# Unifying Grokking and Double Descent

**Xander Davies**[*]
Harvard University

**Lauro Langosco**[*]
Cambridge University

**David Krueger**
Cambridge University

## Abstract

A principled understanding of generalization in deep learning requires unifying disparate observations under a single conceptual framework. Previous work has studied *grokking*, a training dynamic in which a sustained period of near-perfect training performance and near-chance test performance is eventually followed by generalization, as well as the superficially similar *double descent*. These topics have so far been studied in isolation. We hypothesize that grokking and double descent can be understood as instances of the same learning dynamics within a framework of pattern learning speeds, and that this framework also applies when varying model capacity instead of optimization steps. We confirm some implications of this hypothesis empirically, including demonstrating model-wise grokking.

## 1 Introduction

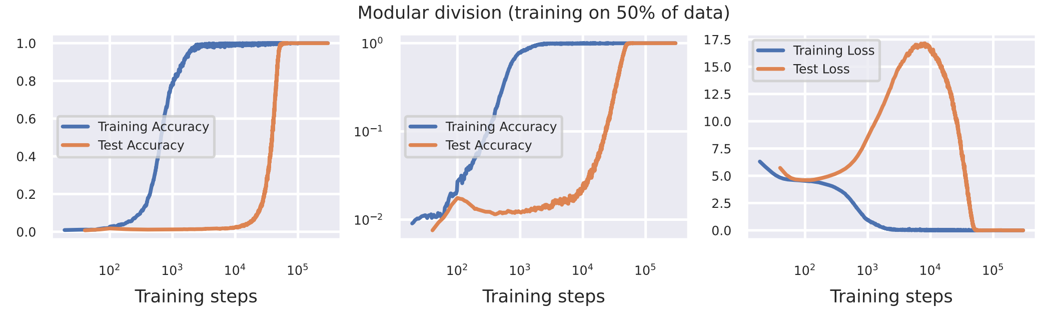

Figure 1: **Modular Division.** Three views of the same training run. *Left:* Typical grokking. *Middle:* When accuracy is put on a log scale, a small double descent becomes visible. *Right:* The double descent is more pronounced in test loss.

On some datasets, neural networks exhibit surprising training dynamics termed *grokking* by Power et al. (2022). In grokking, the model initially overfits, achieving perfect performance on the training set while remaining at near-chance performance on the test set. Later in training, test performance improves and the model eventually achieves perfect test accuracy. Because the model learns to generalize far after perfect classification of the training data, grokking is a well-distilled demonstration of an *inductive bias* that favours well-generalizing solutions (Zhang et al., 2021). Grokking is reminiscent of the *double descent* phenomenon (Belkin et al., 2018; Nakkiran et al., 2021), in which test performance initially improves, then worsens as the model overfits, and then eventually improves again as model capacity increases.

We argue that double descent and grokking are best viewed as two instances of the same phenomenon, in which inductive biases prefer better-generalizing but slower to learn patterns, leading to a transition

[*] Equal contribution. Correspondence to `alexander_davies@college.harvard.edu`.

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

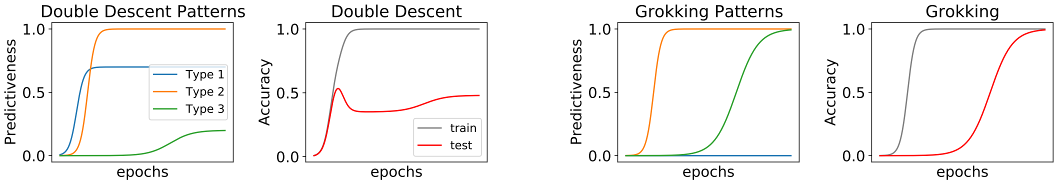

Figure 2: **Toy Model.** Left to right, (1) development of pattern predictiveness during double descent, with well-performing Type 1 patterns and somewhat-predictive Type 3 patterns; (2) resulting train and test accuracy curves for double descent; (3) pattern predictiveness during grokking, with poorly-performing Type 1 patterns and perfectly-predictive Type 3 patterns; (4) resulting train and test curves for grokking.

from poorly-generalizing to well-generalizing patterns. This transition happens both *epoch-wise* as a function of training time (Figure 1) and *model-wise* as a function of model size (Figure 3).

**Contributions.** We provide a conceptual framework (Claim 1, Claim 2) and toy model (Section 3) which unifies the learning dynamics of grokking and double descent. Our Claim 1 is supported by the toy model, and we provide evidence for Claim 2 by showing that grokking can occur as a function of model size (Section 4.1 and Figure 1).

## 2 Background

**Inductive Biases.** In neural network training, there are many different parameter configurations which fit the training data (Zhang et al., 2021). The *inductive bias* of a training procedure determines how our training procedure selects between such solutions (Battaglia et al., 2018), and may be composed of both *explicit* biases (like regularization terms) and *implicit* biases (like certain solutions being easier to reach in parameter-space).

**Double Descent.** The double descent phenomenon describes surprising generalization behavior of neural networks, in which test performance initially improves, then worsens as the model overfits, and then eventually improves again as we increase the *capacity* of our training procedure (Belkin et al., 2018). This capacity measure was originally demonstrated with model size (model-wise); Nakkiran et al. (2021) generalize the capacity notion in *effective model complexity*, and show double descent can occur when varying the number of optimization steps (epoch-wise) and other forms of capacity modulation. Recently, Pezeshki et al. (2021) and Stephenson & Lee (2021) show that epoch-wise double descent can be modeled as different patterns being learned at different speeds.

## 3 Pattern Learning

We first claim that epoch-wise double descent and grokking share the same underlying dynamics, and then claim that these dynamics also transfer to the model-wise setting.

**Claim 1** (Pattern learning dynamics)**.** *Grokking, like epoch-wise double descent, occurs when slow patterns generalize well and are ultimately favored by the training regime, but are preceded by faster patterns which generalize poorly.*

**Toy Model.** To formalize Claim 1, we propose a toy model of neural network training as the learning of *patterns*, or particular input-output functions. In this model, a trained neural network consists of a set of patterns, each of which independently classifies a datapoint correctly with a fixed probability. The network classifies a training datapoint correctly when any of its patterns does so. Test accuracy depends on how well patterns generalize (we give a detailed account of the toy model in Appendix A).

In our model of grokking and double descent, there are three types of patterns learned at different speeds. **Type 1** patterns are fast and generalize well (heuristics). **Type 2** patterns are fast, though slower than Type 1, and generalize poorly (overfitting). **Type 3** patterns are slow and generalize well.

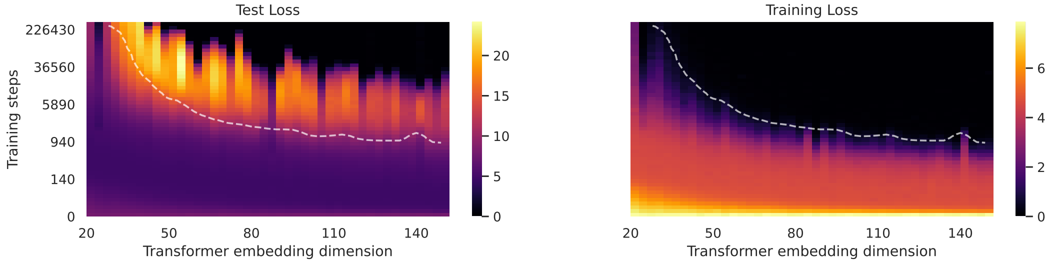

Figure 3: **Model-Wise Grokking.** *Left:* Test loss as function of training steps and model size. *Right:* Training loss. All models are transformers trained on modular division on 50% of the training data. Both model-wise and epoch-wise double descent are visible. Dotted line represents 90% training accuracy.

Type 3 patterns are ultimately *preferred* by the training regime: that is, they are used when multiple patterns can predict the same input (see Appendix A).

As shown in Figure 2, this simple model can produce grokking as well as double descent, and the two can be interpolated between by solely modulating the maximum predictiveness of heuristics (Type 1) and the maximum predictiveness of well-generalizing slow patterns (Type 3).[2]

**Claim 2** (Pattern learning as function of EMC)**.** *In both grokking and double descent, pattern learning occurs as a function of* effective model complexity *(EMC) (Nakkiran et al., 2021), a measure of the complexity of a model that integrates model size and training time.*[3]

If Claim 2 is correct, then grokking (like double descent) depends on EMC rather than training time, and thus should happen model-wise as well as epoch-wise. We confirm that this is indeed the case (Section 4.1).

## 4 Experiments

### 4.1 Model-Wise Grokking

Following the original grokking setting of Power et al. (2022), we train decoder-only transformers (Vaswani et al., 2017) with causal attention masking on the binary operation of division mod 97 (details in B). We explore the effects of regulating effective model complexity by varying parameter-count.

Figure 3 shows train and test loss with respect to optimization steps and embedding dimension. For a range of values, delayed generalization occurs both when moving vertically (epoch-wise grokking), as well as when moving horizontally (model-wise grokking). Figure 5 shows the corresponding accuracy heatmaps, with clear double descent behavior both epoch-wise and model-wise. Figure 6 shows models with range of embedding dimensions trained for 400k epochs.

### 4.2 A Type 1 Pattern in Grokking Setting

In Figure 1, we show that in the modular division setting of Power et al. (2022), we *do* see non-monotonic behavior in test accuracy due to the development. Early in training, the model learns $0/b = 0 \mod n$; this pattern generalizes well and is learned quickly, corresponding to a Type 1 pattern in our toy model. As predicted in our toy model, this leads to an initial spike above chance in test accuracy. Interestingly, the development of poorly-generalizing Type 2 features then leads to *worse-than-chance* performance on the rest of the data, likely due to anti-correlation between train and test set values for every dividend and divisor in modular division, causing a descent in test

[2] Code replicating our results is available at `github.com/xanderdavies/unifying-grok-dd`, including an interactive notebook to explore the toy model and interpolate between grokking and double descent.

[3] In this paper, we use EMC as our metric for capacity of a training regime, but encourage future work exploring alternative metrics. Models that are trained for longer have higher EMC.

performance. This small bump is noticeable in Power et al. (2022), but not noted by the authors. For experimental validation of this theory, see Figure 7.

## 5 Related Work

**Grokking.** Grokking was first demonstrated by Power et al. (2022), who state that grokking is distinct from double descent because generalization occurs far past the interpolation threshold. Liu et al. (2022) demonstrate that grokking generalization speed can be accelerated by using a larger representation step-size. This is inline with findings by Heckel & Yilmaz (2020) about epoch-wise double descent, where decreasing learning rates in later layers (which learn faster) aligns pattern learning speeds. We consider this further evidence that both grokking and epoch-wise double descent occur as a result of similar learning dynamics resulting from different speeds of pattern development. Nanda & Lieberum (2022) investigate grokking through mechanistic interpretability, with findings in line with our results (specifically observing the development of a Type 3 pattern, and analyzing the final algorithm).

**Pattern Learning at Different Speeds.** Heckel & Yilmaz (2020) find different parts of networks learning at different speeds can cause two bias-variance trade-off curves with different minima, leading to double descent in test error; they show that adjusting step-sizes of different layers can align the learning curves and eliminate double descent behavior. Heckel & Yilmaz (2020) argue that epoch-wise double descent occurs due to this difference in learning speed, as opposed to due to regulating model complexity. Pezeshki et al. (2021) use a linear teacher-student model to demonstrate that epoch-wise double descent can be explained by different patterns being learned at different speeds. Stephenson & Lee (2021) find similar, and experimentally demonstrate double descent can be avoided by removing or accelerating slow-to-learn features.

**Weight Decay** Both Nakkiran et al. (2021) and Pezeshki et al. (2021) find that weight decay acts as a capacity constraint, resulting in both weight decay-wise double descent (via EMC regulation) and preventing the learning of slower features. In the grokking setting, however, weight decay plays a significant role in speeding up time to generalization (Power et al., 2022). We speculate that though weight decay does lower model capacity, it also imposes an inductive bias which favors the grokked solution and thus speeds up generalization.

## 6 Conclusion

In this work, we cover two phenomena in deep learning that were previously studied in isolation—grokking and double descent—and argue that they are best understood as instances of the same underlying learning dynamics.

The project of building a principled understanding of how and when networks generalize may be especially important from the point of view of *AGI safety*, the problem of ensuring that advanced AI systems are safe despite pursuing misaligned goals (Carlsmith, 2022; Bostrom, 2014) and incentives to seek power or deceive human operators for instrumental reasons (Turner et al., 2021; Omohundro, 2008). A central problem is that we may need to be certain of the safety of a model before we scale it to a capability level beyond which we cannot control it. Concerningly, it is well-known that the out-of-distribution (OOD) generalization behavior of deep learning systems can be hard to control or predict. A robust theory of generalization and learning in neural networks may be necessary to solve this problem.

## 7 Acknowledgements

Thanks to Antonia Marcu, Dmitrii Krasheninnikov, Robert Kirk, Davis Brown, Peter Barnett, Rohin Shah, Nathan Bernard, Neel Nanda, Max Nadeau, Max Kaufmann, Alan Chan, Michaël Trazzi, and Oam Patel for valuable discussion, feedback on drafts of this paper and / or contributing suggestions for experiments.

## References

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, October 2018. URL http://arxiv.org/abs/1806.01261. arXiv:1806.01261 [cs, stat].

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*, 2018.

Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, Inc., 2014. ISBN 0199678111.

Joseph Carlsmith. Is power-seeking ai an existential risk? *arXiv preprint arXiv:2206.13353*, 2022.

Paul Christiano, Mark Xu, and Ajeya Cotra. Eliciting latent knowledge, 2021. URL https://www.lesswrong.com/posts/qHCDysDnvhteW7kRd/arc-s-first-technical-report-eliciting-latent-knowledge.

Reinhard Heckel and Fatih Furkan Yilmaz. Early Stopping in Deep Networks: Double Descent and How to Eliminate it, September 2020. URL http://arxiv.org/abs/2007.10099. arXiv:2007.10099 [cs, stat].

Tamera Lanham. Externalized reasoning oversight, 2022. URL https://www.lesswrong.com/posts/FRRb6Gqem8k69ocbi/.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *arXiv preprint arXiv:2205.10343*, 2022.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Neel Nanda and Tom Lieberum. A Mechanistic Interpretability Analysis of Grokking, August 2022. URL https://www.alignmentforum.org/posts/N6WM6hs7RQMKDhYjB/a-mechanistic-interpretability-analysis-of-grokking.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. https://distill.pub/2018/building-blocks.

Stephen Omohundro. The basic ai drives. volume 171, pp. 483–492, 01 2008.

Mohammad Pezeshki, Amartya Mitra, Yoshua Bengio, and Guillaume Lajoie. Multi-scale Feature Learning Dynamics: Insights for Double Descent, December 2021. URL http://arxiv.org/abs/2112.03215. arXiv:2112.03215 [cs, stat].

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.

Nate Soares. Visible thoughts project, 2021. URL https://www.alignmentforum.org/posts/zRn6cLtxyNodudzhw/.

Cory Stephenson and Tyler Lee. When and how epochwise double descent happens, August 2021. URL http://arxiv.org/abs/2108.12006. arXiv:2108.12006 [cs].

Alex Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli. Optimal policies tend to seek power. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 23063–23074. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/c26820b8a4c1b3c2aa868d6d57e14a79-Paper.pdf`.

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.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

# A Toy Model Equations

We model learning dynamics as the interaction between *patterns*, or particular input-output functions learned during the training process. At timestep $t$, we say pattern $i$ achieves accuracy $p_i(t)$ on the training set, modeled by a sigmoid function parameterized by a maximum predictiveness $0 \leq \gamma_i \leq 1$, inflection point $b_i \geq 0$, and learning speed $\alpha_i \geq 0$:

$$p_i(t) = \frac{\gamma_i}{1 + e^{-\alpha_i(t-b_i)}}$$

During training, classification is modeled as series of *independent* Bernoulli events (one per pattern), so that for each pattern $i$ at each time step $t$ we have a corresponding Bernoulli event:

$$X_i^t \sim \text{Bern}(p_i(t))$$

Collective training accuracy of all $n$ patterns is modeled as the union of these events:

$$\text{acc}_{\text{train}}(t) = P\left(\bigcup_{i=1}^{n} X_i^t\right)$$

That is, if any pattern successfully classifies the example, the collective model does as well. At test time, we assign generalization parameters $g_i$ and say generalization $G$ occurs with probability $g_i$ if $X_i^t$ is the sole successful event, and $\frac{g_i + \cdots + g_m}{m}$ in the case of $m$ successful events $\{X_i^t, \cdots, X_m^t\}$, computed via a weighting of the inclusion-exclusion expansion of $P\left(\bigcup_{i=1}^{n} X_i^t\right)$:

$$\text{acc}_{\text{test}}(t) = \sum_i g_i P\left(X_i^t\right) - \sum_{i<j} \frac{g_i + g_j}{2} P\left(X_i^t \cap X_j^t\right) + \sum_{i<j<k} \frac{g_i + g_j + g_k}{3} P\left(X_i^t \cap X_j^t \cap X_k^t\right)$$
$$- \ldots + (-1)^{n+1} \frac{g_1 + \cdots g_n}{n} P\left(X_1^t \cap \cdots \cap X_n^t\right)$$

We now add a notion of a *preferred pattern* to our model; we'll later provide an alternative intuition for this behavior (Section A.2).

**Definition A.1** (Preferred Pattern)**.** If pattern $i$ is preferred, we say generalization is given by $P(G_t|X_i^t \cap X_j^t \cap \cdots) = g_i$. That is, generalization always occurs with respect to $i$'s generalization parameter $g_i$ if $X_i^t$ is successful, even if multiple other patterns have successful events.

In the case of a single preferred pattern $p_0$, this induces test accuracy:

$$\text{acc}_{\text{test}}(t) = g_0 P(X_0^t) + \sum_{i>0} g_i P\left(X_i^t \cap \neg X_0^t\right) - \sum_{i<j} \frac{g_i + g_j}{2} P\left(X_i^t \cap X_j^t \cap \neg X_0^t\right)$$
$$+ \ldots + (-1)^{n+1} \frac{g_1 + \cdots g_n}{n} P\left(X_1^t \cap \cdots \cap X_n^t \cap \neg X_0^t\right)$$

See Figure 4 for a depiction of the toy model with two patterns.

## A.1 Heuristics, Memorization, and Slow Well-Generalizing

As described in Section 3, in our paper we model learning as the interaction of three patterns:

- **Type 1:** A well-generalizing pattern which is learned quickly during training (*heuristics*).
- **Type 2:** A poorly-generalizing pattern which is learned slower than heuristics but quicker than category 3 patterns.
- **Type 3:** A well-generalizing pattern which is learned slowly, but is ultimately *preferred* by the training regime (Definition A.1).

## A.2 Alternative intuition for preferred patterns

Rather than thinking about patterns as functions from the entire input distribution $\mathcal{D}$ to the model's output $\mathcal{O}$, we can instead think of patterns as functions from a *subset* of the input distribution $\mathcal{D}_i \subset \mathcal{D}$. We can then say that a *preferred pattern* enforces that its domain $\mathcal{D}_i$ is disjoint from all other pattern domains; that is $\mathcal{D}_i$ is preferred if $\forall j \in \{0, \cdots, n\}, i \neq j \implies \mathcal{D}_i \cap D_j = \emptyset$.

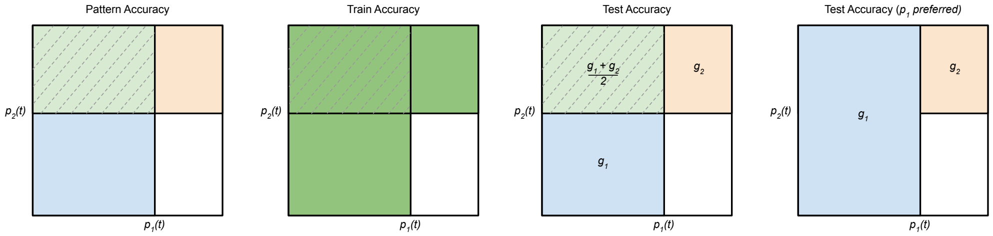

Figure 4: **Two pattern toy model.** Full square represents the data set. Left to right, (1) blue shows training samples correctly classified by $p_1(t)$, red shows training samples correctly classified by $p_2(t)$ correctly classifies, dashed green shows samples correctly classified by both; (2) accuracy is the union of patterns 1 and 2; (3) test accuracy is the sum of correct classifications weighted by generalization parameters, and by their average generalization in the case both correctly classifying; (4) test accuracy given pattern 1 is preferred, causing all examples correctly classified by pattern 1 to generalize according to $g_1$.

## B Experiment Details

In all of our experiments, we train decoder-only transformer with causal attention masking. Each residue is encoded as a symbol, and loss and accuracy are only evaluated on the answer part of the equation. Unless otherwise stated, we use a 2-layer network of width 128, with a single attention head. We typically train for 400 thousand optimization steps via AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.98$), with learning rate of 1e-3 and weight decay of 1e-5.

## C The science of deep learning may be important for safety

A key factor that influences whether we will be able to solve alignment is the degree to which our AI systems are transparent and well-understood. A system is transparent if we are able to ascertain what process lead it to produce a specific output or action; we might try to train a system to produce visible thoughts, truthfully report its knowledge, or probe it with interpretability tools (Christiano et al., 2021; Soares, 2021; Lanham, 2022; Lin et al., 2021; Olah et al., 2018).

Studying deep learning does not directly give us transparency, but it improves our understanding of its inner workings. This may help us build better interpretability tools, or make sure that our training process incentivizes truthful reporting rather than deception (Christiano et al., 2021). An improved understanding of how deep learning systems work and what they are capable of has direct benefits as well. For example, we may want to build a system that can perform a specific feat of engineering, while making sure that it does not have the capability to model humans; or we might try to build a system that optimizes for the short-term approval of its operator, and we would like to know in advance if it will generalize in a way that leads to it deceiving its operator (in order to gain a higher measured approval score) once it gains the ability to do so.

A science of deep learning may also be harmful if theoretical or empirical breakthroughs result in faster advancement of potentially dangerous capabilities. This is a major drawback of this line of research. A tentative argument in favor of scientific / conceptual work is that it seems possible to build human-level AI systems without a radically improved understanding of deep learning; however, building *safe* human-level AI is a harder problem that will likely require a more robust theory. Thus advancing the science of deep learning seems somewhat more critical for solving problems of safety than of capabilities. We note that this argument is speculative, and may turn out to be wrong.

## D Additional Figures

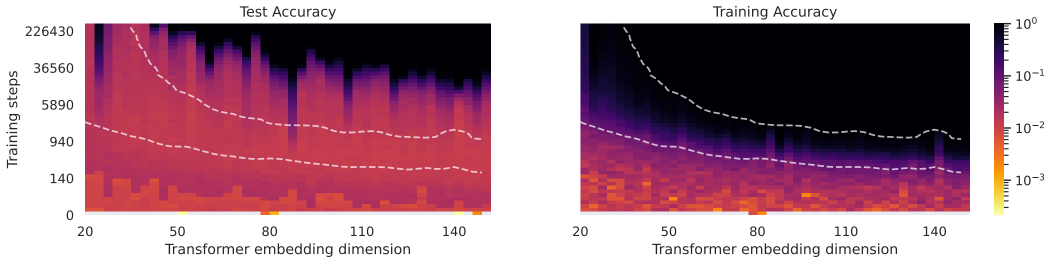

Figure 5: **Left:** Test accuracy as function of training steps and model size. A faint double descent in test accuracy is visible, and delayed generalization is clear both epoch-wise (vertically) and model-wise (horizontally). **Right:** Training accuracy as a function of training steps and model size.

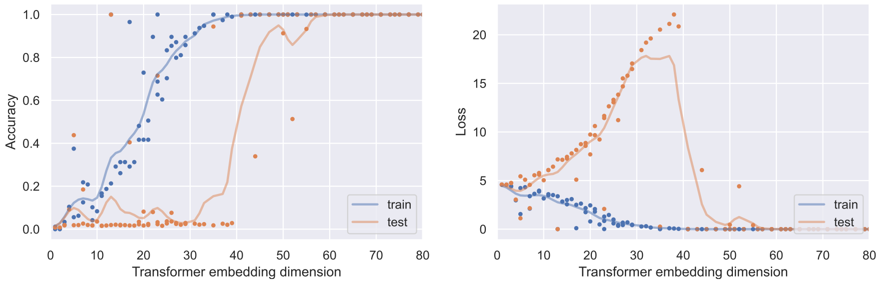

Figure 6: Model-wise grokking, with each point a transformer of the corresponding embedding dimension trained for 400K optimization steps. Lines are smoothed data via a Gaussian filter. **Left:** Accuracy, with delayed generalization. **Right:** Loss, with a rise in test loss followed by a fast descent.

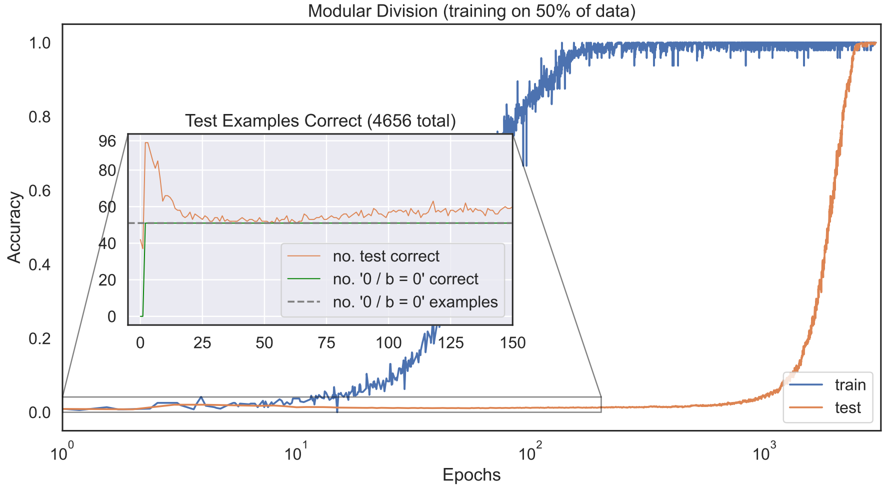

Figure 7: Early in training, the model learns $0/b = 0 \mod n$, as the no. $0/b = 0 \mod n$ correct (green) matches all such examples (dashed gray). This leads to a brief peak of 96 correct examples, corresponding to perfect performance on the $0/b = 0$ and chance performance (1/97) on other data. Test accuracy on all other examples then falls below chance due to memorization.