ECE 826 Project - Grokking

Huanran Li
Department of Electrical Engineering
Wisconsin Institute of Discovery
University of Wisconsin-Madison
Madison, WI 53705
hli488@wisc.edu

Sadman Sakib
Department of Computer Science
University of Wisconsin-Madison
Madison, WI 53705
ssakib@wisc.edu

Abstract

This project explores "grokking," where a neural network model begins to generalize long after overfitting the dataset. This initial exploration aims to replicate findings from existing research and deepen our understanding by addressing key questions: what triggers it, why it occurs, and when it happens. So far, our work has revealed that batch size, weight decay rate, and random noise significantly influence grokking. We have visualized the loss landscape around grokking and non-grokking optima and shown that grokking optima are surrounded by more uneven surfaces. Additionally, we observed that the norm of attention weights tends to spike during the pre-grokking stage, which needs further investigation as a potential predictor of grokking during training. We plan to continue this work, aiming to contribute our findings to a published paper.

1 Opening Questions & Our Short Answers

1. Is there a way to **predict** the occurrence of Grokking in advance?
   Answer: So far we don't know exactly when grokking will happen during training. But we found strong pattern between grokking, training dataset size, and batch size. (See Figure 4)
2. How is grokking influenced by **augmentation and noise**?
   Answer: YES.
   (a) For MNIST, we observe no grokking delay with augmentations (see Table 3).
   (b) For Arithmetic, we add Gaussian noise to the input embeddings, and we see positive correlation (0.123) between the noise level and grokking delay time. (see Table 1).
3. The implications of grokking: **good or bad**?
   Answer: Bad. We perform 3600 runs over different hyperparameter combinations. In summary, small grokking delay implies faster improvement for validation (See Figure 3).
4. **Visualization** of model weights during and post-learning for insights.
   Answer: The loss contour near non-grokking optima is much more smooth than the grokking optima (See Figure 5).
5. What is the impact of **model architecture and dataset** on Grokking?
   Answer: From our experiment, Transformer models are more vulnerable to grokking compared to MLPs.
   (a) By adding small augmentations or switching to a more effective loss functions, grokking on MLP can be easily resolved (See Table 3).
   (b) The grokking on Transformer is related to batch size, learning rate, weight dacay rate, input noise level (see Table 1).

Preprint. Under review.

2 Preliminary

Grokking is a phenomenon when the validation accuracy continues to improve significantly after the training accuracy has plateaued. Grokking usually occurs when the dataset size of training/validation ratio is small (less than 40/60). For later analysis, we denote the grokking delay as the number of steps/epochs between the points when training and validation accuracies reach 95%. (see Figure 1).

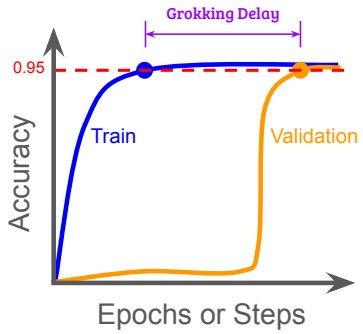


Figure 1: Illustration of Grokking.

Summarizing from the literature (Section 5) and our insights, we are interested in the effect of these hyperparameters on grokking in this project:

- Weight initialization scale (1) & Weight decay rate (1)
- Learning rate (2)
- Training dataset size (3) & Batch size
- Noise/augmentation level

3 Grokking on Arithmetic Dataset with Transformer

We trained a decoder-only transformer model on an arithmetic dataset identical to the setup in (3). The dataset includes operations of the form $(x \, o \, y) \bmod p = z$, where o is an arithmetic operation (addition, subtraction, multiplication, division) and p is a prime number. The model we used for the experiments is a 2-layer transformer model with 4 heads in one transformer block. The dimension of the embeddings is 128, and the hidden layer within the transformer block has an internal width 4 times larger. For optimization, we utilized the AdamW optimizer with Cross Entropy loss and Weight Decay regularization.

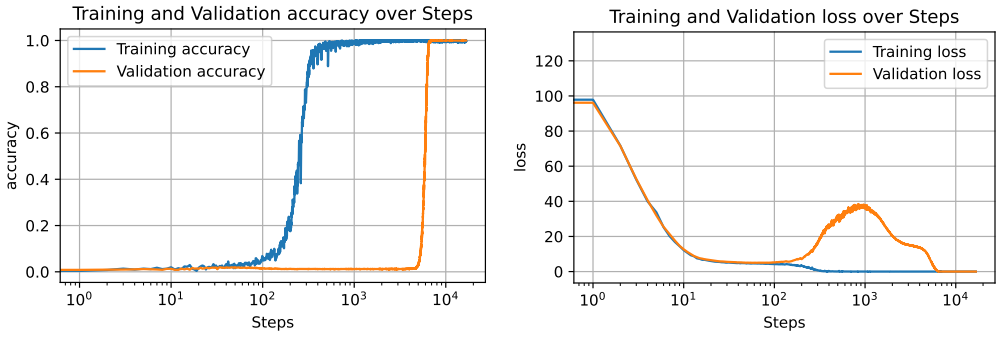


Figure 2: Experiment of grokking on arithmetic dataset (subtraction operation).

We have conducted experiments across all operations. An example is shown in Figure 2, where we observe a grokking delay of 5800 steps between the time of training and validation accuracies

reach 95% in the division operation, with a training/validation ratio of 30/70. Additionally, we have plotted the loss graphs, which display a common pattern observed in transformer models during grokking: the validation loss increases during the pre-grokking stage when training accuracy is high and validation accuracy is low.

3.1 Hyperparameter's Correlation to Grokking Delays

We trained the transformers on a division dataset with varying sizes of training datasets and batch sizes, with the results detailed in Table 1 and Figure 3. The size of the training dataset is controlled by the scale of the prime number p. For all experiments, we set dataset size of training/validation ratio to be 30/70, because this ratio clearly yet swiftly demonstrates the grokking delay within 10,000 training steps. This same setting is consistently applied throughout Sections 3.2 and 3.3 as well. Here are our findings:

- Batch size shows the highest correlation, indicating that a smaller batch size is likely to reduce the grokking delay.
- Weight decay exhibits the lowest negative correlation, suggesting that a larger weight decay is likely to decrease the grokking delay. This finding is consistent with findings on CNN in (1).
- The scale of weight initialization appears to have no impact on Transformer models, which contrasts with the effects reported for CNNs in (1).
- Large grokking delays are typically caused by delayed validation accuracy improvements rather than early training accuracy improvements, suggesting that grokking represents sub-optimal training behavior (See Figure 3).

Parameter	Choices	Correlation to Grokking Delay (Epoch)
Batch Size	$8, 16, 32, 128, 512, 1024$	$\mathbf{0.538}$
Noise Level	$10^{-2}, 10^{-1}, 0, 1, 10$	$\mathbf{0.123}$
Init Scale	$10^{-2}, 10^{-1}, 1, 10, 100$	-0.001
Learning Rate	$10^{-4}, 10^{-3}, 10^{-2}$	-0.063
Weight Decay	$10^{-2}, 10^{-1}, 1$	$\mathbf{-0.26}$

Table 1: All choice of hyperparameter we tried, and its correlation with the grokking delay in epochs.

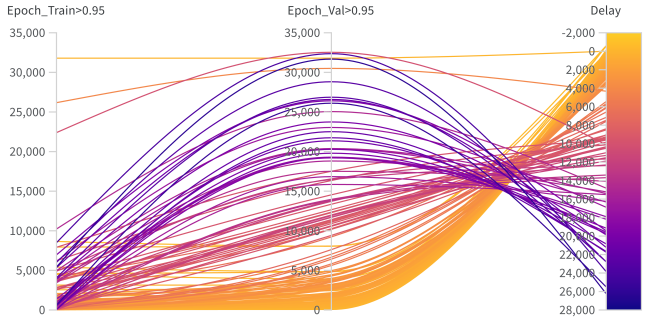


Figure 3: Parallel coordinates plot of runs with different hyperparameter setup. Each line is representing a single run. The first two columns represent the epoch when training/validation accuracy reach 95%.

3.2 The effect of Batch Size, Size of Training Dataset to Grokking

Following this observation, we performed another group of tests focused on batch sizes and training dataset size. The results are shown in Table 2. We observe that:

- **Row "Epochs"** vs **Row "Steps"**: Large batch size and small training set makes the transformers takes MORE epochs but LESS steps to converge.
- **Column "Grokking Delay"**: Comprehension (short grokking delays) forms a narrow band between confusion and grokking. This contradicts our general convention that grokking is the intermediate area between comprehension and confusion.
- The hyperparameter used in (3) is $(1024, 2.8K)$, which is located at the right bottom corner of the heat map with large grokking delays.

We suspect that increasing the training/validation ratio will widen the comprehension band, thereby making grokking less likely to occur.

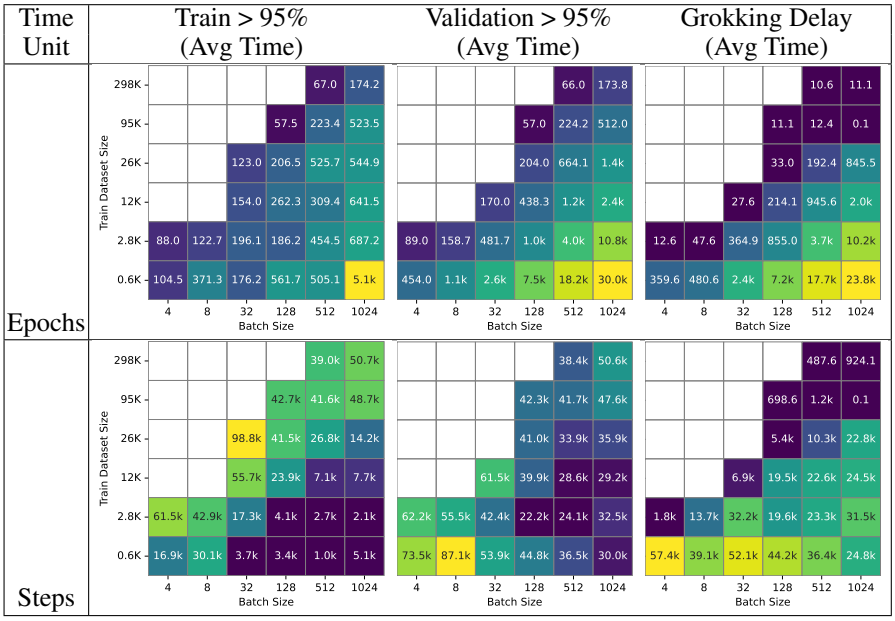


Table 2: Average time of training and validation accuracy reach 95% with different combination of batch size and training dataset size.

Furthermore, we classify all runs into three categories: 1) Grokking (grokking delay > 1000 steps); 2) Comprehension (grokking delay < 1000 steps); 3) Confusion (other cases). For all 3600 runs, we calculated the percentage of each category and plotted the results in Figure 4. We observe that the comprehension region, characterized by a (train size / batch size) ratio mostly ranging from 20 to 800, lies between the confusion and grokking regions.

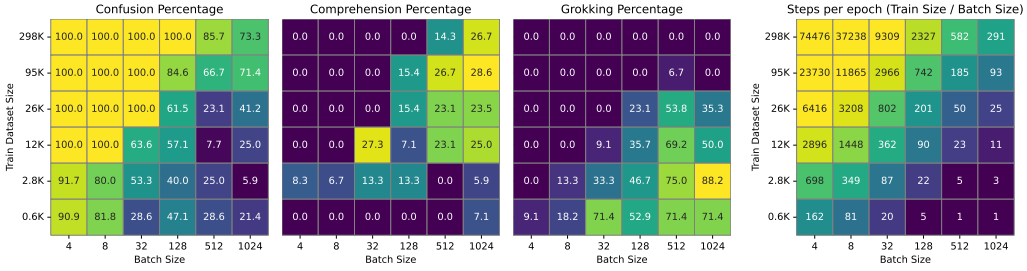


Figure 4: **Left 3:** The percentage of confusion/comprehension/grokking. **Right 1:** The ratio of (train size / batch size) for reference. Comprehension mostly lies in the ratio of [20, 800].

3.3 Loss Landscape Visualization

Following the methods outlined by (4) for visualizing the loss landscape of neural networks, we plotted the contour of loss around the optima. Specifically, after retrieving the final weights post-training, we randomly generated two sets of gradients for each weight from a Gaussian distribution and manually performed gradient descent in these two directions to calculate the loss. Figure 5 presents a landscape comparison between grokking and non-grokking optima. Additional landscapes are detailed in Table 4 in Appendix A.

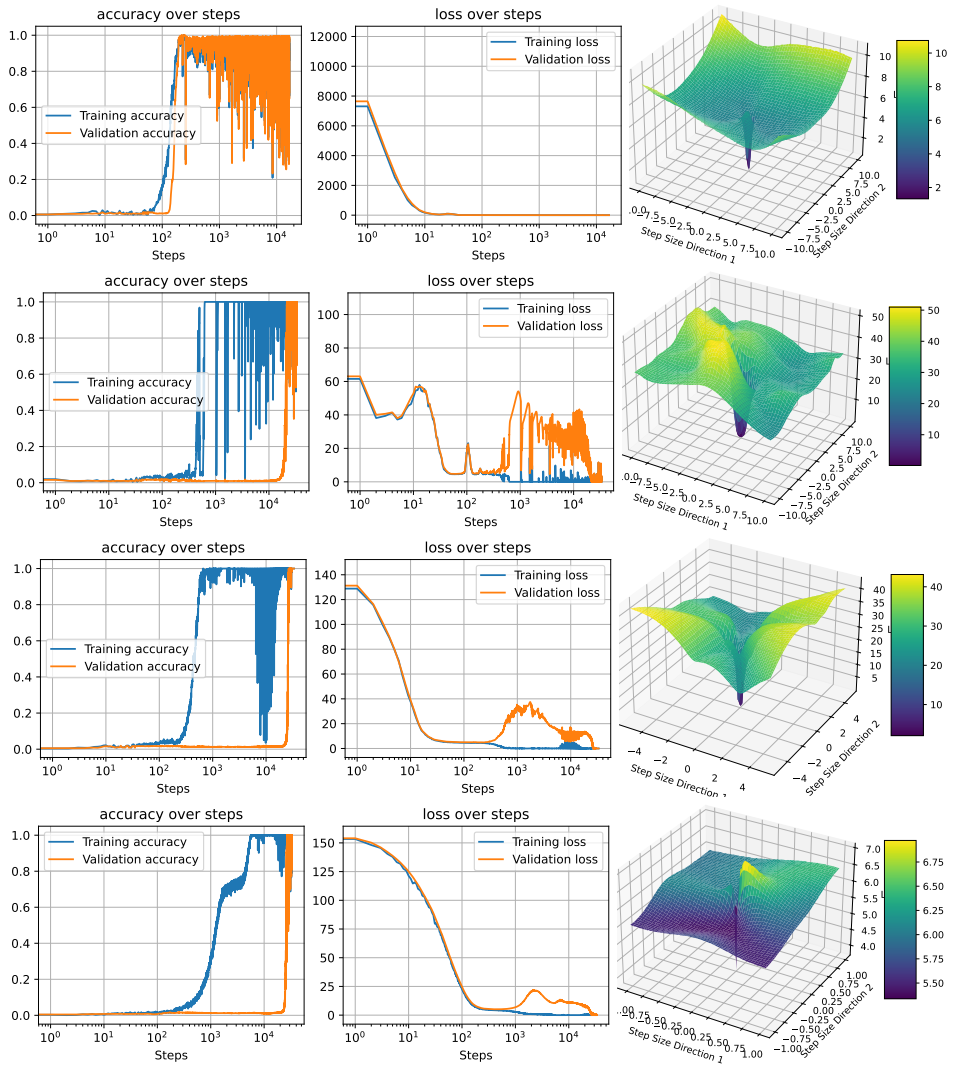


Figure 5: Landscape of loss around optima from non-grokking (Row 1) and grokking (Rows 2 to 4) training processes. Row 2 displays a highly even surface; Row 3 features a large valley leading to the optima; Row 4 presents an ultra-flat surface around the optima, except for a single mountain. These three cases cover 95% of all the scenarios we observed with grokking.

From our analysis of all landscapes, we find that:

- Compared to non-grokking (comprehension) optima, the landscape of grokking optima is more uneven, and the range of loss can be eight times larger.
- Even when non-grokking optima fail to reach 95% accuracy in both training and validation, their landscape remains considerably flatter than that of grokking optima (see Row 2 of Table 4 in Appendix).

- Both grokking and non-grokking solutions experience a certain level of oscillations. This phenomenon should be further studied with (5).

3.4 Model Weights

We investigated how the model weights change during training in different grokking delays. Figure 6 shows the grokking behavior for three training runs on the algorithmic dataset with different training/validation ratios. We observed that norm of attention Q, K, V matrices can take much larger values during training when grokking happens. (1) also indicated that larger weights can facilitate grokking.

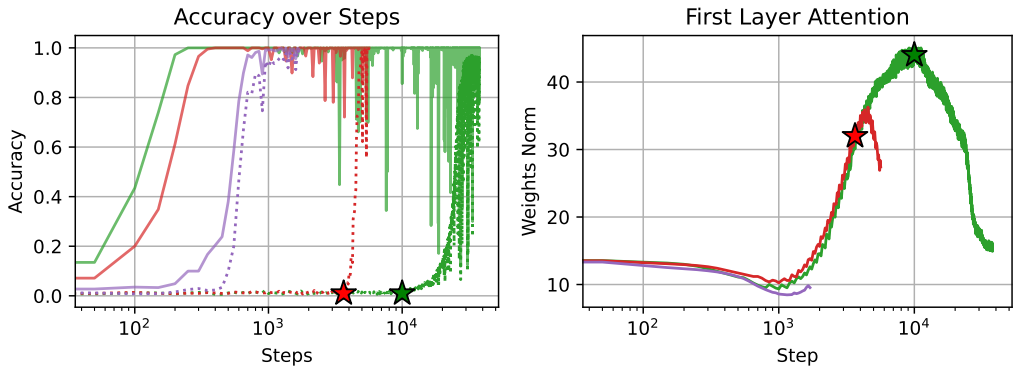


Figure 6: Accuracy and norm of attention weights for grokking (red, green) and non-grokking (purple) process.

Further analysis of the weight norm may predict the onset of grokking, as the mean of the attention weights continues to increase during the pre-grokking stage. Subsequently, as validation accuracy begins to improve, the norm of the weights starts to decrease. We also believe there is a correlation between large weights and a more rugged landscape of the loss function, and this needs further investigation.

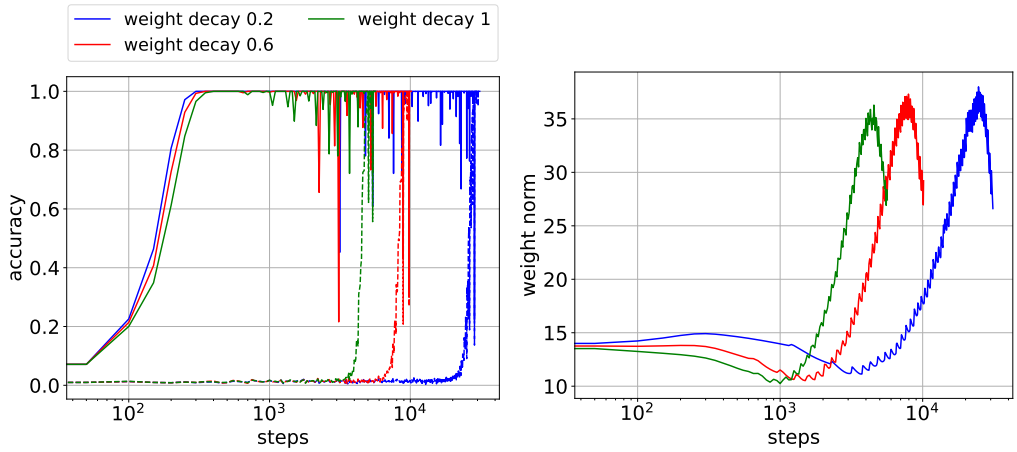


Figure 7: Accuracy and norm of attention weights for weight decay smaller than 1.

In figure 7, we show that for the same training dataset, reducing weight decay can increase grokking delay. This negative correlation was also pointed out in section 3.1. However, we observed that the norm of the attention weights reaches almost same peak value in all cases with smaller weight decay taking longer to reach the peak. We suspect that this is because smaller weight decay makes the model more biased towards training dataset delaying the generalization process.

Figure 8 shows the grokking behavior on same training dataset with weight decay values greater than 1. We saw that increasing the weight decay upto certain value can reduce the grokking delay. The

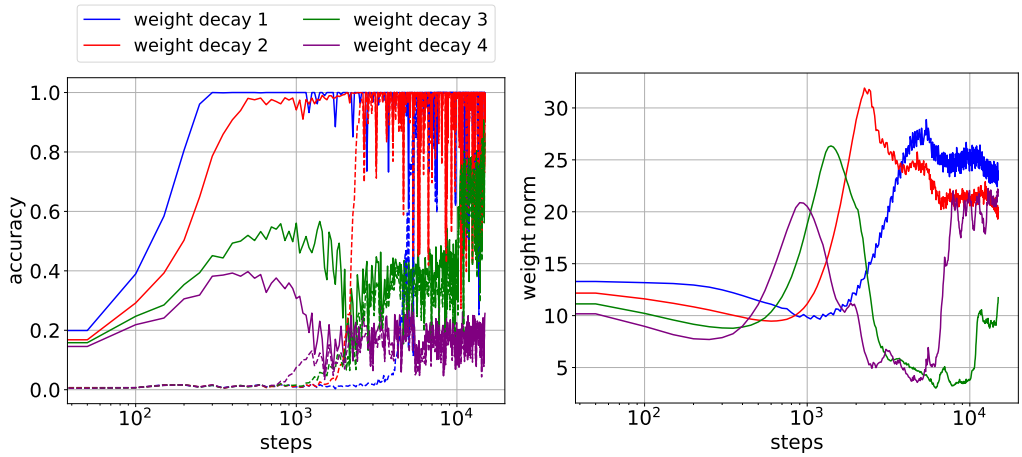


Figure 8: Accuracy and norm of attention weights for weight decay larger than 1.

model with weight decay 2 has about half the grokking delay of weight decay 1. For larger weight decay values, the training and validation accuracy stays at a lower value. The figure also shows that in the overfitted models the peak of attention weight norms reach larger value than the final converged value. On the other hand, in slow-learning models, the peak reaches smaller values and then decreases without converging. We suspect that a certain amount of weight norm is required for a model to generalize from such small training dataset. The penalty from high weight decay hinders reaching this threshold value in early training iterations and so generalization gets delayed.

4 Grokking on MNIST

Following the approach and code outlined in Omnigrokking paper (1), we replicated their experiment using the MNIST dataset. Our efforts successfully reproduced the figure presented in their study, as shown in Figure 9. Additionally, we plotted the Mean Squared Error (MSE) loss functions used by the authors.

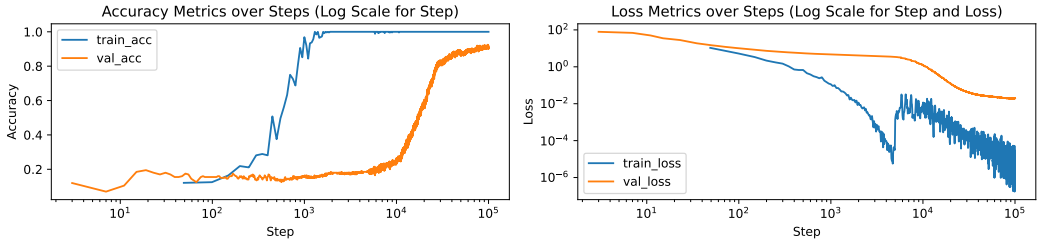


Figure 9: Reproduced Experiment on Grokking of MNIST.

We trained the same MLP on 1k MNIST images but made separate changes to the training dataset size, augmentation strategy, loss function, optimizer, batch size, regularizer, and weight initialization scale. The results are shown under Table 3. Except for the findings from (1) on weight decay rates and weight initialization scale, we observed that:

1. Grokking demonstrates a high sensitivity to image augmentations. The degree of augmentation is specified by two parameters: a Random Rotation with an angle of $\sigma \times 30^\circ$, and a Gaussian Blur characterized by a kernel size of 3 and a standard deviation of $\frac{\sigma}{2}$. Consequently, when the algorithm is subjected to a rotation of 6° and a Gaussian blur with a standard deviation of 0.1, it fails to achieve grokking.

2. Grokking shows a significant dependency on the choice of loss function. Notably, when switching from Mean Squared Error (MSE) to Cross-Entropy (CE), there is a marked improvement: the validation accuracy increases at a rate comparable to the training accuracy.

Data		Training					Results		
Size	Augs(σ)	Loss Fn	Optimizer	BS	Regularizer	Init Scale	Grkin?	Comment	Figure
1K	-	MSE	AdamH	200	W Decay	8.0	✓	Original Method	9
2K	-	MSE	AdamH	200	W Decay	8.0	✓	Exactly Same	
3K	-	MSE	AdamH	200	W Decay	8.0	✓	Grokking Faster, Train slower	13
4K	-	MSE	AdamH	200	W Decay	8.0	✓	Grokking Faster, Train slower	14
5K	-	MSE	AdamH	200	W Decay	8.0	✗	Normal Results	15
1K	0.1	MSE	AdamH	200	W Decay	8.0	✓	Exactly Same	
1K	0.2	MSE	AdamH	200	W Decay	8.0	✗	Test acc little slower	16
1K	0.3	MSE	AdamH	200	W Decay	8.0	✗	Normal Results	
1K	1.0	MSE	AdamH	200	W Decay	8.0	✗	Normal Results	17
1K	-	CE	AdamH	200	W Decay	8.0	✗	**Learn fast!**	18
1K	-	MSE	Adam	200	W Decay	8.0	✓	Lower Acc	19
1K	-	MSE	SGD	200	W Decay	8.0	✓	Slower	20
1K	-	MSE	AdamH	32	W Decay	8.0	✓	Exactly Same	
1K	-	MSE	AdamH	128	W Decay	8.0	✓	Exactly Same	
1K	-	MSE	AdamH	1024	W Decay	8.0	✓	Exactly Same	21
1K	-	MSE	AdamH	200	None	8.0	✓	A little slower	
1K	-	MSE	AdamH	200	W Decay	0.1	✗	Normal Results	

Table 3: Training result with different hyperparameter setup on MNIST.

Building upon these findings, we believe that grokking of MLPs on MNIST dataset is completely different from the transformer's, and it is primarily due to three factors: a scarcity of training samples, suboptimal initialization, and the use of a loss function that leads to slow learning. Our intuition regarding this phenomenon is illustrated in Figure 10.

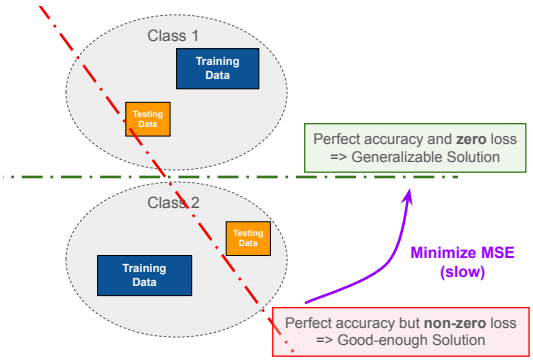


Figure 10: Our intuition about the occurrence of grokking in the MNIST dataset centers on three major factors: the limited amount of training data, poor initialization, and the use of slow-learning functions. These elements collectively contribute to the phenomenon of grokking in this context.

5 Literature Review

5.1 Groking: Generalization beyond overfitting on small algorithmic datasets(3)

Idea. The concept of "Grokking," first proposed in (3), refers to a phenomenon where there is a sudden increase in test accuracy long after the training accuracy has plateaued.

Experiment. In their study, a 2-layer transformer with 40,000 parameters was trained on binary operation tasks, predominantly involving addition, subtraction, division, exponentiation, with the final results modulo 97 to limit the number of possible outcomes.

Result. The findings indicate that Grokking occurs more rapidly when the size of the training dataset is reduced. Additionally, employing weight decay has been found to be the most effective method for accelerating the onset of Grokking.

5.2 Towards understanding grokking: An effective theory of representation learning (2)

Experiment. The study conducted in (1) follows an experimental setup very similar to that of (3). The primary distinction lies in their examination of both regression and classification loss.

Result. The findings reveal that the phenomenon of grokking is chiefly influenced by factors such as the scale of weight decay, the learning rate of the decoder, and the size of the training set. These observations largely concur with those reported in (3). The study categorizes learning outcomes into four distinct stages: Comprehension, Grokking, Memorization, and Confusion. It is demonstrated that Grokking represents a transitional phase between Comprehension and Memorization, influenced variably by weight decay and decoder learning rate. The researchers propose that the ideal learning phase, termed Comprehension, necessitates a balance where representation learning progresses faster than the decoder, but only to a moderate extent.

5.3 Omnigrok: Grokking beyond algorithmic data (1)

Idea. The study in (1) introduces a novel perspective suggesting that grokking is significantly influenced by the position of weight initialization in relation to generalizable solutions. They propose that weights initialized with a large norm are more likely to be in proximity to solutions that lead to overfitting rather than those that are generalizable. Therefore, the transition from overfitting solutions to generalizable ones encapsulates the process of grokking.

Experiment. The initial experiment presented involves a teacher-student setup, employing two identical 4-layer MLP (Multilayer Perceptron) networks. This setup utilizes a regression loss model with 100 training and 100 testing samples, drawn from a Gaussian distribution.

Subsequent experiments focus on more realistic datasets:

- Image classification on the MNIST dataset, using a depth-3, width-200 MLP, trained on a subset of 1,000 images.
- Natural Language Processing tasks on the IMDb reviews dataset, implemented with LSTMs.
- Prediction of isotropic polarizability of molecules on the QM9 dataset, employing a Graph Convolutional Neural Network (GCNN).

Results. In the teacher-student setup, the study illustrates that grokking occurs when the initial weight scale is large, with the phenomenon manifesting more rapidly as the scale of weight decay increases.

Regarding the experiments on realistic datasets, error contours were plotted to examine the interplay between the initial weight norm and the size of the training dataset. It was observed that with a limited training dataset size of 100, the loss contours for training and testing diverge when using the same initial weight scale. Within a specific range of scales, it was noted that while the training contour reaches its optimum, the testing contour does not follow suit, indicating a discrepancy in performance.

References

[1] Z. Liu, E. J. Michaud, and M. Tegmark, "Omnigrok: Grokking beyond algorithmic data," in *The Eleventh International Conference on Learning Representations*, 2022.

[2] Z. Liu, O. Kitouni, N. S. Nolte, E. Michaud, M. Tegmark, and M. Williams, "Towards understanding grokking: An effective theory of representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 34 651–34 663, 2022.

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

[4] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *Advances in neural information processing systems*, vol. 31, 2018.

[5] P. Notsawo Jr, H. Zhou, M. Pezeshki, I. Rish, G. 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.

A Appendix for Algorithmic Dataset Experiements

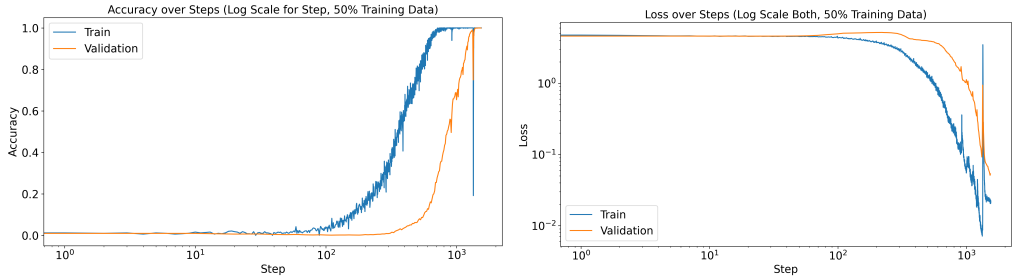


Figure 11: Experiment of grokking on algorithmic dataset (addition operation).

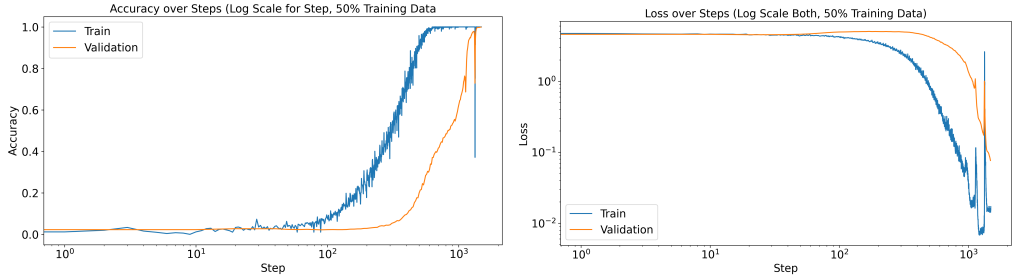


Figure 12: Experiment of grokking on algorithmic dataset (multiplication operation).

B Appendix for MNIST Experiments

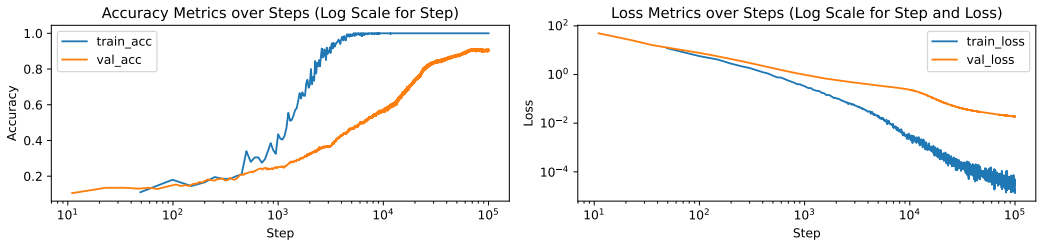


Figure 13: Training 1K to 3K data

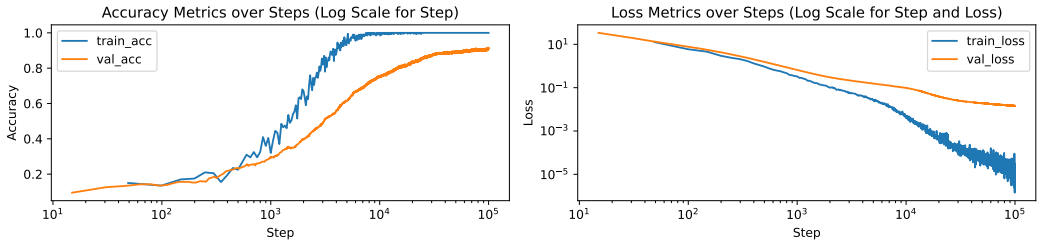


Figure 14: Training 1K to 4K data

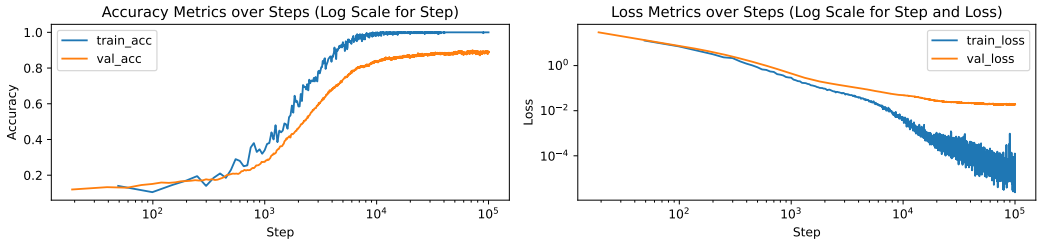


Figure 15: Training 1K to 5K data

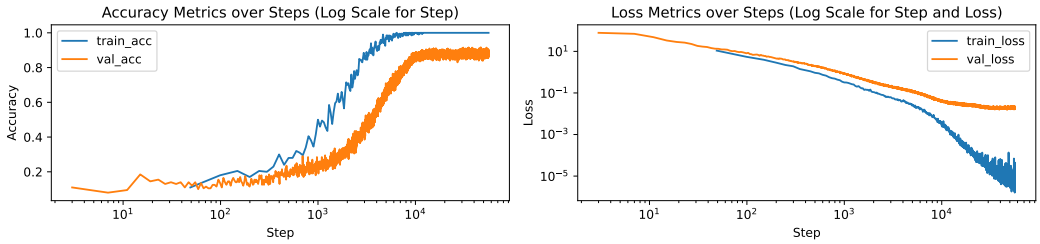


Figure 16: Augmentation of scale 0.2

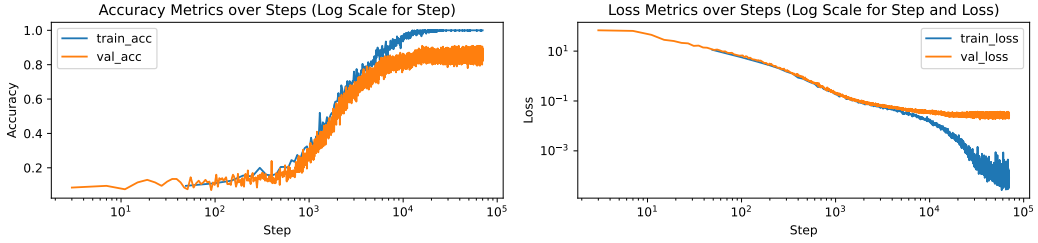


Figure 17: Augmentation of scale 1.0

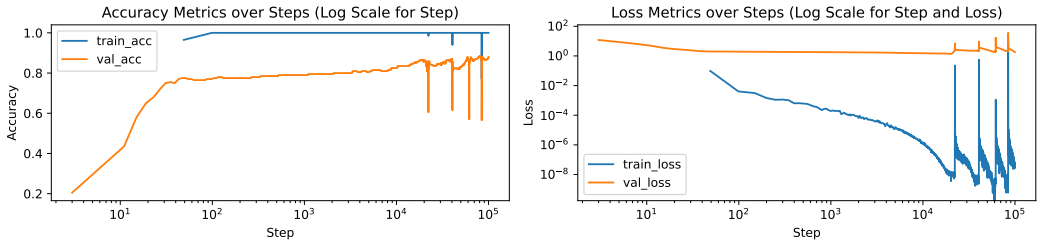


Figure 18: MSE to CE

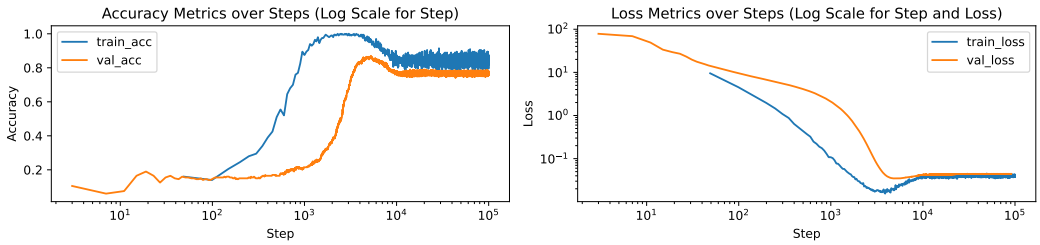


Figure 19: AdamH to Adam

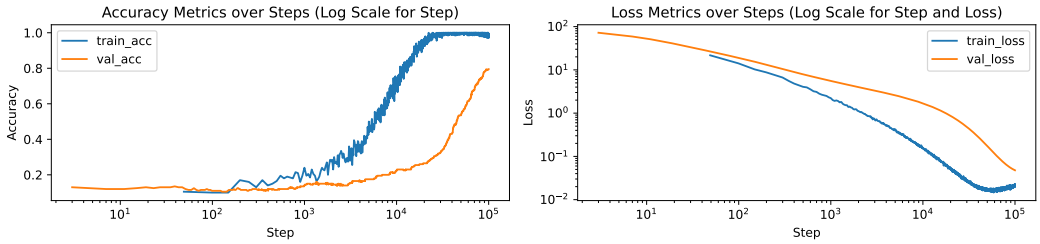


Figure 20: AdamH to SGD

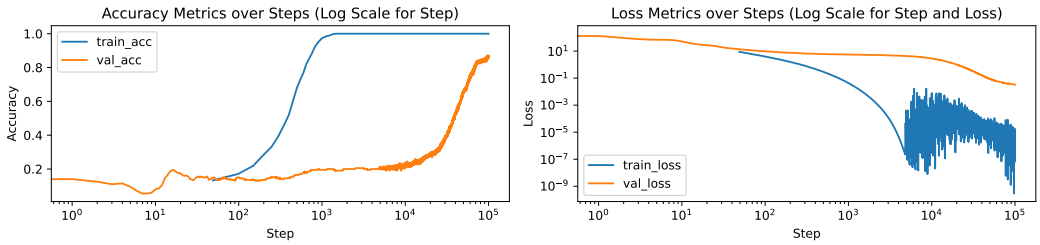


Figure 21: BS 200 to 1024

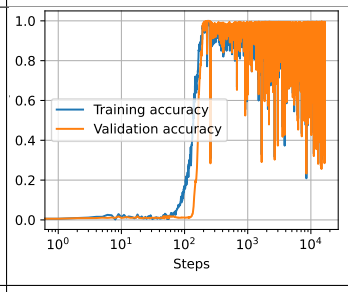
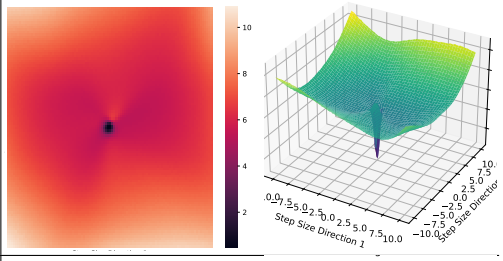
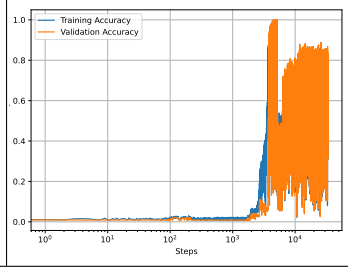
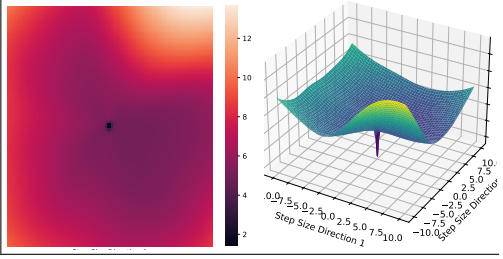
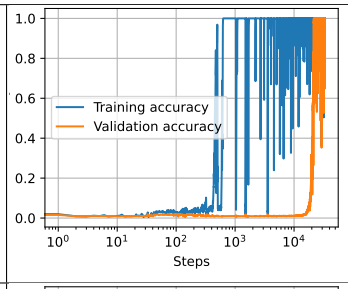
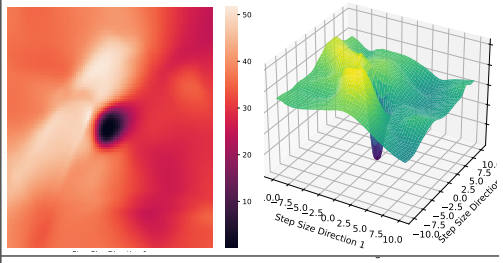
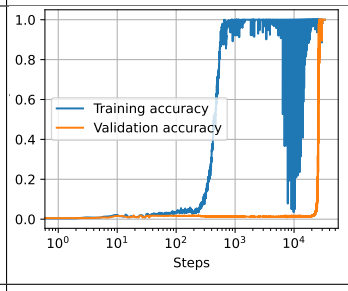
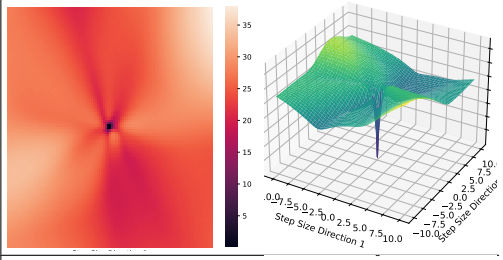
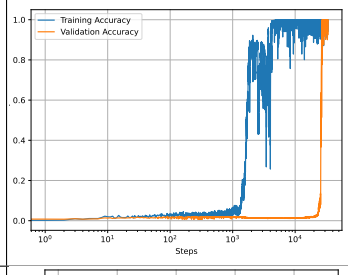
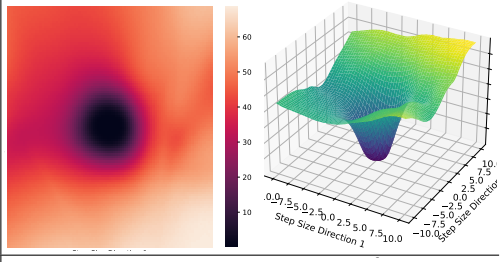
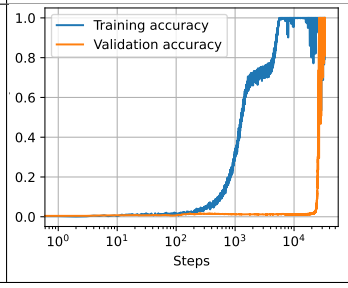
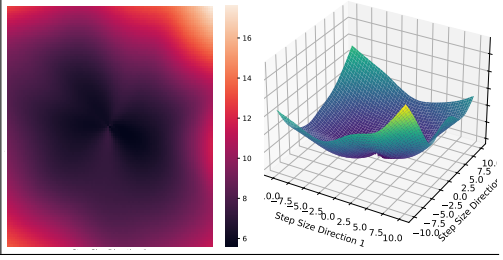
Batch Size	Noise Level	Init Scale	Weight Decay	Accuracies	Loss Contour
512	0.10	100	1		
1024	0.10	0.1	1		
1024	0.01	10	0.01		
1024	0.01	10	0.1		
1024	0.01	100	0.01		
1024	0.01	10	1		

Table 4: Experiment Summary