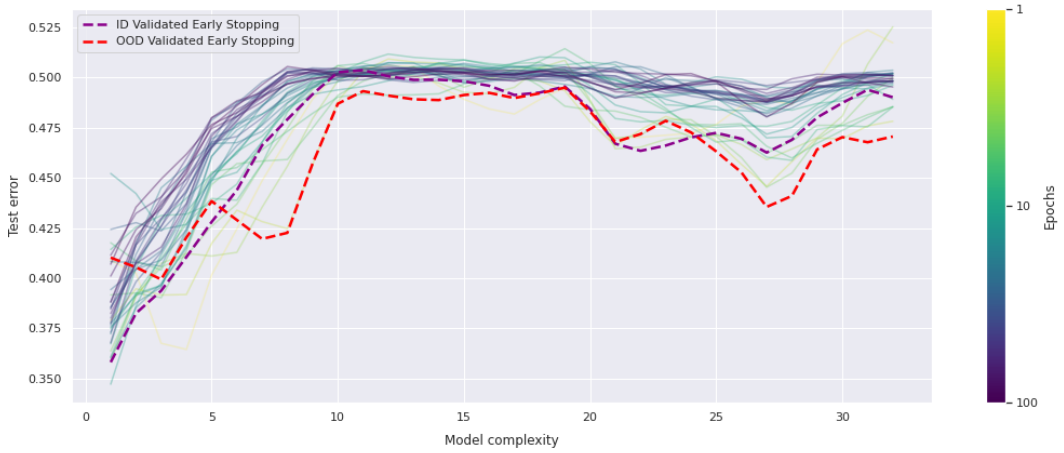


Figure 14. Worst group errors of DenseNet121 with varying complexities in CAMELYON17-WILDS dataset *with data augmentation and ERM optimization*.

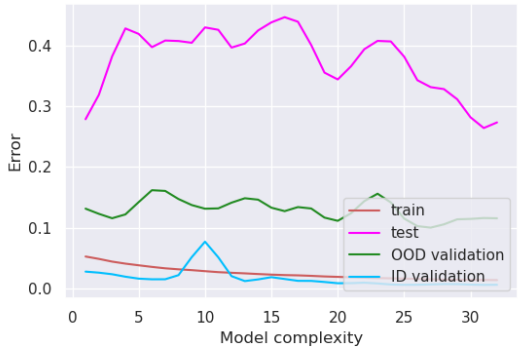


Figure 15. Worst group errors of DenseNet121 with varying complexities in CAMELYON17-WILDS dataset *with data augmentation and Group DRO optimization*.

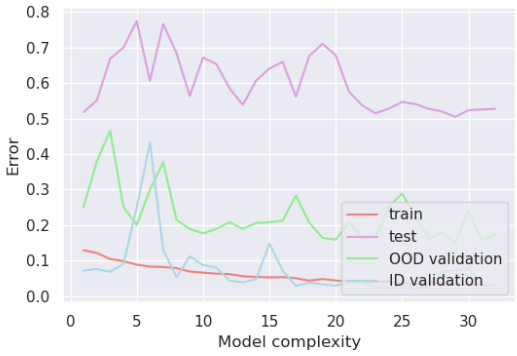


Figure 16. Left: Worst group errors of *6-layer CNN* with varying complexities on CAMELYON17-WILDS dataset with data augmentation and ERM optimization. Right: *6-layer CNN* of three different complexities and their epoch-wised training dynamics for the same setup.