A Survey on Grokking

FRANCESCO BERTOLOTTI, Domyn, Milan, Italy
WALTER CAZZOLA, Computer Science, Università degli Studi di Milano, Milan, Italy

The phenomenon of Grokking, first reported by Power et al. [73], has gained significant attention for its unusual characteristics. Grokking is defined by a delay in generalization, during which the training loss reaches near-zero values while the test performance remains poor. This raises a critical question: if not the training loss, what drives generalization? Understanding the mechanisms behind Grokking is essential for gaining insights into the generalization process in broader deep learning contexts. In recent years, this phenomenon has attracted growing interest, with numerous theories and observations proposed to explain it. In this work, we aim to provide a comprehensive review that consolidates the current research and knowledge surrounding Grokking.

CCS Concepts: • **Computing methodologies → Neural networks;**

Additional Key Words and Phrases: Grokking, generalization, neural networks

ACM Reference Format:
Francesco Bertolotti and Walter Cazzola. 2026. A Survey on Grokking. *ACM Comput. Surv.* 58, 13, Article 327 (June 2026), 25 pages. https://doi.org/10.1145/3814603

1 Introduction

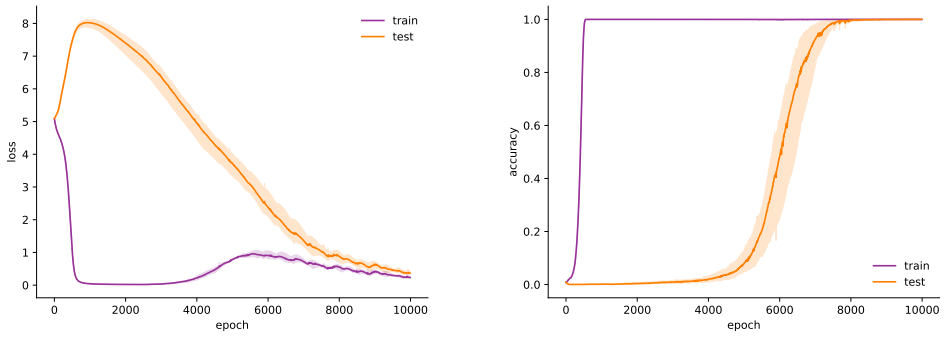
Neural networks, along with gradient descent-based learning, have been transformative in the field of machine learning, demonstrating applications across a wide range of domains (see Abiodun et al. [1] for a review). Despite their success, the underlying behavior of these models remains poorly understood, and many phenomena are yet to be fully explained. In this work, we focus on the phenomenon of Grokking, as described by Power et al. [73].

In recent years, the phenomenon of Grokking has attracted significant research attention due to its intriguing and puzzling nature. Grokking occurs when generalization happens long after the training process has stagnated. A typical scenario is illustrated in Figure 1. This phenomenon is usually characterized by three distinct phases: the memorization phase, the plateau phase, and the generalization phase. The memorization phase is marked by a decrease in training loss and an increase in training accuracy, though test accuracy remains low. The plateau phase is characterized by stagnation in both training loss and accuracy, as well as in test accuracy. Finally, the generalization phase is marked by a sudden increase in test accuracy.

The puzzling nature of Grokking is primarily evident during the plateau phase. Although it may seem that the network is not making any progress for a period of time, it suddenly begins to

This work was partly supported by the MUR project "T-LADIES" (PRIN 2020TL3X8X).
Authors' Contact Information: Francesco Bertolotti, Domyn, Milan, Italy; e-mail: francesco.bertolotti@domyn.com; Walter Cazzola (corresponding author), Computer Science, Università degli Studi di Milano, Milan, Italy; e-mail: cazzola@di.unimi.it.

ACM 0360-0300/2026/06-ART327
https://doi.org/10.1145/3814603



(a) Loss (b) Accuracy

Fig. 1. Grokking phenomenon: Generalization occurs long after training appears to have stagnated. The training loss reaches near-zero values early in the process (within the first 1,000 epochs), and the training accuracy achieves near-perfect levels. However, during this time, the accuracy stays low. After a prolonged period of stagnation (between 1,000 and 4,000 epochs), the test accuracy starts improving, eventually converging between 4,000 and 8,000 epochs.

generalize. This raises a critical question: if the training loss is not responsible (since it is near-zero), what drives generalization? What are the mechanism that allow the network to transition from memorizing solutions to achieving generalization?

Initially observed for a limited set of algorithmically generated tasks, Grokking has now become one of the most studied phenomena in the context of gradient descent learning. It has been observed across various data settings and model architectures [35, 54]. Furthermore, several hypotheses have been proposed regarding the factors that drive networks from memorizing solutions to achieving generalization [44, 54, 55]. However, inconsistent findings have also been reported [28, 48, 54].

Understanding this phenomenon is crucial for the development of more robust and reliable models. In this survey, we provide an overview of the Grokking phenomenon, its implications, and the current state of research. Additionally, we discuss the challenges and open questions that remain to be addressed.

This survey is organized into six main sections. Firstly, we provide a brief background that introduces the key concepts and terminology used in the context of Grokking (Section 3). Secondly, we discuss the mechanisms and hypotheses regarding what drives generalization or explains the observed delay in generalization (Section 4). Thirdly, we offer a broad overview of the main observations and findings related to the Grokking phenomenon (Section 5). Next, we discuss tasks and datasets that have been shown to exhibit the Grokking phenomenon (Section 6). We then explore models and architectures that have been studied in the context of Grokking (Section 7). Finally, we conclude with a discussion of the challenges and open questions that remain to be addressed (Section 9). Related works that mention Grokking, but do not explicitly study it, are discussed in Section 8. Notes on the methodology used in this survey are provided in Section 2.

2 Methodology

This survey was conducted in three phases. In the first phase, we performed a literature search to identify articles that cited the original Grokking article [73] and contained the term “Grokking” in the title or abstract using the *semantic scholar graph* API [39]. We then removed duplicates and examined articles with missing bibliography entries. At this stage, we were left with 241 articles.

In the second phase, we classified the remaining articles based on their relevance to the Grokking phenomenon. We manually removed articles that were irrelevant based solely on the abstract. Finally, we skimmed through potentially relevant articles and removed any that were still deemed irrelevant. The remaining 45 articles were included in the survey.

In the third phase, while organizing the findings among the 45 articles, we applied snowball sampling to identify related works.

3 Background

In this section, we provide a brief overview of the concepts and terminology relevant to the Grokking phenomenon. The purpose of this section is to serve as a reference for the reader, helping them better understand the subsequent sections.

Notation. Let $\mathcal{F} = \{f_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta\}$ be a family of functions parameterized by θ, where $\mathcal{X}$ denotes the input space, $\mathcal{Y}$ denotes the output space, and Θ denotes the set of all possible parameters. Let $f^* : \mathcal{X} \to \mathcal{Y}$ denote the task-specific function that we aim to approximate. Let $D = \{(x_i, y_i)\}_{i=1}^N$ be a training dataset of size N, with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Let $l : \mathcal{F} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}$ be a loss function that measures the discrepancy between the predicted output and the true output. We denote the training loss for a function f as $l(f, D)$. The generalization loss is measured as $l(f, \{(x, f^*(x)) : x \in \mathcal{X}\})$. For brevity, we will denote the training loss as l_t and the generalization loss as l_g and omitting the second argument.

We will refer to overfitting or memorizing solutions as those functions for which $l_t(f) \sim l_t(f^*)$ and $l_g(f) \gg l_g(f^*)$. In contrast, we will refer to generalizing solutions as those functions for which $l_g(f) \sim l_g(f^*)$.

Grokking. The Grokking phenomenon refers to the phenomenon originally observed by [73], where the generalization loss of a neural network suddenly decreases after a long period of stagnation. This phenomenon is typically characterized by three phases: the memorization phase, the plateau phase, and the generalization phase. The memorization phase is marked by a decrease in the training loss, but no improvement in generalization loss (i.e., $l_t(f) \sim l_t(f^*)$ and $l_g(f) \gg l_g(f^*)$). The plateau phase is characterized by stagnation in both training and generalization loss. Finally, the generalization phase is characterized by a sudden decrease in generalization loss, after which $l_g(f) \sim l_g(f^*)$.

Optimal Set. The optimal set, also known as zero-risk manifold, is the set of parameters for which the training loss is minimized. Formally, it is defined as

$$\Theta^* = \{\theta \in \Theta : l_t(f_\theta) = l_t(f^*)\}$$

Generalizing Set. The generalizing set is the set of parameters for which the generalization loss is minimized. Formally, it is defined as

$$\Theta^\star = \{\theta \in \Theta : l_g(f_\theta) = l_g(f^*)\}$$

Note that $\Theta^\star \subseteq \Theta^*$.

Goldilocks Zone Hypothesis. The Goldilocks Zone Hypothesis [24] posits the existence of a hollow spherical shell of parameters in which parameters within the optimal set are more likely to be generalizing ones. We can formalize the Goldilocks zone hypothesis as follows: there exists a preferred weight norm w^* and a threshold ϵ such that the parameter space $\mathcal{G} = \{\theta \in \Theta : |\|\theta\| - w^*| < \epsilon\}$, referred to as the Goldilocks zone, leads to generalizing solutions. That is,

$$\mathbb{E}_{\theta\sim\Theta^*\cap\mathcal{G}}[l_g(f_\theta)] \ll \mathbb{E}_{\theta\sim\Theta^*\cap\bar{\mathcal{G}}}[l_g(f_\theta)],$$

where $\bar{\mathcal{G}}$ denotes $\Theta \setminus \mathcal{G}$.

Occam's Razor Hypothesis. The Occam's Razor Hypothesis posits that simpler models are more likely to generalize. This is a general hypothesis that has been found useful in a variety of domains. In the context of neural networks, it implies the existence of a complexity measure, $g : \Theta \to \mathbb{R}_{\geq 0}$, on the network parameters. Formally, the hypothesis states that for $\theta_1, \theta_2 \in \Theta^*$,

$$g(\theta_1) > g(\theta_2) \implies l_g(f_{\theta_1}) > l_g(f_{\theta_2}).$$

Regularization. The term regularization refers to a set of techniques used to prevent overfitting (see [63] and [90] for a review). Regularization is typically achieved by adding a penalty term to the loss function. The most common form of regularization is L_2 regularization, which adds a penalty term proportional to the square of the L_2 norm of the parameters. Formally, the regularized loss function is given by

$$l_{\text{reg}}(f_\theta, D) = l(f, D) + \lambda ||\theta||_2^2$$

Where λ is a hyperparameter that controls the strength of the regularization.

Kernel Regime. During the training of over parameterized neural networks, small changes in parameter space can lead to large changes in the output space. When, during training, the network does not distance itself from the initialization, the network is said to be in the kernel regime or lazy training [11] (see [29] for a survey). When the network is in the kernel regime, we can approximate its behavior with a linear model (with respect to the feature map $\phi(x) = \nabla f_{\theta_0}(x)$), $\tilde{f}_\theta$, derived from the Taylor expansion, as follows:

$$\tilde{f}_\theta(x) = f_{\theta_0}(x) + \nabla f_{\theta_0}(x)|_{\theta_0}^T (\theta - \theta_0).$$

The error of this approximation is $o(||\theta - \theta_0||^2)$, where θ_0 represents the initial parameters. Moreover, as the width of the network increases, the approximation error decreases [11].

Cross Entropy Loss. The cross-entropy loss is a common loss function used in classification tasks. It is defined as

$$-\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}),$$

where $y_{i,c}$ is the true probability of class c for sample i, and $\hat{y}_{i,c}$ is the predicted probability for the respective class and sample. Usually, the predicted probability is obtained by applying the softmax function to the output of the network, often referred to as logits, $z_{i,c}$:

$$\hat{y}_{i,c} = \frac{e^{z_{i,c}}}{\sum_{c'=1}^{C} e^{z_{i,c'}}}.$$

For numerical stability, the cross-entropy loss is often computed in log-space. That is,

$$-\log(\hat{y}_{i,c}) = -z_{i,c} + \max_c(z_{i,c}) + \log\left(\sum_{c=1}^{C} e^{z_{i,c} - \max_c(z_{i,c})}\right).$$

Absorption Errors. In floating-point arithmetic, absorption errors occur when adding two numbers with vastly different magnitudes. The smaller number is effectively "absorbed" by the larger one. For example, when adding $10^6 + 10^{-4}$ in a floating-point system, the smaller number (10^{-4}) may be rounded off entirely due to the limitations in precision that can be stored.

A minimal Python program to illustrate an absorption error is as follows:

```
print(1e10+1e-10 == 1e10) # True
```

Despite using the more stable log-space computation, absorption errors can still occur when the logits of the network become large. Specifically, when the logits ($z_{i,c}$) corresponding to the correct label become significantly large ($\gg 0$) and the logits corresponding to the incorrect labels

become very small ($\ll 0$), this can result in numerical instability due to precision limitations. More precisely, absorption error can occur when computing

$$\sum_{c=1}^{C} e^{z_{i,c} - \max_c(z_{i,c})}$$

Maximal Margin Solution. When the data is linearly separable, we can select two hyperplanes that separate the data, such that the distance between them is as large as possible. The region between these hyperplanes is referred to as the margin. In this setting, a good classifier choice is the hyperplane that lies halfway between the two hyperplanes. This is known as the maximal margin solution. In practice, the maximal margin solution corresponds to solving the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1 \quad \forall i$$

Where w is the vector orthogonal to the hyperplane, b is the bias term, and (x_i, y_i) is the ith training sample.

For deep networks, the treatment of the maximal margin solution becomes more challenging (see Elsayed et al. [23] for further reference).

4 Explaining Grokking

In this section, we aim to summarize the primary mechanisms and hypotheses that explain the delay and generalizations associated with the Grokking phenomenon. Along with the main mechanisms, we will present observations that either support or contradict these hypotheses. Overall, this will lead to a complex picture in which multiple mechanisms may be at play at different stages.

4.1 Stumbling in the Dark Hypothesis

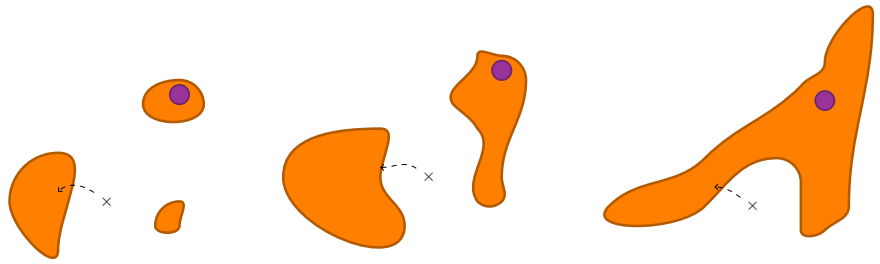
For under-parameterized neural networks, their optimal set, Θ^*, (see Section 3) is small and scattered. As the number of parameters increases, the optimal set also grows. Initially, it forms small and scattered parameter patches with minimal loss. As we further scale the number of parameters, these patches begin to merge, forming a connected region of minimal loss [50, 83] (see Figure 2).

When the network hit the near-zero training-loss region, the parameter exploration became guided by mostly gradient noise. Eventually, during training, the network will stumble upon a set of parameters that generalizes well, leading to the grokking phenomenon [60]. However, this can happen only if the optimal patch at which the network lands is connected to a patch of generalizing solution, $\Theta^\star$. This is depicted in Figure 2 as depicted in Figure 2(c).

Kozyrev [42] supports this hypothesis by proposing that *stochastic gradient descent* (SGD) behaves like Brownian motion on Θ^*, known as the zero-risk manifold. According to the second law of thermodynamics, the network tends to move toward flatter, lower-entropy regions of this manifold, which are typically associated with better generalization [8, 32]. Kozyrev further explains the dependence on training size by noting that each additional training sample imposes new constraints on the zero-risk manifold without altering the region responsible for generalization.

This hypothesis aims to explain the delay in generalization while addressing the reason why over-parameterized networks generalize better than their under-parameterized counterparts.

Barak et al. [3] explore the training of small neural networks for the sparse parity problem (see Section 6) and find evidence against the stumbling in the dark hypothesis. Specifically, they present the following observations:



(a) Under-parametrized (b) Medium-parametrized (c) Over-parametrized

Fig. 2. The optimal set of parameters, Θ^*, in three settings: under-parameterized, medium-parameterized, and over-parameterized, shown in orange. The generalization set, $\Theta^\star$, is shown in purple. The dashed line represents the path taken by the optimization algorithm, and the cross represents the initial set of parameters.

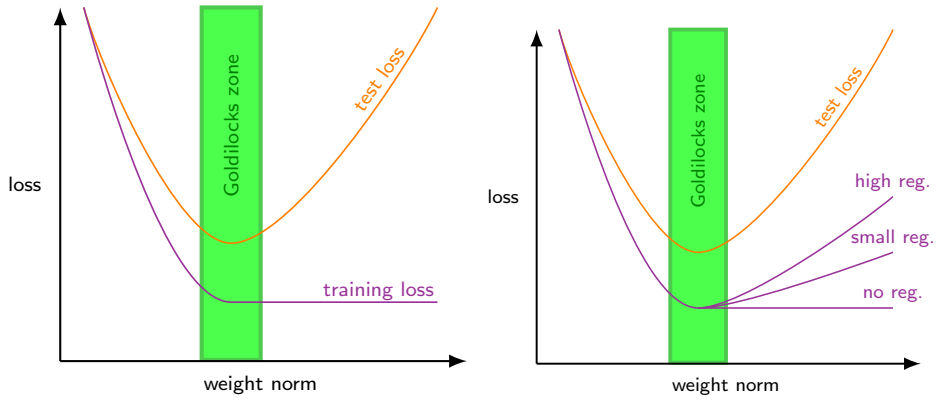
— The stumbling in the dark hypothesis would suggest the existence of a lucky initialization that is close to generalizing solutions that lead to faster generalizations. However, the authors found no such initializations.
— Sampling training batches differently leads to different parameter trajectories. According to the stumbling in the dark hypothesis, one would expect some trajectories to lead to a generalizing solution faster. However, the authors found no such trajectories.
— The time of convergence does not follow a power law with respect to the problem size, as random exhaustive search would predict.
— Running several copies of the network simultaneously does not lead to a speed-up in convergence time. The authors found no evidence of such a speed-up.
— The stumbling in the dark hypothesis suggests that while the network is training within the optimal set of parameters but away from the generalizing set, no progress is made. However, the authors found evidence of a hidden progress measure that continuously improves during training.

4.2 LU Mechanism

Overview. The LU-mechanism describes the relationship between the train and test losses as a function of the weight norm of a neural network. Consider Figure 3(a). The train loss exhibits an L-shaped curve with respect to the weight norm. This implies that in the high-norm region, there are many overfitting solutions but only a few generalizing ones. Conversely, in the low-norm region, overfitting solutions are sparse. On the other hand, the test loss follows a U-shaped curve with respect to the weight norm, reaching its minimum in the so-called "Goldilocks zone", where the generalizing solutions are located.

When the network is initialized with a high weight norm, gradient descent quickly converges to an overfitting solution and remains stuck in this region, as the training loss enters a near-zero regime. In contrast, when the network is initialized in the low-norm region, it progresses slowly toward the Goldilocks zone. This behavior is illustrated in Figure 3(a). These phenomena are also observed in practice.

When regularization is applied, the training loss in the high-norm region increases, as shown in Figure 3(b). This encourages a gradual transition from the high-norm region toward the Goldilocks zone, ultimately leading to a generalizing solution. This observation aligns with the well-established



(a) LU-mechanism. (b) LU-mechanism with weight decay.

Fig. 3. Illustration of the LU-mechanism. (a) The train loss exhibits an L-shaped curve with respect to the weight norm, while the test loss follows a U-shaped curve. (b) The effect of weight decay on the LU-mechanism as its strength increases.

finding that weight decay can improve generalization [73]. Furthermore, Hu et al. [34] report that generalization occurs with "just-right" decreases in the weight norm.

Supporting this hypothesis, recently Musat [65] showed that gradient descent training quickly converges to a zero-training-loss memorizing solution, while weight decay drives movement in this solution space toward lower-norm solutions associated with better generalizations.

In contrast with the LU-mechanisms, Levi et al. [48] analyze the gradient flow dynamics for a simple linear model and conclude that grokking is primarily a result of the slower convergence of the generalizing loss. Their results demonstrate that different initialization scales lead to a rescaling of the training and test losses. This rescaling provides an explanation for why lower-norm initialization results in better generalization.

Similarly, Golechha [28] shows that grokking can occur outside the Goldilocks zone. In their experiments, the authors observe grokking even when negative regularization (which pushes the network toward high norms) is applied. This observation is inconsistent with the LU-mechanism. Similar conclusions are reported in [44, 97].

4.3 Slingshot Mechanism

Thilak et al. [88] note that, when the training accuracy is optimal, the classification layer exhibits a cyclic behavior. Specifically, the norm of the last layer alternates between a plateau phase and a phase of rapid growth. Furthermore, the transition between these two phases triggers spikes in the loss. Figure 4 illustrates this phenomenon. The authors refer to this behavior as the *slingshot mechanism*. They also observe that the slingshot mechanism is often accompanied by a sudden improvement in generalization, measured as an increase in test accuracy.

The authors hypothesize that the slingshot mechanism may be caused by the nature of the cross-entropy loss. Cross-entropy inherently pressures the classification layer to grow in norm to reduce the loss. This norm growth continues until the curvature of the loss surface becomes large, effectively "flinging" (hence the name slingshot) the weights into a different region of parameter space. This abrupt transition leads to a spike in the loss.

Through their experiments, the authors find that each training run may experience multiple slingshots. Moreover, models exhibiting signs of grokking also tend to experience slingshots,

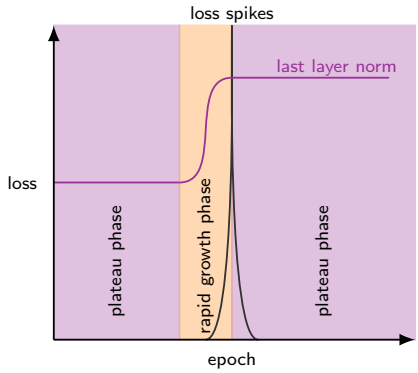


Fig. 4. Idealized representation of the slingshot mechanism. The weight norm of the last layer suddenly grows from a plateau to plateau again. This rapid transition causes a spike in the loss.

whereas models that do not exhibit slingshots show no signs of grokking. Importantly, models that undergo slingshots are observed to generalize better.

The authors also compare the slingshot mechanism to weight decay. Both phenomena prevent the weights from growing unbounded—the slingshot mechanism by "slinging" the weights into a different parameter region and weight decay by directly penalizing large weights. Thus, the slingshot mechanism can be viewed as an implicit form of regularization. However, the authors note that when weight decay is applied, the slingshot mechanism disappears.

Pascal et al. [68] analyze the loss landscape of deep neural networks during training. The authors find that generalizing networks usually occurs when the loss and accuracy quickly oscillate. This observation is then used to predict which model will eventually generalize. The authors argue that the slingshot mechanism is a key factor in the oscillatory behavior. Furthermore, they suggest that the model escapes the lazy regime during these oscillations, leading to generalization.

Similar to the LU mechanism, the grokking phenomenon has been observed in models that do not exhibit the slingshot mechanism [48] and even in the presence of negative regularization [28]. This suggests that the slingshot mechanism is not the sole driver of generalization.

4.4 Kernel and Rich Regime

Overview. The kernel regime refers to an early training phase of over-parameterized neural networks (see Section 3). During this phase, the network behaves like a linear model on the gradient feature map [96] (see Figure 5 for a comparison of training dynamics in the kernel and rich regimes). In the kernel regime, the network tends to overfit the training data [55]. Research indicates that, under specific conditions, a network remains in the kernel regime for a duration proportional to $\mathcal{O}(\log \alpha/\lambda)$, where α represents the initialization weight scale and λ the weight decay strength. This suggests that a lower initialization scale and higher weight decay reduce the time spent in the kernel regime [55].

While the kernel regime is associated with memorization rather than generalization, exiting this regime leads to the rich regime, where the network demonstrates improved generalization capabilities.

The transition between the kernel and rich regimes has been explained by Mohamadi et al. [61] as the network approaching a maximal margin solution on the training data (Gradient descent has been shown to converge slowly towards such solutions [56, 86, 94]).

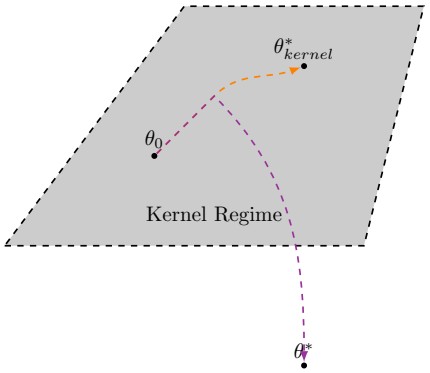


Fig. 5. Idealized representation of the kernel regime compared to the rich regime from [44]. In orange a training trajectory that remains trapped in kernel regime. In purple a that escapes.

Supporting this perspective, Mohamadi et al. [62] demonstrated that neural networks operating in the kernel regime during the modular addition task (see Section 6) fail to generalize unless nearly all training data points are presented. However, once the network escapes the kernel regime, it can generalize with a smaller proportion of the training data while maintaining a bounded weight norm.

Further evidence suggests that label rescaling or logit rescaling can induce the kernel regime, as shown by [11, 26]. This phenomenon was leveraged by Kumar et al. [44], who demonstrated that rescaling controls delays in generalization in grokking scenarios. Additionally, Kumar et al. [44] provided empirical evidence that networks achieve generalization once they escape the kernel regime.

Recently, Tian [89] explains kernel regime as a phase of learning where the network uses random features to quickly fit the training data. After this phase, the network transitions generalizing phases of learning. Similarly, Pascal et al. [67] show that in the first phase of learning parameters remain close to their initialization fitting the training data, while in the second phase, the learning is driven by weight decay or other regularization mechanisms.

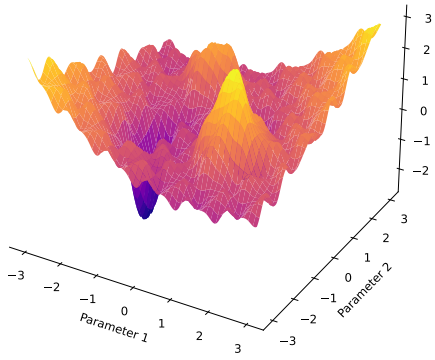
Although there is little evidence against the hypothesis that delayed generalization arises as a consequence of the kernel regime, this framework provides limited insight into the mechanisms driving generalization after the network exits the kernel regime.

4.5 Low Complexity Mechanism

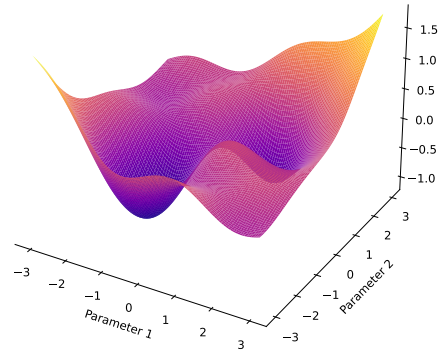
Occam's razor posits that the simplest explanation is the most likely to be correct. In the context of neural networks, this principle is often used to explain generalization, suggesting that generalization arises as networks converge to simpler solutions.

Miller et al. [59] propose that grokking may result from the network transitioning from high-complexity solutions (associated with memorization) to low-complexity solutions (associated with generalization). Figure 6(a) and (b) illustrates this concept. In the first, the high-complexity region contains several low-loss solutions that fit the training data. Small changes in parameters can lead to significant variations in the loss, indicating sensitivity and complexity. In contrast, the low-complexity region has fewer low-loss solutions, and small parameter changes result in minor loss variations, indicating stability and simplicity. Specifically, they characterize typical training trajectories as follows:

(1) At initialization, parameters are in the *high error high complexity* (HEHC) region.



(a) Low complexity region associated with generalization.



(b) High complexity region associated with memorization.

Fig. 6. Comparison of a sharp and flat loss landscape. Often, the sharp region is associated with high complexity solutions, while the flat region is associated with low complexity solutions. These figures are merely illustrative and do not represent actual loss landscapes.

(2) Guided by gradient descent, parameters quickly move into the *low error high complexity* (LEHC) region.
(3) When possible, parameters transition to *low error low complexity* (LELC) solutions, driven by regularizers like weight decay.

The authors support their hypothesis by demonstrating that networks fail to generalize without weight decay and that lower initialization scales reduce delays in generalization.

DeMoss et al. [17] introduce a complexity measure based on the compressibility of network parameters as a proxy for Kolmogorov complexity [41]. Their findings indicate that sudden decreases in test loss (signifying generalization) coincide with abrupt reductions in parameter complexity. Furthermore, they show that explicitly penalizing complexity, rather than weight norm, can reduce delays in generalization.

Other works explore various approximations of complexity:

— Sensitivity to dropout: Golechha [28] approximates complexity using dropout sensitivity, reporting sudden drops in sensitivity at the point of generalization.
— Neuron plane crossings: Humayun et al. [35] measure complexity based on the number of neuron plane crossings within a spatial region. Sudden drops in complexity are observed during generalization.
— Linear region traversals: Liu et al. [53] approximate complexity by counting linear regions traversed when interpolating between two points. Similar to other measures, complexity drops align with generalization.

A complementary perspective is presented in [37], where the authors use a linearization of the network to apply a complexity measure designed for linear models [15]. Their study reports that complexity measured on training data diverges from that on test data at the onset of generalization.

While these studies suggest a strong correlation between generalization and low-complexity solutions, some nuances remain unresolved. For instance, Xu et al. [97] document instances of

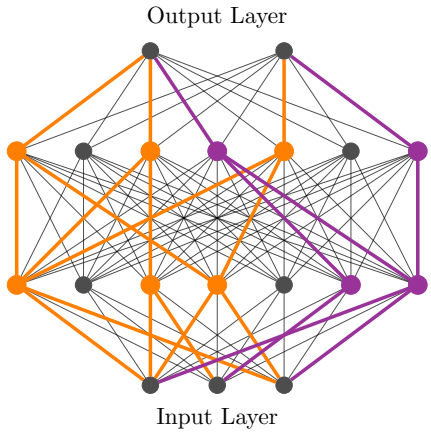


Fig. 7. Two sub-networks (orange and purple) competing.

grokking without explicit regularization, highlighting the need for clearer evidence on what drives networks to converge to low-complexity solutions.

4.6 Competing Sub-Networks Hypothesis

The competing sub-networks hypothesis proposes that during training, different sub-networks within an over-parameterized neural network compete to represent the data. This competition arises because the network can represent the data in multiple ways. Once the network generalizes, one sub-network dominates. In Figure 7, two sub-networks (orange and violet) compete to explain the output given the input.

Merrill et al. [58] suggest that grokking may stem from competition between two sub-networks: a dense one and a sparse one. Initially, the dense sub-network dominates, driving memorization. However, as the model begins to generalize, the sparse sub-network takes over. This hypothesis is supported by experiments identifying the sparse sub-network and tracking its weight norms. Remarkably, during generalization, the weight norm of the sparse sub-network increases sharply, while the weight norms of other components decrease, indicating that the sparse sub-network becomes dominant.

Additional research provides evidence for this hypothesis:

— Reverse engineering algorithms: Chughtai et al. [13] and Nanda et al. [66] reverse engineered the algorithm implemented by neural networks at generalization stage (for group operation task). They develop metrics to track the emergence of the reversed algorithms within the network, which align with the onset of generalization.
— Emergent sub-networks in GPTs: Wang et al. [95] explore grokking-like phenomena in GPT architectures during implicit reasoning tasks. Using techniques such as the logit lens [27, 98] and causal graphs [70], the authors identify generalizing sub-networks that emerge as the model transitions to generalization.
— Davies et al. [16] model grokking with a toy statistical model built on the assumption that the network, during training, learns memorizing/generalizing features at different rates.

A contrasting perspective is offered by Bhaskar et al. [6], who test the competing sub-networks hypothesis using transformer architectures on text datasets. They propose that instead of selecting between competing sub-networks, the model initially learns a "core" set of simple heuristics. Generalization occurs as the model learns to effectively combine these heuristics.

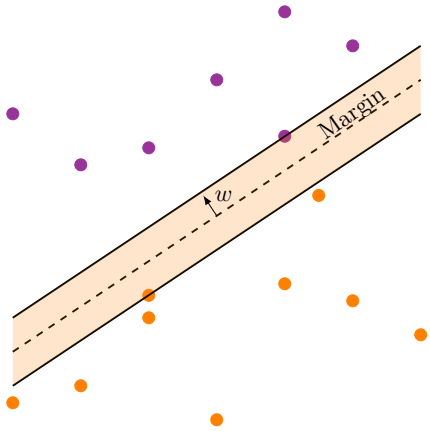


Fig. 8. Max margin solution for a 2 class (orange and purple) problem.

4.7 Maximal Margin Hypothesis

Another potential explanation for grokking is tied to the observation that gradient descent with cross-entropy loss slowly converges to a maximal margin solution [56, 86, 94] (see Section 3 for an overview). Maximal margin solutions are often associated with better and more stable generalization because small perturbations in the output do not alter predictions significantly (see Figure 8). These solutions are also the primary objective of classical machine learning models like support vector machines (SVMs) [7, 14].

Early in training, networks may simply memorize the training data, achieving high training accuracy but poor test accuracy. As training progresses, the network gradually approaches the maximal margin solution, resulting in improved generalization. This hypothesis is supported by Mohamadi et al. [61].

However, while this explanation is plausible, it is important to note that grokking has been observed with loss functions other than cross-entropy [20, 44], which do not inherently benefit from the implicit bias toward maximal margin solutions. This indicates that maximal margin dynamics may not be the sole mechanism driving grokking.

4.8 Different Convergence Rates

Levi et al. [48] analyze the exact gradient flow dynamics for simple linear models. They find that the convergence rates of training loss and generalization loss differ, with the former converging faster than the latter. This discrepancy in convergence rates is proposed as a key factor underlying the delayed generalization observed in grokking. The authors test their hypothesis by plotting the training loss against the analytically derived convergence rate, revealing a strong alignment between the two.

A related observation is reported in [72], which explores the double descent phenomenon. In their teacher-student framework, the authors identify two sets of features: one set is learned quickly, causing an initial descent in loss, while the other set is learned more slowly, resulting in a second descent. Although their work does not directly address grokking, it raises the possibility that a similar mechanism could underlie the delayed generalization observed in grokking. Furthermore, the authors analysis predicts an empirical double descent loss curve in the presence of small l2 regularization.

Davies et al. [16] propose a toy statistical model of the loss curve of grokking and double descent. Their model is built on the assumption that the network, during training, learns

memorizing/generalizing features at different rates. Their model is able to reproduce the loss curve of grokking and double descent, arguing that the two phenomena are related.

This perspective offers a markedly different interpretation of the grokking phenomenon compared to other hypotheses discussed earlier. It suggests that no inherently complex mechanisms are required to explain grokking and that the observed dynamics could be attributed to fundamental differences in learning rates for various features. However, it is important to note that these results are derived from linear models, which may not fully capture the behavior of deep neural networks.

4.9 Detour States

Hu et al. [34] analyze training dynamics by examining various statistics of the weights (such as norm, mean, variance, and trace) using hidden Markov models (HMM) [4]. They decompose the training process into a Markov model, providing insights into the underlying training dynamics. Specifically, they identify the presence of detour states, which, when entered during training, systematically cause delays in convergence.

The existence of detour states indicates that the training process is sensitive to randomness. The authors demonstrate that detour states can be avoided by employing techniques like *layer normalization* [47], *batch normalization* [36], or *skip connections* [31]. When applied correctly, these techniques simplify the Markov model associated with the training process into a straightforward chain, where each state progresses towards the next, ultimately leading to convergence.

Additionally, the authors propose that successful training processes exhibit a "just-right" drop in weight norm—neither too large nor too small. This observation aligns with the Goldilocks hypothesis. They further hypothesize that grokking may emerge from a sharp optimization landscape that facilitates a Slingshot mechanism.

While there is no evidence against the existence of detour states, the idea that grokking results from a sharp optimization landscape is not supported. However, one could argue that a sharp optimization landscape might arise from high-complexity regions of the parameter space. In such a case, the literature presented in Section 4.5 lends support to this hypothesis.

4.10 Good Representation Hypothesis

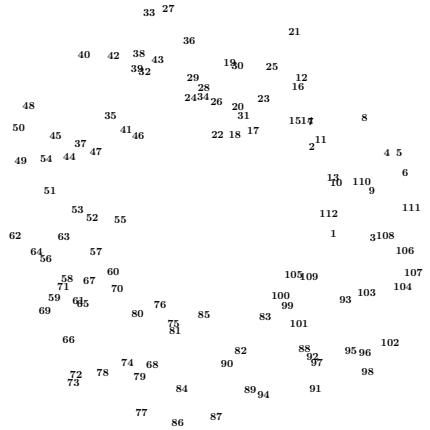
Liu et al. [52] report that generalization on the modular addition (task 6) occurs only when a well-organized representation is achieved within the embedding space (see Figure 9 for a comparison). More specifically, they analyze the training dynamics in relation to an effective loss function that depends solely on the embedding representation. Empirically, they observe that the gradient flow of this effective loss mirrors the training dynamics of the embeddings, suggesting a close relationship between generalization and the organization of the embedding representation.

This hypothesis can be viewed as part of the low complexity mechanism, as well-organized representations are generally considered to have lower complexity compared to chaotic ones.

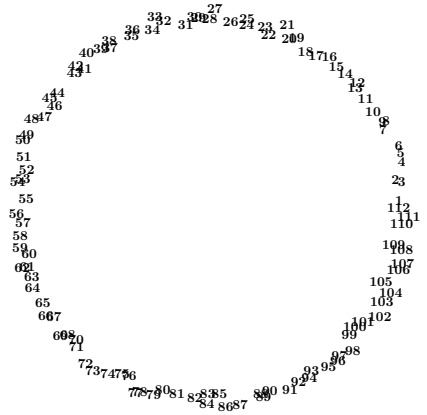
We note that Pearce et al. [69] observed non-generalizing models that, however, introduced a well-organized representation. This suggests that the good representation hypothesis is not sufficient to explain generalization in all cases.

4.11 Numerical Stability

Recently, Beck et al. [5] advanced the hypothesis that the failure in generalization observed in grokking datasets (such as modular addition (see Section 6) or sparse parities (see Section 6)) may be caused by numerical stability issues in the softmax computation within the cross-entropy loss (see Section 3). Specifically, the authors argue that during the plateau phase of training, the cross-entropy loss tends to push the logits to extremely large or small values without affecting the predictions. This update direction, termed naive loss minimization (NLM), reduces the training loss



(a) Chaotic or "bad" representation (a 2-dimensional projection).



(b) Organized or "good" representation (a 2-dimensional projection).

Fig. 9. Idealized representations of the embedding space. The left figure shows a chaotic representation, while the right figure shows an organized representation. This figure is merely illustrative and does not represent actual embeddings.

but does not modify the model's predictions. As a result, absorption errors in Section 3 can occur in the softmax computation, leading to zero-gradient updates and stagnation in generalization loss.

The authors propose two simple fixes to address this issue:

(1) A numerically stable variant of the softmax function, called stable max.
(2) Projecting gradient updates onto the orthogonal space of the NLM direction.

They demonstrate that these two adjustments can significantly improve the delay and generalization behavior observed in the grokking phenomenon. A similar effect can also be achieved by increasing the precision of the floating-point representation.

4.12 Summary

To summarize this section, let us make the following consideration.

— The Stumbling in the Dark hypothesis intuitively explains the delay in generalization but is found to be inconsistent according to Barak et al. [3].
— The LU mechanism accounts for the observed Goldilocks zone and explains why weight decay is necessary for grokking. However, inconsistent findings are reported in [28].
— The kernel regime explains how lower initialization, more data, and weight decay reduce the delay in generalization and account for initial overfitting. However, inconsistent findings are reported in [95] and [28].
— The low-complexity hypothesis invokes Occam's razor and approximates complexity with weight norm to explain the Goldilocks zone.
— The competing subnetworks hypothesis attributes the delay in generalization to the presence of competing memorizing subnetworks vying for the final decision.
— The maximum margin hypothesis explains eventual generalization and potentially the delay.
— Differences in convergence rates between the training and generalization loss have been reported as a potential factor in the delay.

— Detour states are suggested to explain the delay in generalization, as they cause training to stagnate temporarily.
— The good representation hypothesis explains the delay and generalization by proposing that a well-organized representation, which may be difficult to discover, is crucial for generalization.
— Numerical stability errors have been proposed as an explanation for the delay in generalization, particularly in the softmax computation.

It is important to recognize that these mechanisms are not mutually exclusive and may operate at different stages of the training process. For instance, the kernel regime may initially drive the network towards memorizing solutions, which are difficult to remove even as they reduce the loss. Over time, regularization (either implicit, such as the slingshot mechanism, or explicit, such as weight decay) may encourage generalization by discouraging the high-complexity circuits responsible for memorization. Simultaneously, starting in a low-norm region (and by proxy, low-complexity) may help the network avoid overly complex memorizing circuits, as suggested by the LU mechanism and detour states. At the same time, a good representation could reflect the fact that networks that generalize tend to exhibit lower complexity. Finally, numerical stability errors may also contribute to delays in generalization.

5 General Observations

In this section, we summarize general observations about the grokking process. Note that these observations have been made in different contexts (hyperparameters, architectures, data, etc.) and may not be directly comparable. However, they provide a broad understanding of the variables that control the grokking process and how the main theories explain these behaviors. Table 1 summarizes the general observations.

The primary factors influencing the delay in generalization are weight decay strength, initialization scale, and the proportion of training data relative to test data. Notably, some authors find weight decay to be necessary for generalization, while others consider it helpful but not essential. Similarly, smaller initialization scales are observed to reduce the delay in generalization. These observations generally support the Goldilocks hypothesis (Section 3) and, by extension, mechanisms like the LU mechanism (Section 4.2). Supporters of the *low complexity mechanism* (Section 4.5) explain these findings by approximating complexity with weight norm. At the same time, proponents of the *kernel regime mechanism* (Section 4.4) argue that large initialization scales promote the kernel regime, while weight decay acts to discourage it.

A considerable body of literature finds that different measures of complexity increase during the memorization phase of grokking but decrease during the generalization phase. This supports the Occam's Razor principle Section 3 and, by extension, the *low complexity mechanism* (Section 4.5).

In addition, techniques known to smooth the loss landscape, such as noisy gradients, layer normalization, and similar approaches [49], are found to reduce the delay in generalization.

Prakash and Martin [74] recently identified a phenomenon they term *anti-grokking*, where models after grokking suddenly go through a disruptive phase and lose their generalization capabilities.

6 Tasks and Datasets

In this section, we provide an overview of task and dataset that have been studied in the context of Grokking. Table 2 provides a summary. The most commonly studied tasks are algorithmic tasks such as modular addition and sparse parities.

Sparse Parities. (n, k)-sparse parity problem refers to the binary classification tasks of n-length binary strings. The task consists of determining the parity of the relevant k-bits of the input string. The other $n - k$ bits represent noise. To solve this task, the network must learn to recognize the

Table 1. General Observations About the Grokking Process

Observation	References
Goldilocks Hypothesis ("just-right" weight norm)	[34, 50, 65, 73]
Sensitivity to randomness yields slower convergence	[34]
Regularizing architectural choices such as *layer normalization* [47], *batch normalization* [36], skip connections [31], or small batch size lead to faster convergence	[20, 34, 35]
Adding spurious features exacerbates the delay in generalization	[59]
Weight Decay reduces the delay in generalization	[16, 20, 48, 50, 54, 55, 59, 62, 69, 72, 89, 93]
Lower initializations reduce the delay in generalization	[54, 55, 59, 62]
More training data reduces the delay in generalization	[35, 48, 50, 54, 93]
More training data does not reduce the delay in generalization	[95]
Generalization delay can be amplified through data selection	[67]
Increasing the output layer scale increases the delay in generalization	[5, 44]
Dropout decreases the delay in generalization	[20, 28, 50]
Sparsity decreases in the memorization phase, and increases in generalization phase	[28, 58, 66, 69]
Complexity raises in the memorization phase, and decreases in generalization phase	[13, 17, 35, 53, 66]
Noised gradients reduce the delay in generalization	[73]
Weight decay avoids numerical instabilities	[5]
Weight decay is not necessary to exhibit generalization	[5, 28, 44, 55, 64, 67, 97, 101]
Weight decay is necessary to exhibit generalization	[17, 54, 59]
Slingshot mechanism leads to generalizations	[68, 88]
Anti-Grokkinh	[74]

relevant bits and to perform the parity of such bits. However, some natural datasets, such as MNIST (vision modality) and IMDB (text modality), have been shown to exhibit grokking behavior.

Parity. The parity task consists of determining the parity of the input binary string. It is a special case of the sparse parity problem where $k = n$.

Modular Operations. The modular operation task consists in learning to perform arithmetic operations modulo p. The task is defined as follows: given two numbers a and b sampled uniformly from $\{0, 1, \dots, p-1\}$, the network must predict the result of the operation $a \circ b \mod p$, where $\circ$ is a generic operation.

Modular Addition. Particularly popular among the modular operations, the modular addition task consists of learning to add two numbers modulo p. The task is defined as follows: given two numbers a and b sampled uniformly from $\{0, 1, \dots, p-1\}$, the network must predict the sum of a and b modulo p, $a + b \mod p$.

Permutation Composition. The permutation composition task consists of learning to compose permutations. The task is defined as follows: given two permutations σ_1 and σ_2 of $\{0, 1, \dots, p-1\}$, the network must predict the composition of the two permutations, $\pi_1 \circ \pi_2$.

Group Operations. The group operation task consists of learning to perform group operations. Given a group $(G, \circ)$ the task is defined as follows: given two elements a and b from a group G, the network must predict the result of the operation $a \circ b$, where $\circ$ is a generic operation. This is a generalization of the modular operation and permutation composition task.

Addition. The addition task consists of learning to add two numbers. The task is defined as follows: given two numbers a and b from $\{0, 1, \dots, p\}$, the network must predict the sum of $a + b$.

Table 2. Task and Dataset That Have Been Studied in the Context of Grokking

Task	Modality	References
Group Operations	algorithmic	[13]
Modular Operations	algorithmic	[16, 17, 59, 68, 73, 88, 101]
Modular Addition	algorithmic	[5, 13, 17, 20, 34, 44, 50, 53, 55, 59, 61, 62, 65, 67–69, 73, 80, 89]
Addition	algorithmic	[50]
Sparse Parity	algorithmic	[3, 5, 34, 58, 69]
Parity	algorithmic	[51, 97]
Permutation Composition	algorithmic	[13, 53]
Polynomial Regression	algorithmic	[44]
Teacher-Student Setup	algorithmic	[48, 67, 72, 80]
Sampled Data	algorithmic	[54, 54, 55, 93]
MNIST	vision	[5, 28, 44, 50, 54, 67, 74]
CIFAR-10	vision	[88]
CIFA-100	vision	[35]
IMDB	text	[54]
Implicit Reasoning	text	[95]
QM9	other	[54]

MNIST. The MNIST [46] dataset consists of 28×28 gray scale images of handwritten digits from 0 to 9. The task consists of classifying the digit in the image.

CIFAR-10. The CIFAR-10 [43] dataset consists of 32×32 color images of objects from 10 classes. The task consists of classifying the object in the image.

CIFAR-100. The CIFAR-100 [43] dataset consists of 32×32 color images of objects from 100 classes. The task consists of classifying the object in the image.

IMDB. The IMDB dataset [57] consists of movie reviews labeled as positive or negative. The task consists of classifying the sentiment of the review.

QM9. The QM9 dataset [77] comprises approximately 134,000 small organic molecules, each containing up to nine heavy atoms (carbon, nitrogen, oxygen, or fluorine). For each molecule, it provides quantum chemical properties calculated using density functional theory.

Implicit Reasoning. Wang et al. [95] built a custom implicit reasoning dataset. This dataset is made of atomic facts and inferred facts. For example, an atomic fact may be "Barack's wife is Michelle" and "Michelle is born in 1964" while an inferred fact is "Barack's wife is born in 1964".

Polynomial Regression. Kumar et al. [44] studied the grokking phenomenon with the high-dimensional polynomial function $y(\mathbf{x}) = \frac{1}{2}(\beta x)^2$ as target. $x, \beta \in \mathbb{R}^d$.

Teacher-Student Setup. The teacher-student setup corresponds to a broad class of tasks where a network randomly initialized, f_θ^*, perform the role of teacher. Meanwhile, a second network similar in architecture, g_θ is trained to match the teacher's output.

Sample Classification Data. Sample data refer to a broad set of task where input point belonging to different class are sampled from fixed distributions. The task consists in learning to classify the input points.

Table 3. Task and Dataset That Have Been Studied in the Context of Grokking

Model	References
Multi-Layer Perceptron	[3, 5, 20, 28, 34, 35, 44, 50, 53–55, 58, 59, 61, 62, 65, 67, 69, 74, 80, 88, 89, 97]
Transformer	[3, 5, 16, 17, 20, 34, 35, 50, 66, 68, 73, 88, 95, 101]
ResNet	[34, 35]
Convolutional Neural Network	[35]
Gaussian Process	[59]
Bayesian Neural Network	[59]
Linear Model	[48, 72, 93]
Recurrent Neural Network	[54]
Graph Neural Network	[54]

7 Models and Architectures

In this section, we provide an overview of the models and architectures studied in the context of Grokking. Table 3 provides a summary. For a more detailed discussion, we refer the reader to the original articles. The most commonly studied models are transformers [91] and *multi layer perceptrons* (MLP) [2, 78].

Linear Model. The linear model is a simple model that assumes a linear relationship between the input and the output. The model is defined as

$$y = Wx + b.$$

Where W is the weight matrix and b is the bias vector. When $b \neq 0$, the model is affine. The parameters of the model are the weights W and the biases b.

Multi-Layer Perceptron. The MLP is one of the most popular and earliest architecture. It is a feedforward neural network that consists of a sequence of fully connected layers, each connected to the next. The basic layer is an affine transformation followed by a non-linear activation function:

$$h^{(l+1)} = \sigma(W^{(l+1)}h^{(l)} + b^{(l+1)}).$$

Where $h^{(0)}$ represents the input x, and $h^{(L)}$ is the output of the network. The parameters of the network are the weights $W^{(l)}$ and the biases $b^{(l)}$ for each layer l. The activation function σ is typically a non-linear function such as the sigmoid or the ReLU function.

Convolutional Neural Networks (CNNs). CNNs [25, 45] are a type of neural network architecture that is designed to process grid-like data, such as images. CNNs are composed of convolutional layers, pooling layers, and fully connected layers. The convolutional layer is defined as

$$h^{(l+1)} = \sigma(W^{(l+1)} * h^{(l)} + b^{(l+1)}).$$

Where $*$ is the convolution operator, $W^{(l)}$ is the convolutional kernel, and $b^{(l)}$ is the bias. The pooling layer is usually a max-pooling or average-pooling layer that reduces the spatial dimensions of the input. The fully connected layer is the same as in the MLP.

ResNet. ResNet [31] is a deep neural network architecture that introduces residual connections. The idea is to learn the residual of the input and the output of a layer, instead of learning the output directly. This is done by adding the input to the output of the layer:

$$h^{(l+1)} = h^{(l)} + f(h^{(l)}).$$

Where f is the residual function that is learned by the network. This architecture allows to train very deep networks, by mitigating the vanishing gradient problem. In [31], the residual function is a convolutional network.

Transformer. The Transformer [91] is a neural network architecture that is based on alternating multi-head self-attention and feedforward residual layers. A single-head self-attention layer is implemented as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$
$$Q, K, V = W_Q x, W_K x, W_V x.$$

While a feed forward layer is implemented as

$$\text{FFN}(x) = \sigma(W_2\sigma(W_1 x + b_1) + b_2).$$

Popular variant of this architecture are the BERT [18], GPT [75], T5 [76], and ViT [21].

Recurrent Neural Networks (RNNs). RNNs [81] are a type of neural network architecture that is designed to process sequential data. RNNs have a hidden state that is updated at each time step, and the output is computed based on the hidden state. The basic RNN cell is defined as

$$h^{(t)} = \sigma(Wx^{(t)} + Uh^{(t-1)} + b).$$

Where $h^{(t)}$ is the hidden state at time t, $x^{(t)}$ is the input at time t, W and U are the weight matrices, b is the bias, and σ is the activation function. A popular variant of the RNN are *long short-term memory* (LSTM) [32] cell, *gated recurrent unit* (GRU) [12], and *receptance weighted key value* (RWKV) [71].

Graph Neural Networks (GNNs). GNNs [30, 82] are a type of neural network architecture that is designed to process graph-structured data. GNNs have a message passing mechanism that allows nodes to communicate with their neighbors. The basic GNN cell is defined as

$$h_i^{(l+1)} = \sigma\left(W^{(l+1)}h_i^{(l)} + \sum_{j\in\mathcal{N}(i)} W^{(l+1)}h_j^{(l)}\right).$$

Where $h_i^{(l)}$ is the hidden state of node i at layer l, $W^{(l)}$ is the weight matrix, $\mathcal{N}(i)$ is the set of neighbors of node i, and σ is the activation function. Popular variant of the GNN are the *graph convolutional network* (GCN) [40], and *graph attention networks* (GAT) [92].

Gaussian Processes (GPs). GPs are a type of non-parametric probabilistic model used for regression and classification tasks. A GP is defined by a mean function $\mu(x)$ and a covariance function $k(x, x')$. For a target function $f(x)$, the GP is defined as

$$f(x) \sim \mathcal{N}(\mu(x), k(x, x')).$$

Training a Gaussian Process involves learning the hyperparameters of the kernel function (and sometimes the mean function).

Bayesian Neural Networks (BNNs). BNNs are neural networks with probabilistic weights. The weights are treated as random variables, and the network is trained by maximizing the marginal likelihood of the data. This is done by approximating the posterior distribution of the weights using variational inference of *Markov chain Monte Carlo* (MCMC) methods [38].

8 Related Works

While we mainly discuss works focusing on the Grokking phenomenon and its associated mechanisms, in this section, we try to address works that reported phenomena that may be relevant to this study albeit not focusing on Grokking explicitly.

von Oswald et al. [85] report a Grokking-like curve in *in context learning* (ICL) tasks. Murty et al. [64] discuss the phenomenon of Structural Grokking. Grokking-like curves are also reported in the NeQa dataset [99]. Hoffmann et al. [33] study sudden generalizations in multi-hop reasoning tasks. Stander et al. [87] and Zhong et al. [100] discuss an alternative algorithm to [13] implemented by a neural network through gradient descent. A slingshot-like mechanism is proposed in [79]. Dziri et al. [22] discover a Grokking-like phenomenon while studying the limits of transformers in multi-hop reasoning tasks. Dohmatob et al. [19] study how the introduction of synthetic data affects scaling laws. They recover grokking-like curves. Song et al. [84] introduce two information-based metrics to quantify the interplay between data representation and class classification heads. In [9] grokking is further reported on custom datasets. Grokking is also reported in [10] while interpreting transformer results on the greatest common divisor.

9 Conclusion

In this survey, we reviewed a wide range of studies that explore the Grokking phenomenon, focusing on various proposed mechanisms and their influence on generalization during training. Through a detailed examination of the literature, we have identified several key factors that appear to control the grokking process, including weight decay, initialization scale, and the proportion of training data relative to test data. The theories and mechanisms discussed-ranging from the kernel regime to low-complexity hypotheses, offer valuable insights into the complex dynamics of neural network generalization and the delay often observed during grokking.

Despite the progress in understanding these phenomena, it is clear that no single mechanism fully explains the behavior. Rather, multiple mechanisms may be at play at different stages of training, and further empirical investigations are needed to clarify their interactions. For example, slingshot mechanisms, low-complexity transitions, and Goldilocks zone are all compatible with each other and may jointly contribute to the sudden onset of generalization observed in grokking. Additionally, challenges remain in pinpointing the exact causes of delays in generalization, especially in the context of more complex network architectures and diverse datasets.

As the field continues to evolve, future research should aim to integrate these mechanisms into unified models and explore their practical applications and interaction in improving training efficiency and generalization capabilities. Ultimately, a deeper understanding of grokking and the factors that influence it could lead to more robust and generalizable machine learning models across a wide range of tasks.

References

[1] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4, 11 (2018).

[2] Shunichi Amari. 1967. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers* 16, 3 (1967), 299–307.

[3] Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. 2022. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS'23)*. Alekh Argawal, Alice Oh, Danielle Belgrave, and Kyunghyun Cho (Eds.), Curran Associates, Inc., New Orleans, USA.

[4] Leonard E. Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* 37 (1966), 1554–1563.

[5] Alon Beck, Noam Itzhak Levi, and Yohai Bar-Sinai. 2025. Grokking at the edge of linear separability. In *Proceedings of the 42nd International Conference on Machine Learning (ICML'25)*. PMLR, Vancouver, BC, Canada, 3307–3334.

[6] Adithya Bhaskar, Dan Friedman, and Danqi Chen. 2024. The heuristic core: Understanding subnetwork generalization in pretrained language models. *arXiv e-prints* arXiv:2403.03942 (2024), 1–18.

[7] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*. David Haussler (Ed.), ACM, Pittsburgh, PA, USA, 144–152.

[8] Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. *Journal of Machine Learning Research* 2 (2002), 499–526.
[9] Breno W. Carvalho, Artur S. d'Avila Garcez, Luís C. Lamb, and Emílio Brazil. 2025. Grokking explained: A statistical phenomenon. *arXiv e-prints* arXiv:2502.01774 (2025), 1–28.
[10] François Charton. 2024. Learning the greatest common divisor: Explaining transformer predictions. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*. Swarat Chaudhuri, Katerina Fragkiadaki, Mohammad Emtiyaz Khan, and Yizhou Sun (Eds.), ICLR, Vienna, Austria.
[11] Lénaïc Chizat, Edouard Oyallon, and Francis Bach. 2019. On lazy training in differentiable programming. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS'19)*. Curran Associates, Inc., Vancouver, Canada, 2933–2943.
[12] Kyunghyun Cho, B. van Merrienboer, Caglar Gulcehre, F. Bougares, H. Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 19th Edition of the International Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. Bo Pang and Walter Daelemans (Eds.), ACL, Doha, Qatar, 1724–1734.
[13] Bilal Chughtai, Lawrence Chan, and Neel Nanda. 2023. A toy model of universality: Reverse engineering how networks learn group operations. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*. Emma Brunskill, Kyunghyun Cho, and Barbara Engelhardt (Eds.), PMLR, Honolulu, Hawaii, 6243–6267.
[14] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (1995), 273–297.
[15] Alicia Curth, Alan Jeffares, and Mihaela van der Schaar. 2023. A U-turn on double descent: Rethinking parameter counting in statistical learning. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS'22)*. Alice Oh, Tobias Naumann, Globerson. Amir, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.), Curran Associates, Inc., New Orleans, Louisiana, USA, 55932–55962.
[16] Xander Davies, Lauro Langosco, and David Krueger. 2022. Unifying grokking and double descent. In *Proceedings of NeurIPS 2022 Machine Learning Safety Workshop*. NeurIPS Foundation, New Orleans, USA.
[17] Branton DeMoss, Silvia Sapora, Jakob Foerster, Nick Hawes, and Ingmar Posner. 2024. The complexity dynamics of grokking. *arXiv e-prints* arXiv:2412.09810 (2024), 1–17.
[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19)*. Association for Computational Linguistics, Minneapolis, MN, USA, 4171–4186.
[19] Elvis Dohmatob, Yunzhen Feng, Pu Yang, François Charton, and Julia Kempe. 2024. A tale of tails: Model collapse as a change of scaling laws. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*. Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.), PMLR, Vienna, Austria, 11165–111971.
[20] Darshil Doshi, Aritra Das, Tianyu He, and Andrey Gromov. 2024. To grok or not to grok: Disentangling generalization and memorization on corrupted algorithmic datasets. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*. Swarat Chaudhuri, Katerina Fragkiadaki, Mohammad Emtiyaz Khan, and Yizhou Sun (Eds.), ICLR, Vienna, Austria.
[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*. Yan Liu, Chelsea Finn, Yejin Choi, and Marc Deisenroth (Eds.), Curran Associates, Inc., Virtual, 1–22.
[22] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, et al. 2023. Faith and fate: Limits of transformers on compositionality. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS'23)*. Alice Oh, Tobias Naumann, Globerson. Amir, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.), Curran Associates, Inc., New Orleans, Louisiana, USA, 70293–70332.
[23] Gamaleldin F. Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. 2018. Large margin deep networks for classification. In *Proceedings of the 31st Conference on Neural Information Provessing Systems (NeurIPS'18)*. Hanna Wallach, Hugo Larochelle, Kristen Grauman, and Nicolò Cesa-Bianchi (Eds.), Curran Associates Inc., Montréal, Canada, 8420–8431.
[24] Stanislav Fort and Adam Scherlis. 2019. The goldilocks zone: Towards better understanding of neural network loss landscapes. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*. Pascal van Hentenryck and Zhi-Hua Zhou (Eds.), AAAI, Honolulu, HA, USA, 3574–3581.
[25] Kunihiko Fukushima. 1969. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics* 5, 4 (1969), 322–333.

[26] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. 2020. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment* 2020, 11 (2020).
[27] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP'22)*. Trevor Cohn, Yulan He, and Yang Liu (Eds.), Association for Computational Linguistics, Abu Dhabi.
[28] Satvik Golechha. 2024. Progress measures for grokking on real-world tasks. *arXiv e-prints* arXiv:2405.12755 (2024), 1–10.
[29] Eugene Golikov, Eduard Pokonechnyy, and Vladimir Korviakov. 2022. Neural tangent kernel: A survey. *ArXiv e-prints* arXiv:2208.13614 (2022).
[30] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings of the 18th International Joint Conference on Neural Networks (IJCNN'05)*. Daniel S. Levine (Ed.), Vol. 2, IEEE, Montréal, Canada, 729–734.
[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. Lourdes Agapito, Tamara Berg, Jana Kosecka, and Lihi Zelnik-Manor (Eds.), IEEE, Las Vegas, NV, USA, 770–778.
[32] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Flat minima. *Neural Computation* 9, 1 (1997), 1–42.
[33] David T. Hoffmann, Simon Schrodi, Jelena Bratulić, Nadine Behrmann, Volker Fischer, and Thomas Brox. 2024. Eureka-moments in transformers: Multi-step tasks reveal softmax induced optimization problems. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*. Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.), PMLR, Vienna, Austria, 18409–18438.
[34] Michael Y. Hu, Angelica Chen, Naomi Saphra, and Kyunghyun Cho. 2023. Latent state models of training dynamics. *Transactions on Machine Learning Research* (2023), 1–28.
[35] Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. 2024. High-dimensional learning dynamics 2024: The emergence of structure and reasoning. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*. Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.), PMLR, Vienna, Austria, 20722–20745.
[36] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*. Francis Bach and David Blei (Eds.), PMLR, Lille, France, 448–456.
[37] Alan Jeffares, Alicia Curth, and Mihaela van der Schaar. 2024. Deep learning through a telescoping lens: A simple model provides empirical insights on grokking, gradient boosting and beyond. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS'24)*. Angela Fan, Ulrich Paquet, Jakub Tomczak, and Cheng Zhang (Eds.), Curran Associates, Inc., Vancouver, BC, Canada.
[38] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. 2022. Hands-on bayesian neural networks–a tutorial for deep learning users. *IEEE Computational Intelligence Magazine* 17, 2 (2022), 29–48.
[39] Rodney Michael Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, et al. 2023. The semantic scholar open data platform. *arXiv e-prints* arXiv:2301.10140 (2023), 1–8.
[40] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*. ICLR, Toulon, France.
[41] Andrei N. Kolmogorov. 1963. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics* 25, 4 (1963), 369–376.
[42] Sergei Vladimirovich Kozyrev. 2024. How to explain grokking. *arXiv e-prints* arXiv:2412.18624 (2024), 1–6.
[43] Alex Krizhevsky. 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report TR-2009. University of Toronto, Toronto, Canada.
[44] Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. 2024. Grokking as the transition from lazy to rich training dynamics. *arXiv e-prints* arXiv:2310.06110 (2024), 1–26.
[45] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1, 4 (1989), 541–551.
[46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.
[47] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv e-prints* arXiv:1607.06450 (2016), 1–7.
[48] Noam Levi, Alon Beck, and Yohai Bar-Sinai. 2024. Grokking in linear estimators—a solvable model that groks without understanding. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*. Swarat Chaudhuri, Katerina Fragkiadaki, Mohammad Emtiyaz Khan, and Yizhou Sun (Eds.), ICLR, Vienna, Austria.

[49] Hao Li, Zheng Xu, Gavin Taylor, Christoh Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS'18)*. Hanna Wallach, Hugo Larochelle, Kristen Grauman, and Nicolò Cesa-Bianchi (Eds.), Curran Associates Inc., Montréal, Canada.

[50] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. 2022. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis* 59 (2022), 85–116.

[51] Yu Liu, Jiyang Zhang, Pengy Nie, Milos Gligoric, and Owolabi Legunsen. 2023. More precise regression testing selection via reasoning about semantics-modifying changes. In *Proceedings of the 32nd International Symposium on Software Testing Analysis (ISSTA'23)*. Gordon Fraser (Ed.), ACM, Seattle, WA, USA, 664–676.

[52] Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. 2022. Towards understanding grokking: An effective theory of representation learning. In *Processing of the 35th Conference on Neural Information Processing Systems (NeurIPS'22)*. Alekh Argawal, Alice Oh, Danielle Belgrave, and Kyunghyun Cho (Eds.), Curran Associates, Inc., New Orleans, USA.

[53] Ziming Liu, Eric J. Michaud, and Max Tegmark. 2023. Grokking as compression: A nonlinear complexity perspective. *arXiv e-prints* arXiv:2310.05918 (2023), 1–16.

[54] Ziming Liu, Eric J. Michaud, and Max Tegmark. 2023. Omnigrok: Grokking beyond algorithmic data. In *Proceedings of the 11th International Conference on Learning Representation (ICLR'23)*. Yoshua Bengio and Daphne Koller (Eds.), Curran Associates, Inc., Kigali, Rwanda.

[55] Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon Shaolei Du, Jason D. Lee, and Wei Hu Hu. 2024. Dichotomy of early and late phase implicit biases can provably induce grokking. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*. Swarat Chaudhuri, Katerina Fragkiadaki, Mohammad Emtiyaz Khan, and Yizhou Sun (Eds.), ICLR, Vienna, Austria.

[56] Kaifeng Lyu and Jian Li. 2020. Gradient descent maximizes the margin of homogeneous neural networks. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*. Dawn Song, Kyunghyun Cho, and Martha White (Eds.), ICLR, Addis Ababa, Ethiopia.

[57] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*. Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (Eds.), ACL, Portland, OR, USA, 142–150.

[58] William Merrill, Nikolaos Tsilivis, and Aman Shukla. 2023. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. In *Proceedings of the Workshop on Mathematical and Empirical Understanding of Foundation Models (ME-FoMo'23)*. PMLR, Kigali, Rwanda, 1–9.

[59] Jack W. Miller, Charles O'Neill, and Thang D. Bui. 2024. Grokking beyond neural networks: An empirical exploration with model complexity. *Transactions on Machine Learning Research* 2024 (2024).

[60] Beren Millidge. 2022. Grokking 'Grokking'. Post on Beren's Blog. Retrieved May 6, 2026 from https://www.beren.io/2022-01-11-Grokking-Grokking/

[61] Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. 2024. Grokking modular arithmetic can be explained by margin maximization. In *Proceedings of the Workshop on Mathematics of Modern Machine Learning (M3L'24)*. PMLR, Vancouver, Canada.

[62] Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. 2024. Why do you grok? a theoretical analysis on grokking modular addition. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*. Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.), PMLR, Vienna, Austria, 35934–35967.

[63] Reza Moradi, Reza Berangi, and Behrouz Minaei. 2020. A survey of regularization strategies for deep models. *Artificial Intelligence Review* 53, 6 (2020), 3947–3986.

[64] Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D. Manning. 2023. Grokking of hierarchical structure in vanilla transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL'23)*. Jordan Boyd-Graber, Naoaki Okazaki, and Anna Rogers (Eds.), ACL, Toronto, Canada, 439–448.

[65] Tiberiu Musat. 2025. The geometry of grokking: Norm minimization on the zero-loss manifold. *arXiv e-prints* arXiv:2511.01938 (2025), 1–15.

[66] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*. Yoshua Bengio and Daphne Koller (Eds.), Curran Associates, Inc., Kigali, Rwanda, 12572.

[67] Tikeng Notsawo Pascal, Jr, Guillaume Dumas, and Guillaume Rabusseau. 2025. Grokking beyond the Euclidean norm of model parameters. In *Proceedings of the 42nd International Conference on Machine Learning (ICML'25)*. PMLR, Vancouver, BC, Canada, 28552–28618.

[68] Tikeng Notsawo Pascal, Jr., Hattie Zhou, Mohammad Pezeshki, Irina Rish, and Guillaume Dumas. 2024. Predicting grokking long before it happens: A look into the loss landscape of models which grok. In *Proceedings of the 2nd Workshop on Mathematical and Empirical Understanding of Foundation Models (MEU-FM'24)*. ICLR, Vienna, Austria.

[69] Adam Pearce, Asma Ghandeharioun, Nada Hussein, Nithum Thain, Martin Wattenberg, and Lucas Dixon. 2023. Do Machine Learning Models Memorize or Generalize? Post on People + AI Research (PAIR). Retrieved May 8, 2026 from https://pair.withgoogle.com/explorables/grokking/

[70] Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference* (2nd ed.). Cambridge University Press.

[71] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, et al. 2023. RWKV: Reinventing RNNs for the transformer era. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP'23)*. Houda Bouamor, Juan Pino, and Kalika Bali (Eds.), ACL, Singapore, 14048–14077.

[72] Mohammad Pezeshki, Amartya Mitra, Yoshua Bengio, and Guillaume Lajoie. 2022. Multi-scale feature learning dynamics: Insights for double descent. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*. Stefanie Jegelka, Le Song, and Csaba Szepesvari (Eds.), PMLR, Baltimore, USA, 17669–17690.

[73] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv e-prints* arXiv:2201.02177 (2022), 1–10.

[74] Hari K. Prakash and Charles H. Martin. 2025. Grokking and generalization collapse: Insights from HTSR theory. *arXiv e-prints* arXiv:2506.04434 (2025), 1–15.

[75] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *arXiv e-prints* arXiv:1801.06146 (2018), 1–28.

[76] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[77] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. 2014. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* 1, 1 (2014), 1–7.

[78] Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 6 (1958), 386–408.

[79] Elan Rosenfeld and Andrej Risteski. 2024. Outliers with opposing signals have an outsized effect on neural network optimization. In *Proceedings of the 2nd Workshop on Mathematical and Empirical Understanding of Foundation Models (MEU-FM'24)*. ICLR, Vienna, Austria.

[80] Noa Rubin, Imbar Seroussi, and Zohar Ringel. 2024. Grokking as a first order phase transition in two layer networks. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*. Swarat Chaudhuri, Katerina Fragkiadaki, Mohammad Emtiyaz Khan, and Yizhou Sun (Eds.), ICLR, Vienna, Austria.

[81] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.

[82] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.

[83] Alexander Shevchenko and Marco Mondelli. 2020. Landscape connectivity and dropout stability of SGD solutions for over-parameterized neural networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. Emtiyaz Daumé III and Po-ling Loh (Eds.), PMLR, Vienna, Austria.

[84] Kun Song, Zhiquan Tan, Bochao Zou, Huimin Ma, and Weiran Huang. 2024. Unveiling the dynamics of information interplay in supervised learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*. Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.), PMLR, Vienna, Austria, 35934–35967.

[85] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*. PMLR, Honolulu, Hawaii, 35151–35174.

[86] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. 2018. The implicit bias of gradient descent on separable data. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*. Marc'Aurelio Ranzato (Ed.), ICLR, Vancouver, BC, Canada.

[87] Dashiell Stander, Qinan Yu, Honglu Fan, and Stella Biderman. 2024. Grokking group multiplication with cosets. *arXiv e-prints* arXiv:2312.06581 (2024), 1–27.

[88] Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. 2022. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. In *Proceedings of the "Has It Trained Yet?" Workshop (HITY'22)*. New Orleans, USA.

[89] Yuandong Tian. 2025. Provable scaling laws of feature emergence from learning dynamics of grokking. *arXiv e-prints* arXiv:2504.17243 (2025), 1–38.

[90] Yingjie Tian and Yuqi Zhang. 2022. A comprehensive survey on regularization strategies in machine learning. *Information Fusion* 80 (2022), 146–166.

[91] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan M. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS'17)*. Isabel Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), Curran Associates, Inc., Long Beach, CA, USA, 6000–6010.

[92] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *ArXiv e-prints* arxiv:1710.10903v3 (2018).

[93] Bojan Žunkovič and Enej Ilievski. 2024. Grokking phase transitions in learning local rules with gradient descent. *Journal of Machine Learning Research* 25 (2024), 1–52.

[94] Bohan Wang, Qi Meng, Wei Chen, and Tie-Yan Liu. 2021. The implicit bias for adaptive optimization algorithms on homogeneous neural networks. In *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*. Marina Meila and Tong Zhang (Eds.), PMLR, Virtual, 10849–10858.

[95] Boshi Wang, Xiang Yue, Yu Su, and Huan Sun Sun. 2024. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS'24)*. Angela Fan, Ulrich Paquet, Jakub Tomczak, and Cheng Zhang (Eds.), Curran Associates, Inc., Vancouver, BC, Canada.

[96] Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. 2020. Kernel and rich regimes in overparametrized models. In *Proceedings of the 33rd Conference on Learning Theory (COLT'20)*. Jacob Abernethy and Shivani Agarwal (Eds.), PMLR, Vienna, Austria, 3635–3673.

[97] Zhiwei Xu, Yutong Wang, Spencer Frei, Gal Vardi, and Wei Hu. 2024. Benign overfitting and grokking in ReLU networks for XOR cluster data. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*. Swarat Chaudhuri, Katerina Fragkiadaki, Mohammad Emtiyaz Khan, and Yizhou Sun (Eds.), ICLR, Vienna, Austria.

[98] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning?. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL'24)*. Lun-Wei Ku and Vivek Martins, André nd Srikumar (Eds.), Association for Computational Linguistics, Bangkok, Thailand, 10210–10229.

[99] Yuhui Zhang, Michihiro Yasunaga, Zhengping Zhou, Jeff Z. HaoChen, James Zou, Percy Liang, and Serena Yeung. 2023. Beyond positive scaling: How negation impacts scaling trends of language models. In *Proceedings of the Findings of the Association for Computational Linguistics (ACL'23)*. Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.), ACL, Toronto, Canada, 7479–7498.

[100] Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. 2023. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS'23)*. Yoshua Bengio, Daphne Koller, and Ruslan Salakhutdinov (Eds.), NeurIPS Foundation, New Orleans, Louisiana, USA, 1–12.

[101] Xinyu Zhou, Simin Fan, Martin Jaggi, and Jie Fu. 2025. NeuralGrok: Accelerate grokking by neural gradient transformation. *arXiv e-prints* arXiv:2504.17243 (2025), 1–16.

Received 28 February 2025; revised 4 December 2025; accepted 26 April 2026