Where Do Large Learning Rates Lead Us? A Feature Learning Perspective

Ildus Sadrtdinov[1], Maxim Kodryan[2], Eduard Pokonechny[3], Ekaterina Lobacheva[3]*, Dmitry Vetrov[1]*

[1]*Constructor University, Bremen* [2]*HSE University, Moscow* [3]*Independent researcher*

Correspondence to: isadrtdino@constructor.university

Abstract

It is a conventional wisdom that using large learning rates (LRs) early in training improves generalization. Following a line of research devoted to understanding this effect mechanistically, we conduct an empirical study in a controlled setting focusing on the feature learning properties of training with different initial LRs. We show that the range of initial LRs providing the best generalization of the final solution results in a sparse set of learned features, with a clear focus on those most relevant for the task. In contrast, training starting with too small LRs attempts to learn all features simultaneously, resulting in poor generalization. Conversely, using initial LRs that are too large fails to extract meaningful patterns from the data.

Keywords: learning rate, neural networks, feature learning

1. Introduction

In any gradient descent optimizer, the *learning rate* (LR) is probably the most important hyperparameter, especially in the context of deep learning [10]. LR controls the optimization step size and, due to extreme non-convexity of the loss function with a manifold of qualitatively different minima, it is primarily responsible for the type of solution we obtain after training [5, 6, 29, 34, 37, 47, 48, 51]. Using large learning rates, especially at the beginning of training, has become a common practice [18, 32, 52]. Starting with large LR values is known to help avoid poor local minima [12, 28, 33, 40] while the solutions obtained at the end of training often have favorable properties like good generalization and flatter loss landscape around the corresponding optima [20, 29, 43, 47]. Prior work has attempted to explain the benefits of high learning rates mostly from the optimization & loss landscape perspective [5, 9, 22–24, 45, 46, 48, 49]. While suggesting possible mechanisms for why large LR values lead to good solutions, these works still lack mechanistic characterization of these solutions as machine learning models.

Several studies have aimed to address this shortcoming. Some of them have found a tendency, when training with a large LR, to give preference to more sparse solutions w.r.t. network weights and/or activations [2, 6, 11]. For instance, Andriushchenko et al. [6] demonstrated that hovering at some constant loss level when training with large LRs helps optimization to eventually find modes with *sparse features* in terms of sparse activation patterns in hidden layers of deep neural networks. Another line of research examined pattern learning with small and large learning rates. Typically, in specific artificial settings, these works show that more complex patterns are learned

* Shared senior authorship.

© I. Sadrtdinov, M. Kodryan, E. Pokonechny, E. Lobacheva & D. Vetrov.

with smaller LRs, thus, to avoid overfitting, it is beneficial to start training with a large LR and then decay it to smaller values after learning all the "easy-to-fit" patterns from the data [34, 38, 51]. Recently, Rosenfeld and Risteski [44] suggested a possible mechanism for filtering out unreliable spurious features when training with large LRs via so-called "opposing signals" in the data.

Following this research direction, we provide further clarity on what features in the data the model captures after training with different initial LRs. We conduct a detailed empirical analysis in a special setting that allows for precise control of the LR value. Inspired by a recent work of Lobacheva et al. [37] who studied the generalization and geometrical properties of the final solutions obtained after training with different initial LRs, we structure our analysis around the three-regimes taxonomy of the LR values [29, 41] (Figure 1): 1) convergence for small LRs, 2) chaotic equilibrium for medium LRs, and 3) divergence for large LRs. Lobacheva et al. [37] found that the best choice is to start training in the second regime, with moderately high LRs that lead to neither convergence nor divergence, but only a relatively narrow portion of that range, defined as *subregime 2A*, provides consistent optimal results: training in this subregime locates a basin in the loss landscape with well-generalizing solutions, which can be easily obtained by fine-tuning with a small LR or weight averaging. We extend their study from a feature learning perspective. First, we introduce a synthetic example with interpretable features and find that as LR increases up to subregime 2A (including it), models become more specialized in the sense that they rely on fewer features, while further increase in LR gradually impairs the ability to extract any features from the data. Then, we show how our findings translate into a practical image classification setting via Fourier analysis of the inputs and additionally reveal that the model puts more attention on the most useful features for the task in subregime 2A.

2. Methodology

Our methodology is based on that of Lobacheva et al. [37]. They point out that in order to properly study the impact of the initial LR on the final solution, it is required to fix it at the beginning of training. However, this seemingly simple action turns out to be not so trivial due to scale invariance induced by normalization layers, ubiquitously used modern NN architectures, which makes the effective learning rate (ELR) of the model dependent on the parameters norm [5, 7, 29, 35, 36, 41]. In order to eliminate this effect, it was proposed to train normalized models in a fully scale-invariant (SI) way [5, 29, 37, 41] ensuring that fixing an LR leads to a fixed ELR as well. Following this, we make our models fully SI by fixing the last layer and removing trainable affine parameters of normalization layers, and we train them using projected SGD on a sphere of fixed radius. We consider a more conventional setting in Section 4.

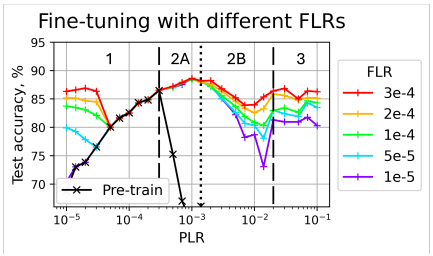

Figure 1: Three training regimes (with subregimes 2A/2B): test accuracy of the pre-trained (black) and fine-tuned solutions (colored) for SI ResNet-18 on CIFAR-10.

Lobacheva et al. [37] applied the same SI setup to study training with different initial LR values. The structure of their analysis is based on the results of Kodryan et al. [29] who investigated training of SI models on the sphere and showed that it typically happens in one of three regimes depending on the LR value: 1) *convergence*, when the optimization simply converges to a minimum, 2) *chaotic equilibrium*, when loss noisily stabilizes at some value, and 3) *divergence*, when a model remains

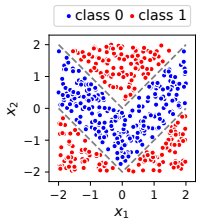

Figure 2: A single 2D "tick" synthetic feature.

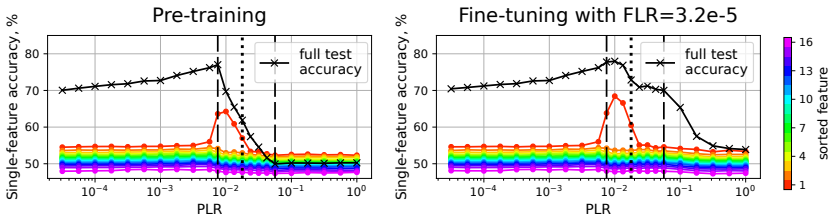

Figure 3: Feature sparsification in the synthetic example for pre-training (left), and fine-tuning (right): single-feature test accuracy (colored) and regular test accuracy (black).

at the random guess level. Lobacheva et al. [37] analyzed the points obtained after initial training in different regimes from the perspective of their utility for subsequent training with small LRs or weight averaging. We further analyze the final solutions in terms of their feature learning ability. Following Lobacheva et al. [37], we divide training into two stages. The first stage is called *pre-training*: we fix the LR, which at this stage we name **P**LR, and train a model for sufficient amount of epochs to ensure stabilization of training dynamics. Then, at the second stage, we either 1) change the learning rate and *fine-tune* the model, i.e., train with a small LR, or 2) continue training with the same PLR and weight-average consequent checkpoints alike stochastic weight averaging (SWA) [21].[1] Small LRs for fine-tuning, called **F**LRs, are taken from regime 1 to ensure model convergence. Further detail on the experimental setup are provided in Appendix A.

In Figure 1, we reproduce the main results of Lobacheva et al. [37] on fine-tuning from different pre-trained points. We can indeed see that the lower part of regime 2, called subregime **2A**, gives the best test accuracy after fine-tuning. For ease of comparison, we similarly divide all plots into three parts corresponding to the three pre-training regimes and also divide the second regime into two subregimes. Below we present a synthetic example with adjustable features and use it to examine feature learning in different regimes. Then, we shift to image classification with a SI ResNet-18 [18] trained on CIFAR-10 [31]. Finally, we demonstrate that our findings remain valid for conventionally trained networks as well. We consider other architectures and datasets in the appendix.

3. Feature learning perspective

Synthetic example We begin to study feature learning in models trained with different initial LRs by introducing a synthetic example with precise control over how features affect the target variable. We consider a binary classification setting with the following two conditions: 1) all three training regimes are present, and 2) the data points contain multiple features, each of which is sufficient to classify the data correctly. The first condition allows for comparison to prior work on training NNs on the sphere [5, 29, 37, 41], while the second one allows for maximally pure feature learning study when all features are equally useful for the task. We use 32-dimensional data vectors, where each pair of coordinates represents a single 2D "tick" feature (Figure 2), all 16 features are sampled from the same distribution. We use a 3-layer MLP with ReLU activation and Layer Norm [8] and make it fully SI. We put further detail in Appendix A. The generalization and geometry properties in this setup are similar to those described in Lobacheva et al. [37], see Appendix C.

1. We have placed the SWA results in the appendix, and in the main text we present only the fine-tuning results.

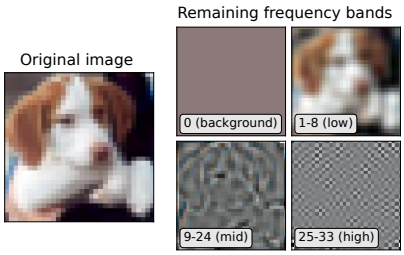

Figure 4: Inverse 2D DFT of four spectrum components.

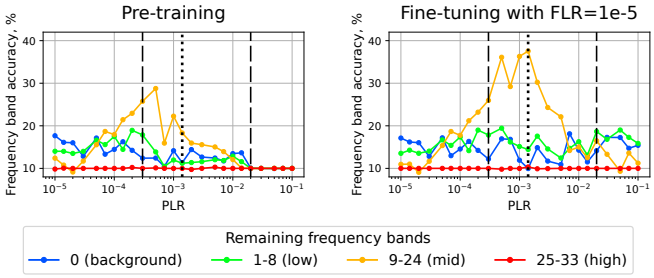

Figure 5: Frequency bands accuracy for pre-training and fine-tuning. SI ResNet-18 on CIFAR-10.

To quantify how features are learned with different LRs, we generate 16 single-feature test sets, each having only one of the 16 total features. The values of other features are distributed along the decision boundary (gray dashed line in Figure 2) to represent missing features. We measure the accuracy of the trained models on these single-feature test sets and relate the obtained values to the importance of the respective features for the model: the higher the accuracy value, the more important the feature. We analyze these values in sorted order for each PLR because models may favor different features over different training runs due to the randomness in initialization [3].

The results are depicted in Figure 3. For small PLRs, there is no feature selection: we observe similar accuracy w.r.t. different features resulting in poor overall generalization, since no target-predicting feature is reliably learned by the model. Closer to the boundary between regimes 1 and 2 we begin to observe a kind of model specialization: the accuracy corresponding to a single feature is significantly higher than for the rest. Moreover, such feature sparsity persists after fine-tuning, indicating completely different feature learning behavior than training with the same FLR from scratch: even in the setting of equally useful features, *the model prefers to focus more on some subset of features instead of trying to learn all features at once*. The sparsity peak in subregime 2A coincides with the peak of the fine-tuning test accuracy (Figure 8), confirming that a sparser set of learned features improves model generalization. This may also be related to the fact that the basin is determined exactly at this range of LRs (see discussion in Appendix B). When the PLR is further increased, the ability to learn any useful patterns from the data is reduced, which manifests itself in degraded accuracy on both regular and single-feature test sets.

Fourier features A similar feature selection effect can be observed in image classification setting. Since for the real-world image data it is generally not clear how to define features [44], we use frequency bands of the 2D Discrete Fourier Transform (DFT) as proxy for features [1]. We divide the full 2D spectrum of an image into 4 components, each consisting of a range of frequency bands: 0 (constant background color), 1-8 (low), 9-24 (mid), and 25-32 (high). For each of the 4 components, we zero out the rest of the spectrum and apply the inverse DFT to obtain images with only one frequency component preserved (Figure 4) by analogy with the single-feature test sets in the synthetic example. We repeat this procedure with every test image and measure the accuracy on the resulting 4 new test sets corresponding to the spectrum components. A more detailed description of the setup can be found in Appendix A. Also, see Appendix D for additional results.

Figure 5 shows the test accuracy w.r.t. each spectrum component after pre-training with different PLRs. As in the synthetic example, small PLRs of regime 1 tend to treat all features approximately equally, paying slightly more attention to the background color and low-frequency features,

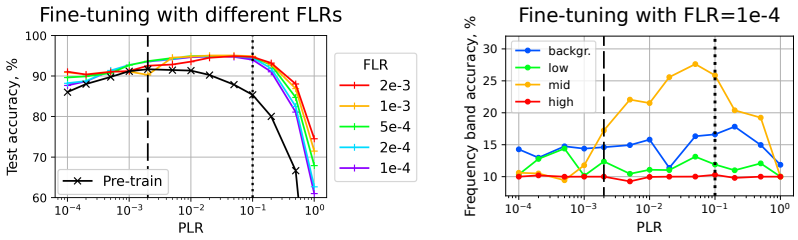

Figure 6: Practical ResNet-18 on CIFAR-10: standard test accuracy of the pre-trained and fine-tuned solutions (left) and accuracy w.r.t. different frequency bands for fine-tuning (right).

while increasing PLR introduces feature sparsity, making the mid-frequency features significantly more important. The peak of the mid-frequency accuracy is achieved in subregime 2A, reaching much higher absolute values than the 0 and 1-8 components in the first regime. Moreover, after fine-tuning, the mid-frequency accuracy improves even further, showing the same bias towards a subset of features as in the synthetic example. A further increase in the PLR reduces the feature sparsity and the mid-frequencies importance. The mid-frequencies are known to play a key role in model generalization and robustness in image classification [1, 50], so their apparent prevalence in subregime 2A indicates that *feature selection is not random but is biased towards the most useful features for the task*. Interestingly, prior work assumed that training with large LRs must prioritize "easy-to-fit hard-to-generalize" features [34, 51] but our results suggest that the model may favor more complex features if they are more helpful in predicting the target.

4. Practical setting

In this section, we validate our results in a more conventional training setting. We train a common ResNet-18 model without the sphere constraint using SGD with momentum, weight decay, and data augmentation; the only deviation from the standard setup is a different LR schedule. As in Lobacheva et al. [37], we can also see only regimes 1 and 2, since regime 3 is too unstable, and draw the boundary between the regimes approximately at the PLR with max pre-train accuracy.

The frequency bands accuracies in Figure 6 (right) show a similar trend as described in Section 3: in subregime 2A, the network captures significantly more mid-frequencies from the inputs than other components, while no similar specification is observed for other PLR ranges. Thus, our main claims remain valid in the practical setting. In Appendices E, F, we provide more practical results.

5. Conclusion

In this work, we studied the influence of different initial LRs on the feature learning properties of the final solution. We discover that using the initial LRs providing the best final generalization leads to a sparse set of the most useful learned features. Using other LR values may lead to suboptimal results: either attemptimg to capture all relevant features at once with smaller LRs or degraded feature learning ability with large LRs. We conduct main experiments in a special setup allowing for more accurate control of the learning rate and validate our key results in a practical setting.

Acknowledgments

The related work analysis and text composition in sections 1 and 2 were done by Maxim Kodryan with the support of the grant for research centers in the field of AI provided by the Analytical Center for the Government of the Russian Federation (ACRF) in accordance with the agreement on the provision of subsidies (identifier of the agreement 000000D730321P5Q0002) and the agreement with HSE University №70-2021-00139. The empirical results were supported in part through the computational resources of HPC facilities at HSE University [30]. Part of the experiments were conducted using the Constructor Research Platform [13].

References

[1] Antonio A. Abello, Roberto Hirata, and Zhangyang Wang. Dissecting the high-frequency bias in convolutional neural networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 863–871, 2021.

[2] Kwangjun Ahn, Sébastien Bubeck, Sinho Chewi, Yin Tat Lee, Felipe Suarez, and Yi Zhang. Learning threshold neurons via edge of stability. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In *International Conference on Machine Learning*, pages 840–902. PMLR, 2023.

[5] Maksym Andriushchenko, Francesco D'Angelo, Aditya Varre, and Nicolas Flammarion. Why do we need weight decay in modern deep learning? In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

[6] Maksym Andriushchenko, Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. Sgd with large step sizes learns sparse features. In *International Conference on Machine Learning*, pages 903–925. PMLR, 2023.

[7] Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkxQ-nA9FX.

[8] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[9] David Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.

[10] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade: Second edition*, pages 437–478. Springer, 2012.

[11] Feng Chen, Daniel Kunin, Atsushi Yamamura, and Surya Ganguli. Stochastic collapse: How gradient noise attracts SGD dynamics towards simpler subnetworks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[12] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

[13] Constructor Research Platform. URL https://constructor.tech/products/research-platform.

[14] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d85b63ef0ccb114d0a3bb7b7d808028f-Paper.pdf.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[16] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

[17] Elisenda Grigsby, Kathryn Lindsey, and David Rolnick. Hidden symmetries of relu networks. In *International Conference on Machine Learning (ICML)*, pages 11734–11760. PMLR, 2023.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Conference on Computer Vision and Pattern Recognition*, 2016.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[20] Nikhil Iyer, V Thejas, Nipun Kwatra, Ramachandran Ramjee, and Muthian Sivathanu. Wide-minima density hypothesis and the explore-exploit learning rate schedule. *arXiv preprint arXiv:2003.03977*, 2020.

[21] P Izmailov, AG Wilson, D Podoprikhin, D Vetrov, and T Garipov. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 876–885, 2018.

[22] Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

[23] Stanislaw Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amost Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019.

[24] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020.

[25] Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. On the maximum hessian eigenvalue and generalization. In *Proceedings on "I Can't Believe It's Not Better! - Understanding Deep Learning Through Empirical Falsification" at NeurIPS 2022 Workshops*, pages 51–65. PMLR, 2023.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[27] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Machine Learning*, 2022.

[28] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *International conference on machine learning*, pages 2698–2707. PMLR, 2018.

[29] Maxim Kodryan, Ekaterina Lobacheva, Maksim Nakhodnov, and Dmitry P Vetrov. Training scale-invariant neural networks on the sphere can happen in three regimes. *Advances in Neural Information Processing Systems*, 35:14058–14070, 2022.

[30] P. S. Kostenetskiy, R. A. Chulkevich, and V. I. Kozyrev. HPC resources of the higher school of economics. *Journal of Physics: Conference Series*, 1740:012050, 2021. doi: 10.1088/1742-6596/1740/1/012050. URL https://doi.org/10.1088/1742-6596/1740/1/012050.

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[33] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[34] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[35] Zhiyuan Li, Kaifeng Lyu, and Sanjeev Arora. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. *Advances in Neural Information Processing Systems*, 33, 2020.

[36] Ekaterina Lobacheva, Maxim Kodryan, Nadezhda Chirkova, Andrey Malinin, and Dmitry P Vetrov. On the periodic behavior of neural network training with batch normalization and weight decay. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21545–21556. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/b433da1b32b5ca96c0ba7fcb9edba97d-Paper.pdf.

[37] Ekaterina Lobacheva, Eduard Pokonechny, Maxim Kodryan, and Dmitry Vetrov. Large learning rates improve generalization: But how large are we talking about? In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

[38] Miao Lu, Beining Wu, Xiaodong Yang, and Difan Zou. Benign oscillation of stochastic gradient descent with large learning rates. *arXiv preprint arXiv:2310.17074*, 2023.

[39] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[40] Amirkeivan Mohtashami, Martin Jaggi, and Sebastian U Stich. Special properties of gradient descent with large learning rates. In *International Conference on Machine Learning*, pages 25082–25104. PMLR, 2023.

[41] Maksim Sergeevich Nakhodnov, Maksim Stanislavovich Kodryan, Ekaterina Maksimovna Lobacheva, and Dmitrii S Vetrov. Loss function dynamics and landscape for deep neural networks trained with quadratic loss. In *Doklady Mathematics*, volume 106, pages S43–S62. Springer, 2022.

[42] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2021.

[43] Yinuo Ren, Chao Ma, and Lexing Ying. Understanding the generalization benefits of late learning rate decay. In *International Conference on Artificial Intelligence and Statistics*, 2024.

[44] Elan Rosenfeld and Andrej Risteski. Outliers with opposing signals have an outsized effect on neural network optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

[45] Bin Shi, Weijie Su, and Michael I Jordan. On learning rates and schrödinger operators. *Journal of Machine Learning Research*, 24(379):1–53, 2023.

[46] Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021.

[47] Yuqing Wang, Minshuo Chen, Tuo Zhao, and Molei Tao. Large learning rate tames homogeneity: Convergence and balancing effect. In *International Conference on Learning Representations*, 2022.

[48] Lei Wu, Chao Ma, et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31, 2018.

[49] Ning Yang, Chao Tang, and Yuhai Tu. Stochastic gradient descent introduces an effective landscape-dependent regularization favoring flat solutions. *Physical Review Letters*, 130(23): 237101, 2023.

[50] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[51] Kaichao You, Mingsheng Long, Jianmin Wang, and Michael I Jordan. How does learning rate decay help modern neural networks? *arXiv preprint arXiv:1908.01878*, 2019.

[52] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

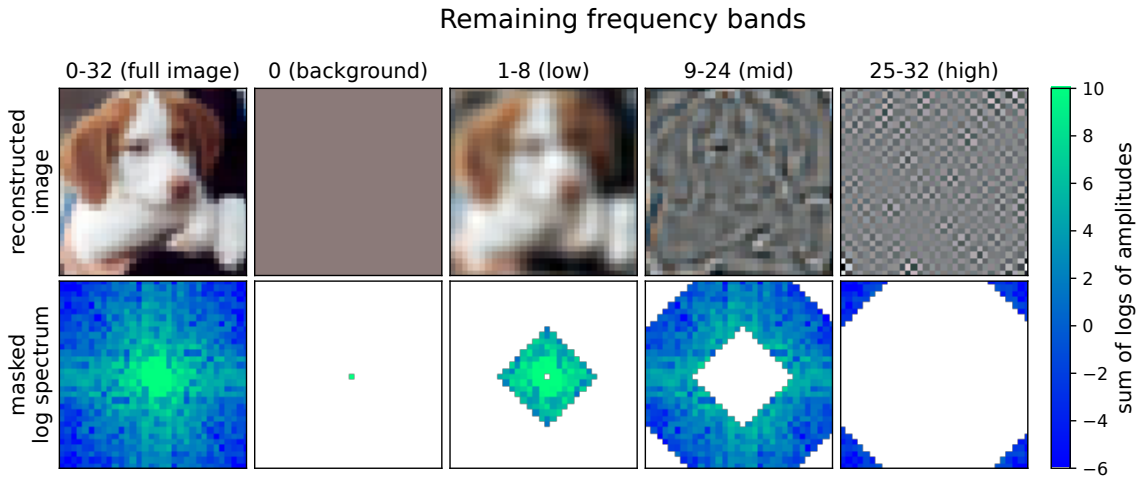

Figure 7: Inverse 2D DFT images (top) and corresponding masked spectra (bottom). When visualizing the *low*, *mid*, and *high* images, we scale each channel to the range 0-1. For the spectra, we plot the logarithm of the absolute values of the amplitudes ($\log |Y[k, l]|$), summed over 3 color channels.

Appendix A. Experimental setup details

Code Our implementation, including the used scale-invariant architectures and training on the sphere, is based on the open-source code of Kodryan et al. [29]: https://github.com/tipt0p/three_regimes_on_the_sphere.

Compute resources We use NVIDIA TESLA V100 and A100 GPUs for computations in our experiments. The total amount of compute spent on all experiments is approximately 1500-2000 GPU hours, while the experiments included in the paper took $\sim$ 1000 GPU hours.

Datasets and architectures Following Lobacheva et al. [37], we conduct experiments with two network architectures, a simple 3-layer convolutional neural network with Batch Normalization layers [19] (ConvNet) and a ResNet-18, on CIFAR-10 and CIFAR-100 datasets. In the scale-invariant setup, we use ResNet models with width factor $k = 32$, and ConvNet models with width factor $k = 32$ and 128 for CIFAR-10 and CIFAR-100, respectively. In the practical setup, we use ResNet with a standard width factor $k = 64$.

Pre-training, fine-tuning, and SWA We train all networks using SGD with a batch size of 128. Both the pre-training and the fine-tuning stages take 200 epochs. This training time is sufficient to either reach a minimum or stabilize the loss in the pre-training stage and to achieve complete convergence in the fine-tuning stage even with the smallest FLR. Fine-tuning is always done with the first regime FLRs to ensure convergence to a minimum. When performing SWA of N models, we continue training for $N - 1$ more epochs with the same PLR and average checkpoints from epochs $200, \ldots, 200 + N - 1$. In the practical setting in Section 4, we use weight decay of $5 \cdot 10^{-4}$, momentum of 0.9, and standard CIFAR augmentations: random crops (size: 32, padding: 4), random horizontal flips, and per-channel normalization.

Synthetic example We use a 3-layer MLP with ReLU activation and Layer Normalization [8] after the first and the second linear layers to make the network scale-invariant. Additionally, we

freeze the final linear layer and set its norm to 10. The size of hidden layers is 32. The trainable weights are initialized with the standard normal distribution and projected to the unit sphere. We take 512 training and 2000 testing samples. We use SGD with batch size 32. Pre-training and fine-tuning stages take 40000 and 20000 iterations, respectively. We consider 10 data sampling seeds and 5 model initialization + SGD batch order seeds, so a total of 50 training runs is done.

Fourier features We use 2D Discrete Fourier Transform (DFT) and frequency masking to create images similar to the single-feature samples in the synthetic example. Let $X \in \mathbb{R}^{N \times M}$ denote an image in the spatial domain (for CIFAR-10/CIFAR-100 we have $N = M = 32$). 2D DFT is defined as:

$$Y[k,l] = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} X[n,m] e^{-2\pi i (\frac{kn}{N} + \frac{lm}{M})}, \quad \begin{cases} -\lfloor \frac{N}{2} \rfloor \le k \le \lceil \frac{N}{2} \rceil - 1 \\ -\lfloor \frac{M}{2} \rfloor \le l \le \lceil \frac{M}{2} \rceil - 1 \end{cases}$$

In our case, $-16 \le k, l \le 15$. The frequency band b is defined as $\{(k,l) : |k| + |l| = b\}$, corresponding to a diamond around the spectrum center $k = l = 0$. This band matches the Fourier basis vectors, which have b oscillations (b "black and white" stripes) rotated at different angles. Then, we apply frequency masking, preserving a range of frequency bands, $a \le b \le c$:

$$Y_{a\text{-}c}[k,l] = \begin{cases} Y[k,l], & a \le |k| + |l| \le c \\ 0, & \text{otherwise} \end{cases}$$

Finally, we use the inverse 2D DFT and obtain the resulting frequency band images:

$$X_{a\text{-}c}[m,n] = \sum_{k=-\lfloor N/2 \rfloor}^{\lceil N/2 \rceil - 1} \sum_{l=-\lfloor M/2 \rfloor}^{\lceil M/2 \rceil - 1} Y_{a\text{-}c}[k,l] e^{2\pi i (\frac{kn}{N} + \frac{lm}{M})}$$

We use 0-0 (constant background color), 1-8 (low), 9-24 (mid), and 25-32 (high) groups of frequency bands. Each of the RGB color channels is processed independently. An example of the application of the described procedure is shown in Figure 7. We omit the per-channel image normalization used during training when evaluating accuracy on low, mid, and high samples (since removing the 0 band centers the resulting images). However, it is still used for the 0-0 sample.

Appendix B. Discussion and future research directions

Loss landscape and feature learning Among other things, Lobacheva et al. [37] have shown that the best LRs for pre-training identify a basin with good solutions, which can be reached via fine-tuning or weight averaging. Complementary to these results, we have found that these LRs also sparsify the learned features, focusing on the most useful ones. A very intriguing question is how exactly are these observations related? Based on the results concerning subregime 2A, it can be assumed that learning some subset of features corresponds to localizing a certain region in the loss landscape, all solutions within which rely to a greater extent on these learned features. In this regard, what features were learned during the pre-training stage can determine the quality of the localized basin of solutions. This conjecture gives rise to a number of very nontrivial but interesting questions. For instance, can we somehow connect the properties of the learned features with certain characteristics of the basin and minima within it, e.g., some notion of sharpness? According to Kodryan et al. [29], the solutions obtained with higher PLRs of the first regime have both better

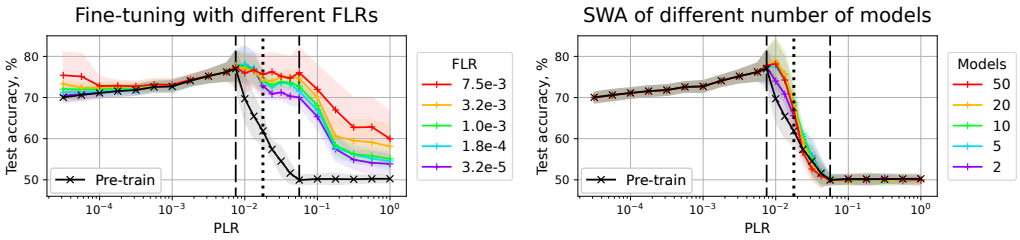

Figure 8: Syntetic example. Test accuracy of different fine-tuned (left) and SWA (right) solutions. Test accuracy after pre-training is depicted with the black line. Dashed lines denote boundaries between the first and second pre-training regimes, dotted line divides the second regime into two subregimes. Mean and standard deviation over 50 seeds are shown.

generalization and lower sharpness, however this trend is more complex for the fine-tuned solutions in regime 2 (see Appendix G). Next, how exactly does feature sparsity affect the properties of the found basin: does it only pre-define a specific set of shared features or can it act as some sort of a regularizer for the solutions inside the basin, e.g., by leading to simpler models [6, 11], which in fact can be represented by smaller networks? The study of the described issues opens an important direction for future research of the neural networks training process, as it may draw links between the optimization perspective and the final model properties.

More complex scenarios We have shown that the choice of the initial LR can lead to different feature learning behavior. In our setting, sparse features and mid-frequency bias were associated with optimal generalization. However, in practice this may not always be the case. For example, some features useful for the training data classification can be spurious for the test data [27]. Or memorization, in general affecting a very small subset of data, is less likely to happen when training with large LRs, while some works show that memorization in neural networks can be useful [16]. Thus, the properties of training with different LRs in more complex practical scenarios with spurious features/benign memorization may lead to more complex relationships between LRs and generalization. We believe this direction is significant for future research to more broadly understand the impact of learning rate on trained models.

Limitations We wish to highlight several limitations of our work. First, our study is primarily empirical in nature; our conclusions do not have direct theoretical support (perhaps only indirect via related work partly mentioned in Section 1). Second, we are limited to a specific setup involving particular datasets and NN architectures and, generally speaking, cannot guarantee that all of our findings will consistently generalize to other settings. Third, although we account for the impact of scale invariance on LR in our main experiments, we may overlook similar effects of other NN invariances, like rescale invariance of homogeneous activations (e.g., ReLU) [17, 39]. Addressing these and other possible limitations is future work.

Appendix C. Additional result for the synthetic example

In this section, we provide additional results for the synthetic example. Figure 8 shows the test accuracy of the fine-tuning with different FLRs and SWA. The overall behavior for different PLRs is similar to that reported by Lobacheva et al. [37], all 3 regimes are clearly observed. In regime 1,

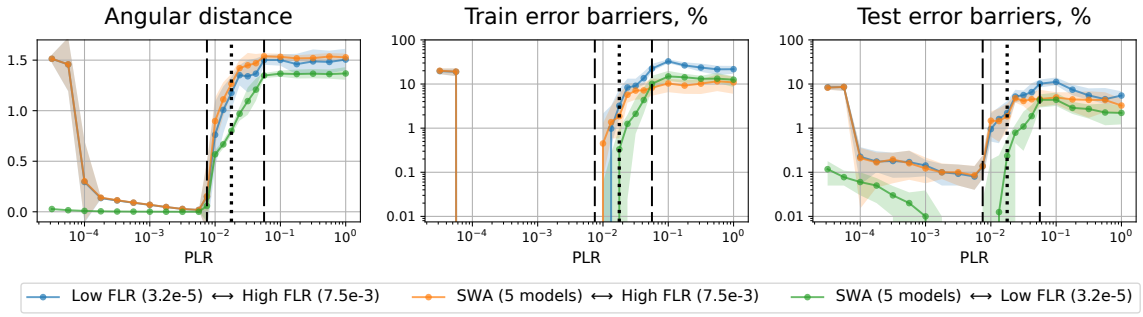

Figure 9: Geometry in the synthetic example between the points fine-tuned with the smallest and the largest FLRs and SWA. Results are aggregated over 50 seeds. For angular distance, mean and standard deviation are shown; for train and test barriers, median and $0.25 - 0.75$ quantile range are plotted.

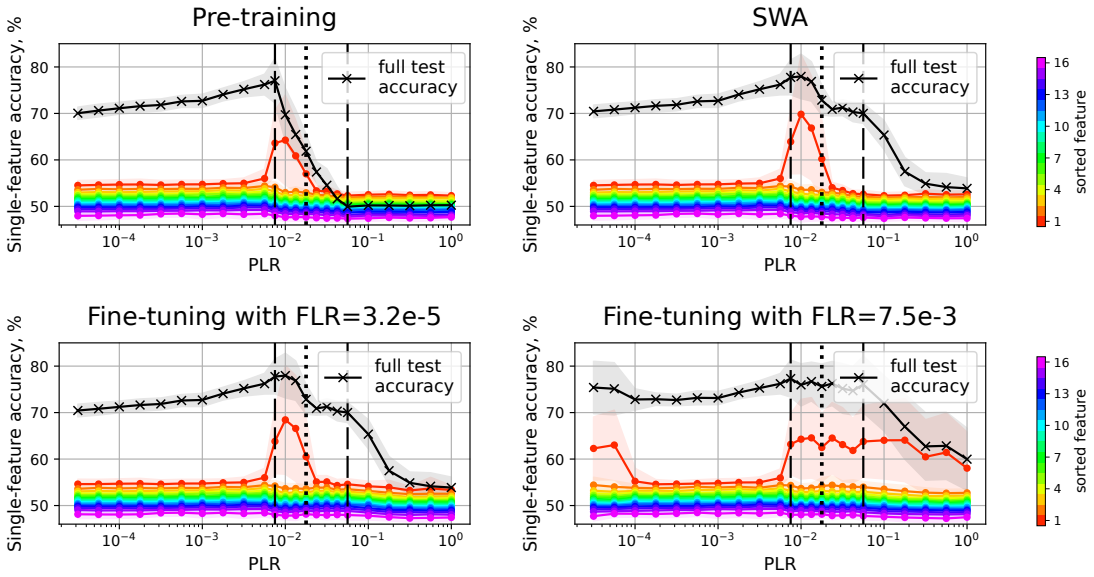

Figure 10: Feature sparsification in the synthetic example for pre-training (top, left), SWA (over 5 models; top, right), and fine-tuning with low FLR $= 3.2 \cdot 10^{-5}$ (bottom, left) and high FLR $= 7.5 \cdot 10^{-5}$ (bottom, right). Mean and standard deviation over 50 seeds are shown.

the pre-training quality is monotonically increased with PLR. Subsequent fine-tuning with smaller, equal, or slightly larger FLRs leads to the same quality, while a significantly larger FLR improves test accuracy. Fine-tuning in subregime 2A gives a slight improvement over training models in regime 1, and all FLRs have the same optimal quality (except for higher FLRs, which experience overfitting given a fixed number of fine-tuning iterations). Fine-tuning with different FLRs in subregime 2B leads to solutions with different accuracy, which is lower compared to 2A. Finally, SWA in subregime 2A produces models with good performance, while averaging models in subregime 2B does not.

Figure 9 shows angular distances and error barriers for the synthetic example. Similarly to Lobacheva et al. [37], we observe the catapult effect for FLR $\gg$ PLR. Although both train and test error barriers emerge in subregime 2A, the barrier values are significantly higher in subregime 2B. It is noteworthy that the angular distance and the error barriers between SWA and low FLR are much smaller than those to the high FLR (at least up to the boundary between subregime 2B and regime 3). Overall, most claims from Lobacheva et al. [37] hold.

Lastly, Figure 10 complements Figure 3 from the main text. We observe that SWA and fine-tuning with a low FLR preserve feature sparsity obtained from pre-training in subregime 2A. Pre-traning with other initial LR values does not allow the model to focus on a single feature. However, fine-tuning with a high FLR restores feature sparsity by either the catapults (when FLR $\gg$ PLR, regime 1) or convergence (when FLR $<$ PLR, regimes 2 and 3).

Appendix D. Additional results on Fourier features

In Figure 11 we present the accuracy of different frequency bands for the rest of scale-invariant setups. Considering SI ResNet-18 on both datasets, the mid-frequency features have higher absolute accuracy values in subregime 2A compared to background and low-frequency features (both in regimes 1 and 2). The specialization on mid frequencies is even more pronounced after weight averaging and fine-tuning. Moreover, the behavior of mid-frequency line after fine-tuning is highly correlated with the test accuracy of the corresponding FLRs in Figure 1.

As for the SI ConvNet architecture, feature specialization is less obvious: we do not observe significant focus on mid frequencies in subregime 2A, perhaps because a combination of 3 convolutional layers is not enough to learn fine-grained image details. This lack of sparsity may be also related to less pronounced effects of the second regime for this architecture, discussed in Lobacheva et al. [37]. Nevertheless, we see a peak of mid-frequency accuracy in subregime 2A for both CIFAR-10 and CIFAR-100, indicating that pre-training with larger PLRs is beneficial for this setup too.

Appendix E. Additional results for the practical setup

In this section, we provide additional results for the practical setup of our experiments: additional plots for the ResNet-18 on CIFAR-10 and ablation on the CIFAR-100 dataset.

Figure 12 depicts the generalization results for the fine-tuned (left plots) and SWA (right plots) solutions in the practical setup on both datasets. SWA follows the same trend as fine-tuning, confirming that the best solutions are obtained in the lower part of the second regime, i.e., subregime 2A. At larger PLRs of the second regime SWA quality rapidly deteriorates. In the first regime, both SWA and fine-tuning are able to improve the accuracy due to incomplete convergence issues discussed in Lobacheva et al. [37]. The results on both datasets are similar.

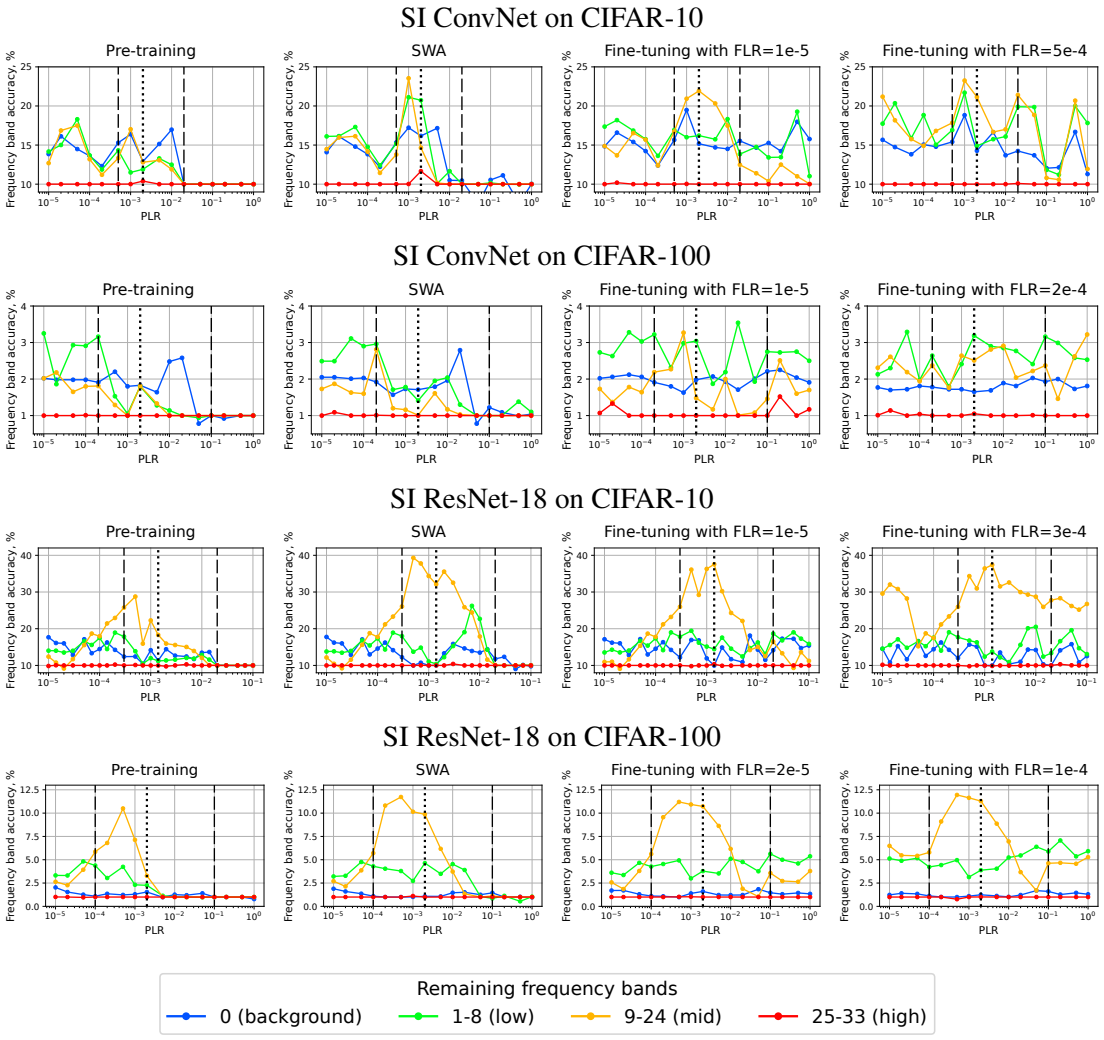

Figure 11: Accuracy of different frequency bands for pre-training (column 1), SWA (over 5 models; column 2), and fine-tuning with low FLR (column 3) and high FLR (column 4). SI ConvNet and SI ResNet-18 on CIFAR-10/CIFAR-100.

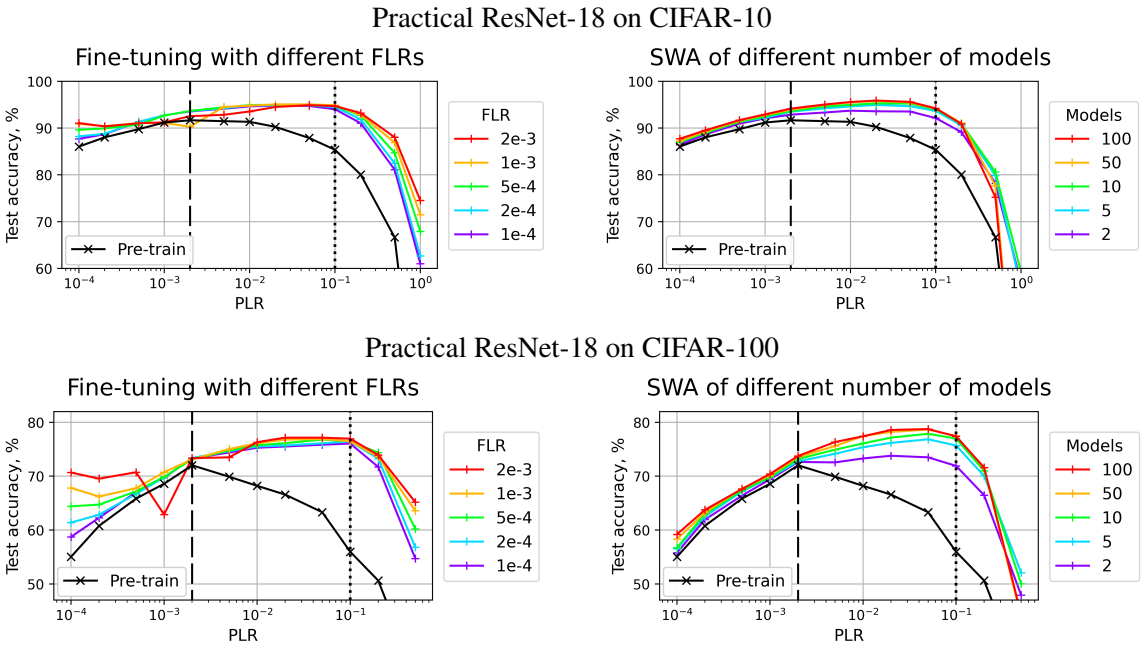

Figure 12: Practical setting. Test accuracy of different fine-tuned (left) and SWA (right) solutions. Test accuracy after pre-training is depicted with the black line. Dashed lines denote boundaries between the pre-training regimes, dotted line divides the second regime into two subregimes. Full version of the results for CIFAR-10 complementing Figure 6 and similar results for CIFAR-100.

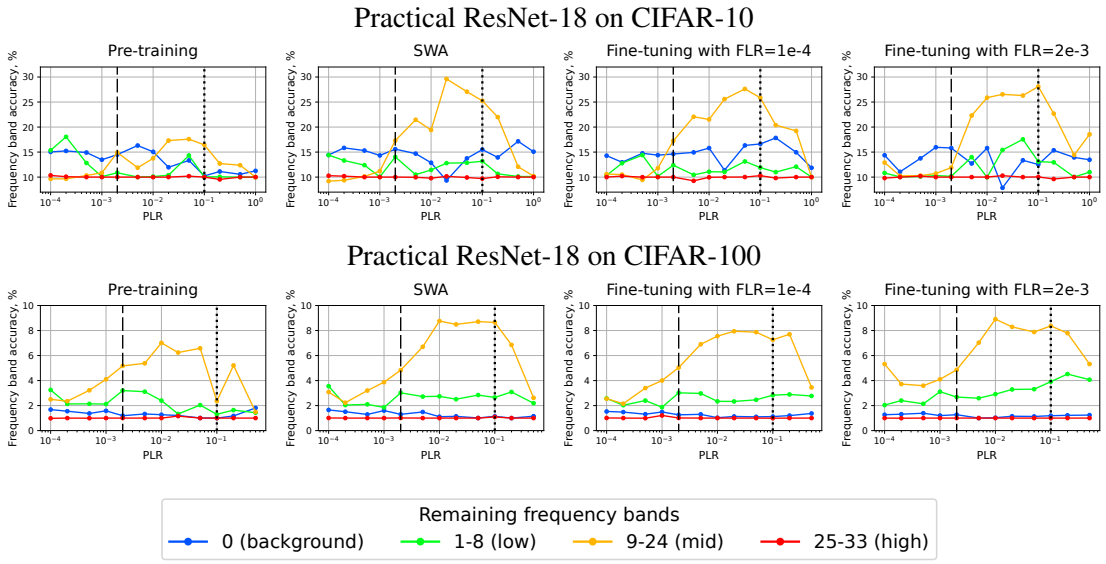

Figure 13: Practical setting. Accuracy of different frequency bands for pre-training (column 1), SWA (over 5 models; column 2), and fine-tuning with low FLR (column 3) and high FLR (column 4). Full version of the results for CIFAR-10 complementing Figure 6 and similar results for CIFAR-100.

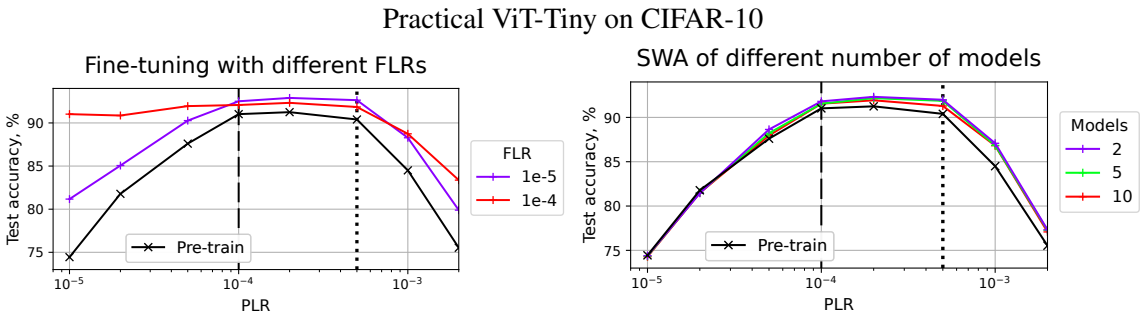

Figure 14: Practical setting on ViT. Test accuracy of different fine-tuned (left) and SWA (right) solutions. Test accuracy after pre-training is depicted with the black line. Dashed lines denote boundaries between the pre-training regimes, dotted line divides the second regime into two sub-regimes.

Figure 13 shows the full panel of results on frequency bands analysis for the practical setting. The mid-frequency bias in subregime 2A and the induced feature sparsity is especially pronounced after fine-tuning or SWA, however it still can be clearly seen already after the pre-training stage on CIFAR-100. Other PLR ranges show no such bias towards a particular spectrum component, however, fine-tuning with a high FLR can partly restore it. Overall, the mid-frequency bias is more consistent on CIFAR-100, possibly due to the complexity of the dataset, resulting in a higher usefulness of this band for classification.

Appendix F. Practical setup on Vision Transformer

In this section, we verify our findings on the Vision Transformer (ViT) architecture [15]. We train the ViT-Small variant of the model on the CIFAR-10 dataset, using the implementation by Kentaro Yoshioka[2]. Moreover, we add Layer Normalization [8] after the linear layers in the Transformer's Feed-Forward Networks to increase the proportion of scale-invariant parameters and ensure that regime 2 is observed in this experimental setup. We train the ViT model with Adam [26] optimizer, using patch size 4, batch size 512, weight decay 10^{-4} and no mixed precision training. Besides the regular CIFAR-10 augmentations mentioned in Appendix A, we use RandAugment [14] with parameters $N = 2, M = 14$. We pre-train and fine-tune both for 500 epochs.

Figure 14 shows the generalization results for both fine-tuning and SWA. Using intensive augmentations requires many more epochs for complete convergence, so fine-tuning with a low FLR still provides an improvement over the pre-training in regime 1. However, fine-tuning with a high FLR gives even higher test accuracy, which is independent of the PLR value. Considering subregime 2A, both FLRs lead to the same optimal quality. SWA also leads to higher accuracy in subregime 2A, which is consistent with Lobacheva et al. [37].

Figure 15 shows the Fourier features analysis for the ViT setup. Contrary to the results for convolutional networks, ViT tends to focus on low-frequency features, which is consistent with findings of Park and Kim [42]. To understand whether ViT prefers learning low-frequency part of the spectrum or it simply has a different boundary between low and mid-frequencies, we evaluate

2. https://github.com/kentaroy47/vision-transformers-cifar10/tree/main

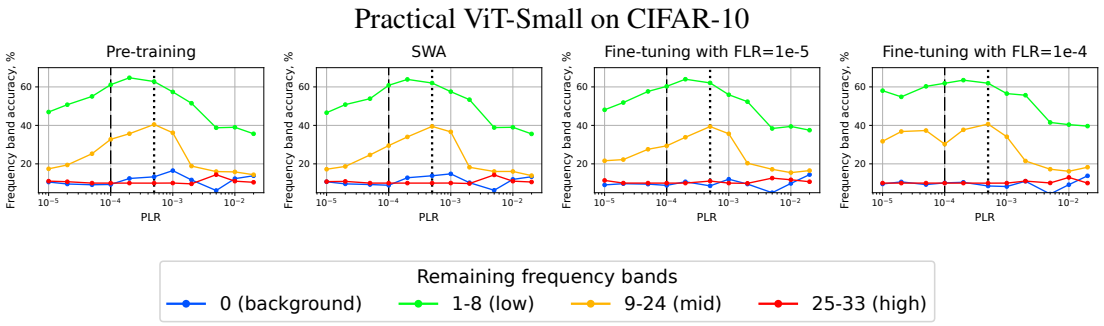

Figure 15: Practical setting on ViT. Accuracy of different frequency bands for pre-training (left), SWA (over 5 models; center left), and fine-tuning with low FLR (center right) and high FLR (right).

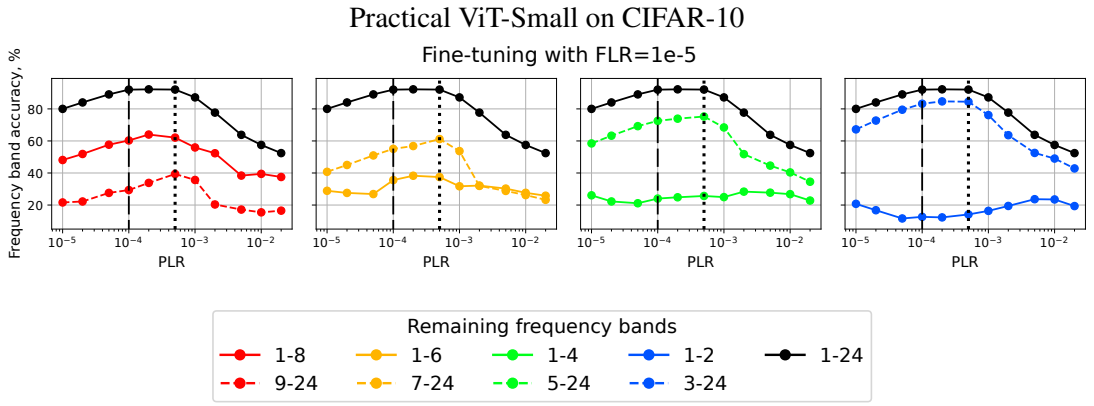

Figure 16: Practical setting on ViT. Accuracy of different frequency bands when varying the boundary between low and mid-frequencies for fine-tuning with low FLR.

the feature band accuracy on different splits of bands into low and mid-frequency groups. The results are presented in Figure 16. Indeed, when switching from the split 1-8 / 9-24 to the split 1-6 / 7-24, we observe that mid-frequencies are more important in the latter case. Moreover, moving the boundary to even smaller values further reduces the importance of the low-frequency features. Thus, the lowest components of the spectrum are still primarily ignored by ViT, but the range of important frequency bands starts at smaller values compared to convolutional networks. Nevertheless, considering the initial 1-8 / 9-24 split, pre-training with large PLRs increases the accuracy of low and mid-frequencies, allowing the transformer to specialize on the parts of the spectrum most relevant to the classification task.

Appendix G. Generalization and sharpness of the fine-tuned solutions

In this section, we extend the results of Kodryan et al. [29], who state that in the first regime higher LRs lead to both better generalizing and less sharp minima, to fine-tuning in the second regime. We adopt their measure of sharpness, which is the mean stochastic gradient norm over batches of the data. For fairness of comparison, since lower loss may naturally lead to lower gradients, we

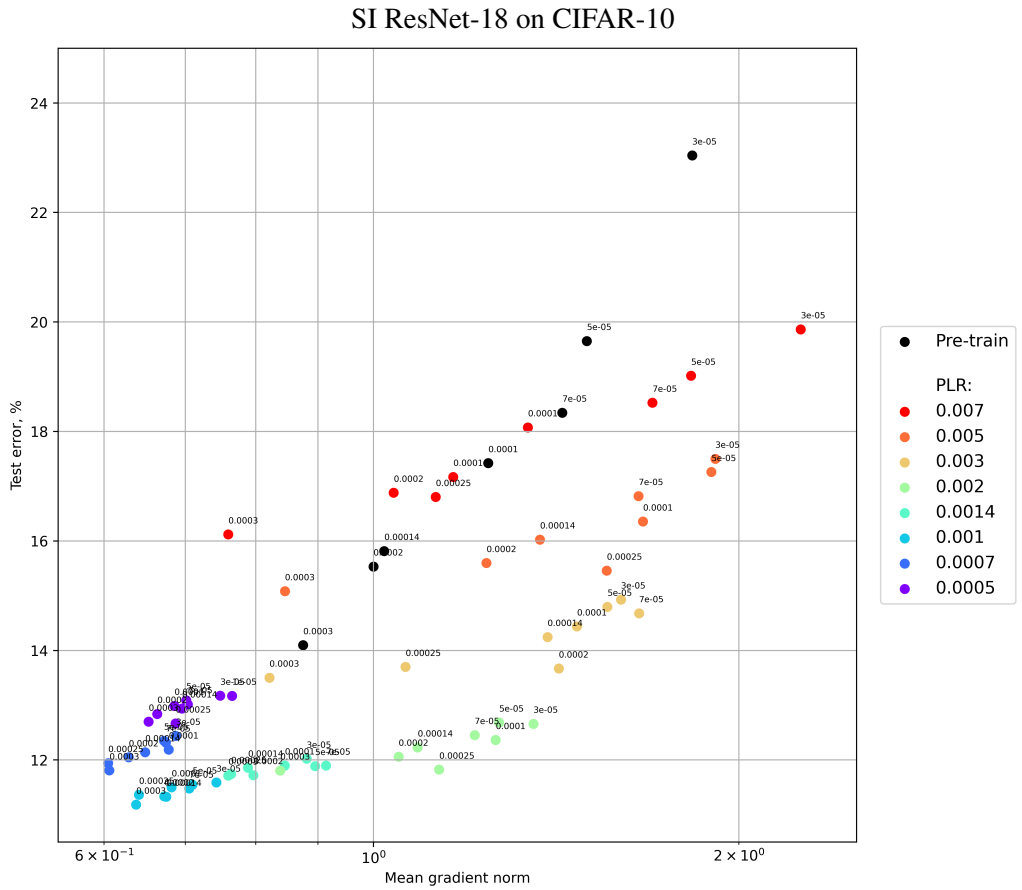

Figure 17: Scatter plot of sharpness vs. test error for the fine-tuned solutions at the same level of the training loss. Points of the same color represent fine-tuned solutions with different FLRs from the same pre-trained point. Different colors denote different PLRs of the second regime: from low (purple) to high (red). Black dots correspond to the pre-trained points of the first regime, replicating the results of Kodryan et al. [29]. SI ResNet-18 on CIFAR-10.

decided to fix the training loss of all fine-tuned solutions at $6 \cdot 10^{-4}$ before calculating test error and sharpness for them. We do this by taking an appropriate weighted average of these values calculated for the fine-tuning checkpoints just before and right after crossing the loss threshold.

The results are presented in Figure 17. At first sight, the picture shows an overall positively correlated trend in the relationship between test error and sharpness. The same trend applies to fine-tuning checkpoints obtained from the same initialization, i.e., from the same pre-trained checkpoint. However, by taking the fine-tuned solutions obtained from different pre-trains we can easily break and even reverse this correlation. Compare, e.g., blue and light green points of the fine-tuned solutions obtained from two different PLRs: with similar test error, their sharpness values differ on average by almost a factor of two.

In sum, we see no direct correlation between sharpness and generalization across fine-tuning runs from different pre-trained points, which accords with recent works in this area [4, 25].