Grokking, Rank Minimization and Generalization in Deep Learning

David Yunis [1] Kumar Kshitij Patel [1] Samuel Wheeler [2] Pedro Savarese [1]
Gal Vardi [3] Karen Livescu [1] Michael Maire [4] Matthew R. Walter [1]

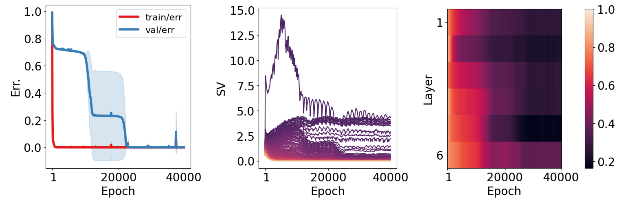
Abstract

Much work has been devoted to explaining the recently discovered *"grokking"* phenomenon, where a neural network first fits the training loss, then many iterations later suddenly fits the validation loss. To explore this puzzling behavior, we examine the evolution of singular values and vectors of weight matrices inside the neural network. First we show that the transition to generalization in grokking coincides with the discovery of a low-rank solution in the weights. We then show that the trend towards rank minimization is much more general than grokking alone and elucidate the crucial role that weight decay plays in promoting this trend. Such analysis leads to a deeper understanding of generalization in practical systems.

1. Introduction

Chief among the puzzling behaviors of neural networks is generalization. While only ever seeing a given training set, they can perform well on unseen data, sometimes even under distribution shifts, and even when perfect memorization solves the training objective. This property has led to an explosion of research and interest in neural networks across a broad set of domains, yet many fundamental questions about their learning behavior remain unanswered.

For instance, despite extensive research, we still lack a complete understanding of the implicit biases of neural networks trained via stochastic optimization (Neyshabur et al., 2014). Even basic questions, such as the role of weight decay (Hanson & Pratt, 1988; Krogh & Hertz, 1991; Zhang et al., 2018a), have only partial answers (Van Laarhoven, 2017; Andriushchenko et al., 2023; Yaras et al., 2023b).



(a) Error (b) SVs (c) Eff. Rank

Figure 1: Grokking experiment reproduced from Nanda et al. (2023). **Left:** Error over time. **Middle:** Singular value evolution for a single matrix in the Transformer where each line is a single singular value and color represents rank. **Right:** Normalized effective rank (Eqn. 1) of all matrices where the y-axis represents the depth of the matrix in the model. We see that the drop in validation error coincides with the discovery of low-rank parameters inside the model, suggesting a connection between rank and generalization. Large error bars and segmented behavior in the singular value evolution are due to averaging over seeds that generalize at different times.

Perhaps most importantly, we still lack a concrete understanding of how neural networks generalize (Zhang et al., 2021), despite having enough capacity to overfit the training data completely.

Power et al. (2022) first noticed a phenomenon they named *"grokking"*, where a model first fits the training loss while performing poorly on the validation loss but eventually *"learns"* to generalize. They propose the setting of this observation as a test bed to study generalization as it has quite unique properties and can also be replicated easily with few resources. Much work has been devoted to explaining the source of this phenomenon (Lyu et al., 2023; Liu et al., 2022; 2023a; Davies et al., 2022). In particular, weight decay seems crucial (Lyu et al., 2023; Liu et al., 2023a), but its precise contribution is still unclear.

Starting with the grokking phenomenon, we study the spectral dynamics of weight matrices, specifically the evolution of singular values and singular vectors, and show that such dynamics are intimately connected to generalization. The contributions of this work are as follows:

[*]Equal contribution [1]Toyota Technological Institute at Chicago, Chicago, IL, USA [2]Argonne National Laboratory, Lemont, IL, USA [3]Weizmann Institute of Science, Israel, work done primarily at TTIC [4]Department of Computer Science, University of Chicago, Chicago, IL, USA. Correspondence to: David Yunis <dyunis@ttic.edu>.

- In Section 3, we see that the validation loss drop in grokking coincides with the discovery of low-rank solutions across all of the weight matrices in the network simultaneously. We observe that, without weight decay, neither grokking nor the discovery of such a low-rank solution occurs. But, given enough data, we again find that low-rank weights and generalization are correlated.

- In Section 4, we provide an empirical overview of the training dynamics of neural network layers through the lens of the SVD. We demonstrate effective rank minimization across various practical neural networks in complex settings. We see a trend in the alignment of singular vectors in consecutive layers, which becomes increasingly strong as training progresses. We also observe early stabilization in the direction of top singular vectors.

- In Section 5, we connect rank minimization to weight decay, showing that weight decay promotes rank minimization and alignment in consecutive layers, extending theoretical work on the topic. Small (Frankle et al., 2020) to large (Biderman et al., 2023) amounts of weight decay are commonly used to improve generalization, so this further suggests a connection between rank and generalization.

- Given the tempting connection between rank and generalization, Section 6 revisits the classic memorization experiments of Zhang et al. (2021). We show that training with random labels leads to high (effective) rank solutions, while with the true labels, the rank is much lower, strengthening the connection.

2. Related Work

2.1. Grokking

Power et al. (2022) first notice a fascinating phenomenon they call ”grokking” where models quickly fit the training loss on toy tasks, then after a long period of training very quickly generalize on the validation loss. Kumar et al. (2023) find that a relaxed definition of grokking for 2-layer MLPs can be observed, even without weight decay, and claim that the transition from kernel (Jacot et al., 2018) to rich (Atanasov et al., 2023) regime is responsible for grokking. Mohamadi et al. (2023) have a similar explanation, and prove sample complexity bounds on generalization for 2-layer networks. Gromov (2023) find grokking for simple 2-layer networks with gradient descent and show the solutions found by GD and Adam agree with a priori implementations. Davies et al. (2022) hypothesize a connection between double descent and grokking, where simple patterns are learned quickly, and generalizing patterns are learned more slowly. Thilak et al. (2022) show a “slingshot” effect with adaptive optimizers, where loss oscillates and flings the parameters into a better generalizing solution, resulting in grokking without weight decay dependent on the epsilon parameter of Adam (Kingma & Ba, 2014). Liu et al. (2023b) find that grokking behavior correlates with a metric that generalizes the number of partitions a network splits the space into. Tan & Huang (2023) prove in a simplified case that weight norm decrease is sufficient for grokking behavior, and find tighter metrics for prediction. Xu et al. (2023) prove grokking behavior for 2-layer ReLU networks on nearly-orthogonal XOR data. Merrill et al. (2023) show that for a simple 1-layer ReLU network, grokking corresponds to finding a sparse model that exactly agrees with the predictions. Notsawo Jr et al. (2023) show that oscillating loss curves in early epochs predict grokking behavior later. Lyu et al. (2023) proved that grokking occurs when a small amount of weight decay is used for simplified settings. Liu et al. (2022) study grokking in a toy system, develop a qualitative picture, and demonstrate grokking on MNIST. Liu et al. (2023a) propose a norm-regularization explanation for grokking, where small norm controls generalizing solutions. None of these works have explicitly examined the connection with rank, which we do in Section 3.

2.2. Singular Value Dynamics

Prior work (Arora et al., 2019; Milanesi et al., 2021) shows that implicit regularization in deep matrix factorization may not necessarily be captured by the matrix norm and rather might be better described as rank regularization. For the full argument, see Arora et al. (2018, Appendix A), but in particular, one critical assumption is “balanced initialization”, namely that at initialization, for two consecutive matrices W_i and W_{i+1} in a product matrix $\prod_j W_j$, we have $W_{i+1}^\top W_{i+1} = W_i W_i^\top$. When substituting the SVDs of these matrices and simplifying through orthogonality, this results in the condition $V_{i+1}\Sigma_{i+1}^2 V_{i+1}^\top = U_i \Sigma_i^2 U_i^\top$ where U_i and V_{i+1} are orthogonal matrices. Now, because these are two orthogonal decompositions of the same matrix, the diagonals must be equivalent up to a permutation of elements with the same value. Thus, $U_i = V_{i+1}O$ up to signs, where O is a block diagonal matrix that may permute the rows of equivalent diagonal elements. In particular, if all the diagonal elements are distinct and U_i and V_{i+1} are square, then we have $U_i = V_{i+1}$ up to signs. As the product matrices are all aligned with the assumption of balanced initialization, the product of the diagonals will evolve in a closed-form fashion, where larger singular values evolve faster than smaller ones. As Arora et al. (2019) demonstrate, the result is a rank-minimizing behavior with deeper and deeper matrix products. The formula is also empirically confirmed for linear matrix factorization problems. Similar results are derived in tensor products and other structured settings (Saxe et al., 2014; Yaras et al., 2023a). In Section 4, we examine the conclusions and assumptions of the developed theory on much larger, practical neural networks.

2.3. Low-Rank Properties

Another line of work focuses on more general low-rank biases. Earlier work looks at norms as an implicit bias (Gunasekar et al., 2017). Theoretical work finds that norms or closed-form functions of the weights may be insufficient to explain implicit regularization but do not rule out rank minimization (Vardi & Shamir, 2021; Razin & Cohen, 2020). Many studies explore low-rank biases of different matrices, like the Jacobian (Pennington et al., 2018), weight matrices (Le & Jegelka, 2021; Frei et al., 2022; Ongie & Willett, 2022; Martin & Mahoney, 2020; 2021), Gram matrix (Huh et al., 2022), or features (Yu & Wu, 2023; Feng et al., 2022). Others show that dynamics influence rank decay (Chen et al., 2023; Wang & Jacot, 2023). Some authors draw a connection between weight decay and rank minimization in ideal settings (Ziyin et al., 2022; Galanti et al., 2022; Zangrando et al., 2024; Ergen & Pilanci, 2023; Parhi & Nowak, 2023; Shenouda et al., 2023). We are interested in the question of how far these connections extend and will present evidence that **sometimes** agrees and deepens connections suggested by theory and small-scale experiments. In Section 5, we empirically demonstrate the connection between weight decay and rank on much larger systems than previously examined.

3. Grokking and Rank Minimization

Motivated by theoretical work that proposes connections between rank and generalization (Razin & Cohen, 2020; Vardi & Shamir, 2021; Timor et al., 2023), weight decay and rank (Galanti et al., 2022; Yaras et al., 2023b; Zangrando et al., 2024), and the importance of weight decay for grokking (Power et al., 2022; Lyu et al., 2023; Liu et al., 2023a) in simple settings, we evaluate the claim that rank might be connected to grokking. Such a description in terms of rank would nicely fit with other descriptions of grokking in terms of the simplification of linear decision boundaries (Humayun et al., 2024), the connection to double descent (Davies et al., 2022), and the discovery of a sparse solution (Merrill et al., 2023).

We mostly follow the setting of Nanda et al. (2023), optimizing a single-layer Transformer for modular addition (see Appendix A for exact details), except we use sinusoidal position embeddings instead of learned. As suggested by work in the deep linear case (Saxe et al., 2014; Arora et al., 2019; Milanesi et al., 2021; Yaras et al., 2023b), we plot singular value evolution for individual weight matrices, and to have a high-level view of all parameter evolutions we compute the (normalized) effective rank of a matrix W (Roy & Vetterli, 2007) with rank R as

$$\mathrm{EffRank}(W) := -\sum_{i=1}^{R} \frac{\sigma_i}{\sum_j \sigma_j} \log \frac{\sigma_i}{\sum_j \sigma_j} \ , \tag{1}$$

$$\mathrm{NormEffRank}(W) := \frac{\mathrm{EffRank}(W)}{R} \ , \tag{2}$$

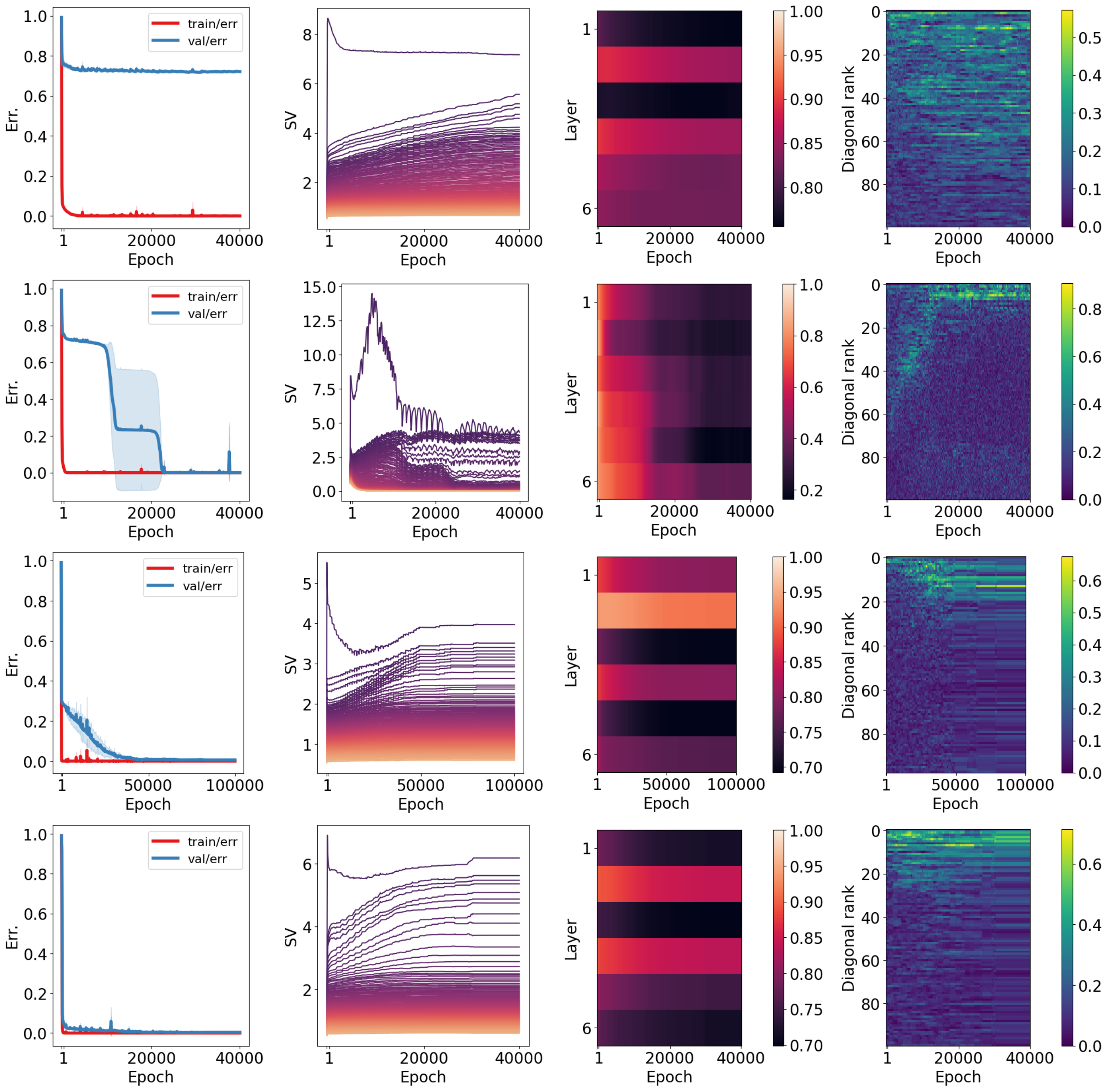
where σ_i are the singular values of matrix W and $\mathrm{EffRank}(W)$ is the entropy of the normalized singular value distribution. As the probability mass concentrates, the effective rank decreases. We plot $\mathrm{NormEffRank}(W)$ to compare across layers and time.

In addition, inspired by the assumptions of balancedness made by prior work (Arora et al., 2018; 2019), we examine the alignment of consecutive weight matrices in the Transformer. To examine and quantify this alignment between consecutive matrices in a neural network $W_i = \sum_{j=1}^{R} \sigma_j(t) u_j(t) v_j(t)^\top$, $W_{i+1} = \sum_{k=1}^{R} \sigma'_k(t) u'_k(t) v'_k(t)^\top$ at training time t, we compute

$$A(t)_{jk} = |\langle u_j(t), v'_k(t) \rangle|, \tag{3}$$

where the absolute value is taken to ignore sign flips in the SVD computation. We then plot the diagonal of this matrix $A(t)_{ii} \ \forall\, i \leq 100$ over time. For exact details on how alignment is computed for different architectures and layers that are more complex than the fully connected case, please see Appendix A.

In Figure 2, we see that the rapid decrease in the validation loss corresponds perfectly with the onset of the low-rank behavior from the perspective of the singular values. In tracking inter-layer alignment over the course of training, we see that the final low-rank solution develops slowly from the middle ranks of the model. By contrast, without weight decay, the grokking phenomenon never occurs, and it does not appear that a low-rank solution will develop. Grokking depends on only using a small portion of the data for training (Power et al., 2022; Nanda et al., 2023). When instead we use 90%, and no weight decay at all, generalization again co-occurs with effective rank minimization, further reinforcing the connection. We also replicate the setting of Thilak et al. (2022) who show a form of grokking without weight decay, but when plotted on a linear-scale x-axis the generalization appears much less sudden than the setting with weight decay. The familiar reader will note that Nanda et al. (2023) previously showed that the particular solution found in modular addition is a low rank fourier decomposition, so our observations on low rank weights will directly follow. While such a description for modular addition is impressively precise, it is difficult to obtain for more complex tasks. In following sections we argue that rank minimization is a perspective that can apply in more complex settings when one does not know what to look for



(a) Error (b) SVs (c) Eff. Rank (d) Alignment

Figure 2: **Top row:** 30% data and no weight decay. **2nd row:** 30% data and weight decay 1.0 (grokking), using hyperparameters from Nanda et al. (2023). **3rd row:** 70% data with no weight decay (slingshot), using hyperparameters from Thilak et al. (2022). **Bottom row:** 90% data and no weight decay. Singular value evolution is visualized for the first attention parameter, where each line represents a single singular value and the color represents the rank. Alignment (Eqn. 3) between the embedding and the first attention parameter is also visualized. One can see that grokking co-occurs with low-rank weights (effective rank is calculated according to Eqn. 1). In addition, there is an alignment that begins early in training that evolves up the diagonal. Without weight decay and with less data, neither grokking nor the other phenomena occur during the entire training budget, but using more data, even without weight decay, leads to low-rank solutions from the beginning of training. The slingshot case follows a similar trend, though the validation loss is fit more gradually. Across generalizing cases alignment is also more prevalent in the top ranks.

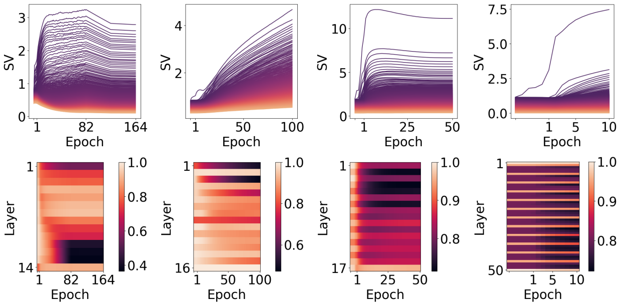
in the weights, and it may even be possible to eventually interpret the neural network via the top ranks (Praggastis et al., 2022).

Though our results do not completely discard the contribution of the norm to generalization, they point out a complication. One can see that without weight decay but with 90% of the data, the generalizing solution that is discovered has many singular values larger than 1, and the maximum singular value is also larger than that of the grokked solution, for which most singular values are near zero. Recall the norm of matrix W is given by $\|W\|_2 = \sqrt{\sum_i \sigma_i^2}$ where σ_i are the singular values. Thus, the large-data solution has a much higher norm than the grokked solution. Still, both settings generalize with a certain low-rank behavior. The confounding factor is that, with high weight decay, the smaller singular values disappear, while without, they do not. We do not know how to precisely tie low-rank behavior to generalization, nor will it always make sense as the solution might necessarily be high-rank, but it seems a tempting explanation from the perspective of Occam's Razor as a low-rank solution is "simpler" from the perspective of dimension.

4. Spectral Dynamics

Grokking is typically observed on quite small-scale systems with very particular hyperparameter settings (Power et al., 2022; Nanda et al., 2023; Gromov, 2023; Kumar et al., 2023), thus we wonder how the observed trend toward rank minimization scales to more complex systems. We also take dual inspiration from prior work on deep linear networks, which studies the evolution of the SVD of the weight matrices (Saxe et al., 2014; Arora et al., 2018; 2019; Milanesi et al., 2021; Yaras et al., 2023a) in simple cases. Thus we apply the same analysis to larger, more practical systems. We show that the trends we saw in the analysis of grokking mostly hold true across networks and tasks at a much larger scale, though our results do not always agree with the theory.

4.1. Methodology

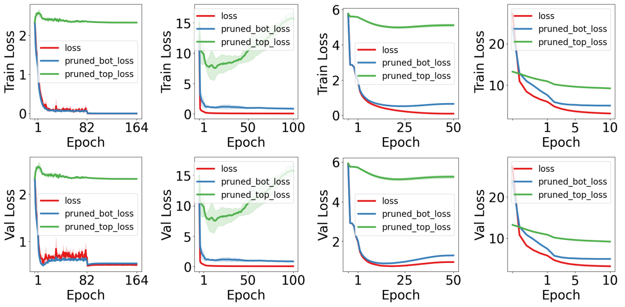


(a) VGG (b) UNet (c) LSTM (d) Transformer

Figure 3: **Top row:** Singular value evolution for a single matrix in the middle of each model. Each line represents a singular value, whereas color represents rank. Notice the unequal evolution where top singular values grow at a disproportionate rate. **Bottom row:** Normalized effective rank (Eqn. 1) evolution visualized in color for different matrices across architectures and time. As we move down the y-axis, the depth of the parameters in the model increases, while the x-axis tracks training time. Notice decreasing effective rank across nearly all parameters, though the magnitude differs across layers. The block-like patterns in the VGG case are likely due to different channel dimension sizes. The banding in the UNet, LSTM, and Transformer cases is due to the differences between convolutional and linear layers, residual block connections, and attention and fully connected layers, respectively. The sharp transition midway through training in the VGG case is likely due to a 10x learning rate decay.

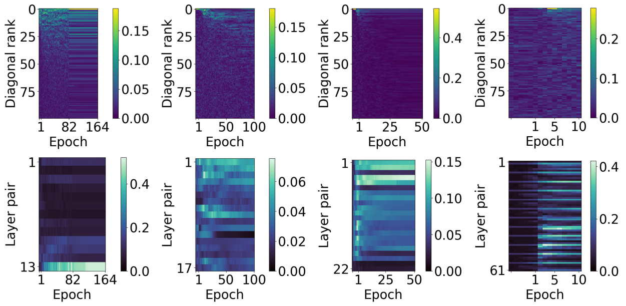
In all our experiments, we aim to study reasonably sized neural networks across a variety of tasks. We choose models and tasks to represent current applications. In particular, we select image classification with CNNs (VGG-16 (Simonyan & Zisserman, 2014)) on CIFAR10 (Krizhevsky, 2009), image generation through diffusion with UNets (Ronneberger et al., 2015) on MNIST (LeCun, 1998), speech recognition with LSTMs (Hochreiter & Schmidhuber, 1997b) on LibriSpeech (Panayotov et al., 2015), and language modeling with Transformers (Vaswani et al., 2017) on Wikitext-103 (Merity et al., 2016). These experiments require training hundreds of runs of the same model variant, so we are limited by computational constraints in the scale of models we examine. We primarily take hyperparameters from existing settings in the literature, making small modifications for simplicity. Thus, we intend that any correlations between settings will be a reflection of common practice as opposed to introduced bias on our part. We hope that the broad scope of these experiments will allow for a more general perspective on neural network optimization.

The bulk of the evidence presented comes from computing singular value decompositions (SVDs) of weight matrices in models. Thus, we ignore 1D bias and normalization parameters entirely in our analysis. There are also previously reported cases where these do not appear crucial for performance (Zhang et al., 2018b; Mohan et al., 2019; Karras et al., 2023). As there are many matrices in the models we study, we provide plots of individual layers' matrix parameters and statistics summarizing behavior across layers for conciseness of presentation. Hundreds of thousands of plots were generated for this study, so it is impossible to provide all the evidence. Please see Appendix A for exact experiment details, including hyperparameters. We will release code for all experiments.



(a) VGG (b) UNet (c) LSTM (d) Transformer

Figure 4: **Top row:** Training losses for all tasks. **Bottom row:** Validation losses for all tasks. Red is the full model. Blue is post-training pruning the bottom half of the SVD for every matrix in the model that is not the final layer. Green is post-training pruning the top half of the SVD. Notice that for all models, keeping the bottom half of the SVD is worse than keeping the top half, supporting the idea that the top directions provide a better approximation to the original function.



(a) VGG (b) UNet (c) LSTM (d) Transformer

Figure 5: Neighboring layer alignment of singular vectors. **Top row:** The diagonal of the alignment matrix $A(t)_{ii}$ (Eqn. 3) vs. training time for a single pair of matrices in the middle of each model. We see a small amount of alignment in the top ranks between layers shortly after training begins, but this becomes more diffuse over time. **Bottom row:** Alignment metric (Eqn. 4) for pairs of matrices for depth vs. training time. It is hard to make out a global trend across models, though the LSTM shows a weak signal around Epoch 1 when the initial alignment occurs, and the Transformer case has a banding pattern with depth due to alignment between the query and key matrices that have no nonlinearity in between.

4.2. Effective Rank Minimization

Expanding on theoretical (Saxe et al., 2014; Arora et al., 2019; Milanesi et al., 2021; Boix-Adserà et al., 2023; Yaras et al., 2023a) and empirical (Boix-Adserà et al., 2023; Martin & Mahoney, 2021; Dittmer et al., 2019) results, we examine effective rank minimization across parameters in both larger models and on a more diverse variety of tasks. In Figure 3, we see that effective rank tends to decrease as training proceeds, regardless of the parameter or network. This suggests that the network is becoming simpler as training proceeds.

To test if the low-rank picture is tightly related to model performance, we also prune either the top or bottom half of the singular values for every matrix in the network and evaluate that pruned model at every timestep. We might expect that the top singular values would be the best approximation to the function of the neural network, and we indeed see that this is the case in Figure 4, where the pruned parameters without any further training can be close approximations of the full parameters. It is a subtle point, but such an approximation might not be valid if pruning lower components led to some critical signal being lost while propagating forward, or if the many small but nonzero singular values were necessary for noise.

4.3. Alignment of Singular Vectors Between Layers

Similar to the work done in the grokking case, we look at alignment between consecutive layers in these larger neural networks. In addition to the alignment matrix derived in Eqn. 3, we derive and plot a scalar measure for alignment in the top diagonal of this matrix:

$$a(t) = \frac{1}{10}\sum_{i=1}^{10} A(t)_{ii} \quad . \tag{4}$$

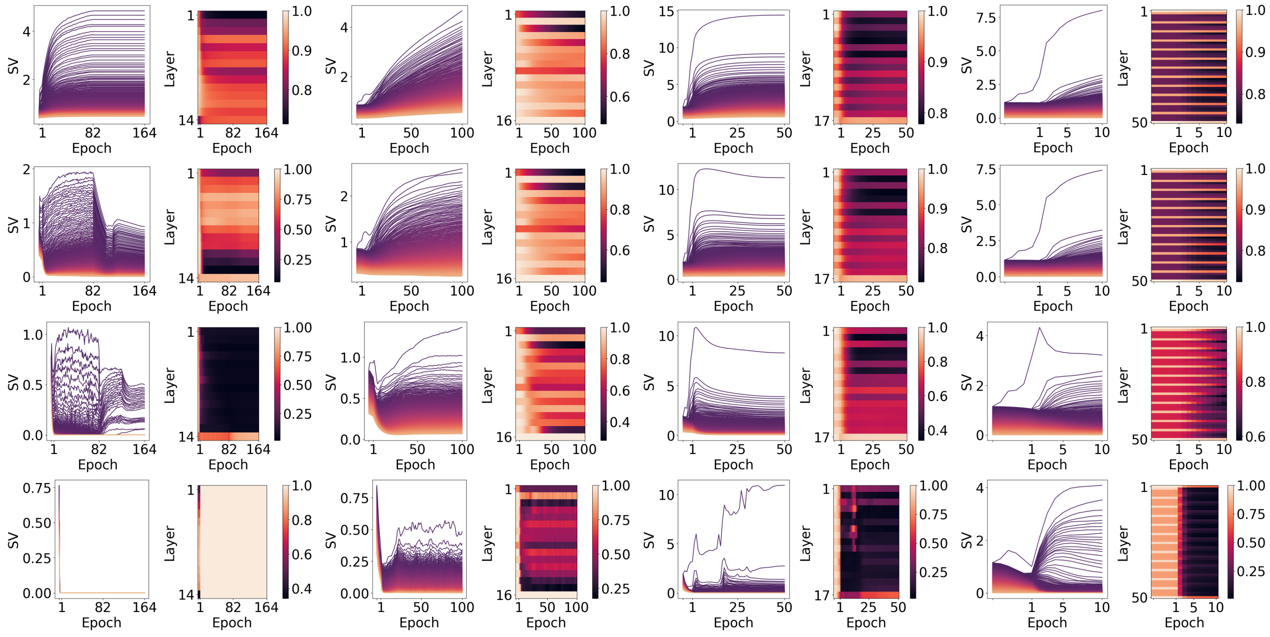
Again, for exact details on how this is computed for different architectures and layers that are more complex than the fully-connected case, please see Appendix A.

Figure 5 establishes that the theoretical assumption of balanced initialization (Arora et al., 2018; Saxe et al., 2014), which assumes aligned SVDs between weight matrices, is **not valid at the beginning of training**. Nor does it appear that alignment is static, like the linear case discussed by Du et al. (2018). All of this points to the fact that the assumptions upon which the theory is based do not hold in these larger-scale nonlinear settings, so the mechanism for rank decrease may be quite different.

We also point out that, even though there is a trend toward rank minimization, the effect is much weaker than in the grokking case. We previously observed that weight decay was a critical factor for the rank decrease, so we further examine weight decay.

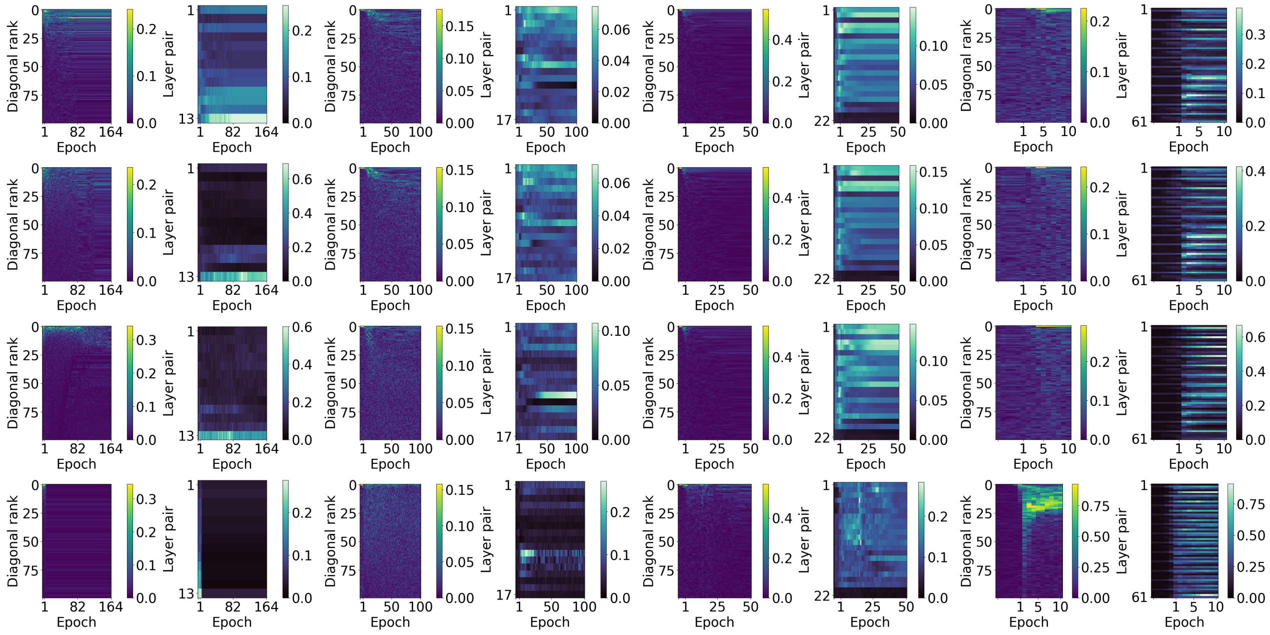
5. The Effect of Weight Decay

In light of the previously observed evolution of singular values, we investigate a proposed effect of weight de-



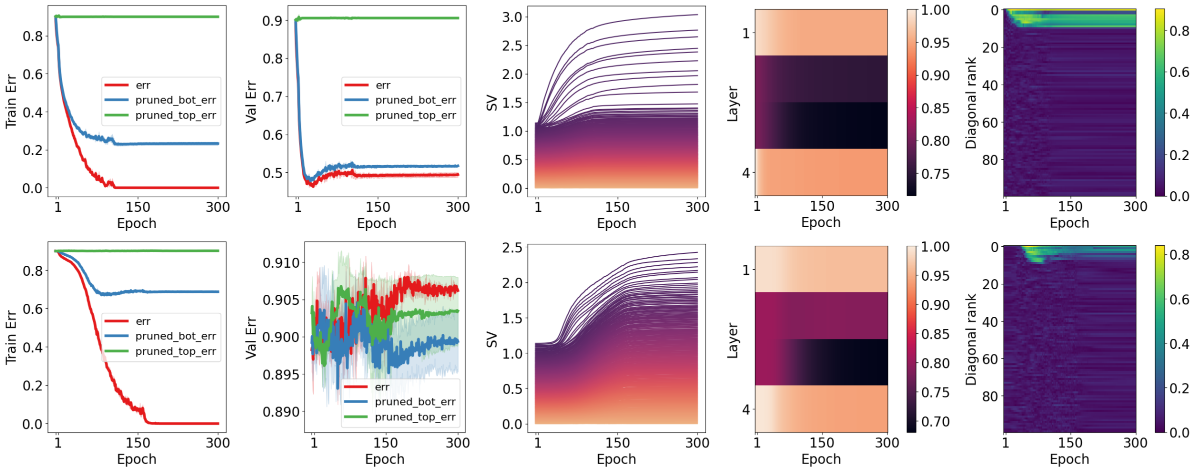
(a) VGG (b) UNet (c) LSTM (d) Transformer

Figure 6: SV evolution for a single matrix and normalized effective rank (Eqn. 1) across matrices over time, where the rows use differing amounts of weight decay. From top to bottom, for VGG we use coefficients $\{0, 0.001, 0.01, 0.1\}$, while for other networks we use coefficients $\{0, 0.1, 1, 10\}$. Higher weight decay coefficients promote more aggressive rank minimization. In the case of VGG, less weight decay is needed before norm collapse.



(a) VGG (b) UNet (c) LSTM (d) Transformer

Figure 7: Diagonal of alignment for a single pair over time (Eqn. 3) and alignment metric across pairs of matrices over time (Eqn. 4) where the y-axis represents depth. From top to bottom, for VGG we use coefficients $\{0, 0.001, 0.01, 0.1\}$, while for other networks we use coefficients $\{0, 0.1, 1, 10\}$. We see that the alignment magnitude is much higher with higher weight decay, and in particular, the Transformer has the strongest alignment even when nonlinearities separate the MLP layers.



(a) Train Err. (b) Val. Err. (c) SVs (d) Eff. Rank (e) Alignment

Figure 8: **Top row:** results with true labels. **Bottom row:** results with random labels. We see that the middle layers have a lower effective rank when using true labels and that alignment in the middle layers persists throughout training, unlike in the random label case.

cay. Though weight decay explicitly penalizes the norm of weights, there is empirical evidence that discards the connection between norm and generalization for neural networks (Razin & Cohen, 2020; Andriushchenko et al., 2023), meaning we do not have a full understanding as to why weight decay may be useful. Alternatively, some theoretical (Boix-Adserà et al., 2023; Razin & Cohen, 2020; Yaras et al., 2023a; Timor et al., 2023; Ongie & Willett, 2022; Galanti et al., 2022; Zangrando et al., 2024) and empirical works (Galanti et al., 2022; Boix-Adserà et al., 2023) propose a connection with the rank of matrices in constrained settings. Still, a comprehensive connection to larger empirical networks has not yet been demonstrated.

We speculate on the intuition of the mechanism in more practical settings. Notice in its simplest form that weight decay asks for arg $\min_W \mathcal{L}(W) + \lambda\|W\|_F^2$, where $\|W\|_F^2 = \sum_{i=1}^{R} \sigma_i^2$ with singular values σ_i of weight matrix W with rank R. We saw that larger singular values of neural networks grow faster (Fig. 3(top row)) and that the top singular vectors are much more useful for minimizing task loss than the bottom ones (Fig. 3). Thus, with minor weight decay regularization, one straightforward solution for the network may be to minimize the rank of a given weight matrix while preserving the top singular values to minimize $\mathcal{L}(W)$.

Figure 6 shows that adding weight decay produces this exact behavior, while too much weight decay leads to complete norm collapse. The exact choice of "too much" varies across architectures and tasks. In addition, even more surprisingly, large amounts of weight decay promote a tighter alignment in the top singular vectors of consecutive layers, which we see in Figure 7. This behavior is quite reminiscent of the balancedness condition (Arora et al., 2018; 2019; Du et al., 2018), though the networks considered here have nonlinearities and much more complex structures. We also provide additional evidence in Appendix A where Figure 9 shows that the solutions with very high weight decay are still performant, even though they are much lower rank.

6. Spectral Dynamics with Random Labels

Given the observations connecting generalization and rank in the grokking case, we are interested in seeing whether the perspective that we have developed sheds any light on the classic random label memorization experiments of Zhang et al. (2021).

Similar to Zhang et al. (2021), we train a simple MLP to fit random or true labels on CIFAR10. Please see Appendix A for the details regarding the experimental setup. Zhang et al. (2021) decay the learning rate to zero, and the random label experiments only converge late in training. Consequently, we use a constant learning rate to control this phenomenon. We see in Figure 8 that both cases are able to achieve zero error, though with different singular value evolution and alignment in the middle layer.

Surprisingly, we see that even without weight decay, with true labels, inner layers align, while with random labels, this alignment occurs and then disappears with more training. This is particularly intriguing as there are nonlinearities that could theoretically separate the network from the linear case, and yet quite strong alignment occurs despite that. We also see that the middle layer of the network trained on true labels has a lower effective rank, which may make sense as the data with true labels likely shares a common structure among classes. This further suggests that viewing generalization through the lens of rank may be fruitful.

7. Discussion

We provide an alternative view on the grokking phenomenon through the lens of SVD evolution, where we see that generalization coincides with the discovery of a low-rank solution in the weight matrices. We then observe that this tendency toward rank minimization exists on a much larger scale across natural tasks. We show that weight decay promotes this tendency toward low rank and provide additional evidence pointing out that generalization and memorization differ in the rank of solutions found by optimization.

Though we do not provide a comprehensive theory that explains these observations, we believe that such observations may form the basis for a deeper understanding of deep learning. Even without restricting assumptions like balancedness, linearity, or small initialization, the spectral dynamics are consistent across settings, so we believe there is likely a common cause.

Many natural questions remain open. There is great interest in the interpretability of models (Nanda et al., 2023), and there is already prior work on interpreting the singular vectors of convolutional weights (Praggastis et al., 2022). One may also wonder on the connection to other unexplained phenomena like double descent (Belkin et al., 2019; Nakkiran et al., 2021; Davies et al., 2022) or the lottery ticket hypothesis (Frankle & Carbin, 2018). Given the generality of our observations, we believe such directions may inform a more precise analysis of neural networks.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

DY would like to thank the following people for helpful discussion at many different stages: Wei Hu for a conversation on connecting LMC to rank dynamics in deep matrix factorization; Jonathan Frankle for very early support and periodic discussions on aiming to understand the source of LMC; Shubham Toshniwal and Shane Settle for support in following this line of work in the early stages; Mikhail Belkin, Tengyu Ma, and Rob Nowak for early conversations and encouragement on rank minimization in small-scale cases; Sudarshan Babu, Sam Buchanan, Haochen Wang, Takuma Yoneda, and Xiao Zhang for discussions and enthusiasm on resuming this work when it had been paused; Anmol Kabra for suggestions on additional experiments and visualizations; and David Mcallester for very helpful comments on presentation. DY would also like to thank Adam Bohlander for his work in maintaining the TTIC cluster, which made the many experiments in this paper possible. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1754881. KKP was supported through the NSF TRIPOD Institute on Data, Economics, Algorithms and Learning (IDEAL) and other awards from DARPA and NSF.

References

Andriushchenko, M., D'Angelo, F., Varre, A., and Flammarion, N. Why do we need weight decay in modern deep learning? *arXiv preprint arXiv:2310.04415*, 2023.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Atanasov, A., Bordelon, B., Sainathan, S., and Pehlevan, C. The onset of variance-limited behavior for networks in the lazy and rich regimes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

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.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

Biewald, L. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

Boix-Adserà, E., Littwin, E., Abbe, E., Bengio, S., and Susskind, J. M. Transformers learn through gradual rank increase. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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

Davies, X., Langosco, L., and Krueger, D. Unifying grokking and double descent. In *NeurIPS ML Safety Workshop*, 2022.

Dittmer, S., King, E. J., and Maass, P. Singular values for ReLU layers. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3594–3605, 2019.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ergen, T. and Pilanci, M. Path regularization: A convexity and sparsity inducing regularization for parallel relu networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Feng, R., Zheng, K., Huang, Y., Zhao, D., Jordan, M., and Zha, Z.-J. Rank diminishing in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Frei, S., Vardi, G., Bartlett, P., Srebro, N., and Hu, W. Implicit bias in leaky ReLU networks trained on high-dimensional data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Galanti, T., Siegel, Z. S., Gupte, A., and Poggio, T. SGD and weight decay provably induce a low-rank bias in neural networks. *arXiv preprint arXiv:2206.05794*, 2022.

Gromov, A. Grokking modular arithmetic. *arXiv preprint arXiv:2301.02679*, 2023.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Hanson, S. and Pratt, L. Comparing biases for minimal network construction with back-propagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1988.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural computation*, 9(1):1–42, 1997a.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997b.

Huh, M., Mobahi, H., Zhang, R., Cheung, B., Agrawal, P., and Isola, P. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2022.

Humayun, A. I., Balestriero, R., and Baraniuk, R. Deep networks always grok and here is why. *arXiv preprint arXiv:2402.15555*, 2024.

Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. *arXiv preprint arXiv:2312.02696*, 2023.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images, 2009.

Krogh, A. and Hertz, J. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1991.

Kumar, T., Bordelon, B., Gershman, S. J., and Pehlevan, C. Grokking as the transition from lazy to rich training dynamics. *arXiv preprint arXiv:2310.06110*, 2023.

Le, T. and Jegelka, S. Training invariances and the low-rank phenomenon: beyond linear networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

LeCun, Y. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

Liu, Z., Kitouni, O., Nolte, N. S., Michaud, E., Tegmark, M., and Williams, M. Towards understanding grokking: An effective theory of representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Liu, Z., Michaud, E. J., and Tegmark, M. Omnigrok: Grokking beyond algorithmic data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023a.

Liu, Z., Zhong, Z., and Tegmark, M. Grokking as simplification: A nonlinear complexity perspective. In *UniReps: the First Workshop on Unifying Representations in Neural Models*, 2023b.

Loshchilov, I. and Hutter, F. Fixing weight decay regularization in Adam. *arXiv preprint arXiv:1711.05101*, 2017.

Lyu, K., Jin, J., Li, Z., Du, S. S., Lee, J. D., and Hu, W. Dichotomy of early and late phase implicit biases can provably induce grokking. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Martin, C. H. and Mahoney, M. W. Heavy-tailed universality predicts trends in test accuracies for very large pre-trained deep neural networks. In *Proceedings of the SIAM International Conference on Data Mining (ICDM)*, 2020.

Martin, C. H. and Mahoney, M. W. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Merrill, W., Tsilivis, N., and Shukla, A. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

Milanesi, P., Kadri, H., Ayache, S., and Artières, T. Implicit regularization in deep tensor factorization. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2021.

Mohamadi, M. A., Li, Z., Wu, L., and Sutherland, D. Grokking modular arithmetic can be explained by margin maximization. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

Mohan, S., Kadkhodaie, Z., Simoncelli, E. P., and Fernandez-Granda, C. Robust and interpretable blind image denoising via bias-free convolutional neural networks. *arXiv preprint arXiv:1906.05478*, 2019.

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), 2021.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Notsawo Jr, P., Zhou, H., Pezeshki, M., Rish, I., Dumas, G., et al. Predicting grokking long before it happens: A look into the loss landscape of models which grok. *arXiv preprint arXiv:2306.13253*, 2023.

Ongie, G. and Willett, R. The role of linear layers in nonlinear interpolating networks. *arXiv preprint arXiv:2202.00856*, 2022.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

Parhi, R. and Nowak, R. D. Deep learning meets sparse regularization: A signal processing perspective. *IEEE Signal Processing Magazine*, 40(6):63–74, 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Pennington, J., Schoenholz, S., and Ganguli, S. The emergence of spectral universality in deep networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Praggastis, B., Brown, D., Marrero, C. O., Purvine, E., Shapiro, M., and Wang, B. The SVD of convolutional weights: a CNN interpretability framework. *arXiv preprint arXiv:2208.06894*, 2022.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Razin, N. and Cohen, N. Implicit regularization in deep learning may not be explainable by norms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Proceeding of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Roy, O. and Vetterli, M. The effective rank: A measure of effective dimensionality. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2007.

Saxe, A., McClelland, J., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Shenouda, J., Parhi, R., Lee, K., and Nowak, R. D. Vector-valued variation spaces and width bounds for DNNs: Insights on weight decay regularization. *arXiv preprint arXiv:2305.16534*, 2023.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

Tan, Z. and Huang, W. Understanding grokking through a robustness viewpoint. *arXiv preprint arXiv:2311.06597*, 2023.

Thilak, V., Littwin, E., Zhai, S., Saremi, O., Paiss, R., and Susskind, J. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

Timor, N., Vardi, G., and Shamir, O. Implicit regularization towards rank minimization in relu networks. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, 2023.

Van Laarhoven, T. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Vardi, G. and Shamir, O. Implicit regularization in relu networks with the square loss. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wang, B. and Vastola, J. ML from scratch: Stable diffusion, day 2, 2022. URL `https://colab.research.google.com/drive/1Y5wr91g5jmpCDiX-RLfWL1eSBWoSuLqO?usp=sharing#scrollTo=9is-DXZYwIIi`.

Wang, Z. and Jacot, A. Implicit bias of SGD in l_{2}-regularized linear DNNs: One-way jumps from high to low rank. *arXiv preprint arXiv:2305.16038*, 2023.

Waskom, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

Xu, Z., Wang, Y., Frei, S., Vardi, G., and Hu, W. Benign overfitting and grokking in ReLU networks for XOR cluster data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Yaras, C., Wang, P., Hu, W., Zhu, Z., Balzano, L., and Qu, Q. Invariant low-dimensional subspaces in gradient descent for learning deep matrix factorizations. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023a.

Yaras, C., Wang, P., Hu, W., Zhu, Z., Balzano, L., and Qu, Q. The law of parsimony in gradient descent for learning deep linear networks. *arXiv preprint arXiv:2306.01154*, 2023b.

Yu, H. and Wu, J. Compressing transformers: Features are low-rank, but weights are not! In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2023.

Zangrando, E., Deidda, P., Brugiapaglia, S., Guglielmi, N., and Tudisco, F. Neural rank collapse: Weight decay and small within-class variability yield low-rank bias. *arXiv preprint arXiv:2402.03991*, 2024.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhang, G., Wang, C., Xu, B., and Grosse, R. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018a.

Zhang, H., Dauphin, Y. N., and Ma, T. Fixup initialization: Residual learning without normalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018b.

Ziyin, L., Li, B., and Meng, X. Exact solutions of a deep linear network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

A. Experimental Details

For all experiments, we use 3 random seeds and average all plots over those 3. This is relatively small, but error bars tend to be very tight, and due to the high volume of runs required for this work we lack the resources to run much more.

In order to compute alignment we consider only pairs of layers that directly feed into each other, and ignore the influence of residual connections so as to cut down on the number of comparisons. Specifics on individual architectures are given below.

A.1. Image Classification with VGG

We train a VGG-16 (Simonyan & Zisserman, 2014) on CIFAR-10 (Krizhevsky, 2009) for 164 epochs, following hyperparameters and learning rate schedule in (Frankle et al., 2020), but without data augmentation. For the optimizer we use SGD with batch size 128, initial learning rate 0.1 and momentum of 0.9. We also decay the learning rate 3 times by a factor of 10 at epoch 82, epoch 120, and finally at epoch 160. We also use a minor amount of weight decay with coefficient 0.0001.

VGG-16 uses ReLU activations and batch normalization (Ioffe & Szegedy, 2015), and includes both convolutional and linear layers. For linear layers we simply compute the SVD of the weight matrix. For convolutional layers, the parameters are typically stored as a 4D tensor of shape $(c_{\text{out}}, c_{\text{in}}, h, w)$ for the output channels, input channels, height and width of the filters respectively. As the filters compute a transformation from each position and input channel to an output channel, we compute the SVD of the flattened tensor $(c_{\text{out}}, c_{\text{in}} \cdot h \cdot w)$, which maps all inputs to outputs, similar to Praggastis et al. (2022). This is not the SVD of the entire transformation of the feature map to the next feature map, but rather the transformation from a set of adjacent positions to a particular position in the next layer. For the individual SV evolution plot, we use the 12th convolutional layer.

In order to compute alignment of bases between consecutive convolutional layers, $V_{i+1}^{\top} U_i$ we need to match the dimensionality between U_i and V_{i+1}. For convolutional layers we are presented with a question as to how to handle the spatial dimensions h and w as naively the input dimension of the next layer will be a factor of $h \cdot w$ larger dimension. We experimented with multiple cases, including aligning at each spatial position individually or averaging over the alignment at all spatial positions, and eventually settled at aligning the output of one layer to the center spatial input of the next layer. That is, for a 3x3 convolution mapping to a following 3x3 convolution, we compute the alignment only for position (1,1) of the next layer. This seemed reasonable to us as on average the edges of the filters showed poorer alignment overall. For the individual alignment plot, we use the alignment between the 11th and 12th convolutional layers at the center spatial position of the 12th convolutional layer.

A.2. Image Generation with UNets

We train a UNet (Ronneberger et al., 2015) diffusion model (Sohl-Dickstein et al., 2015; Ho et al., 2020) on MNIST (LeCun, 1998) generation. We take model design and hyperparameters from (Wang & Vastola, 2022). In particular we use a 4-layer residual UNet and train with AdamW (Loshchilov & Hutter, 2017) with batch size 128, and learning rate of 0.0003 for 100 epochs. This model uses swish (Ramachandran et al., 2017) activations and a combination of linear and convolutional, as well as transposed convolutional layers.

Computing SVDs and alignment is similar to the image classification case described above, except in the case of the transposed convolutions where an extra transpose of dimensions is needed as parameters are stored with the shape $(c_{\text{in}}, c_{\text{out}}, h, w)$. For the individual SV evolution plot, we use the 3rd convolutional layer. For the alignment plot, we use the alignment between the 3rd and 4th convolutional layers at the center spatial position of the 4th convolutional layer.

A.3. Speech Recognition with LSTMs

We train a bidirectional LSTM (Hochreiter & Schmidhuber, 1997a) for automatic speech recognition on LibriSpeech (Panayotov et al., 2015). We tune for a simple and well-performing hyperparameter setting. We use AdamW (Loshchilov & Hutter, 2017) with batch size 32, learning rate 0.0003 and weight decay 0.1 for 50 epochs. We also use a cosine annealing learning rate schedule from 1 to 0 over the entire 50 epochs.

The LSTM only has matrix parameters and biases, so it is straightforward to compute SVDs of the matrices. For individual SV evolution plots, we plot the 3rd layer input parameter. In the case of alignment, we make a number of connections: first down depth for the input parameters, then connecting the previous input parameter to the current hidden parameter in both directions, then connecting the previous hidden parameter to the current input parameter. For the individual layer alignment, we plot alignment between the 3rd and 4th layer input parameters.

A.4. Language Modeling with Transformers

We train a Transformer (Vaswani et al., 2017) language model on Wikitext-103 (Merity et al., 2016). We base hyperparameter choices on the Pythia suite (Biderman et al., 2023), specifically the 160 million parameter configuration with sinusoidal position embeddings, 12 layers, model dimension 768, 12 attention heads per layer, and hidden dimension 768. We use AdamW (Loshchilov & Hutter, 2017) with batch size 256, learning rate 0.0006 and weight decay 0.1. We use a context length of 2048 and clip gradients to a maximum norm of 1. We also use a learning rate schedule with a linear warmup and cosine decay to 10% of the learning rate, like Biderman et al. (2023).

For SVDs, for simplicity we take the SVD of the entire $(3d_{\text{model}}, d_{\text{model}})$ parameter that computes queries, keys and values from the hidden dimension inside the attention layer, without splitting into individual heads. This is reasonable as the splitting is done after the fact internally. We also take the SVD of the output parameters, and linear layers of the MLPs, which are 2 dimensional matrices. For the individual SV evolution plot, we plot the SVs of W_1 of the 8th layer MLP

For alignment, we consider the alignment of W_Q and W_K matrices, W_V and W_O matrices, the alignment between W_O and W_1 of the MLP block, between W_1 and W_2 of the MLP block, and between W_2 and the next attention layer. For the individual layer alignment, we plot alignment between W_1 and W_2 of the 8th layer MLP.

A.5. Weight Decay Experiments

All tasks are trained in exactly the same fashion as mentioned previously, with increasing weight decay in the set $\{0, 0.0001, 0.001, 0.01, 0.1, 1.0, 10.0\}$. For ease of presentation we consider a subset of settings across tasks. In Figure 9 we include trained model performance and pruned model performance to show that, even with high levels of weight decay, models do not entirely break down. More so, the approximation of the pruned model to the full model gets better with higher weight decay.

A.6. Grokking Experiments

We mostly follow the settings and architecture of Nanda et al. (2023), except we use sinusoidal positional encodings instead of learned.

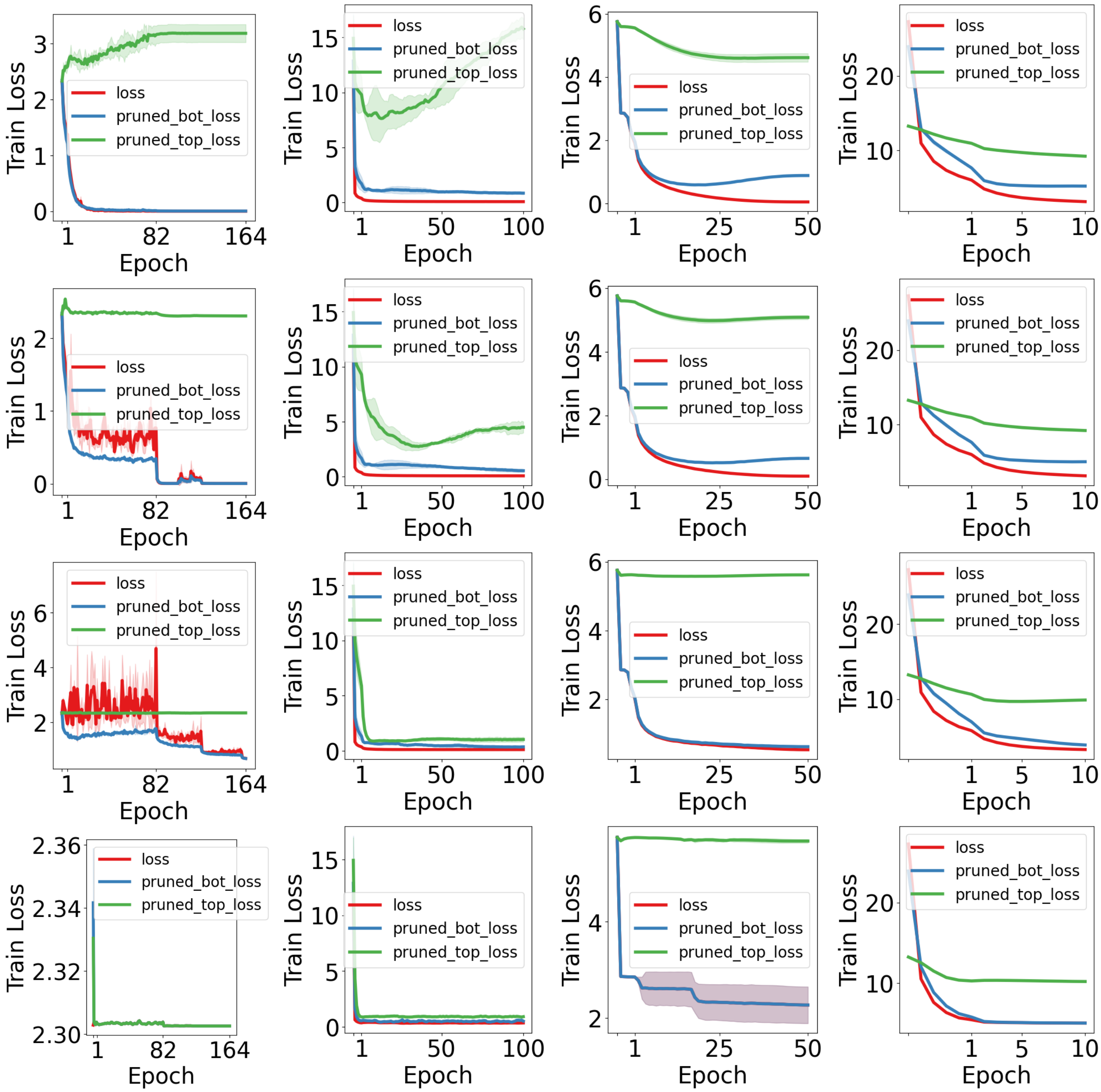
For the slingshot case we follow hyperparameter settings in Thilak et al. (2022), Appendix B except with the 1-layer architecture from Nanda et al. (2023) instead of the 2-layer architecture specified. W perform addition modulo 97. The original grokking plot in Thilak et al. (2022) appears much more dramatic as it log-scales the x-axis, which we do not do here for clarity.

A.7. Random Label Experiments

We train a 4-layer MLP on CIFAR10 (Krizhevsky, 2009) with either completely random labels, or the true labels. We use SGD with momentum of 0.9 and constant learning rate of 0.001, and train for 300 epochs to see the entire trend of training. The major difference to the setting of Zhang et al. (2021) is the use of a constant learning rate, as their use of a learning rate schedule might conflate the results.

B. Limitations

There are a few key limitations to our study. As mentioned, we lack the computational resources to run more than 3 random seeds per experiment, though we do find error bars to be quite tight in general (except for the generalization epoch in the grokking experiments). In addition, as discussed we ignore 1D parameters in the neural networks, which may be particularly crucial (especially normalization). In addition, due to computational constraints we do not consider alignment of layers across residual connections as this quickly becomes combinatorial in depth, thus there may be other interesting interactions that we do not observe. Finally, due to computational constraints we are unable to investigate results on larger models than the 12 layer Transformer, which may have different behavior.



(a) VGG

(b) UNet

(c) LSTM

(d) Transformer

Figure 9: Training loss over time, where the rows use differing amounts of weight decay. From top to bottom, for VGG we use coefficients $\{0, 0.001, 0.01, 0.1\}$, while for other networks we use coefficients $\{0, 0.1, 1, 10\}$. We see that it is still possible to achieve low training loss under high weight decay, and as we increase the amount of weight decay, the gap between pruned and unpruned parameters closes, lending support to the idea that the parameters become lower rank.

C. Compute Resources

All experiments are performed on an internal cluster with on the order of 100 NVIDIA 2080ti GPUs or newer. All experiments run on a single GPU in less than 8 hours, though it is extremely helpful to parallelize across machines. We estimate that end-to-end it might take a few days on these resources to rerun all of the experiments in this paper.

D. Code Sources

We use PyTorch (Paszke et al., 2019), NumPy (Harris et al., 2020) for all experiments and Weights & Biases (Biewald, 2020) for experiment tracking. We make plots with Matplotlib (Hunter, 2007) and Seaborn (Waskom, 2021).

E. Contribution Statement

DY was responsible for, and primarily contributed to, all stages of the project including idea generation, experimentation, plotting and writing. KKP was involved in detailed feedback on experiments and help with presentation. SW proposed certain plots and helped with development. PS provided early feedback and help with preliminary experiments. GV edited and provided guidance on literature the theoretical side. KL provided early feedback and advising. MM provided feedback and advising. MW primarily advised the project and helped with writing.