Grokking Interpretability: Understanding Grokking in Deep Neural Networks

Hugo Latourelle-Vigeant
Department of Statistics and Data Science
Yale University
hugo.latourelle-vigeant@yale.edu

Nahom Seyoum
Department of Statistics and Data Science
Yale University
nahom.seyoum@yale.edu

Abstract

Grokking describes an interesting phenomenon in which a neural network, after long periods of near-random validation performance despite perfect training accuracy, suddenly clicks and achieves excellent generalization. This transition often occurs well beyond the point where overfitting is expected– contrary to what is possible in classical learning theory. Recent works offer multiple, sometimes conflicting, explanations: from implicit regularization in highly overparameterized models, to artifacts of discrete accuracy metrics, to the role of weight decay, and to delayed feature-learning phases. In this report, we outline these explanations and discuss their strengths and limitations.

1 Background

Let $\{(x_i, y_i)\}_{i=1}^n \sim \mathcal{D}^{\otimes n}$ denote a training dataset of n i.i.d. samples drawn from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Then, supervised learning aims to find a predictor $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the *population risk*

$$\mathcal{R}(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f(x), y)] \tag{1}$$

where $\ell : \mathcal{Y}^{\otimes 2} \to \mathbb{R}_{\geq 0}$ is a *loss function*. Because we often do not have access to the population risk, we instead rely on empirical risk minimization (ERM) which minimizes the empirical risk

$$\widehat{\mathcal{R}}_n(f) := \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) \tag{2}$$

over a hypothesis class $\mathcal{F}$.

In this framework, statistical guarantees classically rely on two central ideas: *uniform convergence* and *convexity*. Uniform convergence ensures that, with high probability, the empirical risk $\widehat{\mathcal{R}}_n(f)$ is uniformly close to the population risk $\mathcal{R}(f)$ over the hypothesis class provided $\mathcal{F}$ is suitably constrained in capacity. This leads to a classic tradeoff: $\mathcal{F}$ must be rich enough to model the training data, but not so expressive as to overfit. Convexity ensures that the ERM problem is computationally tractable, and that global minimizers can be efficiently found. Well-known examples of supervised learning models that fall within this theory include splines, support vector machines, and kernel methods.

However, this classical picture breaks down in the context of deep learning. Deep neural networks often operate in highly overparameterized regimes, where the function class is expressive enough to fit arbitrary labels—including pure noise [Zha+16]. Moreover, the associated optimization problems are typically highly non-convex, and models are frequently trained to the point of interpolation, i.e., $\widehat{\mathcal{R}}_n(f) \approx 0$. Yet, such models often generalize remarkably well to unseen data. This empirical success has motivated the development of new theoretical tools to better understand generalization in

deep learning. These include analyses of the *implicit bias* of optimization algorithms and forms of *implicit regularization*. While significant insights have been gained, a complete theory explaining the generalization capabilities of deep neural networks remains one of the most intriguing open problems in modern statistical learning theory.

Among the intriguing phenomena observed is *grokking*, which was introduced in [Pow+22]. Grokking, which means "to understand profoundly and intuitively" [Mer25], refers to the process in which, during training, a model suddenly exhibits improved generalization long after achieving perfect training accuracy. This transition can occur abruptly and well after the point of overfitting, making it particularly interesting in light of practices such as *early stopping*, which aim to prevent overfitting by halting training once the validation error plateaus or deteriorates. Since its discovery, grokking has sparked considerable interest, with several empirical studies seeking to uncover its underlying mechanisms. In this paper, we review some of the prominent explanations proposed in the literature, highlighting their strengths and limitations. We also provide theoretical insights and supporting experiments that shed light on the phenomenon, with the aim of contributing to a deeper understanding of the mechanism enabling grokking.

2 Possible Explanations

In this section, we explore several proposed explanations for the phenomenon of grokking, as discussed in recent literature. We aim to provide an overview of the key ideas, while also highlighting some limitations or gaps in the existing theories.

2.1 Grokking via Implicit Regularization

In [Pow+22], the authors train small transformer models on artificially constructed binary-operation tasks (e.g., modular arithmetic, group multiplication in S_5) and observe that the network rapidly memorizes all training examples—reaching near-100% training accuracy in as few as 10^3 to 10^4 gradient steps—but stays near-chance validation accuracy for up to 10^5 or 10^6 steps. Past some much larger threshold in training iterations, validation accuracy suddenly jumps from near random to nearly perfect, as illustrated in Figure 1. They refer to this sudden jump as grokking.[1].

[1] In addition to grokking, they observe that when they train on a fraction α (e.g., 50%) of all possible equations in a discrete operation table, they see a dramatic increase in training iterations required for generalization as α decreases. Even though fewer samples should, in principle, make memorization easier, it apparently makes discovering a generalizing solution far harder.

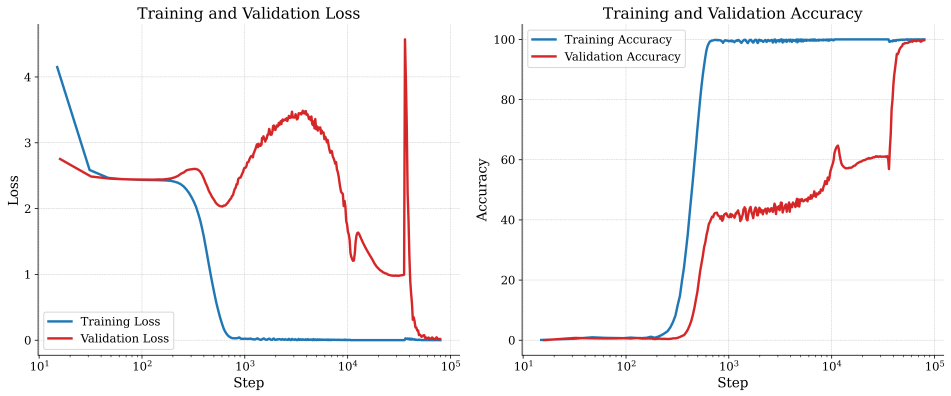

Figure 1: Illustration of grokking in the setting of [Pow+22]. **Left**: Training and validation loss over the course of training. The validation loss exhibits highly non-monotonic behavior, with irregular fluctuations. **Right**: Training and validation accuracy. The training accuracy rapidly saturates, indicating near-perfect memorization after approximately 10^2 iterations. In contrast, the validation accuracy undergoes a sudden jump from trivial to around 40%, followed by a slow increase and a second abrupt transition from 60% to near-perfect accuracy after more than 10^4 iterations. See Appendix A for details on the experimental setup.

From these experiments, they hypothesize that a network first finds a trivial memorizing solutionlikely in a very sharp region of parameter space—and only later drifts toward a simpler function that explains the training data and generalizes. This is precisely the hallmark of implicit regularization.

2.1.1 Implicit Regularization

Let $\theta \in \Theta \subseteq \mathbb{R}^p$ denote the parameters of the transformer architechture under consideration and suppose that the hypothesis class is $\mathcal{F} = \{f_\theta : \mathcal{X} \to \mathcal{Y} : \theta \in \Theta\}$ In [Pow+22], $\mathcal{X}$ is a set of abstract symbols, and $\mathcal{Y}$ is the set of possible outputs for the binary operation and the loss ℓ refers to the cross-entropy loss.

Because these networks have far more parameters than training samples, they can easily interpolate the data, meaning $\widehat{\mathcal{R}}_n(f) = 0$, or equivalently $f_\theta(x_i) = y_i$ for every $i \in [n]$. Let $\mathcal{M} = \{\theta \in \Theta : \widehat{\mathcal{R}}_n(f) = 0\}$ be the set of interpolating solutions. By virtue of overparameterization, $\mathcal{M}$ is expected to contain many interpolationg solutions. Intuitively, some solutions in $\mathcal{M}$ memorize the data in an ad hoc manner (leading to poor validation performance), while others encode the true arithmetic structure and yield near-100% validation accuracy. Implicit regularization then refers to the bias due to the optimization algorithhm to select a "good" solution in $\mathcal{M}$.

2.1.2 A Two-Phase Training Dynamic

In view of implicit regularization, the authors of [Pow+22] suggest that the optimization trajectory can be decomposed into two phases.

During the first *rapid memorization phase*, the network rapidly memorizes the partial operation table. This is typically straightforward for relatively large, flexible models such as the one under consideration here. At the end of this phase, the weights θ reach some $\theta_{\text{mem}} \in \mathcal{M}$. At this point, training accuracy is effectively 100%, yet θ_{mem} may correspond to a "sharp" or complex solution that fails to generalize.

During the second phase, the weights shift from the sharp local minima to a flatter solution which generalizes better. Indeed, since θ_{mem} is thought to correspond to a sharp local minima, the noise from the stochastic optimization procedure may push the model from a narrow (memorizing) basin to a flatter, simpler part of $\mathcal{M}$. In the language of "flat minima," such simpler solutions are often robust to small parameter perturbations and more likely to capture the underlying group-theoretic or arithmetic rule. The moment θ crosses a threshold near this simpler solution, the network's outputs on unseen pairs (x, y) match the correct operation, and validation accuracy jumps.

In order to support their hypothesis, especially regarding the second phase of training described above, the measure the sharpness of the minimum found by a trained network [HS97]. In particular, [Pow+22, Appendix A.5] shows that solutions with near-perfect validation tend to have lower "sharpness" scores.

2.1.3 Limitations of the Implicit Regularization Explanation

We conclude this section by highlighting some potential limitations of the hypothesis that grokking arises solely due to implicit regularization. First, the authors of [Pow+22] note that as the dataset size increases, the training and validation loss curves tend to align more closely.[2] This observation poses a challenge to their framework, as real-world applications typically involve large datasets.

Second, other studies on grokking, such as [Kum+24; LBB23] which we discuss below, have demonstrated the phenomenon using full-batch gradient descent. While this does not preclude the involvement of implicit regularization, it does suggest that grokking cannot be solely attributed to the stochasticity inherent in the optimization process.

[2]Although we did not conduct exhaustive experiments, we attempted to replicate this observation by running their experiment with a larger modulo but did not obtain the same conclusion. We instead observed a similar behavior to that shown in the figure from the article, with the training accuracy approaching perfection later in training.

2.2 Grokking via Accuracy Metric

In [LBB23], the authors provide both empirical and semi-analytic evidence suggesting that grokking, in this setting, is largely an artifact of the accuracy metric and does not correspond to a meaningful notion of sudden "understanding."

2.2.1 Linear Regression Setting

In its simplest form, they demonstrate that even a basic linear regression model can exhibit grokking behavior. Specifically, they consider a dataset where $x_i \sim \mathcal{N}(0, I) \in \mathbb{R}^d$, and the corresponding labels are given by $y_i = Tx_i$, with $T \in \mathbb{R}^d$ being a *teacher vector* whose entries are i.i.d. samples from $\mathcal{N}(0, (2d)^{-1})$.[3] The analysis is carried out in the *proportional regime*, where the ambient dimension and sample size satisfy $d/n = \lambda \in \mathbb{R}_{>0}$. The objective is to recover the teacher vector T from the data.

To do so, the authors consider the natural parametric family of functions $\mathcal{F} = \{x \mapsto S^\top x : S \in \mathbb{R}^d\}$, where S is referred to as the *student vector*. The learning objective is to minimize the mean squared error (MSE) loss $\ell(x, y) = (x - y)^2$. Plugging this into the expressions for the population and empirical risks gives

$$\mathcal{R}(S) = \mathbb{E}_{x\sim\mathcal{N}(0,I_d)}[(T^\top x - S^\top x)^2] = \|D\|^2, \quad \text{and} \quad \widehat{\mathcal{R}}_n(S) = \frac{1}{n}\sum_{j=1}^{n}(S^\top x_j - y_j)^2 = \|D\|^2_{\widehat{\Sigma}},$$

where $D = S - T$ denotes the difference between the student and teacher vectors, and $\widehat{\Sigma} = \frac{1}{n}\sum_{j=1}^{n} x_j x_j^\top \in \mathbb{R}^{d\times d}$ is the *empirical covariance matrix* of the data.[4]

2.2.2 Closed-form Gradient Flow Dynamics

To minimize $\widehat{\mathcal{R}}_n(S)$ over $\mathcal{F}$, the authors consider full-batch gradient descent with step size $\eta \in \mathbb{R}_{>0}$, initialized at $S_0 \in \mathbb{R}^d$, a vector whose entries are i.i.d. samples from $\mathcal{N}(0, (2d)^{-1})$. In the *gradient flow* limit (i.e., setting $\eta = \eta_0\,\mathrm{d}t$), the dynamics of the student-teacher difference evolve according to $\frac{\mathrm{d}}{\mathrm{d}t}D(t) = -2\eta_0\widehat{\Sigma}D(t)$, which admits the closed-form solution

$$D(t) = e^{-2\eta_0\widehat{\Sigma}t}D_0,$$

where $D_0 = S_0 - T$ is the difference at initialization, distributed as a vector with i.i.d. $\mathcal{N}(0, d^{-1})$ entries. Using this closed-form expression, we can express the evolution of both the population and empirical risks during training as

$$\mathcal{R}(S_t) = D_0^\top e^{-4\eta_0\widehat{\Sigma}t}D_0 \quad \text{and} \quad \widehat{\mathcal{R}}_n(S_t) = D_0^\top e^{-4\eta_0\widehat{\Sigma}t}\widehat{\Sigma}D_0.$$

Here, we have used the fact that $\widehat{\Sigma}$ and $e^{-2\eta_0\widehat{\Sigma}t}$ commute, since $X \mapsto e^X$ is a spectral function[5] and $\widehat{\Sigma}$ is symmetric.

Hence, both $\mathcal{R}(S_t)$ and $\widehat{\mathcal{R}}_n(S_t)$ are *linear spectral statistics* of the empirical covariance matrix $\widehat{\Sigma}$. In the high-dimensional regime where $n, d \to \infty$ with $d/n \to \lambda \in \mathbb{R}_{>0}$, the eigenvalue distribution of $\widehat{\Sigma}$ converges (almost surely) to the *MarchenkoPastur distribution* with parameter λ, which we define in Appendix B. Leveraging this, we may derive deterministic approximations for $\mathcal{R}(S_t)$ and $\widehat{\mathcal{R}}_n(S_t)$ in the large-sample limit. The following result, which is heuristically derived in [LBB23] and that we formally proved in Appendix B, makes this precise.

Proposition 1. *As $n, d \to \infty$ with $d/n = \lambda$,*

$$\mathcal{R}(S_t) \to \int e^{-4\eta_0 zt}\mu_{\mathrm{MP}}(\mathrm{d}z) \quad \textit{and} \quad \widehat{\mathcal{R}}_n(S_t) \to \int e^{-4\eta_0 zt}z\mu_{\mathrm{MP}}(\mathrm{d}z)$$

almost surely.

[3] They also analyze a more general model in which $T \in \mathbb{R}^{d_{\text{in}}\times d_{\text{out}}}$ is a *teacher matrix*, and provide empirical evidence that similar phenomena occur in deeper networks with nonlinear activations. For clarity of exposition, we focus solely on the simplest case.

[4] For a positive-definite matrix $C \succ 0$, we denote $\|a\|_C = \sqrt{a^\top Ca}$ the norm induced by the inner product $\langle a, b\rangle_C = a^\top Cb$.

[5] If $X = U\mathrm{diag}(\{\lambda_j\}_{j=1}^d)U^\top$ is a spectral decomposition of X, then $e^X = U\mathrm{diag}(\{e^{\lambda}_j\}_{j=1}^d)U^\top$

2.2.3 Accuracy in Regression Tasks

The authors of [LBB23] next define a notion of *accuracy* as the fraction of points whose prediction error is smaller than $\epsilon \in \mathbb{R}_{>0}$. Then, one may show that

$$\mathcal{A} = \mathbb{E}_{x\sim\mathcal{N}(0,I)}\left[\Theta\left(\epsilon - (D^\top(t)x)^2\right)\right] \to \mathrm{Erf}\left(\sqrt{\frac{\epsilon}{2\mathcal{R}(S_t)}}\right)$$

and

$$\widehat{\mathcal{A}} = \frac{1}{n}\sum_{j=1}^{n}\Theta\left(\epsilon - (D^\top(t)x_j)^2\right) \to \mathrm{Erf}\left(\sqrt{\frac{\epsilon}{2\widehat{\mathcal{R}}_n(S_t)}}\right)$$

where Θ is the heavyside step function, in the large number of sample limit [LMT23].

Using the closed-form expressions for the asymptotic empirical and population losses given in Proposition 1, the authors are able to precisely characterize when $\widehat{\mathcal{R}}_n(S_t)$ falls below a fixed threshold ϵ before $\mathcal{R}(S_t)$ does. Importantly, this phenomenon does not signal any particularly meaningful transition in the dynamics of the loss itself. Instead, the apparent "grokking" behavior emerges due to the discontinuous nature of the accuracy metric, which masks the gradual improvement in population loss until a sharp change in classification performance is observed.

2.2.4 Limitations of the Accuracy Metric Explanation

In our view, there are two main limitations in their work. First, since they focus on linear regression, the model they study semi-analytically has very limited expressivity. In particular, it cannot be meaningfully overparameterized, and therefore fails to exhibit several hallmark phenomena of deep learning, such as double descent. Second, the notion of "accuracy" they use, which relies on thresholding the output to create a classification-like metric, is somewhat artificial in the context of a regression task.

2.3 Grokking via Multiscale Training Dynamics

[Kum+24] instead hypothesized that grokking may arise due to a transition from lazy to rich learning. That is, because during training, a model first acts as "lazy" and attempts to fits the training data without adapting the first layer weights. Then, as the number of steps increases, the model adapts to an underlying structure, which leads to improved generalization error. This is somewhat reminiscent of the decoupling of training phases discussed in Section 2.1.2 and that appeared in other theoretical works such as [MU25].

2.3.1 Polynomial Regression Setting

To study this model, the author study a simple setting in which they train a high dimensional polynomial regression model using gradient descent to fit a two layers neural network.[6] Specifically, they focus on a quadratic target of the form $y = \frac{1}{2}(\beta_\star^\top x)^2$ and consider networks of the form $f(x) = \alpha\sum_{j=1}^{n}\varphi(w_j^\top x)$ with activation function $\varphi(h) = h + \frac{\epsilon}{2}h^2$ and scale parameter α that controls how quickly the model's internal features (the hidden-layer weights w_i) adapt during gradient descent.

2.3.2 Kernel Alignment and the Transition From Lazy To Rich Learning

Crucially, they quantify how well the target y aligns with the neural tangent kernel (NTK) of the network at initialization, denoted K_0 [JGH18]. This is done via the Centered Kernel Alignment (CKA), defined by

$$\mathrm{CKA}(K_0, y) = \frac{y^\top K_0 y}{\|K_0\|_F\|y\|_2}.$$

When this alignment is low, the kernel-based (lazy) dynamics struggle to capture the target function; however, given sufficient training iterations and data, the network eventually breaks out of its kernel-like regime to learn improved internal features. This late-stage feature learning leads to a sudden drop in the test loss – essentially grokking.

[6] The architecture is often referred to as a "two-layer" network, despite the absence of output weights.

Moreover, by adjusting α, one can continuously tune the speed and severity of this transition, thus controlling how long the network remains "lazy" and when it finally switches to a "rich," feature-learning phase. The authors, through a careful loss decomposition (separating contributions from linear vs. quadratic weight statistics) and by tracking the parameter evolution, also show that α and the initial alignment jointly govern whether or not grokking emerges.

2.3.3 Parallel Conclusions for Addition Modulo

The conclusions of [Nan+23] exhibit notable parallels to ours. They investigate the phenomenon of grokking in the context of addition modulo, following earlier work such as [Pow+22]. Their approach centers on reverse engineering the learned representations of a small transformer that exhibits grokking behavior on this task.

A key contribution of their work is the identification of two distinct *progress measures*, which enable a more mechanistic interpretation of training dynamics:

(i) The *restricted loss* measures the error associated with the "optimal" sparse solution, effectively isolating the performance of the generalizable solution during training;

(ii) The *excluded loss* measures the error contributed by the remaining weights, capturing the parts of the model that deviate from this optimal representation.

Using these progress measures, the authors segment training into three phases. In the initial phase, the model memorizes the training data, rapidly reducing the training error to near zero. However, both the restricted and excluded losses remain high during this period, indicating that generalization has not yet emerged. In the second phase, the restricted loss begins to decrease, signaling that the model is starting to align with the sparse, generalizable structure of the task. Simultaneously, the excluded loss increases, suggesting that the model is reallocating capacity away from the less useful components. Finally, in the third phase, the restricted loss continues to decline while the excluded loss eventually plateaus or decreases, indicating a consolidation of the generalizing solution. This gradual shift reflects a smooth transition from memorization to a generalizable representation. This mirrors conclusions in other settings where grokking arises.

2.3.4 Limitations of the Mutliscale Explanation

We find the explanation that training can be meaningfully split into multiple phases to be particularly compelling. This perspective provides a unifying narrative that also encompasses earlier explanations, such as those in [Pow+22].

However, this explanation remains incomplete, as it relies on complex internal mechanisms that are not yet fully understood. While insightful, the phase-based view should be seen as one step toward a deeper mechanistic theory of grokking.

2.4 Grokking via Weight Decay

Earlier works, such as [LMT23], suggest that grokking arises from delayed learning in the networks representations: the model initially memorizes the training data and only later transitions to a more generalizable understanding. They also propose that weight decay facilitates grokking by penalizing large weights, thereby encouraging simpler, more generalizable solutions. Weight decay acts as an explicit regularizer by adding an ℓ_2 penalty term, $\lambda\|w\|_2^2$, to the loss function.

Interestingly, other studies [LBB23; Pow+22] observed that incorporating weight decay can significantly reduces the number of training samples required for generalization to emerge. Additionaly, as shown by [Kum+24] and subsequent works, weight decay is not a necessary condition for grokking. Hence, while it may correlate with the emergence of generalization in some cases, it is not fundamentally required for the phenomenon to occur.

2.5 Conclusion

Grokking remains a deep phenomenon. Multiple explanations (e.g. implicit bias, multiscale dynamics, explicit regularization) capture different facets, but none fully explain it. A complete understanding will require further work unifying these perspectives.

References

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Flat Minima". In: *Neural Computation* 9.1 (Jan. 1997), pp. 1–42. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.1.1. eprint: https://direct.mit.edu/neco/article-pdf/9/1/1/813385/neco.1997.9.1.1.pdf.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.

[Kum+24] Tanishq Kumar et al. *Grokking as the Transition from Lazy to Rich Training Dynamics*. 2024. arXiv: 2310.06110 [stat.ML].

[LBB23] Noam Levi, Alon Beck, and Yohai Bar-Sinai. *Grokking in Linear Estimators – A Solvable Model that Groks without Understanding*. 2023. arXiv: 2310.16441 [stat.ML].

[LMT23] Ziming Liu, Eric J. Michaud, and Max Tegmark. *Omnigrok: Grokking Beyond Algorithmic Data*. 2023. arXiv: 2210.01117 [cs.LG].

[MP67] V. A. Marenko and L. A. Pastur. "Distribution of Eigenvalues for Some Sets of Random Matrices". en. In: *Mathematics of the USSR-Sbornik* 1.4 (Apr. 1967), p. 457. ISSN: 0025-5734. DOI: 10.1070/SM1967v001n04ABEH001994.

[Mer25] Merriam-Webster Dictionary. *Grok*. Accessed: 2025-04-05. 2025. URL: https://www.merriam-webster.com/thesaurus/grok.

[MU25] Andrea Montanari and Pierfrancesco Urbani. *Dynamical Decoupling of Generalization and Overfitting in Large Two-Layer Networks*. 2025. arXiv: 2502.21269 [stat.ML].

[Nan+23] Neel Nanda et al. *Progress measures for grokking via mechanistic interpretability*. 2023. arXiv: 2301.05217 [cs.LG].

[Pow+22] Alethea Power et al. *Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets*. 2022. arXiv: 2201.02177 [cs.LG].

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press, 2018. ISBN: 978-1-108-41519-4. DOI: 10.1017/9781108231596.

[Zha+16] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *arXiv preprint arXiv:1611.03530* (2016).

A Experimental Details

To produce Figure 1, we closely follow the experimental setup of [Pow+22], using their publicly available code at: https://github.com/openai/grok. The task involves learning addition modulo 127, and the dataset is generated accordingly. We use 50% of the dataset for training and evaluate on the remainder. The model is trained with no weight decay and run for a limited number of iterations on a single GPU. Apart from the modifications mentioned, we retain the default hyperparameters provided in the original implementation.

B Proof for Proposition 1

Before giving the proof of Proposition 1, note that the MarchenkoPastur distribution with parameter λ is defined as

$$\mu_{\mathrm{MP}}(A) = \begin{cases} \left(1 - \frac{1}{\lambda}\right) \delta_{0 \in A} + \nu(A) & \text{if } \lambda > 1 \\ \nu(A) \end{cases},$$

$$\mathrm{d}\nu_{\mathrm{MP}}(x) = \begin{cases} \frac{1}{2\pi\sigma^2} \frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}\lambda x}{\mathrm{d}} x & \text{if } x \in [\lambda_-, \lambda_+] \\ 0 & \text{otherwise} \end{cases}$$

where $\lambda_\pm = (1 \pm \sqrt{\lambda})^2$ denote the support of the absolutely continuous part.

We will focus on showing that, as $n, d \to \infty$ with $d/n = \lambda$,

$$\widehat{\mathcal{R}}_n(S_t) = D_0^\top e^{-4\eta_0 \widehat{\Sigma} t} \widehat{\Sigma} D_0 \to \int e^{-4\eta_0 z t} z \mu_{\mathrm{MP}}(\mathrm{d}z)$$

almost surely. The other part of Proposition 1 follows from a similar argument.

To begin, note that D_0 is a vector with i.i.d. $\mathcal{N}(0, d^{-1})$ entries. By Hanson-Wright inequality (see e.g. [Ver18, Theorem 6.2.1]),

$$\mathbb{P}_{D_0}\left(\left|D_0^\top M D_0 - \frac{1}{d}\operatorname{tr}(M)\right| \geq t\right) \leq 2\exp\left(-c_1 \min\left\{\frac{d^2t^2}{\|M\|_F^2}, \frac{dt}{\|M\|_2^2}\right\}\right)$$

where we wrote $M = e^{-4\eta_0\widehat{\Sigma}t}\widehat{\Sigma}$ for simplicity and $c_1 \in \mathbb{R}_{>0}$ is an absolute constant. Furthermore, by [Ver18, Theorem 4.4.5], $\|\widehat{\Sigma}\|_2 \leq c_2$ with probability at least $1-2e^{-n}$ for some constant $c_2 \in \mathbb{R}_{>0}$. Consequently, $\|M\|_2 \leq c_3 := e^{-4\eta_0 c_2 t}c_2$ and $\|M\|_F \leq \sqrt{d}\|M\|_2 \leq \sqrt{d}c_3$ with probability at least $1-2e^{-n}$, which in turn implies that

$$\mathbb{P}_{D_0}\left(\left|D_0^\top M D_0 - \frac{1}{d}\operatorname{tr}(M)\right| \geq t\right) \leq 2\exp\left(-c_4 dt \min\left\{t^2, t\right\}\right) + 2\exp(-n)$$

In other words,

$$D_0^\top M D_0 \to \frac{1}{d}\operatorname{tr}(M)$$

almost surely as $n, d \to \infty$.

Now, consider the spectral decomposition $\widehat{\Sigma} = U\Lambda U^\top$ where U is an orthonormal matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. Then,

$$\frac{1}{d}\operatorname{tr}(M) = \frac{1}{d}\operatorname{tr}(Ue^{-4\eta_0\Lambda t}U^\top U\Lambda U^\top) = \frac{1}{d}\sum_{j=1}^{d}\lambda_j e^{-4\eta_0\lambda_j t} = \int z e^{-4\eta_0 z t}\mu_n(\mathrm{d}z)$$

where $\mu_n = d^{-1}\sum_{j=1}^{d}\delta_{\lambda_j}$ is the *empirical spectral distribution* of $\widehat{\Sigma}$.

By the celebrated Marčhenko-Pastur law, to whom is attributed the name of the distribution μ_{MP}, the random probability measure μ_n converges weakly to μ_{MP} almost surely as $n, d \to \infty$ [MP67]. Since the function $z \mapsto ze^{-4\eta_0 z t}$ is bounded and continuous on $[0, \infty)$, the support of μ_n and μ_{MP}, it follows that $\int ze^{-4\eta_0 z t}\mu_n(\mathrm{d}z) \to \int ze^{-4\eta_0 z t}\mu_{\text{MP}}(\mathrm{d}z)$ almost surely as $n, d \to \infty$.