MISALIGNMENT OF LOW-LOSS REGIONS CAUSES GROKKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Grokking refers to the delayed emergence of validation-set generalization after a model has already overfit the training set. Although first observed in small algorithmic tasks trained with transformers, its underlying mechanism remains unsettled. In this work, we develop an analysis framework based on mode connectivity and the geometry of low-loss regions. The framework predicts that the standard modular-arithmetic setting does not always produce grokking: under a symmetry-preserving train/validation split, we observe a stable anti-grokking case in which validation performance does not recover. This counterexample challenges several existing correlational explanations of grokking. More broadly, our analysis framework and results further suggest that grokking arises when the low-loss regions induced by the training and validation partitions are misaligned. Once these regions become well aligned, training hyperparameters alone cannot produce grokking and the observed dynamics collapse to either trainable or non-trainable behavior.

1 INTRODUCTION

Grokking is the phenomenon in which a neural network generalizes to a validation set only after a long period of apparent overfitting to the training set. It was first reported by OpenAI when training a two-layer transformer on modular arithmetic tasks (Power et al., 2022). This behavior contrasts with the usual expectation that training and validation accuracy improve on comparable timescales, and it has been discussed as a form of emergent generalization in which a model eventually discovers the structure underlying the data (Nanda et al., 2023; Wei et al., 2022).

Prior work has approached grokking from several perspectives, including Fourier circuit (Nanda et al., 2023), representation learning (Liu et al., 2022), numerical stability (Prieto et al., 2025), Hessian geometry (Zhang et al., 2024), weight decay (Xu et al., 2026), the Slingshot effect (Thilak et al., 2022), and information leakage (Carvalho et al., 2025). These studies reveal useful patterns related to grokking, but they do not provide a full causal account of why grokking occurs.

In this paper, we develop a geometric framework for analyzing grokking, based on empirical studies of Mode Connectivity (MC) (Draxler et al., 2018; Tian et al., 2025). Singular Learning Theory (SLT) suggests that Low-Loss Regions (LLRs)[1] in neural-network parameter spaces are better understood as extended valleys than as isolated basins (Li et al., 2018; Ainsworth et al., 2023; Lion et al., 2024). MC provides empirical tools for identifying low-loss paths between independently trained minima.

Our analysis framework predicts that the OpenAI modular-arithmetic setting does not necessarily produce grokking. We validate this prediction by constructing an anti-grokking split that reliably prevents delayed validation improvement while leaving the model architecture, optimizer, dataset size, and task family unchanged. We then use this case as a counterexample for evaluating whether existing explanations can predict the absence of grokking. Several of them fail this test, suggesting that they capture correlates of grokking rather than its underlying cause.

Our analysis suggests that grokking requires two ingredients. First, the low-loss regions defined by the training and validation partitions must be sufficiently misaligned. Second, the training trajectory must enter the overlap of these regions in a delayed manner, a condition shaped by hyperparameters such as learning rate and weight decay. We regard the first ingredient as the root cause because

[1] The sets of model weights whose loss is below a specified threshold.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

grokking does not occur when these low-loss regions are well aligned, whereas the second ingredient controls how training moves through the low-loss regions. In practical terms, low-loss regions are easier to misalign when the information carried by the training partition is small relative to the expressive capacity of the model.

This paper makes the following contributions:

1. We provide a geometric analysis framework that accounts for several proposed causes of grokking through the misalignment of low-loss regions.
2. Based on this framework, we identify counterexamples within the OpenAI grokking setup that do not exhibit grokking. These anti-grokking cases contradict several phenomena previously proposed as reliable indicators of grokking.
3. Based on this framework, we show that low-loss regions shrink and become more aligned as the information carried by the training partition increases. When the low-loss regions are fully aligned, grokking cannot occur.

The paper is organized as follows. Section 2 introduces the terminology and reviews prior work on grokking and mode connectivity. Section 3 presents our low-loss-region framework and the anti-grokking example, showing that grokking does not always occur in the OpenAI grokking setup. Section 4 studies the alignment of low-loss regions. Section 5 discusses the implications of these results, and Section 6 concludes the paper.

2 Terminology and Existing Work

This section introduces the terminology used throughout the paper, reviews the OpenAI modular-arithmetic setup that we reuse in our experiments, summarizes several influential grokking explanations, and then describes mode connectivity, which is the empirical starting point for our framework.

2.1 Common Terminology

Consider a model architecture $\mathcal{M}$ with D trainable parameters, a fixed dataset $\mathcal{D}$, and a loss function $\mathcal{L}$. Each parameter vector $P \in \mathbb{R}^D$ induces a loss value on $\mathcal{D}$. For a threshold L_{th}, we define the corresponding low-loss region $S_{L \leq L_{\text{th}}}$ as

$$S_{L \leq L_{\text{th}}} \coloneqq \{P \in \mathbb{R}^D \mid \mathcal{L}(P, \mathcal{D}) \leq L_{\text{th}}\} \tag{1}$$

This region is determined by the model architecture $\mathcal{M}$, dataset $\mathcal{D}$, loss function $\mathcal{L}$, and threshold L_{th}. It is independent from the optimizer and training hyperparameters. In principle, evaluating the loss at every point in parameter space would determine the boundary of this region. We therefore write Equation 1 as the following mapping:

$$S_{L \leq L_{\text{th}}} = \text{LLR}(\mathcal{M}, \mathcal{D}, \mathcal{L}, L_{\text{th}}) \tag{2}$$

where LLR stands for Low-Loss Region. Since the loss function is coupled with the architecture and the threshold is adjusted to select a contour level, we use the concise notation $S_{L \leq L_{\text{th}}} = \text{LLR}(\mathcal{M}, \mathcal{D})$ when these choices are clear from context.

We use the following notation for the overlap of the low-loss regions induced by two datasets[2]:

$$\text{LLR}(\mathcal{M}, \mathcal{D}_1 + \mathcal{D}_2) \coloneqq \{P \in \mathbb{R}^D \mid P \in \text{LLR}(\mathcal{M}, \mathcal{D}_1) \wedge P \in \text{LLR}(\mathcal{M}, \mathcal{D}_2)\}$$

Similarly, we define the part of $\text{LLR}(\mathcal{M}, \mathcal{D}_1)$ that does not overlap with $\text{LLR}(\mathcal{M}, \mathcal{D}_2)$ as

$$\text{LLR}(\mathcal{M}, \mathcal{D}_1 + !\mathcal{D}_2) \coloneqq \{P \in \mathbb{R}^D \mid P \in \text{LLR}(\mathcal{M}, \mathcal{D}_1) \wedge P \notin \text{LLR}(\mathcal{M}, \mathcal{D}_2)\}$$

2.2 Set Up Grokking Experiments

Grokking was first observed when training a simple transformer on a modular arithmetic dataset (Power et al., 2022). We reuse this experimental setup and denote the corresponding model architecture by $\mathcal{M}_{grok}$. The full architecture and training recipe are provided in Appendix A.

[2]The threshold L_{th} is omitted for readability.

The modular arithmetic dataset maps integer pairs (x, y) to $f(x, y) \bmod m$, where m is the modulus and $x, y \in \{0, \ldots, m-1\}$. We write this dataset as $\mathcal{D}_{mod}(f(x, y), m)$. Modular addition is the most commonly studied function in grokking research (Nanda et al., 2023; Thilak et al., 2022). We illustrate $\mathcal{D}_{mod}(x + y, 10)$ in Figure 1(a).

The original OpenAI setup uses $\mathcal{D}_{mod}(x + y, 97)$ with randomly 50% of the samples assigned to training and the remainder assigned to validation (Power et al., 2022). We follow this setup and reproduce grokking[3], see Figure 1(b).

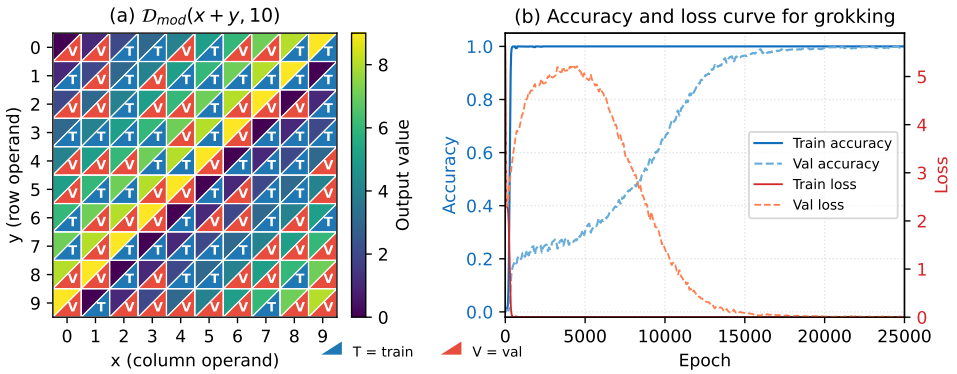


Figure 1: Two-panel illustration of the grokking setup. (a) Illustration of the modular dataset $\mathcal{D}_{mod}(x + y, 10)$. The x-axis and y-axis represent the values of x and y for the mapping function $f(x, y) = x + y$. Each square indexed by (x, y) contains two triangles: the upper-left triangle gives the value of $f(x, y)$, and the lower-right triangle indicates whether the sample belongs to the training or validation partition. This small example is shown only for illustration purposes, and our experiments use the larger dataset $\mathcal{D}_{mod}(x+y, 97)$. (b) Illustration of grokking. Validation accuracy increases only long after the model has overfit the training partition. The result is reproduced with $\mathcal{M}_{grok}$ on $\mathcal{D}_{mod}(x + y, 97)$.

2.3 Existing Work on Grokking

Prior work has proposed several mechanisms and empirical indicators associated with grokking. We summarize the main indicator claims most relevant to our analysis below.

1. **Slingshot effect** Grokking has been reported to occur near the onset of the Slingshot effect and to be absent without it (Thilak et al., 2022). The Slingshot Effect refers to periodic spikes in training loss during optimization. Figure 2(a) visualizes this effect for the experiment in Figure 1(b).
2. **Fourier circuit** In transformers trained on $\mathcal{D}_{mod}(x + y, m)$, grokking has been linked to the formation of a Fourier-based generalizing circuit (Nanda et al., 2023). This circuit has also been proposed as a progress measure for grokking. Figure 2(b) visualizes the corresponding Fourier components.
3. **Training partition threshold** Several studies argue that the training partition must exceed a critical size before a model can learn a generalizing representation and exhibit grokking (Varma et al., 2023; Liu et al., 2022; Zhu et al., 2024; Abramov et al., 2025).
4. **Grokking in MNIST/CIFAR** Grokking is not limited to transformers or modular arithmetic. It has also been observed when training Vision Transformers (ViTs) on CIFAR and MLPs on MNIST (Liu et al., 2022; Deng, 2012; Krizhevsky, 2012; Thilak et al., 2022; Dosovitskiy et al., 2021).

2.4 Existing Work on Mode Connectivity

Mode connectivity is the empirical observation that two independently trained models can be connected by a low-loss path (Draxler et al., 2018). This property has been observed for ResNet (He

[3]The full training recipe is provided in Appendix A.

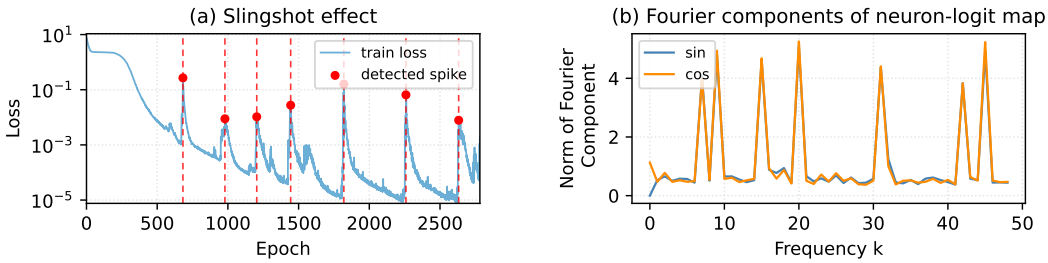


Figure 2: Two-panel illustration of grokking indicators. (a) Slingshot effect: red dots mark spike locations on the training-loss curve. The curve is taken from the experiment in Figure 1(b) and plotted on a log scale to make the spikes visible. Prior work reports that such spikes are positively correlated with grokking (Thilak et al., 2022). (b) Fourier components of the neuron-logit map for a well-trained grokked model. The model is obtained at the end of the experiment in Figure 1(b). Prior work suggests that the formation of this Fourier circuit can be used as a progress measure for grokking (Nanda et al., 2023).

et al., 2015), VGG (Simonyan & Zisserman, 2015), and DenseNet (Huang et al., 2018) models trained on CIFAR-10. More recently, the Low-Loss Path Finding (LLPF) method extended this observation to transformer architectures (Tian et al., 2025).

We empirically validate mode connectivity for $\mathcal{M}_{grok}$ on $\mathcal{D}_{mod}(x+y, 97)$ using LLPF. Configuration details and results are provided in Appendix B. For two independently trained $\mathcal{M}_{grok}$ models, the maximum loss along the discovered path remains below 1.5×10^{-3} and the minimum accuracy remains above 99.85%. Repeating this experiment ten times yields only small standard deviation across runs. These empirical results motivate Conjecture 1.

Conjecture 1. *The low-loss region LLR*$(\mathcal{M}_{grok}, \mathcal{D}_{mod}(x+y, 97), \mathcal{L}, 1.5 \times 10^{-3})$ *is a single path-connected region.*

This conjecture is consistent with the LLPF interpretation that repeated success across different pairs of modes trained on random seeds provides evidence for a path-connected low-loss region. The "path" should be understood as evidence of a higher-dimensional connected manifold rather than a literal one-dimensional structure: LLPF identifies one-dimensional curves, but the underlying connected set is expected to have dimension greater than one (Benton et al., 2021).

There is also theoretical support for this perspective, please see Appendix L. We also clarify that MC should not be conflated with linear mode connectivity, see Appendix M.

3 Analysis Framework and Anti-Grokking Example

This section introduces our analysis framework and uses it to identify a corner case in the OpenAI modular-arithmetic setup where grokking does not occur. We call this behavior *anti-grokking*. We then use the anti-grokking case to test whether several existing grokking indicators can predict the absence of grokking.

3.1 Analysis Framework

Consider a dataset $\mathcal{D}$ split into a training partition $\mathcal{D}_{train}$ and a validation partition $\mathcal{D}_{val}$. These two partitions induce two low-loss regions, LLR$(\mathcal{M}, \mathcal{D}_{train})$ and LLR$(\mathcal{M}, \mathcal{D}_{val})$. Because $\mathcal{D}_{train} \neq \mathcal{D}_{val}$, the corresponding low-loss regions are not completely identical.

In the grokking experiment shown in Figure 1(b), the training partition is randomly sampled from $\mathcal{D}_{mod}(x+y, 97)$ and contains 50% of the whole dataset. We therefore expect the two induced low-loss regions to have a comparable volume: $V(\text{LLR}(\mathcal{M}, \mathcal{D}_{train})) \approx V(\text{LLR}(\mathcal{M}, \mathcal{D}_{val}))$.

When the two regions are comparable in volume but not identical, some weight vectors P may lie in LLR$(\mathcal{M}, \mathcal{D}_{train})$ without lying in LLR$(\mathcal{M}, \mathcal{D}_{val})$. We call this set the *non-overlap low-loss region*

and write the non-empty condition as

$$\text{LLR}(\mathcal{M}, \mathcal{D}_{train} + !\mathcal{D}_{val}) \neq \varnothing. \tag{3}$$

The same reasoning applies with $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ interchanged.

The second question is whether the overlap between the two low-loss regions is non-empty:

$$\text{LLR}(\mathcal{M}, \mathcal{D}_{train} + \mathcal{D}_{val}) \neq ?\varnothing. \tag{4}$$

For $\mathcal{M}_{grok}$ and $\mathcal{D}_{mod}(x + y, 97)$, this overlap is not empty because Figure 1(b) shows that some parameter vectors achieve low loss on both training and validation partitions. In other settings, however, the overlap may be empty, for example when the architecture is inappropriate for the task or when the labels are randomly generated.

Combining Conjecture 1 with Equations 3 and 4 yields the geometric illustration in Figure 3. This illustration is schematic and does not encode dimensionality. For example, point A should not be interpreted as a zero-dimensional point in $\mathbb{R}^D$, and curve $\overset{\frown}{AB}$ should not be interpreted as a literal one-dimensional curve. Likewise, the figure does not imply that exactly one $\text{LLR}(\mathcal{D}_{train})$ path intersects exactly one $\text{LLR}(\mathcal{D}_{val})$ path. Several high-dimensional branches may intersect in the overlap region.

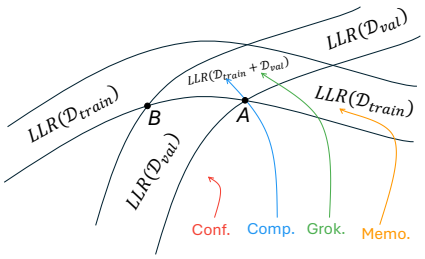


Figure 3: Geometric illustration of $\text{LLR}(\mathcal{M}_{grok}, \mathcal{D}_{mod}(x + y, 97))$ in the weight space. Motivated by Conjecture 1, we draw connected paths representing $\text{LLR}(\mathcal{D}_{train})$ and $\text{LLR}(\mathcal{D}_{val})$. Equations 3 and 4 indicate that both overlap and non-overlap regions exist for the grokking setup, so the two low-loss regions are not aligned. The four learning trajectories (Comprehension, Grokking, Memorization, and Confusion) correspond to the learning phases identified in (Liu et al., 2022).

Existing grokking work identifies four learning phases (Liu et al., 2022):

- Comprehension: accuracy on $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ increases at approximately the same time.
- Grokking: validation accuracy increases much later than training accuracy.
- Memorization: only training accuracy increases.
- Confusion: neither training nor validation accuracy increases.

These trajectories are also shown in Figure 3. Because the illustration is dimension-free, it should not be read as implying that comprehension is less likely than other phases merely because the drawn trajectory appears to pass through a point.

The grokking trajectory in Figure 3 has two stages: (1) entering $\text{LLR}(\mathcal{D}_{train})$ and (2) moving from $\text{LLR}(\mathcal{D}_{train})$ into $\text{LLR}(\mathcal{D}_{train} + \mathcal{D}_{val})$. We find that the second stage lacks an explanation, because the first stage is the ordinary outcome of minimizing training loss. Since low-loss regions are determined by $\mathcal{M}$ and $\mathcal{D}$, modifying the train/validation split might be able to alter whether the model can move from the training-only low-loss region into the overlap region.

3.2 Anti-Grokking Example

Based on our analysis framework and prior discussion, we find one way to prevent the second stage using the notion of surprisal from Shannon information theory (Shannon, 1948). Each sample in a binary modular arithmetic dataset maps two variables a and b to an output c. For $\mathcal{D}_{mod}(x + y, 97)$, the function $f(x, y) = x + y$ satisfies the symmetry $f(a, b) = f(b, a)$. We treat this symmetry

as task ***knowledge***. If the fact $f(a, b) = c$ is known together with this symmetry, then observing $f(b, a) = c$ should have lower surprisal than it would without the symmetry. This suggests that a train/validation split can preserve the same training ratio while reducing the effective surprisal of the training partition. Empirically, we find such a corner case: grokking is eliminated by enforcing this symmetry in the partition. We call this the *anti-grokking* example.

We randomly generate a train/validation partition for $\mathcal{D}_{mod}(x + y, 97)$ such that paired samples $f(a, b)$ and $f(b, a)$ always belong to the same partition, as shown in Figure 4(a). Training $\mathcal{M}_{grok}$ with the same hyperparameters on this anti-grokking split does not produce grokking, as shown in Figure 4(b). The only change relative to Figure 1(b) is the train/validation split.

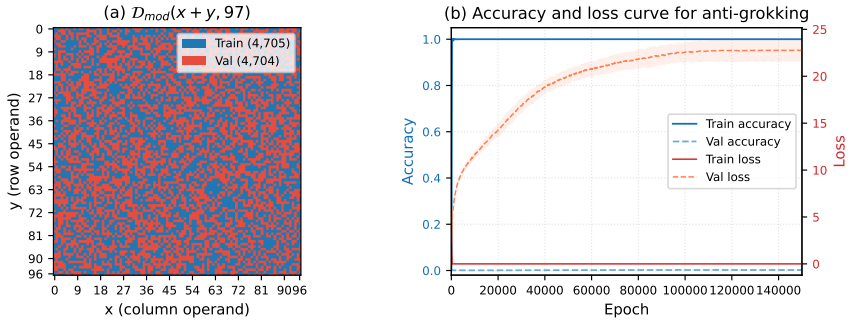


Figure 4: Anti-grokking setup. (a) Dataset partition for the anti-grokking example on $\mathcal{D}_{mod}(x + y, 97)$. Unlike a random partition, samples located at (a, b) and (b, a) always belong to the same partition. The numbers 4705 and 4704 indicate the sample counts in the two partitions. (b) Training $\mathcal{M}_{grok}$ with the same hyperparameters on the split shown in Figure 4(a) does not produce grokking: validation accuracy remains low, and validation loss does not decrease. We repeat this experiment with ten random seeds and different splits. The shaded region indicates the standard deviation across runs.

Section 2 summarized three indicators often associated with grokking: (1) the Slingshot effect, (2) the Fourier-based generalization circuit, and (3) a critical training-set size. We revisit these indicators in the anti-grokking example to test whether they faithfully predict the absence of grokking.

In the anti-grokking example, Slingshot spikes are still observed, see Figure 5(a). Thus the Slingshot effect alone is not a faithful predictor, because it would still suggest that grokking should occur. The Fourier component plot in Figure 5(b) also shows that some frequencies have larger components than others, although the concentration is much weaker than in Figure 2(b). However, using this signal as a progress measure is misleading in this case: Figure 4(b) shows no progress toward grokking, whereas the Fourier components suggest slight progress.

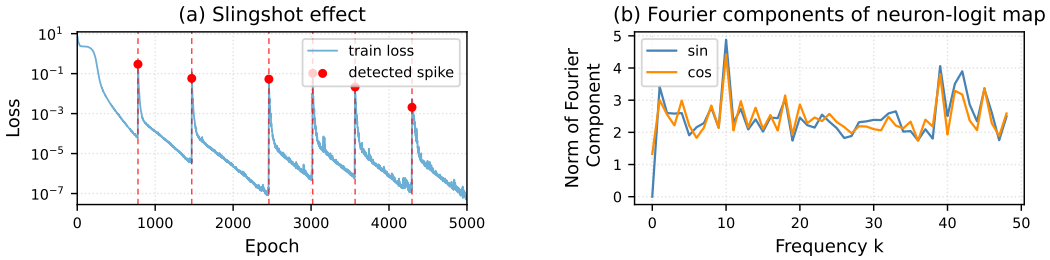


Figure 5: Anti-grokking diagnostics. (a) A Slingshot pattern similar to Figure 2(a) is observed in the anti-grokking example. Thus, the Slingshot effect is not by itself a faithful explanation of grokking. (b) Compared with Figure 2(b), several frequencies still have larger components than others, but the concentration is much weaker than in the grokking case.

The claim that the training set must exceed a size threshold is also incomplete, because the anti-grokking example keeps the training-set size fixed. A more precise statement is that ***grokking requires the effective information entropy carried by the training partition to exceed a architecture-***

dependent threshold. This effective information entropy is not determined solely by the raw number of samples: it should account for task knowledge that makes some samples informative about others. Appendix H.1 operationalizes this idea for the symmetry $f(a, b) = f(b, a)$ and shows that symmetry-aware training coverage provides a useful empirical proxy for effective information entropy. Prior work has used entropy-based notions for datasets (Shi et al., 2025; Schwalbe-Koda et al., 2025; Zhang et al., 2020), but has not explicitly incorporated task knowledge in this way. A formal entropy measure that incorporates task knowledge is left for future work.

Finally, the requirement that samples at (a, b) and (b, a) appear in the same partition is not an all-or-nothing condition. In the anti-grokking split, 100% of such pairs share a partition. As this ratio is reduced toward 70%, grokking gradually reappears, see Appendix H. Anti-grokking is rarely observed under the original random partitioning policy because the same-partition ratio concentrates around 50%.

4 Alignment of Low-Loss Regions

This section analyzes how low-loss regions change as the dataset becomes larger. We first argue that adding data imposes additional constraints and therefore tends to shrink low-loss regions. We then empirically show that, as these regions shrink, the train and validation low-loss regions become increasingly aligned. Under such alignment, grokking disappears and training dynamics collapse to either trainable or non-trainable.

4.1 Shrinking of Low-Loss Region

The volume of a low-loss region should decrease, or at least not increase, as the dataset grows. When one dataset is a subset of another, this monotonic relation can be written as

$$\mathrm{LLR}(\mathcal{M}, \mathcal{D}_1) \subseteq \mathrm{LLR}(\mathcal{M}, \mathcal{D}_2) \quad \text{if } \mathcal{D}_2 \subseteq \mathcal{D}_1 \tag{5}$$

Equation 5 follows from interpreting each data point as a constraint on the parameter vector. If $\mathcal{D}_1$ contains every sample in $\mathcal{D}_2$ plus additional samples, then any parameter vector that has low loss on $\mathcal{D}_1$ also satisfies the constraints imposed by $\mathcal{D}_2$. This intuition is consistent with empirical work showing that models typically incur higher loss on additional out-of-distribution data than on in-distribution data (Zhang et al., 2017; Salman & Liu, 2019). However, with average loss, this relationship should be understood as a geometric heuristic rather than a strict theorem, because errors on different samples can compensate for one another.

We slightly modify the modular-dataset construction so that it satisfies the nested-dataset condition in Equation 5. We first generate a single master dataset $\mathcal{D}_{\mathrm{mod1997}}(x^y, 1997)$, and use the same tokenizer for all derived datasets. For each $n \leq 1997$, we then define $\mathcal{D}_{\mathrm{mod1997}}(f, n)$ by retaining only those samples whose operands satisfy $x, y \in \{0, \ldots, n-1\}$. The targets remain $f(x, y) \bmod 1997$ for every n; thus, n indexes the operand-domain size rather than the modulus. Consequently, for any $m < n$, $\mathcal{D}_{\mathrm{mod1997}}(f, m) \subseteq \mathcal{D}_{\mathrm{mod1997}}(f, n)$. Increasing n therefore adds labeled examples and imposes additional constraints. Accordingly, Equation 5 suggests that the volume $V(\mathrm{LLR}(\mathcal{M}_{\mathrm{grok}}, \mathcal{D}_{\mathrm{mod1997}}(f, n)))$ should be non-increasing as n grows.

4.2 Alignment of Low-Loss Regions

The grokking setup has two low-loss regions, $\mathrm{LLR}(\mathcal{M}_{grok}, \mathcal{D}_{\mathrm{mod1997,train}}(f, n))$ and $\mathrm{LLR}(\mathcal{M}_{grok}, \mathcal{D}_{\mathrm{mod1997,val}}(f, n))$. As n increases, both regions shrink. The key question is how the overlap region, $\mathrm{LLR}(\mathcal{M}, \mathcal{D}_{train} + \mathcal{D}_{val})$, and the non-overlap region, $\mathrm{LLR}(\mathcal{M}, \mathcal{D}_{train} + !\mathcal{D}_{val})$, change during this shrinkage.

Our analysis is that the overlap region should shrink but would never become an empty set as long as the task remains learnable. Otherwise, no model could fit both the training and validation partitions. By contrast, the non-overlap region should decrease as the two low-loss regions shrink. This process is illustrated in Figure 6.

To test this analysis, we design an empirical experiment that indirectly probes the overlap region, as shown in Figure 7. The procedure is inspired by prior work studying the effect of training hyperparameters on grokking (Liu et al., 2022). Specifically, we sweep learning rate and weight decay, train

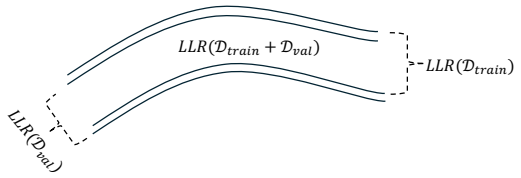


Figure 6: Low-loss regions defined by the training and validation partitions become increasingly aligned as partition size increases. The overlap region is illustrated as a connected path because mode connectivity is empirically verified to persist for $\mathrm{LLR}(\mathcal{M}_{grok}, \mathcal{D}_{train} + \mathcal{D}_{val})$ in Appendix B.

on $\mathcal{D}_{\mathrm{mod}1997}(x^y, n)$, and classify the resulting learning phase for each hyperparameter pair[4]. We then count how many cells belong to each learning phase as n increases. A decrease in Memorization cells together with an increase in Comprehension cells suggests that fewer trajectories terminate in the training-only low-loss region and more trajectories directly reach the overlap of $\mathrm{LLR}(\mathcal{D}_{train})$ and $\mathrm{LLR}(\mathcal{D}_{val})$. A complete explanation of why phase-cell counts provide an indirect probe of low-loss-region alignment is given in Appendix D.

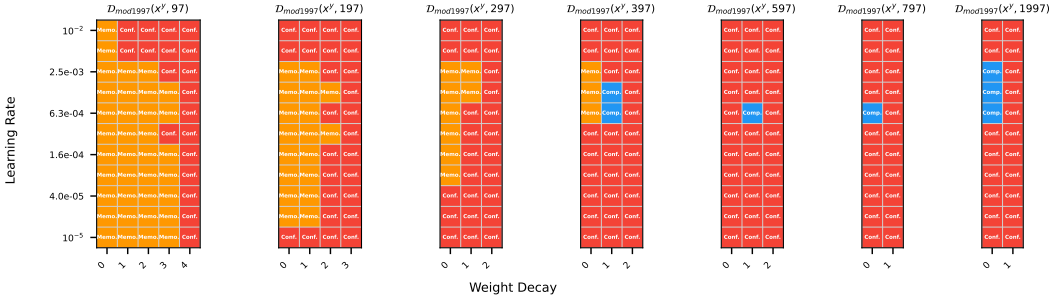


Figure 7: Phase diagrams for $\mathcal{D}_{\mathrm{mod}1997}(x^y, n)$ as the operand-domain size n increases. Each cell represents one learning-rate/weight-decay pair and is classified as Comprehension, Grokking, Memorization, or Confusion according to the criteria in Appendix C. As n increases from 97 to 1997, the number of Memo. cells decreases and eventually vanishes, while Comp. cells emerge and become more frequent. This shift indicates that fewer training trajectories remain in the training-only low-loss region and more directly reach the overlap of the training and validation low-loss regions, suggesting that the two regions become increasingly aligned. No Grokking cells are observed, consistent with the anti-grokking analysis that the effective information carried by the training partition remains below the threshold required for grokking. Appendix D explains why phase-cell counts provide an indirect probe of alignment.

We perform these experiments on $\mathcal{D}_{\mathrm{mod}1997}(x^y, n)$ for $n \in \{97, 197, 297, 397, 597, 797, 1997\}$ and the results are shown in Figure 7. The four learning phases follow Figure 3, and the empirical classification criteria are given in Appendix C.

This experiment is not a direct measurement of low-loss-region volume. Precisely measuring such a volume is computationally infeasible because it would require exploring a high-dimensional parameter space. The phase diagrams in Figure 7 should therefore be interpreted as an approximate probe based on sweeping weight decay and learning rate.

Even as approximate probes, the diagrams make the shrinking and alignment trends visible. As n increases, the number of Memo. cells decreases and eventually reaches zero, indicating that fewer training trajectories terminate in the training-only low-loss region. At the same time, Comp. cells appear and become more frequent, indicating that a larger fraction of trainable trajectories directly reaches the overlap of $\mathrm{LLR}(\mathcal{M}, \mathcal{D}_{train})$ and $\mathrm{LLR}(\mathcal{M}, \mathcal{D}_{val})$. Together, these trends suggest that the two low-loss regions become increasingly aligned as the nested dataset grows. The prevalence of Conf. cells at larger n is consistent with the additional data constraints shrinking the accessible low-loss regions. Appendix K shows the complementary effect that larger models make the low-loss

[4]Our setup is close to Figure 6 of (Liu et al., 2022).

regions more misaligned. Appendix E provides a corresponding comparison using the non-nested datasets $\mathcal{D}_{\mathrm{mod}}(x^y, m)$.

We make several additional experimental arrangements to obtain the results in Figure 7:

- The function f affects the geometry of the low-loss regions. We turn to use $f(x, y) = x^y$ because it has a much smaller overlap low-loss region than $f(x, y) = x + y$, see Appendix G.
- All cells within the same panel are obtained using the same training and validation partitions, generated by the random-partition policy.
- We strictly control randomness during training, see Appendix F for details.

5 Discussion

The Cause of Grokking Prior work has linked grokking to several factors, including Fourier circuit (Nanda et al., 2023), numerical stability (Prieto et al., 2025), weight decay (Xu et al., 2026) and the Slingshot effect (Thilak et al., 2022).

Instead, these factors can be separated into primary and secondary factors. The primary factor is the misalignment of low-loss regions, which is determined by the model architecture $\mathcal{M}$ and dataset $\mathcal{D}$, see Equation 2. The secondary factor is the training trajectory, or more precisely, how optimization approaches the overlap of the low-loss regions, see Figure 3. This trajectory is jointly controlled by weight decay, learning rate, weight initialization, and related hyperparameters. These factors are secondary because they cannot produce grokking when the primary factor already yields well-aligned low-loss regions.

The primary factors ($\mathcal{M}$ and $\mathcal{D}$) also include several implicit choices. For example, $\mathcal{M}$ should include numerical precision, because model weights in empirical experiments have finite precision rather than ideal real-valued coordinates. For modular arithmetic, $\mathcal{D}$ includes the dataset-generating function f, which strongly affects the geometry of low-loss regions. Our experiments show that $\mathcal{D}_{mod}(x^2 + y^2, 97)$ is more aligned than $\mathcal{D}_{mod}(x + y, 97)$, see Appendix J for details.

Interpretation of Grokking Prior work often treats grokking as a form of emergent generalization (Nanda et al., 2023; Wei et al., 2022). This description can suggest that, after sufficiently long training, transformers uncover the underlying knowledge hidden in the data. Our results suggest a different interpretation. Since the root cause of grokking is the misalignment of low-loss regions, delayed generalization indicates that the information carried by the training partition is insufficient relative to the model architecture. In this sense, grokking is not necessarily evidence of a hidden ability of neural networks. Rather, it should be viewed as a warning sign that the training data are insufficient for the model to reach stable generalization without passing through a delayed transition.

Similarly, the observation of memorization also suggests that the training dataset is too small relative to the model architecture. This is because both grokking and memorization can only happen when the training and validation low-loss regions are sufficiently misaligned.

6 Conclusion

We argue that grokking is governed by two hierarchical factors. The primary factor is the misalignment of the low-loss regions induced by the training and validation partitions, which is determined by the architecture and dataset. Such misalignment is more likely when the information carried by the training partition is small relative to the expressive capacity of the model. The secondary factor is the training trajectory, which is shaped by weight decay, learning rate, initialization, and related hyperparameters. When the primary factors induce well-aligned low-loss regions, grokking cannot be produced by secondary factors alone.

AI USE STATEMENT

In this work, we used generative AI tools to revise our paper to improve readability, implement methods, and write plotting scripts to visualize results and data. We have not used generative AI tools for formulating mathematical claims, helping develop theoretical models or conceptual frameworks, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, interpreting results, or assisting in the writing of proofs. The rest of the required disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools for sourcing/searching for information and summarizing or analyzing existing literature. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the source code and simulation configuration files as supplementary material, including the settings used for dataset generation and partitioning, model training, hyperparameter sweeps, phase classification, mode-connectivity experiments, and figure generation.

REFERENCES

Roman Abramov, Felix Steinbauer, and Gjergji Kasneci. Grokking in the wild: Data augmentation for real-world multi-hop reasoning with transformers. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 231–245. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/abramov25a.html.

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CQsmMYmlP5T.

Gregory Benton, Wesley Maddox, Sanae Lotfi, and Andrew Gordon Gordon Wilson. Loss surface simplexes for mode connecting volumes and fast ensembling. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 769–779. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/benton21a.html.

Liam Carroll. Phase transitions in neural networks. Master's thesis, The University of Melbourne, October 2021. URL http://therisingsea.org/notes/MSc-Carroll.pdf.

Breno W. Carvalho, Artur S. d'Avila Garcez, Luís C. Lamb, and Emílio Vital Brazil. Grokking explained: A statistical phenomenon, 2025. URL https://arxiv.org/abs/2502.01774.

Francesco D'Angelo, Maksym Andriushchenko, Aditya Varre, and Nicolas Flammarion. Why do we need weight decay in modern deep learning?, 2024. URL https://arxiv.org/abs/2310.04415.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1309–1318. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/draxler18a.html.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=dNigytemkL.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M. Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel, 2020. URL https://arxiv.org/abs/2010.15110.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/frankle20a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. URL https://arxiv.org/abs/1608.06993.

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

Aitor Lewkowycz and Guy Gur-Ari. On the training dynamics of deep networks with l_2 regularization, 2021. URL https://arxiv.org/abs/2006.08643.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6391–6401, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/a41b3bb3e6b050b6c9067c67f663b915-Abstract.html.

Kai Lion, Lorenzo Noci, Thomas Hofmann, and Gregor Bachmann. How good is a single basin?, 2024. URL https://arxiv.org/abs/2402.03187.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning, 2022. URL https://arxiv.org/abs/2205.10343.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, 2023. URL https://arxiv.org/abs/2301.05217.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

Lucas Prieto, Melih Barsbey, Pedro A. M. Mediano, and Tolga Birdal. Grokking at the edge of numerical stability, 2025. URL https://arxiv.org/abs/2501.04697.

Shaeke Salman and Xiuwen Liu. Overfitting mechanism and avoidance in deep neural networks, 2019. URL https://arxiv.org/abs/1901.06566.

Daniel Schwalbe-Koda, Sebastien Hamel, Babak Sadigh, Fei Zhou, and Vincenzo Lordi. Model-free estimation of completeness, uncertainties, and outliers in atomistic machine learning using information theory. *Nature Communications*, 16(1):4014, Apr 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-59232-0. URL https://doi.org/10.1038/s41467-025-59232-0.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

Lianghe Shi, Meng Wu, Huijie Zhang, Zekai Zhang, Molei Tao, and Qing Qu. A closer look at model collapse: From a generalization-to-memorization perspective. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 40658–40691. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/39d3b980155d3d52613e412b018004a4-Paper-Conference.pdf.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL https://arxiv.org/abs/1409.1556.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua M. Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the \emph{Grokking Phenomenon}. In *Has it Trained Yet? NeurIPS 2022 Workshop*, 2022. URL https://openreview.net/forum?id=lY1e0PNkSJ.

Yongding Tian, Zaid Al-Ars, Maksim Kitsak, and Peter Hofstee. Connecting independently trained modes via layer-wise connectivity, 2025. URL https://arxiv.org/abs/2505.02604.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency, 2023. URL https://arxiv.org/abs/2309.02390.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Susan Wei, Daniel Murfet, Mingming Gong, Hui Li, Jesse Gell-Redman, and Thomas Quella. Deep learning is singular, and that's good. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10473–10486, 2023. doi: 10.1109/TNNLS.2022.3167409.

Mingyue Xu, Gal Vardi, and Itay Safran. To grok grokking: Provable grokking in ridge regression, 2026. URL https://arxiv.org/abs/2601.19791.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017. URL https://arxiv.org/abs/1611.03530.

Peiliang Zhang, Huan Wang, Nikhil Naik, Caiming Xiong, and richard socher. DIME: An information-theoretic difficulty measure for AI datasets. In *NeurIPS 2020 Workshop: Deep Learning through Information Geometry*, 2020. URL https://openreview.net/forum?id=kvqPFy0hbF.

Zhenshuo Zhang, Jerry Weihong Liu, Christopher Re, and Hongyang R. Zhang. A hessian view of grokking in mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=BfLVCoov0b.

Zhanpeng Zhou, Yongyi Yang, Xiaojiang Yang, Junchi Yan, and Wei Hu. Going beyond linear mode connectivity: The layerwise linear feature connectivity, 2023. URL https://arxiv.org/abs/2307.08286.

Xuekai Zhu, Yao Fu, Bowen Zhou, and Zhouhan Lin. Critical data size of language models from a grokking perspective, 2024. URL https://arxiv.org/abs/2401.10463.

A Model Architecture and Training Configuration

We mostly follow the model architecture and training configuration of OpenAI (Power et al., 2022), with slight modifications. We denote the resulting architecture by $\mathcal{M}_{grok}$. The architecture is specified as follows:

- Model type: decoder-only causal transformer.
- Number of decoder blocks: 2.
- Model width: $d_{\text{model}} = 128$.
- Attention heads per block: 4.
- Per-head key/value dimension: $d_k = d_v = 32$.
- Feed-forward hidden dimension: $d_{\text{ff}} = 512$.
- Feed-forward nonlinearity: ReLU.
- Dropout probability: 0.1.
- Vocabulary size used by the model: 2000.
- Input sequence length for binary modular arithmetic data: 6 tokens.
- Positional encoding: fixed sinusoidal encoding.
- Attention mask: causal lower-triangular mask.
- Normalization: LayerNorm after the attention residual and after the feed-forward residual.
- Output head: bias-free linear projection from 128 features to 2000 logits.

The training configuration is specified as follows:

- Optimizer: AdamW.
- Learning rate: 10^{-3}.
- Weight decay: 0.0.
- Batch size: $\max(\text{len}(\mathcal{D}_{mod}), 65536)$
- AdamW momentum parameters: $\beta_1 = 0.9$, $\beta_2 = 0.98$.
- AdamW numerical stability constant: $\epsilon = 10^{-8}$.
- Total training epochs: 150,000 for $D_{mod}(x + y, 97)$, linearly scales down as the number of training samples increase for other datasets.
- Warmup epochs: 10.
- Minimum learning rate: 10^{-4}.
- Learning-rate schedule: linear warmup followed by cosine annealing.
- Warmup schedule: LinearLR with start factor 10^{-8} and end factor 1.0.
- Cosine schedule: CosineAnnealingLR with minimum learning rate 10^{-4}.
- Loss function: cross-entropy on the right-hand side tokens after the equals sign.

B Empirical Validation of Mode Connectivity for $\text{LLR}(\mathcal{M}_{grok}, \mathcal{D}_{mod}(x + y, 97))$

We empirically validate mode connectivity for $\mathcal{M}_{grok}$ on $\mathcal{D}_{mod}(x + y, 97)$. For each trial, we train a pair of independently initialized models and then connect the resulting modes using the Low-Loss Path Finding Algorithm (LLPF) introduced in (Tian et al., 2025). Figure 8 shows the resulting paths.

The LLPF hyperparameters used to reproduce Figure 8 are listed below. The notation follows Algorithm 1 and Table 4 of (Tian et al., 2025).

- Phase iterations: $T = 400$.

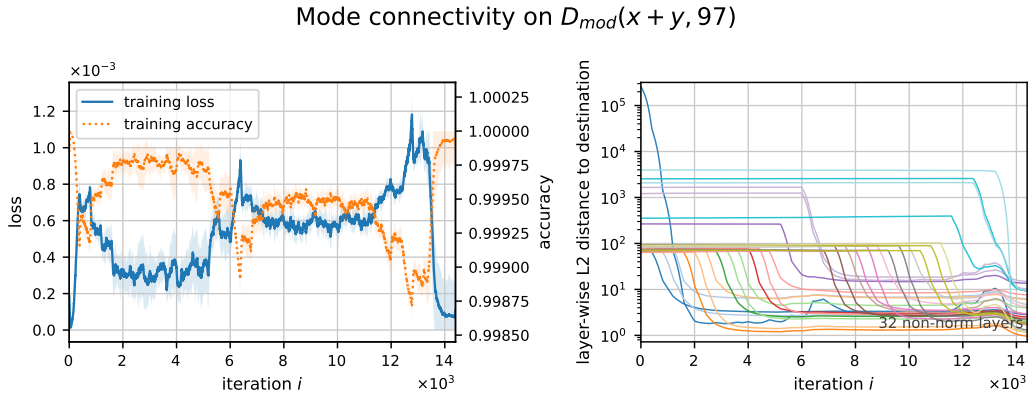


Figure 8: Results of applying LLPF to connect independently trained $\mathcal{M}_{grok}$ models. The left panel reports training accuracy and loss on the full $\mathcal{D}_{mod}(x + y, 97)$ dataset along the discovered path. The right panel reports layer-wise L_2 distance along the same path. The training loss remains below 1.5×10^{-3}, and the L_2 distance decreases gradually, supporting mode connectivity for $\mathcal{M}_{grok}$ on $\mathcal{D}_{mod}(x+y, 97)$. The final point has nearly the same training accuracy and loss as the starting point because both endpoints are obtained through SGD. The experiment is repeated ten times, and the shaded region in the top panel indicates the standard deviation across runs.

- Step parameter: $\text{step}_a = 0.002$.
- Step parameter: $\text{step}_f = 0.002$.
- Maximum training rounds: $r = 400$.
- Loss threshold: 0.001.
- Batch size: $\mathcal{B} = 9409$ (full batch).
- Optimizer: AdamW with $\eta = 0.0001$ and $\beta = (0.9, 0.999)$.

LLPF updates selected layers in a prescribed order. We therefore move one layer per phase and use the following hierarchical layer order. Layer names follow PyTorch naming conventions:

1. Top-level order: embedding $\rightarrow$ block0 $\rightarrow$ block1 $\rightarrow$ linear.
2. Within each block (block0 and block1): head0 $\rightarrow$ head1 $\rightarrow$ head2 $\rightarrow$ head3 $\rightarrow$ ffn.
3. Within each attention head (head0 to head3): query $\rightarrow$ key $\rightarrow$ value.

C Empirical Criteria for Learning Phases

Table 1 defines the empirical criteria used to assign each training trajectory to a learning phase.

Table 1: Empirical definitions of the four learning phases.

	Criteria		
Phase	Final train. acc. $\geq 95\%$	Final val. acc. $\geq 95\%$	Val. acc. at $t^{95\%}_{\text{train}} \geq 50\%$
Comprehension	Yes	Yes	Yes
Grokking	Yes	Yes	No
Memorization	Yes	No	Not applicable
Confusion	No	No	Not applicable

D PHASE CELLS AS AN INDIRECT PROBE OF ALIGNMENT

The phase diagram provides an indirect way to probe the alignment of the training and validation low-loss regions. Each cell corresponds to one pair of training hyperparameters, such as the learning rate and weight decay, and therefore to one optimization trajectory under the fixed model, dataset, initialization, and training protocol. The phase label summarizes the qualitative behavior of that trajectory according to the criteria in Appendix C.

The upper panels of Figure 9 schematically show two possible geometries. When the training and validation low-loss regions are misaligned, the training-only region is relatively large and different hyperparameter choices can produce Memorization or Grokking. When the two regions become more aligned, the training-only region becomes smaller relative to their overlap, so fewer trajectories are expected to produce these two phases. The corresponding phase diagrams in the lower panels therefore contain fewer Memorization and Grokking cells and a larger fraction of trajectories classified as Comprehension or Confusion.

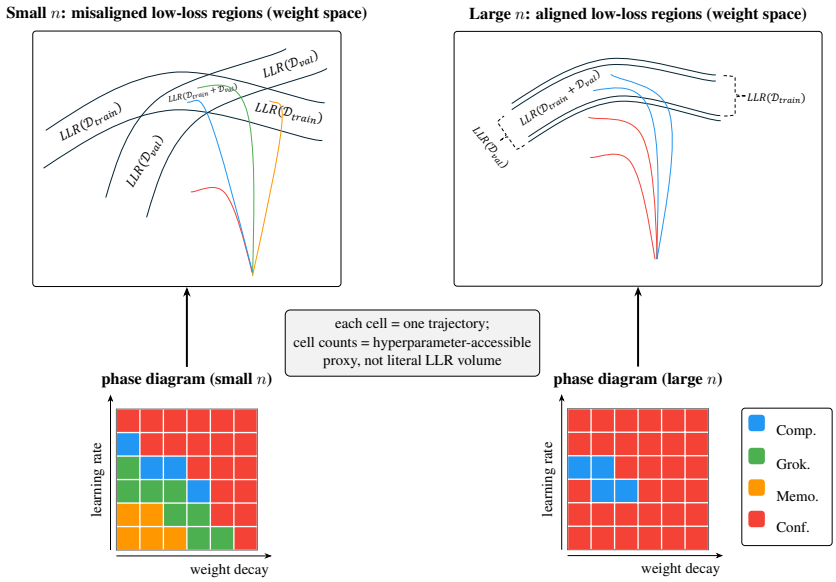


Figure 9: Schematic explanation of why phase-cell counts can serve as an indirect probe of low-loss-region alignment. The upper panels show a misaligned case, in which the training and validation low-loss regions have a relatively large training-only portion, and an aligned case, in which their overlap is relatively larger. Each cell in the corresponding phase diagram represents one learning-rate/weight-decay pair and therefore one training trajectory. As the nested dataset grows, the low-loss regions contract while the optimization problem remains otherwise controlled. The resulting changes in the phase-cell distribution provide a hyperparameter-accessible proxy for alignment rather than a literal measurement of low-loss-region volume.

In our construction, the full labeled datasets are nested: for $m < n$, $\mathcal{D}_{\text{mod1997}}(f, m) \subseteq \mathcal{D}_{\text{mod1997}}(f, n)$. Therefore, increasing n adds constraints while retaining all constraints from the smaller dataset, and the corresponding low-loss region is expected to contract rather than move to an unrelated part of parameter space. This inclusion relation is a geometric heuristic when the loss is averaged over samples, as discussed in Section 4, but it provides the basis for comparing the phase diagrams across n.

Moreover, with the model architecture, initialization, optimizer, randomness control, and training budget fixed, SGD remains driven, on average, toward lower training loss. Changing n therefore changes the set of constraints and the target low-loss region, but does not replace the optimization problem with an unrelated one. We consequently expect the training trajectories to change primarily through their access to the shrinking low-loss regions, rather than through arbitrary changes in

trajectory geometry. This controlled change is what makes the distribution of phase cells useful for comparing alignment across dataset sizes.

E Phase Diagrams for Non-Nested Modular Datasets

We also report phase diagrams for the conventional, non-nested datasets $\mathcal{D}_{\text{mod}}(x^y, m)$ with $m \in \{97, 197, 297, 397, 597, 797, 1997\}$. Unlike the fixed-modulus family in the main experiment, changing m changes the operand domain and labels, so these panels provide a qualitative cross-dataset comparison rather than a nested-dataset test. We use the same phase-classification criteria as in the main experiment.

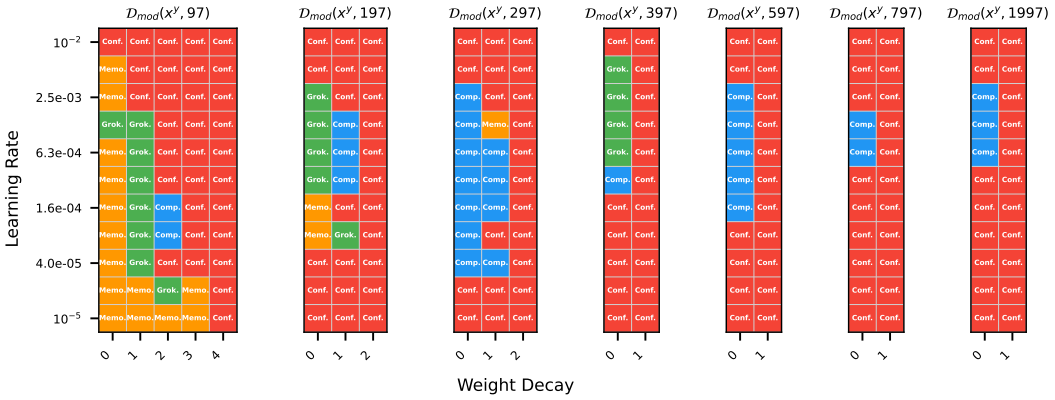


Figure 10: Phase diagrams for the conventional modular datasets $\mathcal{D}_{\text{mod}}(x^y, m)$ as the modulus m increases. Each cell represents one learning-rate/weight-decay pair and is classified using the criteria in Appendix C. As m increases, Memorization cells decrease and eventually disappear, while Comprehension cells emerge and become more prominent; Grokking cells are mainly observed at intermediate values of m. The resulting shift toward Comprehension and Confusion is qualitatively consistent with increasing low-loss-region alignment. Because the datasets are not nested across m, however, this figure should be interpreted as a cross-dataset comparison rather than direct evidence of monotonic low-loss-region shrinkage.

F Experimental Arrangements for Phase Diagram Sweeps

The phase-diagram experiment in Figure 7 uses several additional arrangements compared with the grokking experiment in Figure 1(b):

- The CosineAnnealingLR scheduler is replaced with a constant learning-rate schedule to isolate the effect of the swept hyperparameters.
- The batch size is set to $\max(\text{len}(\mathcal{D}), 65536)$ to limit GPU memory consumption.
- All cells for the same dataset start from the same initialized weights to reduce randomness across the sweep.
- The random seed is fixed across cells for the same dataset, reducing stochasticity from mini-batching and SGD.
- For each learning rate, we iterate over weight decay values $\lambda = 0 \ldots 9$. If two consecutive weight-decay values yield confusion, the remaining cells for that learning rate are also treated as confusion. This shortcut is motivated by empirical work showing that loss often follows a convex relationship with weight decay (Lewkowycz & Gur-Ari, 2021; D'Angelo et al., 2024).
- We use the tokenizer file from the largest modulus. For every operand-domain size n, we use the tokenizer copied from the fixed-modulus-1997 master dataset $\mathcal{D}_{\text{mod1997}}(x^y, 1997)$.

G Phase-Diagram Comparison Between $\mathcal{D}_{mod}(x+y, 997)$ and $\mathcal{D}_{mod}(x^y, 997)$

Figure 11 illustrates that the low-loss region of $\mathcal{D}_{mod}(x^y, 997)$ is smaller than that of $\mathcal{D}_{mod}(x+y, 997)$. Both datasets are randomly partitioned into training and validation sets using a 50% split.

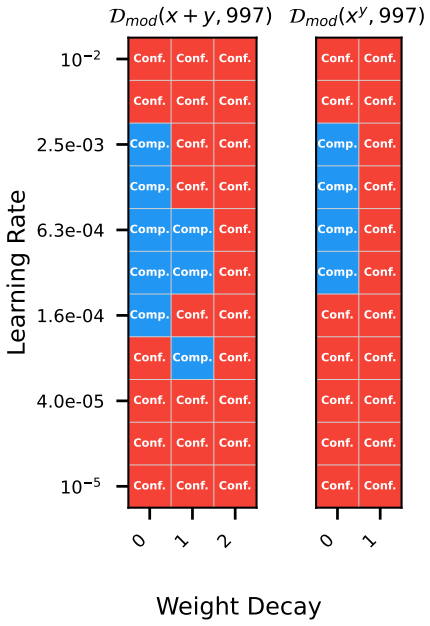


Figure 11: Comparison of phase diagrams for $\mathcal{D}_{mod}(x+y, 997)$ and $\mathcal{D}_{mod}(x^y, 997)$. The visible trainable region is larger for $\mathcal{D}_{mod}(x+y, 997)$, indicating a larger low-loss region.

H Anti-Grokking Ablation by Same-Partition Ratio

Figure 12 shows that grokking can recover when the same-partition ratio is reduced from the anti-grokking value of 100%. At 100%, every paired sample located at (a, b) and (b, a) belongs to the same partition. Lowering this ratio moves the split away from the fully symmetry-preserving anti-grokking construction.

H.1 Symmetry-aware training coverage

The main text argues that the raw size of the training partition is not the fundamental quantity governing grokking. Instead, the relevant quantity is the effective information entropy carried by that partition. To make this idea operational, we use the symmetry of the modular-addition task, $f(a, b) = f(b, a)$. Let $\mathcal{T}$ denote the set of operand pairs in the training partition, and let $\mathcal{O}(a, b) = \{(a, b), (b, a)\}$ denote the corresponding symmetry orbit. We call an orbit *covered* when $\mathcal{O}(a, b) \cap \mathcal{T} \neq \varnothing$; thus, observing one member of the orbit provides information about both ordered samples. The symmetry-aware training coverage is then the fraction of symmetry orbits covered by $\mathcal{T}$:

$$C_{\text{sym}}(\mathcal{T}) = \frac{|\{\mathcal{O}(a, b) : \mathcal{O}(a, b) \cap \mathcal{T} \neq \varnothing\}|}{|\{\mathcal{O}(a, b)\}|}. \tag{6}$$

We compare this coverage with the raw training-partition size in two experiments. First, under the random-partition policy, we vary the training-partition size. As shown in Figure 13, 40% training does not produce a complete grokking trajectory, whereas 45% training produces a clear delayed rise in validation accuracy. The corresponding coverages are 0.637 and 0.695, respectively. Second, in the anti-grokking experiment, we vary the fraction of symmetric pairs assigned to the same partition,

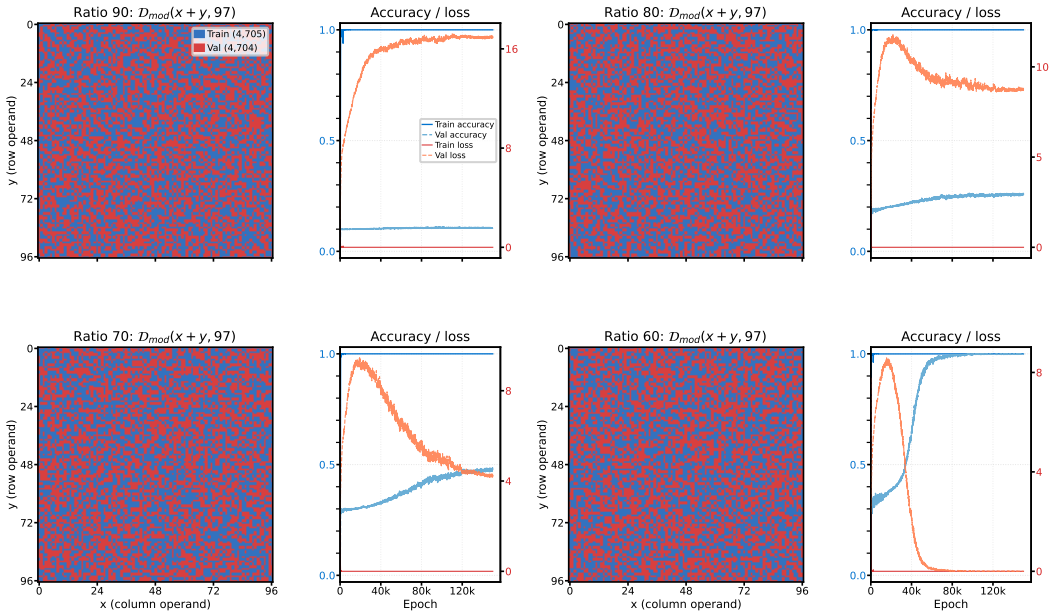


Figure 12: Grokking gradually recovers as the same-partition ratio decreases from 100% toward lower ratios.

as shown in Figure 12. No complete grokking trajectory is observed at a same-partition ratio of 70%, whereas a clear grokking trajectory is recovered at 60%; the corresponding coverages are 0.648 and 0.680. Thus, the transition intervals in the two experiments overlap substantially. In particular, the coverage at 45% training in the random-partition experiment is close to the coverage at a 60% same-partition ratio in the anti-grokking experiment, even though the raw partition parameters differ. This agreement suggests that symmetry-aware training coverage captures task-relevant information more accurately than the raw number of training samples and can serve as an empirical proxy for effective information entropy. This interpretation is empirical and limited to the modular addition task. A formal entropy measure that incorporates task knowledge remains an interesting direction for future work.

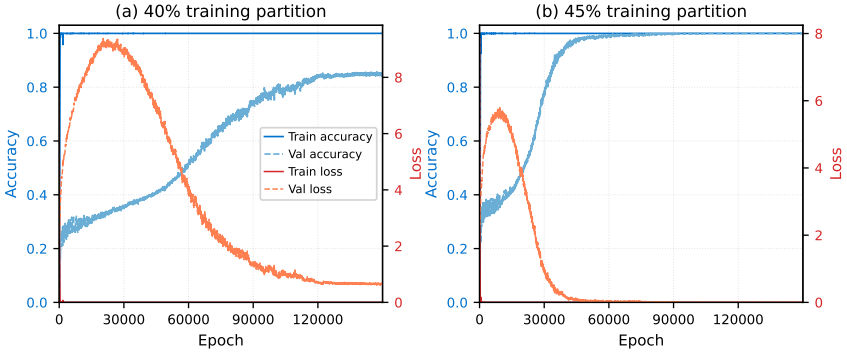


Figure 13: Training dynamics for the normal grokking case on $\mathcal{D}_{\mathrm{mod}}(x+y, 97)$. With a 40% training partition (a), validation accuracy remains below one and no complete grokking transition is observed. With a 45% training partition (b), training accuracy reaches one first and validation accuracy rises substantially later, producing a clear grokking trajectory.

Table 2: Symmetry-aware training coverage for the random-partition grokking experiments and the same-partition-ratio anti-grokking experiments. For the random partitions, the reported values are the expected coverages $p/m + (1 - 1/m)(2p - p^2)$ with $m = 97$ and training fraction p; for the symmetry-controlled partitions, coverage is computed directly from the realized split using the same symmetry-closure definition. In the random-partition setting, the transition from no complete grokking to grokking occurs between 40% and 45% training, corresponding to coverages of 0.637 and 0.695. In the anti-grokking setting, the corresponding transition occurs between same-partition ratios of 70% and 60%, corresponding to coverages of 0.648 and 0.680. The close coverage values suggest that symmetry-aware coverage is a more informative comparison across the two experimental controls than the raw partition parameters.

Grokking case		Anti-grokking case	
Train partition (%)	Training coverage	Same-partition ratio (%)	Training coverage
50	0.747	100	0.500
45	**0.695**	90	0.550
40	0.637	80	0.599
35	0.575	70	0.648
30	0.508	**60**	**0.680**
25	0.436	50	0.747

I Additional Phase Diagrams for $m = 997$

This section reports phase diagrams for four additional datasets: $\mathcal{D}_{mod}(x+y, 997)$, $\mathcal{D}_{mod}(xy, 997)$, $\mathcal{D}_{mod}(x^2+2y^2, 997)$, and $\mathcal{D}_{mod}(x^2+2y^2+xy, 997)$. These diagrams use a narrower weight-decay range than Figure 7, giving higher resolution and revealing more Comp. cells.

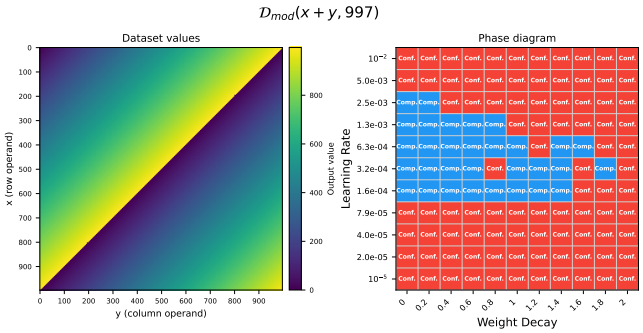


Figure 14: Dataset illustration (left) and phase diagram (right) for $\mathcal{D}_{mod}(x + y, 997)$.

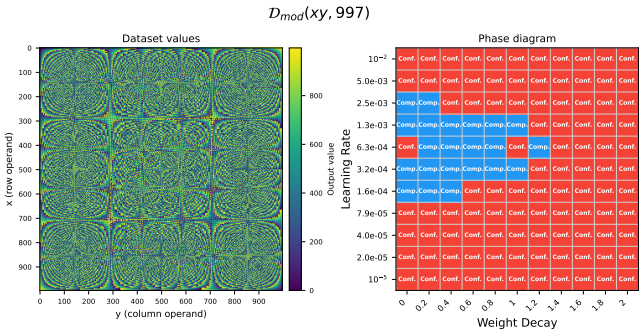


Figure 15: Dataset illustration (left) and phase diagram (right) for $\mathcal{D}_{mod}(xy, 997)$.

Figures 14–17 show that combining learnable functions does not guarantee trainability. At $m = 997$, both $f_1(x, y) = xy$ and $f_2(x, y) = x^2 + 2y^2$ are trainable because they produce a visible number

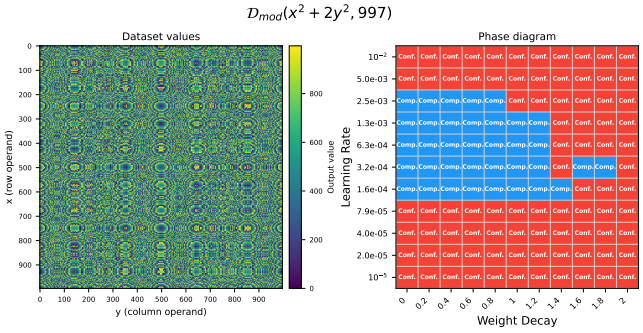


Figure 16: Dataset illustration (left) and phase diagram (right) for $\mathcal{D}_{mod}(x^2 + 2y^2, 997)$.

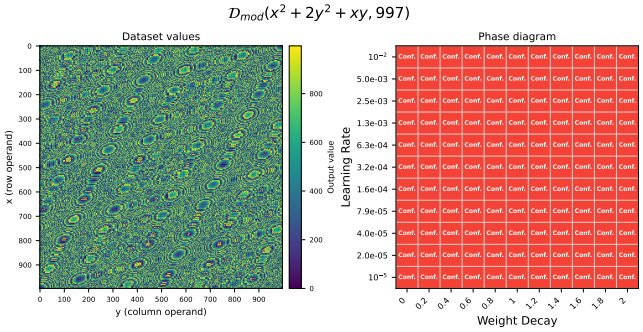


Figure 17: Dataset illustration (left) and phase diagram (right) for $\mathcal{D}_{mod}(x^2 + 2y^2 + xy, 997)$.

of Comp. cells. However, their sum $f(x, y) = f_1 + f_2 = x^2 + 2y^2 + xy$ is not trainable under the same sweep, because no Comp. cells are observed.

J Additional Phase Diagrams for $m = 97$

This section reports phase diagrams for $\mathcal{D}_{mod}(x+y, 97)$, $\mathcal{D}_{mod}(x^2+y^2, 97)$, and $\mathcal{D}_{mod}(2x^2+y, 97)$. Under $m = 97$, the low-loss regions for $f(x, y) = x^2 + y^2$ are the most aligned among these three functions, those for $f(x, y) = x + y$ are intermediate, and those for $f(x, y) = 2x^2 + y$ are the least aligned.

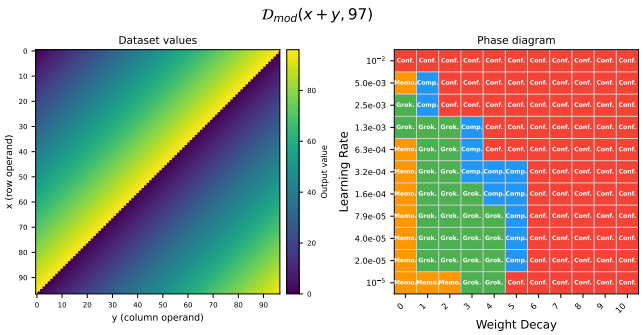


Figure 18: Dataset illustration (left) and phase diagram (right) for $\mathcal{D}_{mod}(x + y, 97)$.

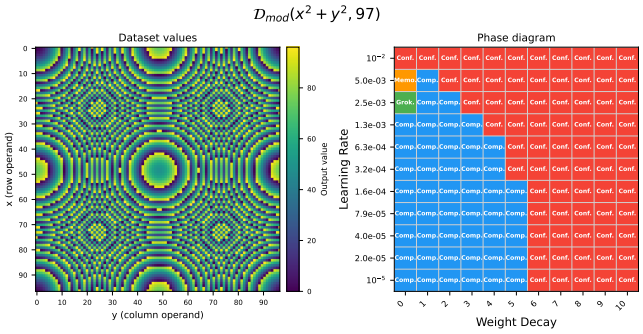


Figure 19: Dataset illustration (left) and phase diagram (right) for $\mathcal{D}_{mod}(x^2 + y^2, 97)$.

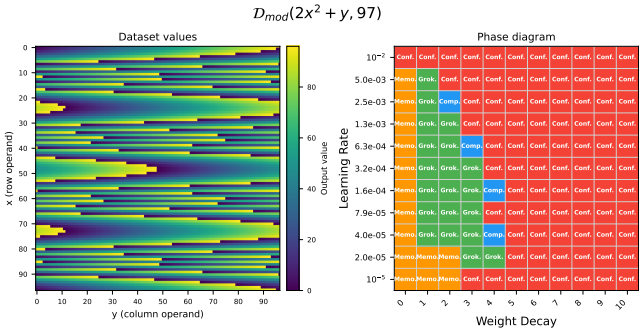


Figure 20: Dataset illustration (left) and phase diagram (right) for $\mathcal{D}_{mod}(2x^2 + y, 97)$.

K Effect of Model Size on Phase Diagrams

This section illustrates how model size influences low-loss-region alignment.

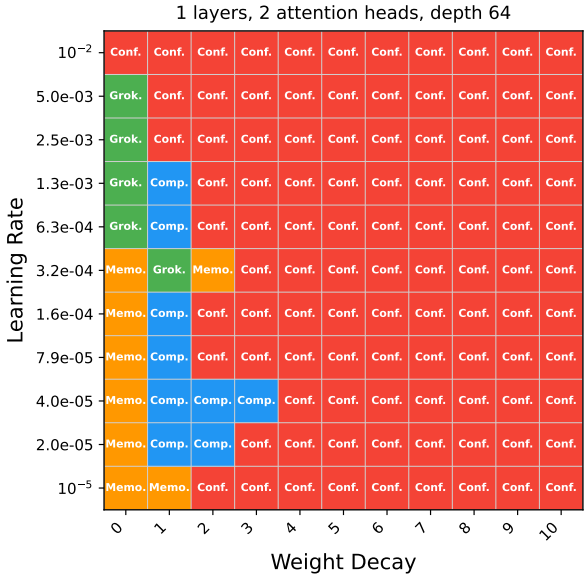


Figure 21: Smaller model with 1 decoder layer, 2 attention heads, and width 64. The fraction of Comprehension cells is higher than in the baseline model, indicating that the low-loss regions are more aligned.

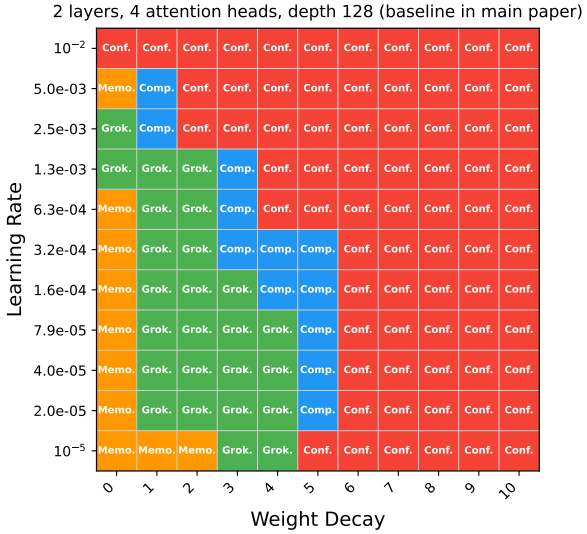


Figure 22: Baseline model with 2 decoder layers, 4 attention heads per layer, and width 128.

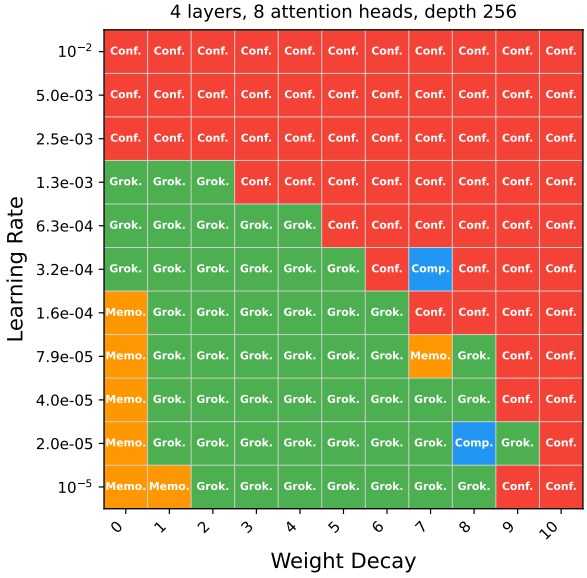


Figure 23: Larger model with 4 decoder layers, 8 attention heads per layer, and width 256. Comprehension cells are rare, indicating that the low-loss regions are more misaligned than in the baseline model.

The baseline $\mathcal{M}_{grok}$ model has 2 decoder layers, 4 attention heads per layer, and width 128. We construct smaller and larger variants by multiplying these values by 0.5 and 2, respectively. We then train each model on $\mathcal{D}_{mod}(x + y, 97)$ to examine low-loss-region alignment. Since a larger model has greater representational capacity, increasing model size can play a role similar to reducing the effective dataset size, making the low-loss regions more misaligned. The results in Figures 21–23 support this analysis.

L Singular Learning Theory

Singular Learning Theory(SLT) (Wei et al., 2023) treats layered neural networks as singular models, implying that directions can exist in $\mathbb{R}^D$ along which a model moves while preserving loss. This view suggests that SGD solutions do not simply lie at the bottom of isolated basins, as in traditional loss-landscape visualizations (Li et al., 2018), but instead lie along extended valleys (Carroll, 2021).

M Mode Connectivity vs Linear Mode Connectivity

We distinguish Mode Connectivity (MC) from Linear Mode Connectivity (LMC). LMC works do not investigate mode connectivity for independently trained models (Zhou et al., 2023). LMC typically studies whether two low-loss models can be connected by a linear low-loss path when the models are obtained by **spawning** or **permutation**. **Model spawning** initializes and trains one model for a few epochs, copies it into two descendants, and then trains the descendants independently to convergence (Frankle et al., 2020; Fort et al., 2020). **Model permutation** trains two models independently and then permutes the neurons of one model to align it with the other, producing a functionally equivalent representation (Ainsworth et al., 2023; Entezari et al., 2022).

N Limitations

Several aspects of the modular-arithmetic setup may not transfer directly to general datasets.

1. Modular arithmetic datasets are noise-free, whereas real datasets may contain noise. This can make low-loss regions less clean and may change the observed training dynamics.
2. The symmetry-based knowledge concept used in Section 3 may not have a direct analogue in general datasets.
3. We use a 50% training split, which is uncommon in many practical settings.

O Computational Resources

We report the GPU cost of producing the main results. A single grokking experiment, corresponding to one cell in Figure 7, takes approximately 25 minutes on an NVIDIA RTX 5090 GPU for 150,000 epochs. Sweeping a full $11 \times 11 = 121$ grid for one dataset takes approximately 2.1 days. To reduce this cost, we skip the remaining cells for a learning rate after two consecutive weight-decay values fall in the confusion region. With this optimization, the runtime for one phase diagram is reduced to approximately one day.