How Learning Rates Shape Neural Network Focus: Insights from Example Ranking

Ekaterina Lobacheva[1,2] **, Keller Jordan**[3] **, Aristide Baratin**[1,4] **, Nicolas Le Roux**[2]
[1] Université de Montréal [2] Mila - Quebec AI Institute [3] Independent researcher
[4] Samsung - SAIT AI Lab Montreal
Correspondence to: ekaterina.lobacheva@mila.quebec

Abstract

The learning rate is a key hyperparameter that affects both the speed of training and the generalization performance of neural networks. Through a new *loss-based example ranking* analysis, we show that networks trained with different learning rates focus their capacity on different parts of the data distribution, leading to solutions with different generalization properties. These findings, which hold across architectures and datasets, provide new insights into how learning rates affect model performance and example-level dynamics in neural networks.

1 Introduction

The learning rate is a critical hyperparameter in neural network training, with a significant impact on both convergence speed and generalization performance. It is widely recognized that using a high initial learning rate improves generalization [4, 15]. A common explanation is that higher learning rates introduce more noise from the stochastic optimizer, inducing feature sparsity [1–3, 16] and leading to flatter minima [6, 10, 11] with better generalization properties [5, 8, 9].

In this work, we offer a new perspective by examining how learning rates affect the network's focus on different parts of the data distribution. By analyzing *loss-based example ranking*–the ordering of examples in the dataset based on their loss values at the end of training–we show that varying the learning rate shifts the network's emphasis toward certain examples. We further give evidence that these shifts in focus directly influence generalization performance.

The key contributions of this work are:

1. We introduce loss-based example ranking to explore how learning rate affects the network's focus on different parts of the data distribution (Section 2).
2. We demonstrate that different learning rates lead to distinct example rankings, showing that the learning rate alters which examples receive more or less emphasis (Section 3.1 and 3.2).
3. Through experiments with reweighting examples during training based on shifts in example ranking between different learning rates, we illustrate that changes in the network's focus induces by the learning rate influence generalization (Section 3.3).

2 Methodology

2.1 Loss-based Example Ranking

Let us first introduce the concept of **loss-based example ranking** as a way to quantify how neural networks prioritize learning examples at different learning rates. Given a dataset $D =$

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

$\{(x_1, y_1), ..., (x_N, y_N))\}$, a trained network f, and a loss function $L(f(x), y)$, we define the rank of each example based on its loss value at the end of training. Specifically, for a given example (x_i, y_i), its rank r_i is defined as the number of examples in the dataset on which network f has lower or equal loss:

$$r_i = \sum_{j=1}^{N} I[L(f(x_j), y_j) \leq L(f(x_i), y_i)] \tag{1}$$

where I is the indicator function. The example with the lowest loss is assigned a rank of 1, and the example with the highest loss is assigned a rank of N. This ranking is invariant to the overall scale of the loss values which can vary for different learning rates, and allows us to focus on the relative difficulty of examples when comparing networks. Throughout the paper, we refer to examples as 'easy' or 'hard' for the network based on their ranks, with lower-ranked examples considered easier.

2.2 Normalized Rank Correlation

To compare the ranking of examples across different networks, we use the standard **Kendall rank correlation coefficient**, which measures the similarity between two ranked lists. Let f^{LR_1} and f^{LR_2} be two networks trained with different learning rates. The Kendall rank correlation τ between two rankings can be computed as:

$$\tau(f^{LR_1}, f^{LR_2}) = \frac{2}{N(N-1)} \sum_{i<j} \text{sign}(r_i^{LR_1} - r_j^{LR_1}) \text{sign}(r_i^{LR_2} - r_j^{LR_2}) \tag{2}$$

where $r_i^{LR_1}$ and $r_i^{LR_2}$ are the ranks of example i in networks trained with learning rates LR_1 and LR_2, respectively.

In practice, the rankings of two networks trained with the same learning rate do not correlate perfectly due to the noise introduced by random initialization and stochastic training. Moreover, the magnitude of the noise may vary for different learning rates, making the comparison based on the Kendall rank correlation less stable. To account for this noise, we define a **normalized rank correlation** as:

$$\tau_{\text{norm}}(f^{LR_1}, f^{LR_2}) = \frac{\tau(f^{LR_1}, f^{LR_2})}{\sqrt{\mathbb{E}_{s_1 \neq s_2}[\tau(f_{s_1}^{LR_1}, f_{s_2}^{LR_1})] \mathbb{E}_{s_1 \neq s_2}[\tau(f_{s_1}^{LR_2}, f_{s_2}^{LR_2})]}} \tag{3}$$

where the expectation is taken over different random seeds s_1 and s_2. This normalization accounts for random fluctuations between training runs while focusing on differences caused by the learning rates.

3 Results and Analysis

In this section, we present the results of our experiments, which assess the effect of learning rates on loss-based example rankings.We report results obtained with ConvNet and ResNet-18 on CIFAR-10 [12], and ResNet-18 on CIFAR-100 [13] and Tiny-ImageNet [14]. We use a learning rate schedule in all experiments to ensure convergence to high-quality solutions while varying the maximum learning rate during training as a parameter, which is either the initial learning rate for ResNet-18 or the learning rate after the warm-up for ConvNet. Experimental details are given in Appendix A.

3.1 Effect of Learning Rate on Example Ranking

We first examine the rankings of training examples for networks trained with different learning rates. For each learning rate, we train five networks with different random seeds and compute the loss-based example rankings at the end of training. Figure 1 shows the normalized rank correlations between example rankings for different learning rates, averaged over seeds, along with standard deviations.

Key Observation: Correlations between example rankings for different learning rates are lower than correlations between networks trained with the same learning rate; the larger the difference between learning rates, the lower the rank correlation. This result indicates that the choice of learning rate can significantly impact the example prioritization and lead the network to focus on distinct parts of the data distribution. Analogous results hold for rankings of examples in the test set (see Figures 5 and 6 in Appendix B). We also discuss how the noise between the training runs affects these results in Appendix B.

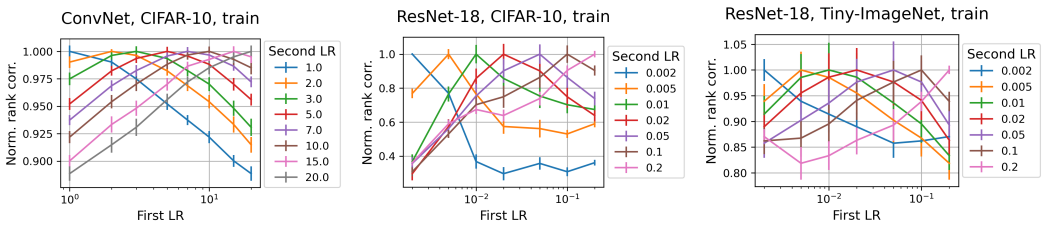

Figure 1: Normalized rank correlations between rankings of training examples for different pairs of learning rates for various datasets and network architectures. The x-axis corresponds to the values of the first learning rate from the pair, while the different colors of the curves correspond to different values for the second one. For each learning rate, five networks were trained using different random seeds. The mean and standard deviation values for each pair of learning rates are shown.

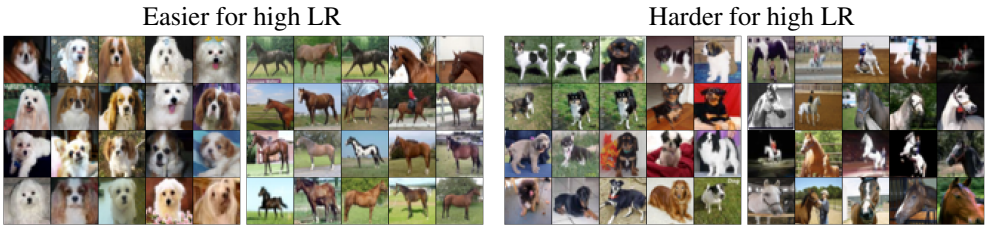

Figure 2: Examples with the highest and the lowest rank difference between LR 1 and LR 20.

3.2 Example-Level Analysis

To better understand the effect of learning rate, we perform an example-level analysis of rank changes between networks trained with different learning rates. To reduce the noise in example rankings caused by variability across training runs, we analyze **logit-ensembles** instead of individual networks. For each learning rate, we train many neural networks and then construct a logit-ensemble by averaging them in the logit space. This approach enables us to capture the primary effects of the learning rate while minimizing the influence of training noise. The airbench package [7] we use for experiments with ConvNets on CIFAR-10 allows us to efficiently train a large number of networks. For maximally stable results, we use 5000-network logit-ensembles in our analysis.

Figure 2 displays the 20 examples from two CIFAR-10 classes with the largest changes in ranking between ensembles trained with the lowest (1.0) and the highest (20.0) learning rates in our setup. Upon close inspection, we find that examples that become easier with a higher learning rate tend to be more typical of their class, such as front-face views of dogs or side views of horses. In contrast, examples that become harder are generally less typical or more complex, requiring the network to focus on finer details. However, we emphasize that this is only a general trend, and ranking changes for more localized learning rate increases may diverge significantly (see Appendix C for details).

To explore this further, we compare the ranking changes induced by different pairs of learning rates. Specifically, for each example i, we compute the changes $\Delta r_i := r_i^{LR_{\text{lower}}} - r_i^{LR_{\text{higher}}}$ for different pairs $(LR_{\text{lower}}, LR_{\text{higher}})$ of learning rates. Figure 3 presents these comparisons. As shown in the left plot, the changes in example rankings for different increases of the learning rate from the same low value are positively correlated: examples that become easier with a moderate increase in learning rate tend to also become easier with larger increases. However, the degree of correlation decreases as the gap between learning rates grows. In the right plot, we compare consecutive learning rate increases and observe that while neighboring learning rate increases influence example rankings very similarly, more distant learning rate increases show almost no correlation in their example rankings.

Key Finding: Learning rates smoothly affect example rankings, however the effect is not uniform. Different learning rate changes can cause substantially different shifts in which examples become easier or harder, indicating that each learning rate causes the network to focus on different aspects of the data.

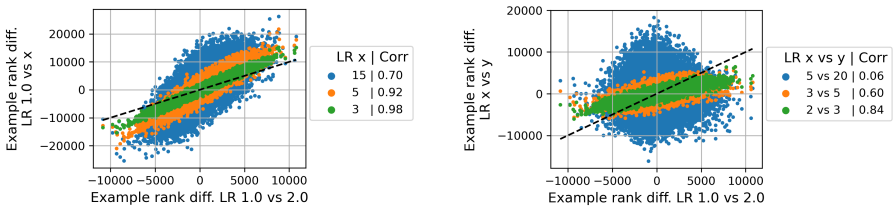

Figure 3: Comparison of the changes in example rankings for different learning rate increases from the same low learning rate (left) and for consecutive learning rate increases (right). Each point corresponds to an example and depicts changes in its rank for specified learning rate increases along the x- and y-axis. The diagonal $x = y$ is indicated with a dashed line. To compare the changes for different learning rate increases, we measure the Pearson correlation (shown in legends).

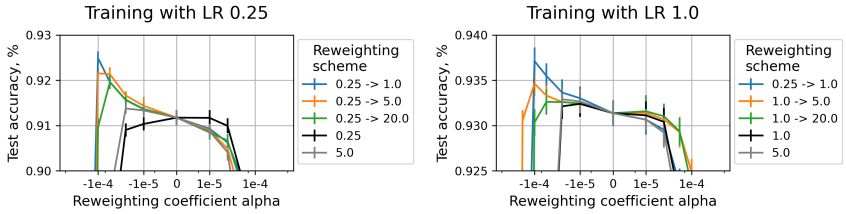

Figure 4: Low learning rate training with example reweighting. Colored lines correspond to reweighting examples based on the difference in example rankings between a pair of learning rates (negative α correspond to upweighting/downweighting examples which become easier/harder for higher learning rate), while black/gray lines depict baselines of reweighting examples based on their ranking for a specific learning rate (negative α corresponds to upweighting/downweighting examples which are easier/harder for this learning rate).

3.3 Link to Generalization

We hypothesize that changes in example ranking caused by the learning rate can partially explain the generalization benefits of training with higher learning rates. To test this hypothesis, we reweight examples during low learning rate training based on how their rankings shift compared to networks trained with higher learning rates. Specifically, we adjust the loss for each training example i by $1 + \alpha \cdot (r_i(f) - r_i(g))$, where $r_i(f)$ and $r_i(g)$ represent the rankings of the example in two 5000-network ensembles, f (trained with a high learning rate) and g (trained with a low learning rate). Negative values of α put more emphasis on examples that become easier at higher learning rates while downweighting examples that become harder. To ensure that possible improvements from such reweighting are not merely the result of upweighting/downweighting easy/hard examples for a fixed learning rate, we also include baselines where examples are reweighted based on the example ranking for a specific learning rate using $1 + \alpha \cdot (r_i(f) - N/2)$ weights. Negative values of α, in this case, correspond to upweighting/downweighting examples, which are easier/harder for this learning rate. Figure 4 shows test accuracies for various values of α.

Key Finding: This reweighting scheme improves generalization, confirming that the positive effect of higher learning rates can be at least partially explained through the changes in the network's focus over the data distribution. However, we observe that reweighting based on small, localized changes in learning rates (e.g., from 0.25 to 1.0) yields significantly better results than reweighting based on large differences (e.g., from 0.25 to 20.0). This nuance highlights the higher importance of specific ranking changes for each learning rate increase over the broader trend of focusing more on typical examples.

4 Conclusion

In this paper, we investigated how learning rates impact the functions learned by neural networks using loss-based example ranking, revealing where the network focuses its capacity within the data distribution. Our findings indicate that increasing the learning rate smoothly alters example ranking.

These ranking changes are meaningful and partially responsible for the improved generalization for higher learning rates. Our qualitative analysis suggests that the change in ranking correlates with the typicality of examples in the dataset, with more typical examples becoming easier for networks trained with higher learning rates. However, this is only a general trend; we also found that generalization properties are connected to more nuanced changed in ranking. Future work will explore these aspects further.

Acknowledgments and Disclosure of Funding

We thank Arsenii Kuznetsov and Maxim Kodryan for valuable discussions. NLR is supported by the Canada CIFAR AI Chair Program. We also acknowledge support from the Canada Excellence Research Chairs Program. This research was enabled by compute resources and technical support provided by Mila - Quebec AI Institute (mila.quebec).

References

[1] Ahn, K., Bubeck, S., Chewi, S., Lee, Y. T., Suarez, F., and Zhang, Y. (2024). Learning threshold neurons via edge of stability. In *Advances in Neural Information Processing Systems*.

[2] Andriushchenko, M., Varre, A. V., Pillaud-Vivien, L., and Flammarion, N. (2023). Sgd with large step sizes learns sparse features. In *International Conference on Machine Learning*.

[3] Chen, F., Kunin, D., Yamamura, A., and Ganguli, S. (2023). Stochastic collapse: How gradient noise attracts SGD dynamics towards simpler subnetworks. In *Advances in Neural Information Processing Systems*.

[4] Frankle, J., Schwab, D. J., and Morcos, A. S. (2020). The early phase of neural network training. In *International Conference on Learning Representations*.

[5] Hochreiter, S. and Schmidhuber, J. (1994). Simplifying neural nets by discovering flat minima. In *Advances in Neural Information Processing Systems*.

[6] Jastrzębski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. (2019). On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*.

[7] Jordan, K. (2024). 94% on CIFAR-10 in 3.29 seconds on a single GPU.

[8] Kaur, S., Cohen, J., and Lipton, Z. C. (2023). On the maximum hessian eigenvalue and generalization. In *Proceedings of Machine Learning Research*, volume 187, pages 51–65. PMLR.

[9] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*.

[10] Kleinberg, B., Li, Y., and Yuan, Y. (2018). An alternative view: When does sgd escape local minima? In *International Conference on Machine Learning*.

[11] Kodryan, M., Lobacheva, E., Nakhodnov, M., and Vetrov, D. P. (2022). Training scale-invariant neural networks on the sphere can happen in three regimes. In *Advances in Neural Information Processing Systems*.

[12] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

[13] Krizhevsky, A., Nair, V., and Hinton, G. (2009). CIFAR-100 (Canadian institute for advanced research).

[14] Le, Y. and Yang, X. S. (2015). Tiny imagenet visual recognition challenge.

[15] Li, Y., Wei, C., and Ma, T. (2019). Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*.

[16] Sadrtdinov, I., Kodryan, M., Pokonechny, E., Lobacheva, E., and Vetrov, D. (2024). Where do large learning rates lead us? In *Advances in Neural Information Processing Systems*.

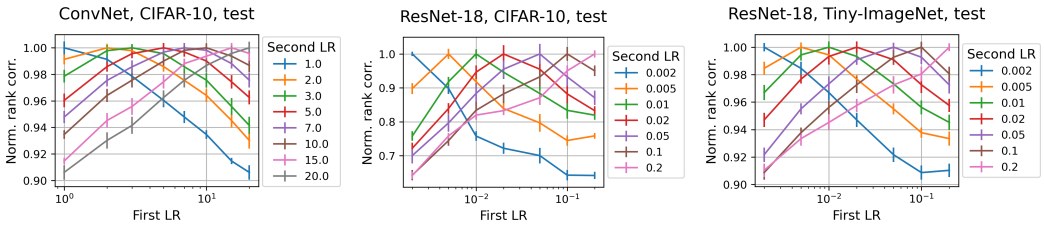

Figure 5: Normalized rank correlations between *test* example rankings for different pairs of learning rates for various datasets and network architectures. For each learning rate, five networks were trained using different random seeds. The mean and standard deviations of the normalized rank correlations for each pair of learning rates on the training data are shown.

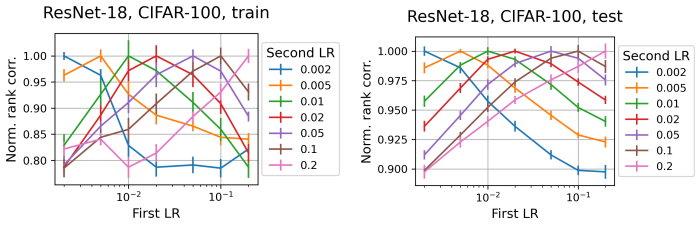

Figure 6: Normalized rank correlations between train (left) and test (right) example rankings for different pairs of learning rates for ResNet-18 on CIFAR-10. For each learning rate, five networks were trained using different random seeds. The mean and standard deviations of the normalized rank correlations for each pair of learning rates on the training data are shown.

A Experimental Details

ConvNet experiments We conduct most of the experiments using a simplified version of airbench package [7] available at `https://github.com/KellerJordan/research-airbench/tree/master`. It allows us to train a ConvNet architecture on CIFAR-10 dataset [12] to 94% test accuracy in several seconds. We use the original training procedure from clean_airbench and vary the LR parameter in our experiments. We train 5000 networks with different random seeds for each LR parameter value from $[0.25, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20]$. Table 1 presents the mean train and test accuracy values of the resulting networks. For the reweighting experiments, we additionally train 30 networks for each reweighting scheme and coefficient α.

ResNet experiments We conduct additional experiments with ResNet-18 on CIFAR-10 [12], CIFAR-100 [13], and Tiny-ImageNet [14]. We train ResNet-18 networks using stochastic gradient descent with batch size 128, weight decay 0.0005, momentum 0.9, standard crop and flip data augmentations, and 200-epochs cosine LR schedule with maximal LRs from $[0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2]$. For each LR we train 5 networks with different random seeds. We provide the mean train and test accuracy values of the resulting networks in Table 2.

B Additional Results on Example Rank Correlation for Different LRs

In this section, we provide additional results on the example rank correlation between networks trained with different learning rates complementary to Figure 1 in the main text. In Figure 5, we show the normalized rank correlations between test example rankings for ConvNet on CIFAR-10, ResNet-18 on CIFAR-10, and ResNet-18 on Tiny-ImageNet. Additionally, Figure 6 depicts the same results for ResNet-18 on train and test sets of CIFAR-100. The learning rate changes the example ranking for all considered dataset-architecture pairs on both train and test sets.

Interestingly, the results for ConvNet on train and test sets are almost identical, while results for ResNet-18 are much more noisy on train sets. The noise level in correlation results highly depends on the variance of training runs with different random seeds. We include values of mean example

Table 1: Accuracy and non-normalized example rank self-correlation of ConvNet trained with different learning rates on train and test subsets of CIFAR-10. Results on five individual networks, five 1000-network ensembles, and a 5000-network ensemble are shown.

ConvNet on CIFAR-10

LR	1.0	2.0	3.0	5.0	7.0	10.0	15.0	20.0
Mean train acc., %	98.16	98.70	98.94	99.06	99.11	99.02	98.82	98.58
Mean test acc., %	93.26	93.63	93.68	93.80	93.87	93.80	93.81	93.53
Train ex. rank self-corr.	0.70	0.71	0.71	0.72	0.73	0.74	0.75	0.75
Test ex. rank self-corr.	0.75	0.75	0.75	0.76	0.77	0.77	0.78	0.78

1000-network ensemble of ConvNets on CIFAR-10

LR	1.0	2.0	3.0	5.0	7.0	10.0	15.0	20.0
Mean train acc., %	98.63	99.14	99.31	99.42	99.45	99.41	99.27	99.04
Mean test acc., %	94.12	94.57	94.73	94.80	94.80	94.68	94.62	94.39
Train ex. rank self-corr.	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Test ex. rank self-corr.	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

5000-network ensemble of ConvNets on CIFAR-10

LR	1.0	2.0	3.0	5.0	7.0	10.0	15.0	20.0
Train acc., %	98.63	99.14	99.34	99.43	99.45	99.42	99.27	99.03
Test acc., %	94.13	94.57	94.74	94.81	94.79	94.66	94.62	94.41

Table 2: Accuracy and non-normalized example rank self-correlation of ResNet-18 trained with different learning rates on train and test subsets of CIFAR-10, CIFAR-100, and Tiny-ImageNet. Results on five individual networks are shown.

ResNet-18 on CIFAR-10

LR	0.002	0.005	0.01	0.02	0.05	0.1	0.2
Mean train acc., %	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Mean test acc., %	92.53	93.91	94.38	94.88	95.17	95.16	94.90
Train ex. rank self-corr.	0.45	0.35	0.26	0.22	0.26	0.29	0.33
Test ex. rank self-corr.	0.59	0.54	0.47	0.42	0.44	0.45	0.48

ResNet-18 on CIFAR-100

LR	0.002	0.005	0.01	0.02	0.05	0.1	0.2
Mean train acc., %	86.88	88.44	88.58	87.56	83.95	78.01	69.30
Mean test acc., %	71.44	74.37	75.65	76.70	77.85	78.10	77.20
Train ex. rank self-corr.	0.35	0.33	0.21	0.20	0.25	0.29	0.35
Test ex. rank self-corr.	0.64	0.65	0.64	0.65	0.67	0.67	0.66

ResNet-18 on Tiny-ImageNet

LR	0.002	0.005	0.01	0.02	0.05	0.1	0.2
Mean train acc., %	84.87	87.10	86.10	83.76	76.63	67.73	55.42
Mean test acc., %	50.68	52.71	52.88	52.64	52.21	49.58	42.97
Train ex. rank self-corr.	0.35	0.22	0.21	0.24	0.26	0.33	0.43
Test ex.rank self-corr.	0.64	0.63	0.64	0.65	0.65	0.67	0.66

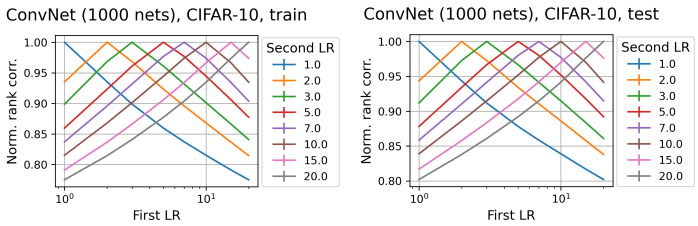

Figure 7: Normalized rank correlations between example rankings for 1000-network ensembles of ConvNets trained with different LRs on train and test sets of CIFAR-10. Five ensembles are trained with each LR, mean and standard deviations of normalized rank correlation for each pair of LRs are shown (standard deviation is very low in this experiment).

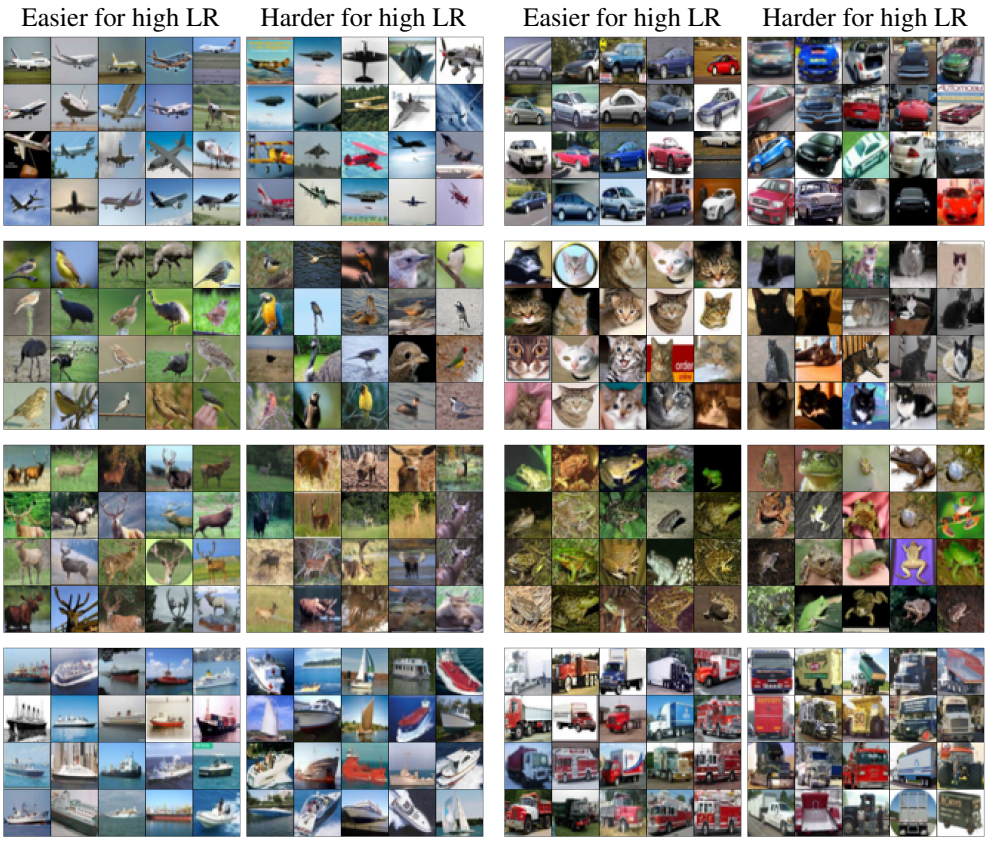

Figure 8: Examples with the highest and the lowest rank difference between LR 1 and LR 20.

rank self-correlation for all networks and LRs in Tables 1, 2 (mean non-normalized rank correlation between networks trained with the same learning rate but different random seeds). The higher values of this metric correspond to a lower variance of training runs with different random seeds. ResNets indeed demonstrate lower self-correlation than ConvNets in general, and especially on train sets.

In Figure 7, we additionally demonstrate that normalized rank correlations demonstrate the same behavior in a low-noise setting, where instead of five ConvNet networks, we compare the example rankings of five 1000-network ConvNet ensembles. The results are very clean in this case due to the very low variance of training runs (the example rank self-correlation is $> 99\%$, see Table 1).

Figure 9: Examples with the highest and the lowest rank difference between different pairs of low and high LRs.

C Additional Results on Example-Level Analysis

In this section, we provide additional results and analysis of example groups with the highest and lowest rank difference between networks trained with different LRs. Complementary to Figure 2 from the main text, in Figure 8, we present images from all other CIFAR-10 classes with the most significant rank changes between the lowest and highest LRs. For most classes, the example groups with opposite rank changes look qualitatively different, with more typical examples becoming easier for networks trained with high LR.

In Figure 9, we compare the example groups with the highest and lowest rank difference for various changes in LR. For most LR changes, the examples in both groups are similar to the ones characterizing the general trend between low to high LRs from Figure 2. However, the example groups are not identical for different LR changes and for some LR changes they divert a lot from the general trend (e.g., LR 10 vs LR 20 for dogs and LR 1 vs LR 3 for horses). We leave a further analysis of how exactly different LR changes influence the example ranking and which changes are the most influential for generalization for future work.