When Data Falls Short: Grokking Below the Critical Threshold

Vaibhav Singh[1,2] **Eugene Belilovsky**[1,2] **Rahaf Aljundi**[3]
[1]Concordia University [2]Mila [3]Toyota Motor Europe

Abstract

In this paper, we investigate the phenomenon of *grokking*, where models exhibit delayed generalization following overfitting on training data. We focus on data-scarce regimes where the number of training samples falls below the critical threshold, making grokking unobservable, and on practical scenarios involving distribution shift. We first show that Knowledge Distillation (KD) from a model that has already grokked on a distribution (p_1) can induce and accelerate grokking on a different distribution (p_2), even when the available data lies below the critical threshold. This highlights the value of KD for deployed models that must adapt to new distributions under limited data. We then study training on the joint distribution (p_1, p_2) and demonstrate that while standard supervised training fails when either distribution has insufficient data, distilling from models grokked on the individual distributions enables generalization. Finally, we examine a continual pretraining setup, where a grokked model transitions from p_1 to p_2, and find that KD both accelerates generalization and mitigates catastrophic forgetting, achieving strong performance even with only 10% of the data. Together, our results provide new insights into the mechanics of grokking under knowledge transfer and underscore the central role of KD in enabling generalization in low-data and evolving distribution settings.

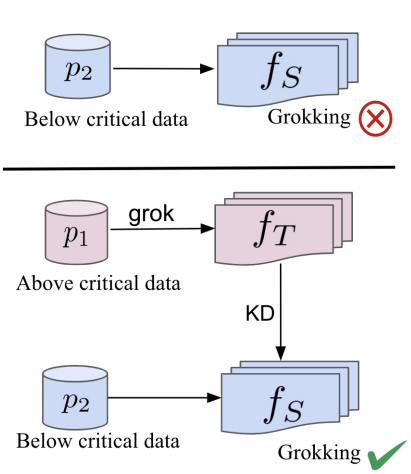
1 Introduction

Generalizing across varying data distributions remains a core challenge in machine learning, as standard training often fails under distribution shifts or data scarcity [33, 7, 30, 29, 15]. The phenomenon of *grokking* [26] sheds light on this problem, showing how models can suddenly generalize after prolonged overfitting [1]. Explanations range from implicit regularization, such as weight decay [2, 23], to training dynamics that enable generalization even at zero loss [12, 19, 9]. A common finding is that grokking occurs only when training data exceeds a *critical threshold* [26, 38].

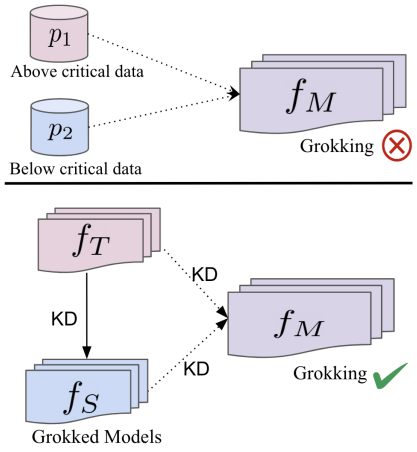
Building on these findings, we explore grokking in scarce data regimes, under distribution shift. Specifically, we ask: **Can a grokked model be leveraged to *train* another model on a different distribution?** To test this, we train a one-layer Transformer [35] on p_1 as a Teacher (f_T) and distill its knowledge into a Student (f_S) on p_2. We find that f_S not only groks on p_2 but also requires fewer steps under distillation. This enables faster adaptation when p_2 has limited data, demonstrating the practical value of pre-grokked models for distribution shift, continual learning, multi-task learning, and domain generalization.

We also show that generalization does not depend on smaller weight norms or weight decay. While prior work linked grokking to a cleanup phase driven by weight decay [23, 17, 34], our experiments refute this claim. Consistent with recent findings [27], we find that weight decay mainly mitigates floating-point errors. Grokking occurs even with zero weight decay and increasing weight norms, ruling out these factors as primary explanations.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Continual and Compatible Foundation Model Updates (CCFM).



(a) Grokking a model on p_2 using KD, which otherwise fails to generalize.



(b) Distilling from multiple grokked models f_T, f_S yields grokking on a larger model f_M below critical data.

Figure 1: Figure 1a shows that $f_S$groks below the critical data size when trained via KD from a grokked model f_T, whereas training from scratch fails. In Figure 1b, a larger model $f_M$trained jointly on p_1 and p_2 cannot generalize when either dataset is below the threshold. Distilling from the smaller grokked models $f_S$and f_T, however, enables f_M to grok and generalize effectively even under scarce data.

We further ask: **Is grokking possible when training data falls *below* the *critical threshold* ?** Our results show that distillation from the Teacher model (f_T) not only reduces the steps required for grokking but also enables it in regimes where data is below the critical size. The critical data size, as defined in [17, 34, 38], is the minimum amount of data below which generalization does not occur. Our study further demonstrates that KD helps mitigate forgetting when models adapt to new domains. Together, these contributions outline a practical framework for building efficient and adaptable learning systems.

2 Related Work

Grokking was first identified in algorithmic tasks by [26]. Subsequent work has aimed to explain this phenomenon. [23] reverse-engineered a grokked modular-addition transformer and found it learns a composition of trigonometric and inverse-trigonometric functions. [21] attributed grokking to competition between sparse (generalizing) and dense (memorizing) subnetworks. From a theoretical lens, [28] framed grokking as a first-order phase transition in feature learning, while [14] provided analytical solutions for loss and accuracy dynamics in linear networks. Studying polynomial regression, [12] linked grokking to a shift from lazy to rich learning. [19] further suggested that the sharp test-accuracy jump arises from differing implicit biases in early vs. late training.

Grokking has also been observed in practical settings, e.g., CNNs trained on CIFAR-10 [8, 11]. [8] describe delayed robustness, where models eventually grok adversarial examples long after interpolation. Early prediction of grokking has been attempted using Fourier spectral signatures [24]. It has been linked to slow formation of useful representations within a "Goldilocks zone" between memorization and confusion [16], and to gradual amplification of structured weights followed by removal of memorized components [23]. Other explanations include hidden SGD-driven amplification of a Fourier gap [2] and the "Slingshot mechanism," where training cycles between stable and unstable phases [32].

Relationship to Dataset Size: Circuit-efficiency analysis shows that generalization is slower but more efficient, implying a critical data size below which models memorize rather than generalize [34]. Training below this threshold yields semi-grokking, and fine-tuning grokked models on such small data can cause "ungrokking." Regularization has been proposed to correct training-sample errors [6], and loss-landscape analysis links grokking to data size, weight decay, and representation learning [17].

Accelerating Grokking: Several methods speed up grokking: gradient decomposition and amplification [13], lottery-ticket approaches [22], transferring embeddings from a weaker to a stronger model [36], and replacing softmax with stable-max [27]. In contrast, our method removes the phase transition without extra data or redundant pretraining, and— to our knowledge— is the first to accelerate grokking in data-scarce settings under distribution shift.

3 Experimental Setup

We trained a decoder only transformer to perform experiments on algorithmic tasks of the form $((a@b)\%P)$, where @ represents operator for any of the binary operations. In this work, we focus on addition and subtraction tasks following previous studies [23, 34, 17, 26, 16] which consistently report grokking on these tasks. The model input is $[a, b, @, P]$, and the output c is read from the final token P. In our experiments, each arithmetic modulo-P task is denoted as p_1 for a specific prime P. A distribution shift is introduced by changing the modulus while keeping the operator fixed. For example, in addition modulo P, the task $(a + b)\%P$ with $P = P_1$ is referred to as distribution p_1, while $(a + b)\%P$ with $P = P_2 \neq P_1$ is denoted p_2. Our results remain consistent regardless of the choice of P_1 and P_2.

We begin training with 35% of the dataset to first observe grokking, as demonstrated in prior work [26, 17]. We then progressively reduce the data fraction to 30%, 25%, and 10%. For algorithmic addition and subtraction tasks, prior studies define the critical data size to be around 25% of the dataset [34, 38]. Consistent with this, our observations show that grokking does not occur when 25% or less of the data is available, confirming that 25% marks the critical threshold below which generalization becomes impossible. We utilize StableMax Cross Entropy [27] since cross entropy with softmax function causes numerical instability, given as:

$$L_{\text{StCE}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\log \text{StableMax}(f_S^y(x;\theta))\right] \tag{1}$$

where $f_S(\cdot;\theta)$, is the Student Network: parameterized by θ and StableMax(x) = Softmax($g(x_i)$) where $g(x_i)$ is defined as

$$g(x) = \begin{cases} \log(x+1) & \text{if } x \geq 0 \\ -\log(-x+1) & \text{if } x < 0 \end{cases} \tag{2}$$

We observe that the usage of StableMax [27] already gives a prior speedup in inducing grokking, needing around 6000 iterations to grok, which otherwise would have been in order of $1e4$ [17, 26, 23]

For knowledge distillation, we use Kullback-Leibler (KL) Divergence Loss:

$$L_{\text{KL}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_X}\left[D_{\text{KL}}\left(q_T(x)\|q_S(x;\theta)\right)\right] \tag{3}$$

where $D_{\text{KL}}(p\|q) = \sum_{i=1}^{K} p_i \log\left(\frac{p_i}{q_i}\right)$. This takes softened outputs as $q_T(x) = \text{softmax}\left(\frac{f_T(x)}{t}\right)$, and $q_S(x;\theta) = \text{softmax}\left(\frac{f_S(x;\theta)}{t}\right)$ where f_T :, represents the Teacher model, and $t > 0$ is the Temperature used to soften probabilities.

The total distillation loss is therefore realised as:

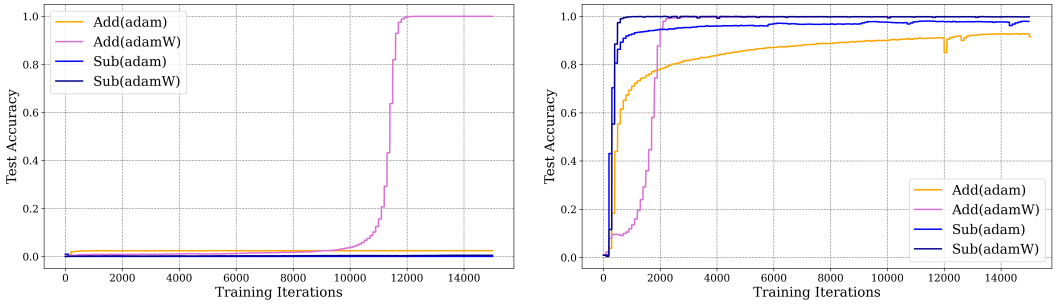
$$L(\theta) = (1-\alpha)L_{\text{CE}}(\theta) + \alpha L_{\text{KL}}(\theta) \tag{4}$$

where α controls the proportion of each loss component.

For demonstrating the efficacy of our distillation method and to negate the dependency of weight norm and weight decay theories, we compare both Adam without weight decay & AdamW(with weight decay) optimizer [18] with a learning rate $\gamma = 1e-3$. For AdamW we set the weight decay parameter $\lambda = 1$. We perform 15,000-30,000 epochs of training with a batch size of 2048 on NVIDIA V100 GPU.

4 Accelerating Grokking through Knowledge Distillation (KD)

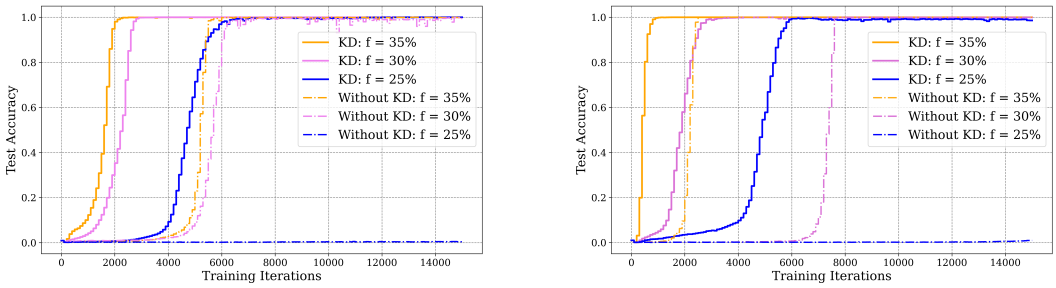
KD has been shown to provide multiple benefits in improving training dynamics. [20] provided a statistical perspective on distillation, that providing the true class-probabilities from the teacher



(a) Training on p_2 without KD (f = 30%)

(b) Training on p_2 with KD from p_1 (f = 30%)

Figure 2: Figure 2b shows the effectiveness of KD **irrespective of the optimizer choice** for both addition and subtraction modulo task. In Figure 2a typical grokking phenomena on distribution p_2 on 30% of training data (denoted as f), without KD. We observe that weight decay is helpful in showing grokking but its not the only underlying cause. When trained with Adam, grokking is not observed for both tasks when trained for 15000 iterations. This concurs with [26]. However Figure 2b demonstrates a $Student$ model trained on a different distribution p_2 with same fraction, but now with KD from the $Teacher$ model trained on p_1 displays accelerated generalization irrespective of the optimizer choice.



(a) Accelerated Grokking on p_2 for Task: $(a + b)\%107$

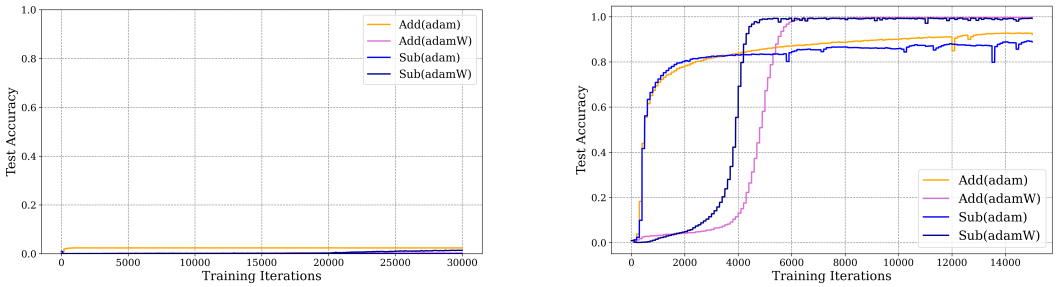
(b) Accelerated Grokking on p_2 for Task: $(a - b)\%107$

Figure 3: Dashed lines inFigure 3a and Figure 3b show typical grokking on p_2 $(P = 107)$ with different training fractions (f). Training from scratch below 30% shows no grokking. With KD from a grokked model on p_1 $(P = 113)$, grokking is accelerated and occurs with as little as 25% of p_2. Distillation is applied to probability outputs from the operator token, enabling generic operator-level representations rather than P-specific ones.

model can lower the variance of the student objective, and thus improve performance. Further [25] provides a generalization bound that establishes fast convergence of the expected risk of a distillation-trained linear classifier. In [4, 3], a theoretical framework is given to analyze model distillation into decision trees through PAC-learning statistical theory. They show that if teacher model $f_T$is perfectly distillable into a student model f_S, then with a probability of at least $1 - \delta$, $f_S$generalizes with training samples no greater than $\left\lceil \frac{\log(1/\delta)}{\epsilon} \right\rceil$. It can be inferred from these studies [31, 5, 37] that KD brings the following advantages towards training dynamics,

- **Regularization Effect through Label Smoothing:** KD smooths the labels, which acts as a regularizer and prevents overfitting.
- **Domain Knowledge Injection:** The teacher model imparts class relationships that shape the geometry of the student's logit layer.
- **Instance-Specific Knowledge:** The teacher adjusts the student model's per-instance gradients based on the difficulty of each sample, facilitating more effective learning.

We first grok a 1 layer Transformer model with 35% of training data for modular addition and subtraction tasks $((a \pm b)\%P)$ with $P = 113$ (Choice of P was aritrary). We call this as data distribution p_1. This model will act as the teacher model (f_T). Next we train another model on a



(a) Without KD on 20% of p_2

(b) With KD on 20% of p_2

Figure 4: Figure 4a demonstrates that its impossible to observe grokking when the data fraction goes below a certain critical threshold(20%.), even with 2X iterations (30,000) In such a case, the model does not learn anything regardless of the optimizer. In Figure 4b, it can be clearly seen that with KD, grokking is observed for all tasks, even without weight decay. However we notice that weight decay helps in achieving a better generalisation.

distribution p_2, by modifying the modulo prime ($P = 107$), and compare the impact of KD under different fractions for p_2 on both these tasks. As seen in Figure 3, KD significantly accelerates the grokking process for both tasks, even in scenarios where the proportion of training data is below critical dataset size. This observation is independent of the optimizer used, as shown in Figure 2. This demonstrates a practical utility of grokked models, illustrating their effectiveness in training models on varying distributions through KD. It is important to note that distillation occurs on the probability outputs from the operator token rather than the P token. This approach aims to learn generic operator-level representations instead of task-specific representations, which would depend on the choice of P.

Building on these observations, we ask: ***Can KD enable generalization below the critical data threshold?*** To test this, we repeat the experiments with only 20% of the data. Without KD, grokking does not occur even after 30,000 iterations, regardless of weight decay. In contrast, with KD, generalization emerges rapidly at this reduced data fraction (Figure 4). Notably, the weight norm continues to increase (more details on weight norm given in A.1), reinforcing our earlier claim that neither weight decay nor decreasing weight norms are essential for grokking. These results underscore the value of a grokked Teacher model (f_T) in data-scarce settings. KD not only accelerates grokking but also makes it possible below the critical threshold, highlighting its practical utility for efficient training under limited data and shifting distributions.

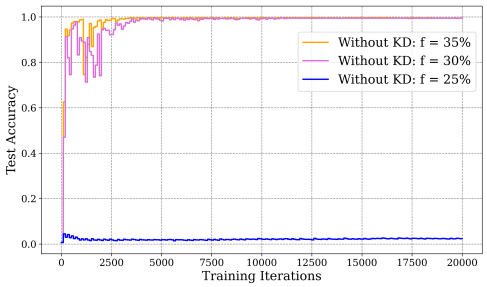
5 Leveraging Grokked Models for Distillation and Continual Transfer

We extended our study by checkpointing grokked models on fractions $(0.35, 0.3, 0.25)$ of p_2 trained via distillation (Section 4). The model grokked on p_1 with 35% data is denoted f_{p_1}, and the KD-trained models on p_2 as f_{p_2}. Our goal was to train a new transformer (f_M) that generalizes across both p_1 and p_2. In joint training, f_M failed to generalize when p_2 was below the critical size, showing that scarcity in any distribution limits learning. In contrast, training f_M solely via KD from f_{p_1} and f_{p_2} enabled simultaneous generalization, even when either distribution had limited data (Figure 5)

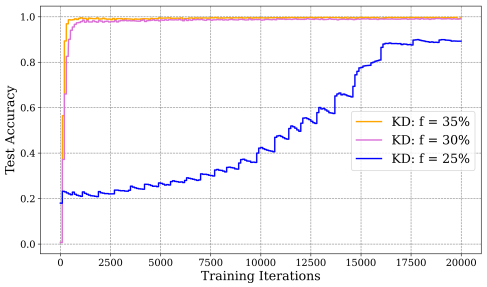
Remarkably, f_M exhibited grokking ***only when trained via KD***, generalizing even when p_1 or p_2 was below the critical size. This shows that KD over the joint distribution (p_1, p_2) provides a stronger learning signal than ground-truth labels, enabling grokking under limited data. Notably, the effect persists even when the teacher f_{p_2} itself was trained on a small fraction of p_2, distilled from f_{p_1}.

Building on recent work in continual pretraining [10], we evaluated the transition of a grokked model from p_1 to p_2, focusing on the role of knowledge distillation (KD) in mitigating catastrophic forgetting. A model grokked on p_1 was further pretrained on p_2 under two conditions: with and without KD.

As shown in Figure 6, training without KD led to immediate and severe forgetting of p_1, though the model quickly generalized to p_2. With KD, however, the model retained near-perfect accuracy on p_1 while also achieving rapid generalization on p_2. KD thus prevented delayed generalization

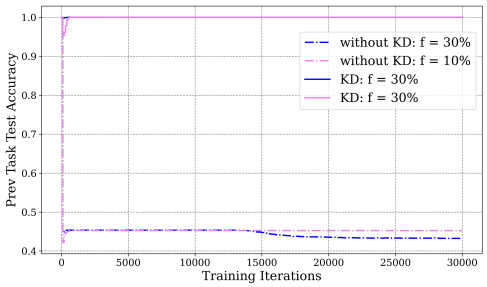


(a) without KD

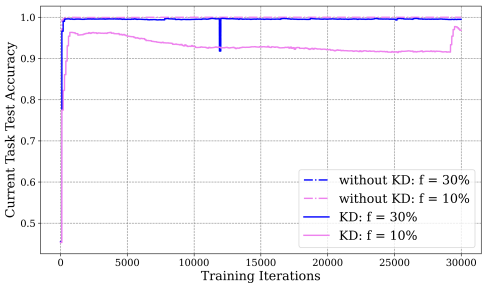


(b) With only KD, without entropy minimization

Figure 5: Performance comparison of training strategies for a larger transformer model f_M on distributions of $p_1(35\%)$ and different fractions $(0.35, 0.3, 0.25)$ of p_2. In the Joint Training regime (Figure 5a), the model fails to generalize via cross-entropy when data from either distribution falls below the critical threshold. In contrast, training solely with distillation enables grokking even with 25% of p_2 (Figure 5b). At this low fraction, generalization does not reach unity due to the imperfect f_{p_2} trained under data scarcity, while for 0.35 and 0.3 fractions, generalization is rapid with no grokking.



(a) Previous Task Accuracy for different fractions of data, with and without KD.



(b) Current Task Accuracy for different fractions of data, with and without KD.

Figure 6: Continual pretraining where a grokked model on p_1 is further trained on p_2. Without KD, performance on p_1 drops rapidly while generalization on p_2 is quick. With KD (Figure 6b), accuracy on the current task is preserved and forgetting is mitigated. Training from a grokked model enables fast generalization without delayed grokking, though for data below the critical size we observe a sudden phase transition from $\sim 92\%$ to 100% accuracy at around $28K$ steps.

and preserved prior knowledge, underscoring its role in balancing retention and adaptation during continual pretraining.

These findings highlight the utility of KD in enabling generalization when data from multiple distributions is scarce, a scenario common in practice due to privacy, security, or resource constraints. Leveraging KD from pre-trained grokked models offers an effective solution in such settings. Finally, across all experiments we observed increasing weight norms despite successful grokking, challenging theories that link grokking to weight norm reduction. Instead, our results suggest that mechanisms like representation transfer via KD play a more central role, opening new directions for understanding the fundamental drivers of grokking.

6 Conclusions and Future Work

This study advances the understanding of grokking by examining its behavior below the critical data regime. Unlike prior work centered on single distributions and weight-norm dynamics, we show that Knowledge Distillation (KD) can induce and accelerate grokking without relying on weight decay or decreasing norms. Our results demonstrate that KD enables generalization even below the critical threshold and across varying distributions, offering a practical solution in data-scarce settings where traditional training fails. Future work may extend these insights to more complex tasks, deepen our understanding of grokking's mechanisms, and explore broader uses of pre-grokked models that generalize reliably and adapt efficiently to dynamic real-world data.

References

[1] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 2017.

[2] Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

[3] Shalev Ben-David and Shai Ben-David. Learning a classifier when the labeling is known. In *International Conference on Algorithmic Learning Theory*, pages 440–451. Springer, 2011.

[4] Enric Boix-Adsera. Towards a theory of model distillation, 2024.

[5] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[6] Darshil Doshi, Aritra Das, Tianyu He, and Andrey Gromov. To grok or not to grok: Disentangling generalization and memorization on corrupted algorithmic datasets. *arXiv preprint arXiv:2310.13061*, 2023.

[7] Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. *Advances in neural information processing systems*, 33:11996–12007, 2020.

[8] Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Deep networks always grok and here is why. *arXiv preprint arXiv:2402.15555*, 2024.

[9] Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do we need zero training loss after achieving zero training error? *ArXiv*, abs/2002.08709, 2020.

[10] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual learning of language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[11] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[12] Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024.

[13] Jaerin Lee, Bong Gyun Kang, Kihoon Kim, and Kyoung Mu Lee. Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*, 2024.

[14] Noam Itzhak Levi, Alon Beck, and Yohai Bar-Sinai. Grokking in linear estimators – a solvable model that groks without understanding. In *The Twelfth International Conference on Learning Representations*, 2024.

[15] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pages 1–34, 2024.

[16] Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022.

[17] Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *The Eleventh International Conference on Learning Representations*, 2022.

[18] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[19] Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon Shaolei Du, Jason D. Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7632–7642. PMLR, 18–24 Jul 2021.

[21] William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *arXiv preprint arXiv:2303.11873*, 2023.

[22] Gouki Minegishi, Yusuke Iwasawa, and Yutaka Matsuo. Bridging lottery ticket and grokking: Is weight norm sufficient to explain delayed generalization? *arXiv preprint arXiv:2310.19470*, 2023.

[23] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[24] Pascal Notsawo Jr, Hattie Zhou, Mohammad Pezeshki, Irina Rish, Guillaume Dumas, et al. Predicting grokking long before it happens: A look into the loss landscape of models which grok. *arXiv preprint arXiv:2306.13253*, 2023.

[25] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5142–5151. PMLR, 09–15 Jun 2019.

[26] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[27] Lucas Prieto, Melih Barsbey, Pedro A. M. Mediano, and Tolga Birdal. Grokking at the edge of numerical stability. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] Noa Rubin, Inbar Seroussi, and Zohar Ringel. Grokking as a first order phase transition in two layer networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[29] Vaibhav Singh, Rahaf Aljundi, and Eugene Belilovsky. Controlling forgetting with test-time data in continual learning. *arXiv preprint arXiv:2406.13653*, 2024.

[30] Vaibhav Singh, Anna Choromanska, Shuang Li, and Yilun Du. Wake-sleep energy based models for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4118–4127, 2024.

[31] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*, 2020.

[32] Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

[33] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[34] Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023.

[35] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[36] Zhiwei Xu, Zhiyu Ni, Yixin Wang, and Wei Hu. Let me grok for you: Accelerating grokking via embedding transfer from a weaker model. In *The Thirteenth International Conference on Learning Representations*, 2025.

[37] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[38] Xuekai Zhu, Yao Fu, Bowen Zhou, and Zhouhan Lin. Critical data size of language models from a grokking perspective. *CoRR*, abs/2401.10463, 2024.

A Appendix

A.1 Evolution of L2 weight norm

L2 weight norm trained with adam continuously increases for both addition and subtraction tasks as observed in Figure 7, yet grokking still occurs. These findings challenge the theories proposed by [23] who suggest that the abrupt transition to perfect test accuracy during grokking occurs in the cleanup phase (where weight decay removes memorization components), following the establishment of the generalizing mechanism. Our empirical evidence contradicts these claims by demonstrating grokking even without weight decay and with increasing weight norms.

Similarly [17] induce grokking by increasing the initial weight norm and conclude that generalizing solutions lie on smaller norm spheres in parameter space. While we acknowledge that an initially higher weight norm can facilitate grokking, our results indicate that generalizing solutions do not necessarily lie on smaller norm spheres. Our modular arithmetic tasks serve as counterexamples, where the final generalizing solutions exhibit larger parameter weight norms than their initial states, and grokking occurs without the application of weight decay.

Furthermore [34] claim that the transition from memorizing to generalizing circuits occurs because the generalizing circuit is more "efficient" than the memorizing circuit, in the sense that it can produce equivalent loss with a lower parameter norm. In contrast, our studies show that modular arithmetic tasks can achieve generalizing solutions with higher parameter norms without any weight decay, disproving the necessity of norm reduction for grokking.

We empirically demonstrate that parameters' (L2) weight norm trained with adam continuously increases for both addition and subtraction tasks as observed in Figure 7, yet grokking still occurs. This challenges the notion that decreasing weight norm is fundamental to grokking. Therefore, we assert that neither parameter weight decay nor a decreasing weight norm during optimization is inherently fundamental to observing grokking, contrary to its purported necessity in previous studies [17, 23, 34] on modular arithmetic tasks.

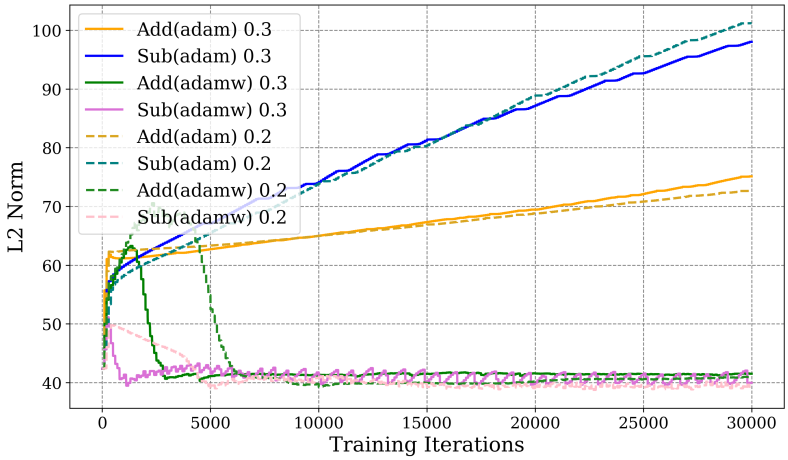


Figure 7: Evolution of the L2 weight norm for $Student$ model f_S trained with Adam(without weight decay) and AdamW(with weight decay) on different fractions of p_2 distribution. f_S is trained via KD from a grokked model f_T . Notably, training without weight decay the L_2-weight norm increases continuously, while giving generalised solutions. This rules out the necessity of decreased weight norm condition for exhibiting grokking given by [17, 34, 23]