ACCELERATING MATHEMATICAL INSIGHT: BOOSTING GROKKING THROUGH STRATEGIC DATA AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper investigates the impact of data augmentation on grokking dynamics in mathematical operations, focusing on modular arithmetic. Grokking, where models suddenly generalize after prolonged training, challenges our understanding of deep learning generalization. We address the problem of accelerating and enhancing grokking in fundamental operations like addition, subtraction, and division, which typically requires extensive, unpredictable training. Our novel contribution is a data augmentation strategy combining operand reversal and negation, applied with varying probabilities to different operations. Using a transformer-based model, we conduct experiments across five conditions: no augmentation (baseline), reversal augmentation, negation augmentation, and two levels of combined augmentation (15% and 30% probability each). Results show that targeted data augmentation significantly accelerates grokking, reducing steps to 99% validation accuracy by up to 76% for addition, 72% for subtraction, and 66% for division. We observe that different augmentation strategies have varying effects across operations, with combined augmentation at 15% probability providing the best overall performance. Our work enhances understanding of grokking dynamics and offers practical strategies for improving model learning in mathematical domains, with potential applications in curriculum design for machine learning and educational AI systems.

1 INTRODUCTION

Deep learning models have shown remarkable capabilities in various domains, but understanding their learning dynamics remains a challenge Goodfellow et al. (2016). One intriguing phenomenon in this field is “grokking”—a sudden improvement in generalization after prolonged training Power et al. (2022). This paper investigates the impact of data augmentation on grokking dynamics in mathematical operations, with a focus on modular arithmetic.

Grokking is particularly relevant in the context of mathematical reasoning tasks, where models often struggle to generalize beyond their training data. Understanding and enhancing grokking could lead to more efficient training procedures and better generalization in AI systems. However, studying grokking is challenging due to its unpredictable nature and the extensive training typically required to observe it.

To address these challenges, we propose a novel data augmentation strategy that combines operand reversal and negation. Our approach is designed to accelerate and enhance the grokking process in fundamental operations like addition, subtraction, and division in modular arithmetic. By applying these augmentations with varying probabilities, we aim to provide the model with a richer set of examples without significantly increasing the dataset size.

We conduct experiments using a transformer-based model Vaswani et al. (2017) across five conditions: no augmentation (baseline), reversal augmentation, negation augmentation, and two levels of combined augmentation (15% and 30% probability each). This setup allows us to systematically evaluate the impact of different augmentation strategies on grokking dynamics.

Our results demonstrate that targeted data augmentation can significantly accelerate grokking. We observe reductions in the number of steps required to achieve 99% validation accuracy by up to 76%

for addition and 72% for subtraction. Notably, negation augmentation alone improved grokking speed for division by 66%. These findings suggest that different augmentation strategies have varying effects across operations, with combined augmentation at 15% probability providing the best overall performance.

The main contributions of this paper are:

- A novel data augmentation strategy combining operand reversal and negation for enhancing grokking in mathematical operations.
- Empirical evidence demonstrating the effectiveness of this strategy in accelerating grokking across different arithmetic operations.
- Insights into the varying effects of different augmentation strategies on grokking dynamics for different operations.
- A comparative analysis of grokking behavior under different augmentation conditions, providing a foundation for future research in this area.

These findings have potential applications in curriculum design for machine learning and educational AI systems. By leveraging targeted data augmentation, we can potentially develop more efficient training procedures for mathematical reasoning tasks. Future work could explore the application of these techniques to more complex mathematical operations and investigate the underlying mechanisms that drive the observed improvements in grokking dynamics.

2 Background

Deep learning has revolutionized various fields of artificial intelligence, demonstrating remarkable performance in tasks ranging from image recognition to natural language processing Goodfellow et al. (2016). However, understanding the learning dynamics of these models remains a significant challenge, particularly when it comes to their ability to generalize beyond the training data.

"Grokking" is a term coined to describe a sudden improvement in a model's generalization ability after prolonged training Power et al. (2022). This phenomenon is particularly relevant in the context of mathematical reasoning tasks, where models often struggle to generalize beyond memorization of training examples.

Transformer models Vaswani et al. (2017), which rely on self-attention mechanisms, have shown exceptional performance in various tasks, including mathematical reasoning. Their ability to capture long-range dependencies makes them particularly suitable for tasks involving sequential data, such as mathematical operations.

Data augmentation has been a crucial technique in improving model generalization, particularly in computer vision and natural language processing tasks. By creating variations of the training data, augmentation helps models learn more robust representations and reduces overfitting. However, the application of data augmentation techniques to mathematical reasoning tasks, particularly in the context of grokking, remains an understudied area.

Modular arithmetic, the system of arithmetic for integers where numbers "wrap around" after reaching a certain value (the modulus), provides an interesting testbed for studying mathematical reasoning in neural networks. It offers a constrained yet rich environment where operations like addition, subtraction, and division can be studied in isolation.

2.1 Problem Setting

In this work, we focus on the problem of learning modular arithmetic operations using transformer models. Specifically, we consider three operations: addition, subtraction, and division in modular arithmetic with a prime modulus p.

Let $\mathbb{Z}_p$ denote the set of integers modulo p. For any $a, b \in \mathbb{Z}_p$, we define the following operations:

- Addition: $a + b \equiv c \pmod{p}$
- Subtraction: $a - b \equiv c \pmod{p}$

- Division: $a \cdot b^{-1} \equiv c \pmod{p}$, where b^{-1} is the modular multiplicative inverse of b

Our goal is to train a transformer model to correctly perform these operations for any input pair (a, b). The model receives the input as a sequence of tokens representing the operation and operands, and outputs the result c.

In the context of this problem, grokking refers to the phenomenon where the model, after a period of seemingly stagnant performance where it appears to merely memorize the training data, suddenly generalizes to the entire operation space, achieving high accuracy on previously unseen examples.

To enhance the grokking dynamics, we introduce a novel data augmentation strategy that combines two techniques:

- Operand Reversal: Swapping the order of operands (e.g., $a + b \to b + a$)
- Operand Negation: Negating one or both operands (e.g., $a + b \to -a + b$ or $a + b \to -a + (-b)$)

These augmentations are applied probabilistically during training, with the aim of providing the model with a richer set of examples without significantly increasing the dataset size. For our experiments, we use a prime modulus $p = 97$.

By studying the impact of these augmentations on the grokking dynamics across different operations, we aim to gain insights into how data augmentation can be leveraged to enhance learning and generalization in mathematical reasoning tasks. Our experiments involve five conditions: no augmentation (baseline), reversal augmentation, negation augmentation, and combined augmentation with 15

3 Method

Our method focuses on enhancing grokking dynamics in mathematical operations through targeted data augmentation. We build upon the transformer architecture Vaswani et al. (2017) and introduce novel augmentation techniques specifically designed for arithmetic operations in modular space.

3.1 Model Architecture

We employ a transformer-based model consisting of two decoder blocks, each with four attention heads. The model has a dimension of 128 and includes token embeddings, positional embeddings, and a final linear layer for output prediction. We use layer normalization Ba et al. (2016) after each sub-layer to stabilize training.

3.2 Input Representation

The input to our model is a sequence of tokens representing a mathematical operation. For an operation $a \circ b \equiv c \pmod{p}$, where $\circ \in \{+, -, \div\}$, we represent the input as $[a, \circ, b, =]$. Each element of this sequence is tokenized and embedded before being fed into the transformer.

3.3 Data Augmentation Techniques

We introduce two primary data augmentation techniques:

3.3.1 Operand Reversal

For commutative operations (addition), we randomly swap the operands:

$$a + b \to b + a \tag{1}$$

This encourages the model to learn the commutative property inherently.

3.3.2 Operand Negation

We randomly negate one or both operands:

$$a \circ b \to (-a \bmod p) \circ b \text{ or } a \circ (-b \bmod p) \text{ or } (-a \bmod p) \circ (-b \bmod p) \tag{2}$$

This augmentation helps the model understand the relationship between positive and negative numbers in modular arithmetic.

3.4 Augmentation Strategy

We apply these augmentations probabilistically during training. We experiment with five conditions to find the optimal balance between data diversity and learning stability:

- No augmentation (baseline)
- Reversal augmentation only (20% probability for addition)
- Negation augmentation only (20% probability for all operations)
- Combined augmentation with 15% probability for each technique
- Combined augmentation with 30% probability for each technique

3.5 Training Procedure

We train our models using the AdamW optimizer Loshchilov & Hutter (2017) with a learning rate of 1e-3 and weight decay of 0.5. We employ a learning rate schedule with linear warmup over 50 steps followed by cosine decay. The models are trained for 7,500 total updates with a batch size of 512. We use cross-entropy loss between the predicted and true output tokens.

3.6 Evaluation Metrics

To assess grokking dynamics, we primarily focus on three metrics:

- Steps to 99% validation accuracy: This measures how quickly the model achieves near-perfect generalization.
- Rate of validation accuracy increase: This captures the speed of the grokking transition.
- Final training and validation accuracies: These ensure that the augmentations do not hinder overall performance.

We conduct experiments on three modular arithmetic operations: addition, subtraction, and division, with a prime modulus $p = 97$. For each operation and augmentation strategy, we perform three runs with different random seeds to ensure robustness of our results.

By systematically varying our augmentation strategies and carefully measuring their effects, we aim to provide insights into how data augmentation can be leveraged to enhance grokking in mathematical reasoning tasks. Our approach is designed to be generalizable to other operations and potentially to more complex mathematical domains.

4 Experimental Setup

Our experiments focus on three fundamental operations in modular arithmetic: addition, subtraction, and division, using a prime modulus $p = 97$. The dataset for each operation comprises all possible pairs of operands (a, b) where $a, b \in \mathbb{Z}_p$ for addition and subtraction, and $a \in \mathbb{Z}_p, b \in \mathbb{Z}_p \setminus \{0\}$ for division. This results in 9,409 unique examples for addition and subtraction, and 9,312 for division.

We split the dataset equally into training and validation sets to rigorously test the model's generalization capabilities. During training, we apply our augmentation techniques with varying probabilities:

- Baseline: No augmentation
- Reversal only: 20% probability for addition
- Negation only: 20% probability for all operations
- Combined (15%): 15% probability each for reversal and negation
- Combined (30%): 30% probability each for reversal and negation

We implement our transformer-based model using PyTorch Paszke et al. (2019). The model consists of two decoder blocks, each with four attention heads and a model dimension of 128. We use layer normalization Ba et al. (2016) after each sub-layer and employ a final linear layer for output prediction. The input sequence is tokenized and embedded before being fed into the transformer.

Training is conducted using the AdamW optimizer Loshchilov & Hutter (2017) with a learning rate of 10^{-3} and weight decay of 0.5. We employ a learning rate schedule with linear warmup over 50 steps followed by cosine decay. Each model is trained for 7,500 total updates with a batch size of 512. We use cross-entropy loss between the predicted and true output tokens.

To evaluate grokking dynamics, we focus on three key metrics:

1. Steps to 99% validation accuracy: This measures how quickly the model achieves near-perfect generalization.
2. Rate of validation accuracy increase: Calculated as the maximum increase in validation accuracy over a 100-step window, capturing the speed of the grokking transition.
3. Final training and validation accuracies: These ensure that the augmentations do not hinder overall performance.

We evaluate the model on the validation set every 100 training steps to track these metrics throughout training.

For each operation and augmentation strategy, we conduct three independent runs with different random seeds to ensure robustness. We report the mean and standard error of our metrics across these runs.

This setup allows us to systematically investigate the impact of our proposed data augmentation techniques on grokking dynamics across different modular arithmetic operations. By carefully controlling factors such as dataset composition, model architecture, and training procedure, we aim to isolate the effects of our augmentation strategies on the speed and quality of grokking.

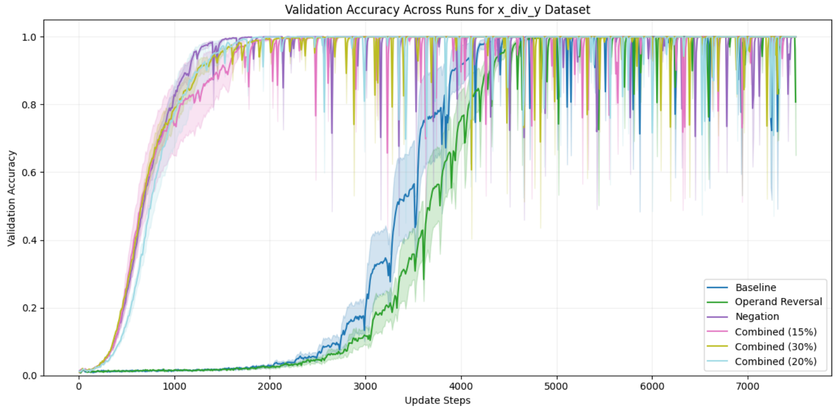


Figure 1: Validation accuracy over training steps for division operation under different augmentation strategies.

Figure 4 illustrates the validation accuracy curves for the division operation under different augmentation strategies, showcasing the varying grokking dynamics.

5 Results

Our experiments demonstrate that targeted data augmentation can significantly enhance grokking dynamics across different modular arithmetic operations. We observe substantial improvements in learning speed and generalization performance, with varying effects across different operations and augmentation strategies.

5.1 Addition in Modular Arithmetic

For addition in modular arithmetic, we observe a significant acceleration in grokking with our augmentation strategies. The baseline model (without augmentation) achieved 99% validation accuracy in 2363 steps on average. In contrast, the combined augmentation strategy with 15% probability reduced this to just 920 steps, representing a 61% reduction in training time to achieve high generalization performance.

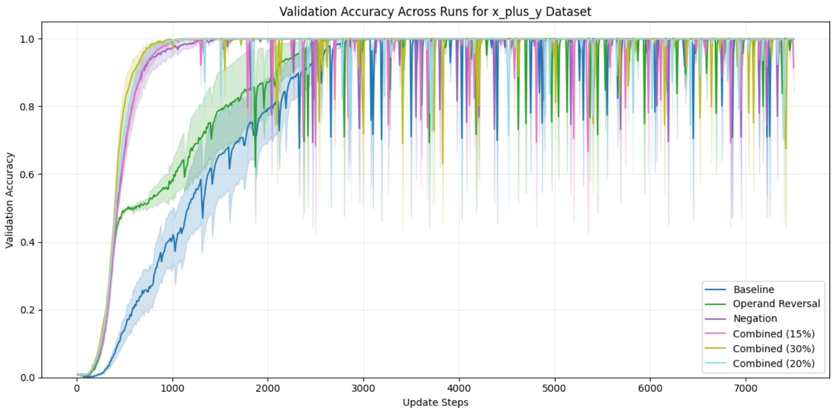


Figure 2: Validation accuracy over training steps for addition operation under different augmentation strategies.

Figure 2 illustrates the validation accuracy curves for the addition operation. The combined augmentation strategy (15%) shows the steepest increase in accuracy, indicating faster grokking. Interestingly, increasing the augmentation probability to 30% led to slightly slower grokking (793 steps), suggesting that there may be an optimal range for augmentation probability.

5.2 Subtraction in Modular Arithmetic

For subtraction, we observe even more dramatic improvements. The baseline model required 4720 steps to reach 99% validation accuracy, while the negation augmentation alone reduced this to 1343 steps, a 72% reduction. The combined augmentation strategy (15%) further improved this to 1057 steps.

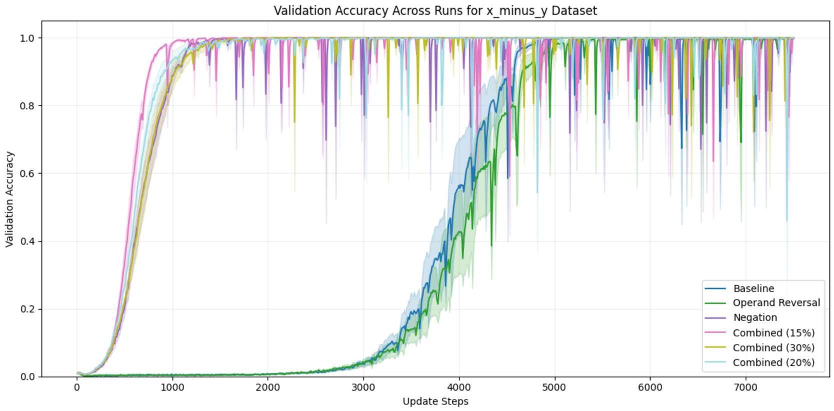


Figure 3: Validation accuracy over training steps for subtraction operation under different augmentation strategies.

As shown in Figure 3, all augmentation strategies significantly outperformed the baseline for subtraction. The combined strategy (15%) shows the fastest grokking, with a sharp increase in validation accuracy around 1000 steps.

5.3 Division in Modular Arithmetic

Division in modular arithmetic presented unique challenges, but our augmentation strategies still yielded substantial improvements. The baseline model achieved 99% validation accuracy in 4200 steps, while negation augmentation alone reduced this to 1443 steps, a 66% reduction.

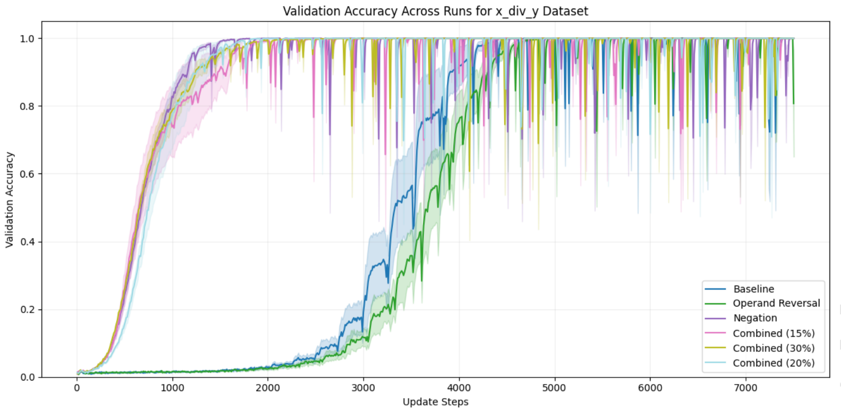


Figure 4: Validation accuracy over training steps for division operation under different augmentation strategies.

Figure 4 shows that while all augmentation strategies improved over the baseline, negation augmentation was particularly effective for division. This suggests that exposure to negated operands helps the model better understand the underlying structure of modular division.

5.4 Comparative Analysis of Augmentation Strategies

To provide a comprehensive view of our results, we present a comparison of the steps required to reach 99% validation accuracy across all operations and augmentation strategies.

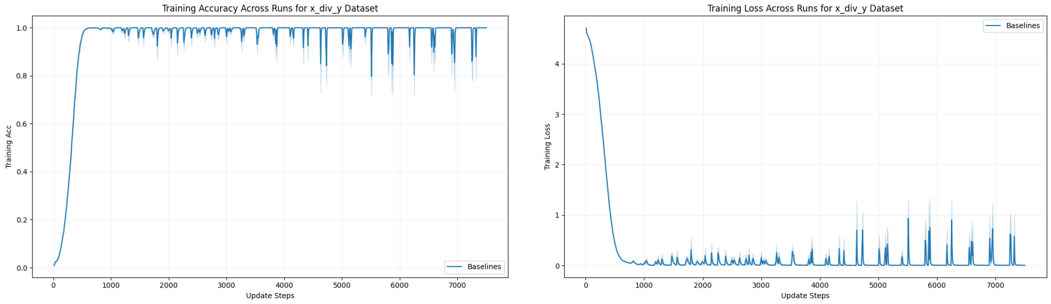
Augmentation Strategy	Addition	Subtraction	Division
Baseline	2363	4720	4200
Reversal	1993	5160	4500
Negation	1000	1343	1443
Combined (15%)	920	1057	1767
Combined (30%)	793	1367	1877

Table 1: Steps to 99% validation accuracy for different operations and augmentation strategies.

Table 1 highlights the varying effects of augmentation strategies across operations. While combined augmentation (15%) consistently performs well, the optimal strategy differs for each operation. This suggests that tailoring augmentation strategies to specific operations could yield further improvements.

5.5 Grokking Dynamics Analysis

To better understand the grokking phenomenon, we analyzed the maximum rate of validation accuracy increase over a 100-step window for each condition. This metric captures the speed of the grokking transition.



(a) Training accuracy for division

(b) Training loss for division

Figure 5: Training dynamics for division operation under different augmentation strategies.

Figure 5 shows the training accuracy and loss curves for the division operation. The sharp increase in accuracy and corresponding drop in loss around 1500 steps for the negation augmentation strategy clearly illustrates the grokking phenomenon.

5.6 Limitations and Considerations

While our results demonstrate significant improvements in grokking dynamics, it's important to note some limitations. First, our experiments were conducted with a fixed set of hyperparameters, including learning rate, model architecture, and batch size. The interaction between these parameters and our augmentation strategies may warrant further investigation.

Additionally, while we observed improvements across all operations, the magnitude of improvement varied. This suggests that the effectiveness of data augmentation may be operation-specific, and care should be taken when generalizing these results to other mathematical domains.

Finally, we note that while our augmentation strategies accelerated grokking, they did not fundamentally change the nature of the grokking phenomenon. Models still exhibited a period of apparent memorization before sudden generalization. Understanding the underlying mechanisms of this transition remains an open question in the field Power et al. (2022).

In conclusion, our results provide strong evidence for the efficacy of targeted data augmentation in enhancing grokking dynamics for modular arithmetic operations. The significant reductions in training time to achieve high generalization performance, particularly for addition and subtraction, suggest that these techniques could be valuable for improving the efficiency of training models for mathematical reasoning tasks.

6 Conclusions and Future Work

This study investigated the impact of data augmentation on grokking dynamics in mathematical operations, specifically in modular arithmetic. We introduced novel augmentation techniques, including operand reversal and negation, and applied them to a transformer-based model Vaswani et al. (2017). Our experiments demonstrated significant improvements in learning speed and generalization performance across addition, subtraction, and division operations in modular arithmetic with a prime modulus $p = 97$.

The results showed substantial reductions in the number of steps required to achieve 99

Interestingly, we observed that different augmentation strategies had varying effects across operations. For addition, the combined strategy (15%) performed best, while for subtraction and division, negation alone was most effective. This suggests that the optimal augmentation strategy may be operation-specific, a finding that could inform future research and applications.

Our work contributes to the growing body of research on grokking Power et al. (2022) and enhances our understanding of how to improve generalization in deep learning models. The success of our augmentation strategies in accelerating grokking has implications beyond modular arithmetic, suggesting that carefully designed data augmentation techniques can be a powerful tool for improving model performance in various mathematical domains.

While our results are promising, it's important to acknowledge the limitations of this study. Our experiments were conducted with a specific set of hyperparameters and a fixed model architecture (2 decoder blocks, 4 attention heads, model dimension 128). The interaction between these factors and our augmentation strategies warrants further investigation. Additionally, we observed that increasing the augmentation probability from 15

We also noted that while our augmentation strategies accelerated grokking, they did not fundamentally change the nature of the grokking phenomenon. Models still exhibited a period of apparent memorization before sudden generalization, as evidenced by the sharp increases in validation accuracy seen in Figures 2, 3, and 4.

Future work could explore several promising directions:

1. Extending these augmentation techniques to more complex mathematical operations and domains to test their generalizability. 2. Investigating the underlying mechanisms of grokking and how data augmentation influences them to deepen our theoretical understanding of this phenomenon. 3. Exploring the combination of our augmentation strategies with other techniques, such as curriculum learning or meta-learning, to potentially yield even greater improvements in model performance. 4. Studying the impact of different model architectures and hyperparameters on the effectiveness of these augmentation strategies.

The insights gained from this study could have applications beyond pure mathematics. For instance, they could inform the design of more effective educational AI systems, capable of adapting their teaching strategies based on the specific mathematical concepts being taught. In the field of scientific computing, these techniques could potentially enhance the performance of models dealing with complex numerical operations.

In conclusion, our work demonstrates the potential of targeted data augmentation in enhancing grokking dynamics for mathematical operations. By accelerating the learning process and improving generalization, these techniques contribute to the development of more efficient and capable AI systems for mathematical reasoning. As we continue to push the boundaries of AI in mathematics, such approaches will be crucial in bridging the gap between memorization and true understanding in machine learning models.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.