HiLD 2026: 4th Workshop on High-dimensional Learning Dynamics 1–18

Model Capacity Determines Grokking through Competing Memorisation and Generalisation Speeds

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

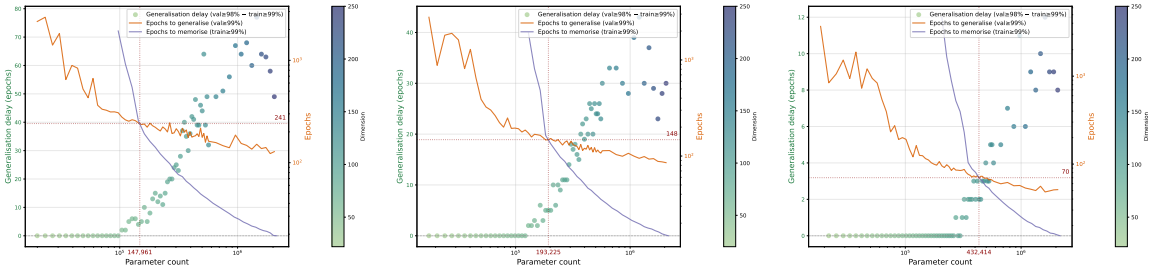
Existing accounts of grokking explain the phenomena in terms of mechanistic frameworks such as circuit efficiency or lazy-to-rich transitions. However, despite a known dependence between grokking and model size, how model capacity shapes grokking remains an open question. We give an information-theoretic account of this relationship on the task of modular arithmetic, showing that grokking does not immediately occur when a model becomes large enough to memorise the training set, but rather emerges as the outcome of a competition between two measurable timescales: a memorisation speed $T_{\text{mem}}(P)$ and a generalisation speed $T_{\text{gen}}(P)$, both of which are functions of model parameter count P. Adapting the information capacity framework of Morris et al. (2025), we estimate $T_{\text{mem}}(P)$ on random-label data of equivalent complexity and $T_{\text{gen}}(P)$ on the modular task itself, and show that grokking emerges close to the parameter scale where these timescales intersect. The framework also suggests an empirical model for predicting memorisation speed given model capacity and dataset complexity, recovering the previously reported empirical observation that larger models memorise faster. Overall, we motivate the formalisation of different learning timescales as important abstractions to study when explaining how model capacity shapes grokking on algorithmic tasks.

1. Introduction

The grokking phenomenon (Power et al., 2022) on modular arithmetic, where training accuracy saturates long before test accuracy, is a popular testbed for memorisation and generalisation in overparameterised networks. Existing accounts of grokking identify why a generalising solution is preferred at convergence (Lyu et al., 2024; Kumar et al., 2024; Varma et al., 2023; Huang et al., 2024; Merrill et al., 2023; Mohamadi et al., 2024; Nanda et al., 2023), but say less about when during training each solution is reached or about the parameter scale at which grokking appears.

Empirically, very small Transformers do not grok at all, generalising immediately (Liu et al., 2022; Huang et al., 2024); only past some larger scale does the delay appear. A natural prediction from bits-per-parameter capacity (Morris et al., 2025) is that grokking should begin as soon as model capacity is large enough to memorise the training set. However, this is not what happens empirically: models well above this threshold continue to generalise immediately. The onset of grokking sits strictly above P_{mem}, which we define as the parameter count at which a memorising solution first becomes representable.

We argue that the missing piece is a quantitative treatment of learning timescales. Adapting the random-label measurement protocol of Morris et al. (2025), we define two



(a) $p = 97$ (b) $p = 113$ (c) $p = 139$

Figure 1: Generalisation delay (scatter, left axis: epochs between training saturation and validation saturation; colour encodes embedding width d) and learning speeds (orange: T_{gen}; purple: T_{mem} on a random dataset of matched complexity) versus parameter count for three primes. The dashed crosshair marks the predicted grokking point $\widehat{P}_{\text{cross}}(p)$, where the two speed curves cross; this coincides closely with the empirical onset of non-zero generalisation delay. For small models, generalisation is faster than memorisation and the model generalises immediately; once the curves cross, delays become non-zero and grokking begins.

measurable speeds for a Transformer of parameter count P: a memorisation speed $T_{\text{mem}}(P)$, the epochs to fit a random-label dataset of equivalent information content, and a generalisation speed $T_{\text{gen}}(P)$, the epochs to reach high validation accuracy on a task. Our central finding (Fig. 1) is that grokking onset coincides closely with the parameter count P_{cross} at which these two speeds intersect, and that $P_{\text{cross}} \gg P_{\text{mem}}$. Capacity sufficient to represent a memorising solution is not sufficient to make memorisation the path that gradient descent selects; the deciding factor is whether memorisation is faster than generalisation.

We observe three regimes as we scale P. **Under-capacity** ($P \ll P_{\text{mem}}$): neither solution is well-represented and both train and test accuracy are low. **Immediate generalisation** ($P_{\text{mem}} \lesssim P < P_{\text{cross}}$): although the memorising solution is representable, $T_{\text{gen}}(P) < T_{\text{mem}}(P)$, so gradient descent reaches the algorithmic solution first and there is no delay. **Grokking** ($P > P_{\text{cross}}$): $T_{\text{mem}}(P) \lesssim T_{\text{gen}}(P)$, so the model memorises first and generalises later.

Contributions. (i) A mechanistic account of how capacity controls grokking, via competing memorisation and generalisation speeds: $\widehat{P}_{\text{cross}}$ predicts the empirical onset across primes (Spearman ρ=0.976) and is a robust statistic across hyperparameters. (ii) The previous observations that small models do not grok (Liu et al., 2022; Huang et al., 2024) and that larger models memorise faster (Tirumala et al., 2022) fall out as separate consequences of our framework (Section C).

2. Related work

Existing accounts explain grokking by identifying mechanisms that select the generalising solution: circuit efficiency under weight decay (Varma et al., 2023; Huang et al., 2024), selective norm growth (Merrill et al., 2023), lazy-to-rich transitions (Kumar et al., 2024; Lyu

et al., 2024), kernel-regime impossibility (Mohamadi et al., 2024), and reverse engineering (Nanda et al., 2023); phase-diagram and complexity-based progress measures appear in (Liu et al., 2022; Huang et al., 2024; Clauw et al., 2024; DeMoss et al., 2025), and Manir and Rupa (2026) isolate optimisation effects. The conceptual ancestor of our framing is Davies et al. (2023), who argue that grokking arises because different patterns are learned at different rates; we make this concrete by measuring two timescales whose crossover is the empirical phase boundary. Optimiser-side interventions (Lee et al., 2024; Thilak et al., 2022) and epoch-wise double descent (Nakkiran et al., 2020) reinforce that training time is a meaningful axis. The random-label memorisation lineage runs (Zhang et al., 2017; Arpit et al., 2017; Carlini et al., 2022) to Morris et al. (2025), whose C_{model} we adopt to characterise speeds rather than the static threshold.

3. Setup and definitions

Modular arithmetic and architecture. Fix a prime p and operation $\circ \in \{+, -, \times, /\}$ on $\mathbb{Z}_p$ (with $a/b \equiv a \cdot b^{-1} \pmod{p}$, $b \neq 0$). Each example is a length-4 token sequence $[a, \texttt{op}, b, \texttt{=}]$ with label $a \circ b$ over vocabulary $V = p + 2$. For modular division there are $p(p-1)$ pairs; we train on a random fraction $\alpha \in (0, 1)$, with $\alpha = 1/2$ throughout the central sweep. Adapting the information-theoretic framework of Morris et al. (2025), the training labels carry

$$K_{\text{mem}}(p, \alpha) = \alpha p(p-1) \log_2(p+2) \tag{1}$$

bits under a uniform-prior code. All experiments use a depth-2, single-head, decoder-only Transformer of width $d \in [10, 1000]$ trained with AdamW (Section A).

Memorisation and capacity. Adapting Morris et al. (2025) to discrete classification, the total memorisation of p_θ on $D = \{(x_i, y_i)\}_{i=1}^n$ is

$$M_T(\theta; D) = \sum_{i=1}^{n} \big(\log_2 V + \log_2 p_\theta(y_i \mid x_i)\big) \in [0,\, n \log_2 V], \tag{2}$$

where each term the code-length reduction relative to a uniform baseline. Training to saturation on random labels yields a capacity curve $M_T(n)$ whose plateau scales linearly with P; the slope C_{model} (bits per parameter) is the per-architecture capacity constant. We measure $C_{\text{model}} \approx 2.16$ for the central sweep, allowing us to define the capacity threshold

$$P_{\text{mem}}(p, \alpha) = K_{\text{mem}}(p, \alpha)/C_{\text{model}} \tag{3}$$

above which memorisation is representable. The capacity measurement itself is not a contribution of this paper (Roberts et al., 2020; Lu et al., 2024; Allen-Zhu and Li, 2024; Morris et al., 2025); we re-run it only to obtain C_{model} per architecture Section B.

Learning speeds. Capacity tells us what is representable; speeds capture what gradient descent actually finds first. We operationalise both as first-passage times to a 99% training-accuracy threshold under standard AdamW training.

Definition 1 (Memorisation speed) *For a model of size P trained on n random-label points, $T_{\text{mem}}(P, n)$ is the first epoch at which training accuracy exceeds 99%. We write*

$T_{\text{mem}}(P) := T_{\text{mem}}(P, n_{\text{equiv}})$ *where* $n_{\text{equiv}} = K_{\text{mem}}(p, \alpha)/\log_2 V$ *matches the modular-task complexity in bits.*

Definition 2 (Generalisation speed) *For a model of size* P *trained on* $\mathcal{D}_p^{\text{train}}$, $T_{\text{gen}}(P)$ *is the first epoch at which validation accuracy on* $\mathcal{D}_p^{\text{test}}$ *exceeds* 99%.

We additionally use the *capacity fraction* $f(P, n) = n \log_2 V/(C_{\text{model}} P)$, the ratio of dataset complexity to model capacity.

Quantifying grokking. The generalisation delay is $\Delta E(P) = \max\{0, E_{\text{val}}(P) - E_{\text{train}}(P)\}$, with E_{train} and E_{val} the first epochs at which training and validation accuracy exceed 99% and 98% respectively.[1] We average the speeds across 10 seeds and take the minimum generalisation delay across seeds, because we are interested in finding the smallest model size for which we consistently observe grokking. The grokking onset $P_{\text{onset}}(p)$ for a prime p is the smallest P for which $\Delta E(P') > 0$ at every measured parameter count $P' \geq P$.

4. Results

4.1. The speed intersection tracks grokking onset

For each prime p we measure $T_{\text{mem}}(P)$ on a random-label dataset of size $n_{\text{equiv}}(p, \alpha)$ and $T_{\text{gen}}(P)$ on the modular-division dataset. Fig. 1 overlays the speeds and the generalisation delay versus parameter count for $p \in \{97, 113, 139\}$. Three patterns are consistent across primes: at small P, $T_{\text{gen}}(P) < T_{\text{mem}}(P)$ and delays are zero; as P grows, $T_{\text{mem}}(P)$ decreases more steeply than $T_{\text{gen}}(P)$; the two cross at a characteristic $P_{\text{cross}}(p)$ that lies strictly above the capacity threshold $P_{\text{mem}}(p, \alpha)$, and this crossing coincides closely with the onset of non-zero $\Delta E(P)$. For several primes (including $p = 113, 139$) the smallest models attaining near-perfect *train and test* accuracy have $P \lesssim P_{\text{mem}}$, indicating that gradient descent finds an algorithmic solution more compact than the raw lookup.

Quantitative tests. We decompose our hypothesis into three falsifiable sub-claims: predictiveness, calibration to $y{=}x$, and sufficiency against baselines, evaluated on grokking onset over 11 primes from 97 to 150 (Table 1; definitions in Section D). The rank order is essentially perfect ($\rho{=}0.976$), the log–log slope is $\hat{b}{=}0.985$ (no proportional bias), and dropout adds no explanatory power beyond $\widehat{P}_{\text{cross}}$ (M_3 vs. M_1, $F{\approx}0$). The only residual is a small constant offset, where empirical onset arrives roughly 30% earlier than predicted (median log-residual -0.16), so the intersection captures the scaling of grokking onset with prime p up to a single multiplicative correction.

4.2. Robustness across hyperparameters, architectures, and tasks

We test the framework's robustness against changes in weight-decay (Section E.1), learning rate (Section E.4), initialisation scale (Section E.3), training fraction (Section E.2), depth scaling (Section F.1), and operation choice (Section G.1). Across all six, our framework

1. We picked a slightly lower threshold for validation accuracy to remove noise from empirical data. When generalisation delay is close to zero, we observed many runs where models reached 99% train accuracy and $> 98\%$ validation accuracy, but subsequently saw a large number of epochs elapse before validation accuracy matched train accuracy.

Table 1: Tests of the intersection hypothesis on 11 primes in $[90, 150]$, $\circ = /$, $\alpha = 0.5$, in $\log_{10}$ onset coordinates. p_{perm}: 10^4-shuffle permutation. Nested OLS: M_0 intercept-only, M_1=$\{\log_{10} \widehat{P}_{\text{cross}}\}$, M_2=$\{\text{dropout}\}$, M_3=$M_1 \cup M_2$. See Section D.

Sub-claim / Test	Statistic	p-value
(1) Rank predictiveness		
Spearman ρ	0.976	$p_{\text{perm}} < 10^{-3}$
Kendall τ	0.911	3.0×10^{-5}
(2) Calibration to $y = x$		
Lin's CCC (95% CI)	0.74 [0.51, 0.80]	—
Slope $\hat{b}$ (log–log OLS)	0.985	—
Intercept $\hat{a}$	-0.076	—
Joint F-test for $(a = 0, b = 1)$	$F = 199$	1.5×10^{-7}
Wilcoxon on log-residuals	median -0.16	2.0×10^{-3}
(3) Sufficiency vs. baselines (nested OLS)		
M_1 vs. M_0	$F = 569$	1.0×10^{-8}
M_3 vs. M_2	$F = 498$	9.2×10^{-8}
M_3 vs. M_1	$F \approx 0$	1.0

holds within any one consistent setting of hyperparameter, architecture, and task; analysing how the framework calibrates across settings is a separate question we leave for future work.

5. Discussion

Capacity influces grokking through pattern learning speeds. A natural application of information-theoretic frameworks (Morris et al., 2025) to grokking is a threshold-based prediction: grokking begins once $C_{\text{model}}P \gtrsim K_{\text{mem}}$. Our data say otherwise on modular arithmetic: over an extended range of parameter counts where $C_{\text{model}}P \gtrsim K_{\text{mem}}$ the models continue to generalise immediately, with no generalisation delay. The transition into the grokking regime instead coincides with the parameter count at which the measured $T_{\text{mem}}(P)$ and $T_{\text{gen}}(P)$ curves cross — a count strictly larger than P_{mem}. What controls whether a model groks is not whether the lookup is representable, but the relative speeds of memorisation and generalisation as a function of P.

Connection to prior accounts and limitations. Mechanistic accounts (Varma et al., 2023; Merrill et al., 2023; Huang et al., 2024; Kumar et al., 2024) identify which solution is preferred at convergence; we add a quantitative account of which solution gradient descent encounters first as a function of P. The "small models do not grok" regime (Liu et al., 2022; Huang et al., 2024) falls out as the $T_{\text{gen}} < T_{\text{mem}}$ region; the "larger models memorise faster" regularity (Tirumala et al., 2022) is recovered because T_{mem} depends primarily on f (Section C). The framework also formalises the speed intuition of Davies et al. (2023) into two separately measurable timescales. However, note that $T_{\text{gen}}(P)$ is measured on actual task data rather than analytically derived and that both speeds vary with hyperparameter settings, so each new configuration requires re-running the speed measurements.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*, 2024.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arpit17a.html.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Kenzo Clauw, Sebastiano Stramaglia, and Daniele Marinazzo. Information-theoretic progress measures reveal grokking is an emergent phase transition. *arXiv preprint arXiv:2408.08944*, 2024. doi: 10.48550/arXiv.2408.08944. URL https://arxiv.org/abs/2408.08944.

Xander Davies, Lauro Langosco, and David Krueger. Unifying grokking and double descent. *arXiv preprint arXiv:2303.06173*, 2023.

Branton DeMoss, Silvia Sapora, Jakob Foerster, Nick Hawes, and Ingmar Posner. The complexity dynamics of grokking. *arXiv preprint arXiv:2412.09810*, 2025. URL https://arxiv.org/abs/2412.09810.

Yufei Huang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. Unified view of grokking, double descent and emergent abilities: A perspective from circuits competition. *arXiv preprint arXiv:2402.15175*, 2024. doi: 10.48550/arXiv.2402.15175. URL https://arxiv.org/abs/2402.15175.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06110. arXiv:2310.06110.

Jaerin Lee, Bong Gyun Kang, Kihoon Kim, and Kyoung Mu Lee. Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*, 2024. doi: 10.48550/arXiv.2405.20233. URL https://arxiv.org/abs/2405.20233.

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

Xingyu Lu, Xiaonan Li, Qinyuan Cheng, Kai Ding, Xuan-Jing Huang, and Xipeng Qiu. Scaling laws for fact memorization of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11263–11282, 2024.

Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon S. Du, Jason D. Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2311.18817. arXiv:2311.18817.

Shalima Binta Manir and Anamika Paul Rupa. A systematic empirical study of grokking: Depth, architecture, activation, and regularization. *arXiv preprint arXiv:2603.25009*, 2026. URL https://arxiv.org/abs/2603.25009.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *arXiv preprint arXiv:2303.11873*, 2023. ICLR Workshop on Mathematical and Empirical Understanding of Foundation Models.

Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. Why do you grok? A theoretical analysis on grokking modular addition. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 35934–35967. PMLR, 2024.

John X. Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize? *arXiv preprint arXiv:2505.24832*, 2025.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.02292. arXiv:1912.02292.

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

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, Vedant Misra, Aaron Mishkin, Johannes Kramer, Joar Skalse, Marcin Andrychowicz, Ilya Sutskever, et al. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*, 2020.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, and Roni Paiss. The slingshot mechanism. *arXiv preprint arXiv:2206.04817*, 2022. doi: 10.48550/arXiv.2206.04817. URL https://arxiv.org/abs/2206.04817.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

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

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1611.03530.

Appendix A. Experiment details

Primes. The central sweep uses 10 primes in $[90, 140]$: {97, 101, 103, 107, 109, 113, 127, 131, 137, 139}; sweeps in Sections E to G use five primes drawn from this set unless otherwise stated.

Architecture and optimisation. Decoder-only Transformer in PyTorch with depth $L_{\text{depth}} = 2$, a single attention head, RoPE on queries and keys, gated FFN of hidden width $4d$, RMSNorm in both sub-blocks, final RMSNorm, and a linear unembedding to $\mathbb{R}^V$. Optimiser is AdamW with learning rate 10^{-3}, betas $(0.9, 0.98)$, weight decay 1.0, batch size 512, for up to 5,000 epochs. Dropout is 0.2 in grokking and speed runs and 0 in capacity runs.

Compute. NVIDIA A100 / H100, single GPU per training run, dispatched in parallel via a YAML-configured job scheduler (`gc-dispatch`). The full set of capacity, speed, and grokking experiments required on the order of 10^3 GPU-hours; preliminary and discarded sweeps used a comparable additional amount.

Capacity runs. For each $d \in \{10, 12, 14, 16, 18, 20, 22\}$ and n from a geometric grid of 8 values between 10^3 and 10^4 (specifically $\{1000, 1374, 1891, 2601, 3576, 4917, 6761, 9300\}$) we sample a random dataset uniformly over the vocabulary $V = p + 2$ at $p = 113$ and train as in Section A with dropout 0, stopping when training loss fails to improve by more than $\Delta = 10^{-4}$ for a patience window of 100 epochs. We report M_T via Eq. (2). Within-arch capacity curves are stable across seeds, so we use a single seed (42) per cell.

Speed runs. For each prime p and each $d \in \{20, 24, 28, \ldots, 128\} \cup \{136, 144, \ldots, 256\}$ we generate a random-label dataset of size $n_{\text{equiv}}(p, \alpha)$ and train as in Section A (dropout 0.2, weight decay 1.0). We declare saturation once training accuracy exceeds 99% and report $T_{\text{mem}}(P)$ as the seed-mean over 10 seeds (42–51).

Grokking runs. For each prime p and each $d \in \{20, 22, \ldots, 128\} \cup \{130, 140, \ldots, 1000\}$ (truncated to $d \leq 256$ for per-prime intersection figures via `max_dim`) we construct a modular-division dataset with $\alpha = 0.5$ and a random train/test split, training as in Section A. We use 10 seeds per cell (42–51); $T_{\text{gen}}(P)$ is the seed-mean and $\Delta E(P)$ the per-seed minimum.

Sweeps beyond the central setting. Each sweep reuses the architecture, optimiser, and aggregation conventions above; only the swept axis changes and any data-dependent quantities (e.g. K_{mem}, n_{equiv}) are recomputed. For the depth sweep, d is selected per cell to hit fixed parameter-count targets. For the training-fraction sweep, n_{equiv} is recomputed per α while the architecture and bits-per-parameter constant inherited from the matched arch group are held fixed.

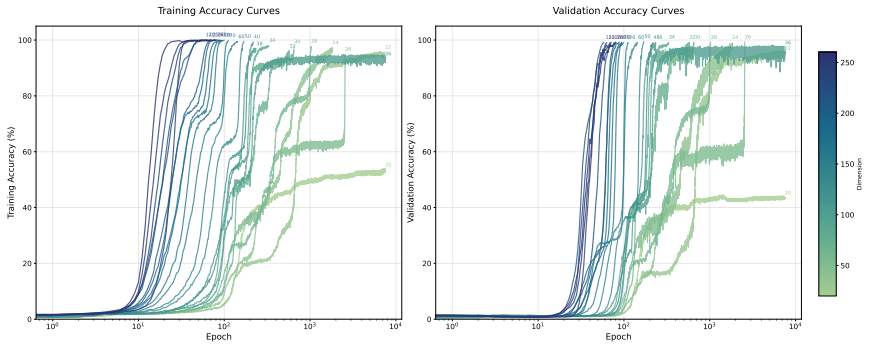
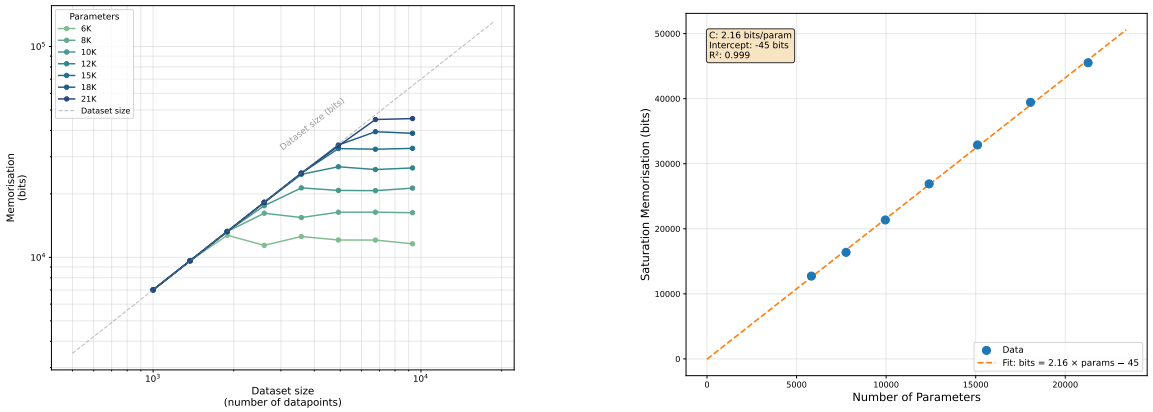


Figure 2: Training (left) and validation (right) accuracy for modular division at $p = 127$ across model sizes. Small models underfit, intermediate models generalise immediately, larger models grok.



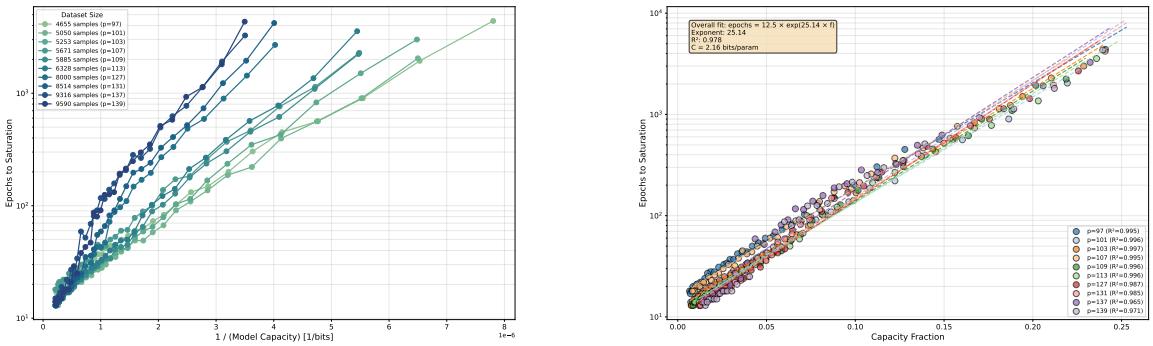
(*a*) M_T vs. dataset complexity.

(*b*) $\widehat{\text{Cap}}$ vs. parameter count.

Figure 3: Random-label capacity experiments following Morris et al. (2025). Left: $M_T(n)$ on random-label datasets at several model sizes; dashed grey line is the dataset complexity in bits. Right: highest M_T achieved vs. parameter count; the slope of the linear fit is $C_{\text{model}} \approx 2.16$.

Appendix B. Information capacity

We re-run the capacity protocol of Morris et al. (2025) only to obtain the per-architecture C_{model} used in Eq. (3) and to define the capacity fraction f used in Section C. Fig. 3 shows $M_T(n)$ on random data: for small n the curves grow linearly with slope close to $\log_2 V$; for larger n they saturate at plateaus that scale linearly with P (right panel). Fitting $\widehat{\text{Cap}}_k \approx C_{\text{model}} P_k + b$ yields $C_{\text{model}} \approx 2.16$ with a small intercept and high R^2, matching prior reports (Roberts et al., 2020; Lu et al., 2024; Allen-Zhu and Li, 2024; Morris et al., 2025). We treat this measurement as background; it is not a contribution of this paper.



(*a*) Saturation time vs. $1/(C_{\text{model}}P)$.

(*b*) Saturation time vs. capacity fraction f.

Figure 4: Memorisation-speed measurements across primes. Plotted against inverse model capacity (left), datasets give different curves; plotted against capacity fraction (right), all datasets collapse onto a single trend, supporting the empirical regularity that T_{mem} depends primarily on f.

Appendix C. Memorisation speeds

Fig. 4 (left) shows that memorisation time increases with inverse model capacity $1/(C_{\text{model}}P)$: smaller models take longer to memorise. The right panel re-plots the same data against the capacity fraction f, and points across primes collapse roughly onto a single curve, suggesting that T_{mem} is determined by f rather than by P or K separately. We empirically fit $T_{\text{mem}}(P) \approx b\, e^{af}$ for $f \in [0, 0.25]$. The exponential is a functional fit, not a mechanistic claim; combined with Eq. (1), it recasts the "larger-models-memorise-faster" observation of Tirumala et al. (2022) as a near-linear scaling in capacity fraction (a fixed dataset complexity occupies a smaller fraction of a larger model's capacity).

Appendix D. Quantitative tests of the intersection hypothesis

We turn the qualitative claim "$\widehat{P}_{\text{cross}}(p)$ predicts $P_{\text{onset}}(p)$" into three falsifiable sub-claims and report a test for each. All tests operate on $\{(\widehat{P}_{\text{cross}}(p),\, P_{\text{onset}}(p))\}_p$ in $\log_{10}$ coordinates.

Predictiveness. We test for any monotone relationship via Spearman's ρ (with both analytic and 10^4-shuffle permutation p-values) and Kendall's τ.

Calibration. Rank correlation alone allows a strongly correlated predictor with the wrong slope or a constant offset to pass. We add three statistics: Lin's concordance correlation coefficient $\text{CCC} = 2\rho\sigma_p\sigma_e/(\sigma_p^2 + \sigma_e^2 + (\mu_p - \mu_e)^2)$ (95% CI from 10^4 cell-level bootstrap resamples), an OLS fit $\log_{10} P_{\text{onset}} = a + b \log_{10} \widehat{P}_{\text{cross}}$ with a joint F-test for $(a = 0, b = 1)$, and a Wilcoxon signed-rank test on the log-residuals against zero. Reporting $(\hat{a}, \hat{b})$ separately makes the *direction* of any miss legible: a slope < 1 means the predictor over-extrapolates large onsets, an intercept > 0 means it systematically under-predicts.

Sufficiency. The strongest version of the claim is that the intersection is a sufficient statistic for the empirical onset, in the sense that no simpler predictor does as well. We use nested OLS comparisons against $\log_{10} P_{\text{onset}}$:

M_0 **(null):** intercept only.

M_1 **(intersection):** $\log_{10} \widehat{P}_{\text{cross}}$.

M_2 **(hyperparams):** every column in the config's swept axes.

M_3 **(combined):** $M_1 \cup M_2$ predictors.

M_1 vs. M_0 confirms the intersection has any signal. M_3 vs. M_2 confirms it adds information *over and above* the swept hyperparameters. M_3 vs. M_1 is the sufficiency test proper: do the hyperparameters add anything beyond the intersection?

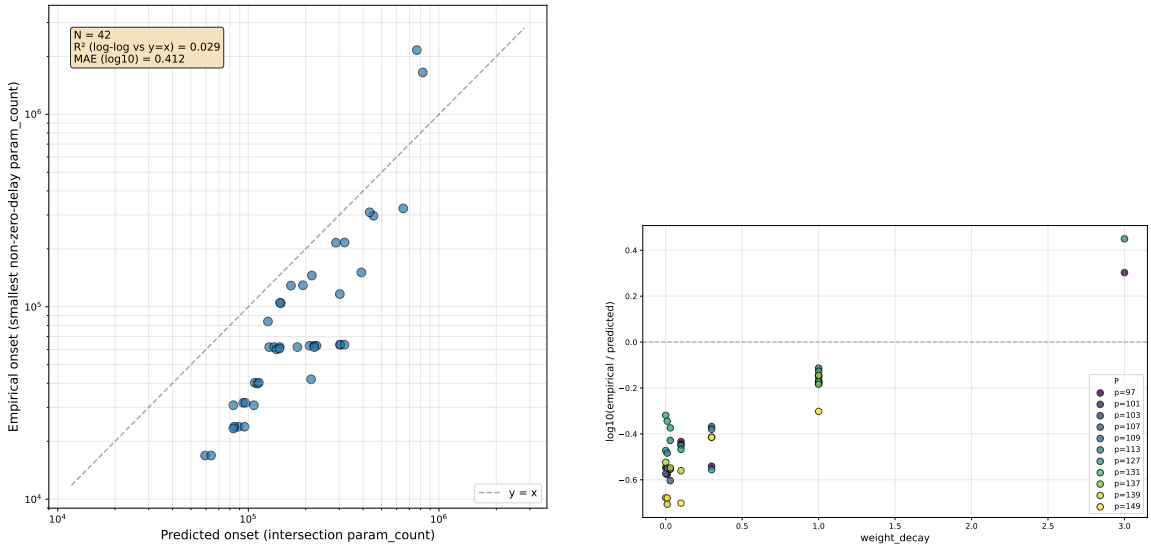
Robustness. For configurations with multiple swept axes we additionally regress the per-cell log-residual against each axis and Holm-correct p-values across axes, flagging systematic mispredictions along any axis.

What this is not. With ~ 10 cells per figure only large effects are detectable; the point is to prevent overclaiming, not to certify the predictor as perfect. The setup is not cross-validated — with this N, k-fold leaves 5–6 training points per fold and variance dominates bias, so the nested-model F-tests use in-sample fits and report adjusted R^2. Per-seed variability is folded into the seed-min onset estimate via the aggregation procedure of Section 4 rather than modelled explicitly.

Central-sweep numerics. On the central sweep, RSS values give in-sample / adjusted R^2 of $M_0 = 0$, $M_1 = 0.986/0.984$, $M_2 = 0/-0.125$, $M_3 = 0.986/0.982$. The corresponding F-tests in Table 1 reject M_0 in favour of M_1 ($F = 569$, $p = 10^{-8}$) and M_2 in favour of M_3 ($F = 498$, $p = 9 \times 10^{-8}$); the upgrade from M_1 to M_3 is null ($F \approx 0$, $p = 1$). The OLS slope is $\hat{b} = 0.985$ and the intercept $\hat{a} = -0.076$. The Wilcoxon median log-residual is -0.16 ($p = 2 \times 10^{-3}$), consistent with the OLS residual model evaluated at typical onsets ($\log_{10} \widehat{P}_{\text{cross}} \approx 5$ gives a per-cell offset ≈ -0.15): a single ~ 0.16 dex constant offset rather than a structural miscalibration.

Appendix E. Hyperparameter invariance

Reading guide. The central sweep fixes every hyperparameter except p and d. To assess robustness we re-run the full pipeline at each setting of a swept axis and apply the framework of Section D. Each subsection below uses a fixed template (*scope*, headline figures, *pooled tests* table, *within-setting* log-residual table, *verdict*) so that further axes slot in without restructuring. The framework's central claim — that within a fixed setting the parameter count at which T_{mem} and T_{gen} cross predicts P_{onset} — does not require a single calibration constant to apply across settings; we therefore evaluate each sweep on *within-setting predictiveness* and on *cross-setting calibration*.



(*a*) Predicted vs. empirical onset. (*b*) Per-cell log-residual vs. λ.

Figure 5: Weight-decay sweep.

Table 2: Weight-decay sweep: pooled hypothesis tests on the 42 valid cells (out of 47).

Sub-claim / Test	Statistic	p-value
Spearman ρ	0.900	$p_{\text{perm}} = 10^{-4}$
Kendall τ	0.774	9.8×10^{-13}
Lin's CCC (95% CI)	0.531 [0.35, 0.65]	—
Slope $\hat{b}$ / Intercept $\hat{a}$	1.439 / -2.683	—
Joint F-test for $(a = 0, b = 1)$	$F = 72.4$	5.1×10^{-14}
Wilcoxon on log-residuals	median -0.431	2.0×10^{-9}
M_1 vs. M_0	$F = 150$	4.0×10^{-15}
M_3 vs. M_2	$F = 167$	1.8×10^{-15}
M_3 vs. M_1	$F = 92.0$	2.7×10^{-15}
λ (numeric, residual slope)	$+0.321$/unit	1.9×10^{-18}

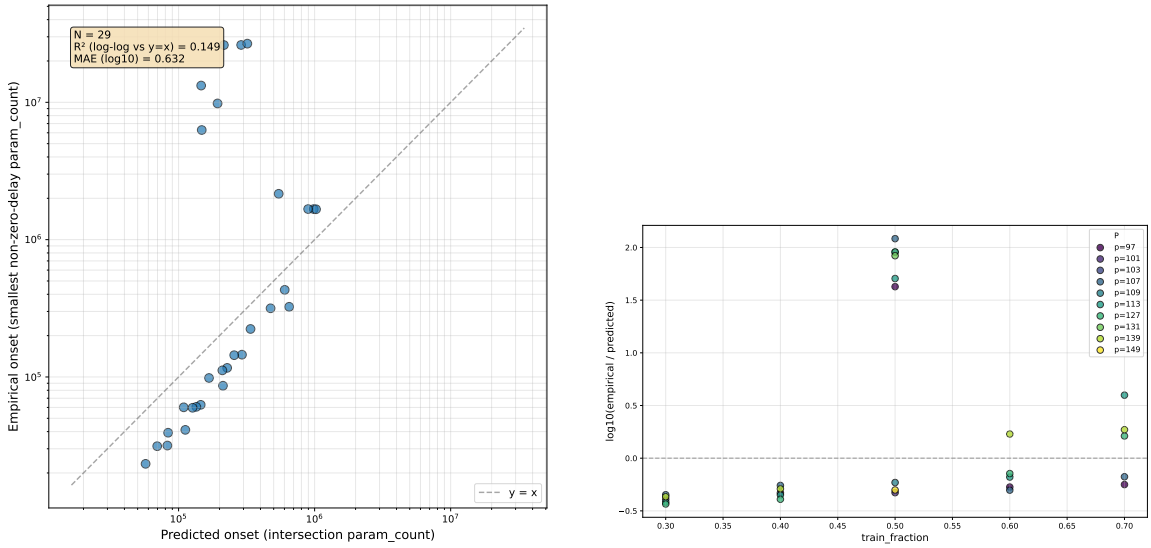
E.1. Weight decay

Scope. $\lambda \in \{0, 0.01, 0.03, 0.1, 0.3, 1.0, 3.0\}$, matched between speed and grokking runs. Six primes per cell ($p \in \{97, 107, 113, 127, 139, 149\}$), with the central-sweep primes also present at λ=1.0; 5 seeds per cell except the λ=1.0 baseline (central-sweep seed pool). Capacity is re-measured at $\lambda \in \{0.01, 0.1, 1.0\}$ and pinned at $C_{\text{model}} = 2.16$ otherwise.

Verdict. Within-setting predictiveness is best at λ=1.0 and degrades smoothly as λ moves away. The per-cell offset becomes more negative at small λ and turns positive at λ=3.0 ($+0.32$ dex per unit λ, p=10^{-18}). Pooled rank predictiveness remains high (ρ=0.900); λ does add information beyond the intersection (M_3 vs. M_1, F=92), consistent with weight decay reshaping the speed curves rather than simply translating them. The qualitative claim holds at every λ where both curves are well-resolved.

Table 3: Weight-decay sweep: within-setting log-residual summary.

λ	n_p	median $\log_{10}(P_{\text{onset}}/\widehat{P}_{\text{cross}})$	std	verdict
0.0	6	-0.535	0.119	weak
0.01	6	-0.564	0.131	weak
0.03	5	-0.548	0.096	moderate
0.1	6	-0.458	0.105	moderate
0.3	6	-0.415	0.082	moderate
1.0	11	-0.171	0.049	strong
3.0	2	$+0.376$	0.105	under-resolved



(*a*) Predicted vs. empirical onset. (*b*) Per-cell log-residual vs. α.

Figure 6: Training-fraction sweep.

E.2. Training fraction

Scope. $\alpha \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$, applied to the modular-division dataset; $n_{\text{equiv}}(p, \alpha)$ recomputed per α. Five primes per cell, 4 seeds per cell except $\alpha = 0.5$ (11 primes / central-sweep seed pool). C_{model} is reused from the matched arch group (λ=1.0, dropout= 0.2).

Verdict. Within each fixed α the predictor ranks the primes correctly: per-cell log-residuals are tight at $\alpha \in \{0.3, 0.4, 0.5\}$ ($\sigma_{\log} \leq 0.05$). The residual spread widens at $\alpha \in \{0.6, 0.7\}$, partly real and partly a measurement artefact (at α=0.7, p=113 the empirical onset lands at the d=256 cap of the swept width range). Pooled, predictiveness is essentially perfect (ρ=0.977); calibration shifts smoothly with α (+1.23 dex/unit) and the M_3 vs. M_1 test is borderline (p=0.044). Same pattern as weight decay: within α tight, across α a smooth recalibration.

Table 4: Training-fraction sweep: pooled hypothesis tests on all 31 cells (5 primes × 4 train-fractions plus the 11-prime α=0.5 central-sweep row).

Sub-claim / Test	Statistic	p-value
Spearman ρ	0.977	$p_{\text{perm}} < 10^{-4}$
Kendall τ	0.890	2.2×10^{-12}
Lin's CCC (95% CI)	0.804 [0.70, 0.86]	—
Slope $\hat{b}$ / Intercept $\hat{a}$	1.452 / −2.605	—
Joint F-test for $(a = 0, b = 1)$	$F = 29.95$	8.8×10^{-8}
Wilcoxon on log-residuals	median −0.179	4.9×10^{-4}
M_1 vs. M_0	$F = 263.9$	4.4×10^{-16}
M_3 vs. M_2	$F = 79.6$	1.1×10^{-9}
M_3 vs. M_1	$F = 4.45$	0.044
α (numeric, residual slope)	+1.23/unit	9.4×10^{-6}

Table 5: Training-fraction sweep: within-setting log-residual summary.

α	n_p	median $\log_{10}(P_{\text{onset}}/\widehat{P}_{\text{cross}})$	std	verdict
0.3	5	−0.390	0.036	strong
0.4	5	−0.329	0.051	strong
0.5	11	−0.171	0.049	strong
0.6	5	−0.179	0.214	moderate
0.7	5	+0.211	0.348	moderate

E.3. Initialisation scale

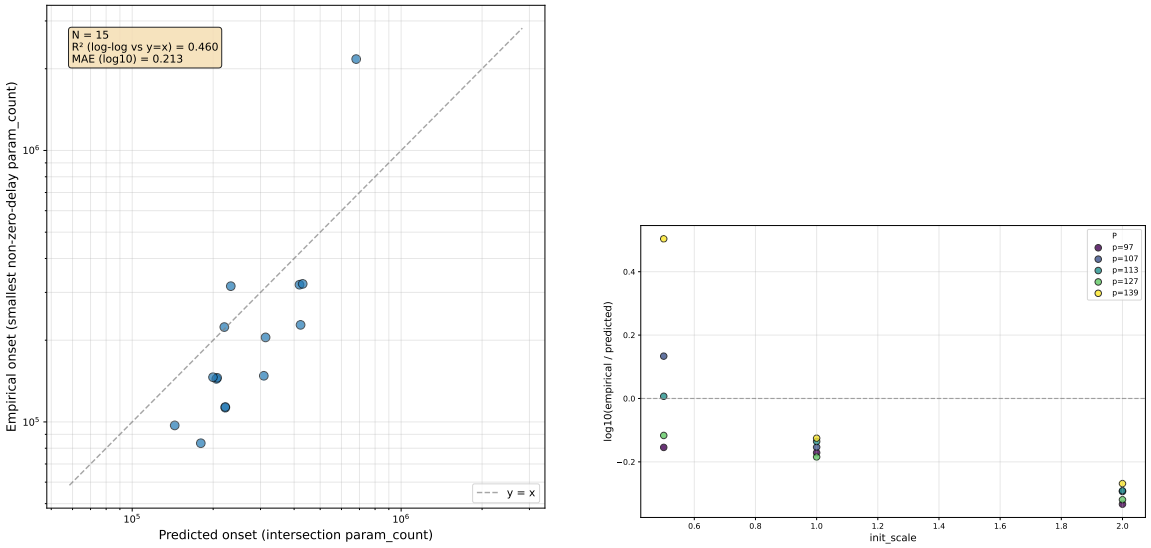
Scope. `init_scale` $\in \{0.5, 1.0, 2.0\}$ as a post-init multiplicative scaling of all weights; 5 primes per cell, 10 seeds per cell.

Verdict. For `init_scale`$\in \{1.0, 2.0\}$, log-residuals are tight ($\sigma_{\log} \leq 0.03$); at `init_scale`$=$ 0.5 the residual variance widens (driven by p=139 in the small-init / under-trained corner). Across the sweep, predictiveness is preserved (ρ=0.839) and the calibration constant trends monotonically (-0.24 dex/unit) — a smooth setting-specific recalibration rather than a structural failure.

E.4. Learning rate

Scope. $\eta \in \{10^{-4}, 3{\times}10^{-4}, 10^{-3}, 3{\times}10^{-3}, 10^{-2}\}$, matched between speed and grokking runs. The two largest η do not produce grokking within the explored width range ($d \leq 256$); pooled tests operate on the 15 cells from the lower three.

Verdict. Across the three η that produce grokking, within-setting predictiveness is preserved ($\sigma_{\log} \leq 0.08$), the predictor ranks the five primes correctly at every fixed η, and the calibration constant is essentially flat ($p_\eta = 0.62$). Sufficiency passes: given $\widehat{P}_{\text{cross}}$, η adds no further explanatory power. At η=3×10^{-3} no model in the swept width range groks within the epoch budget; at η=10^{-2} training never saturates — both regimes lie outside the optimisation window where the speed measurements themselves are well-defined.



(*a*) Predicted vs. empirical onset. (*b*) Per-cell log-residual vs. `init_scale`.

Figure 7: Initialisation-scale sweep.

Table 6: Initialisation-scale sweep: pooled hypothesis tests (15 cells).

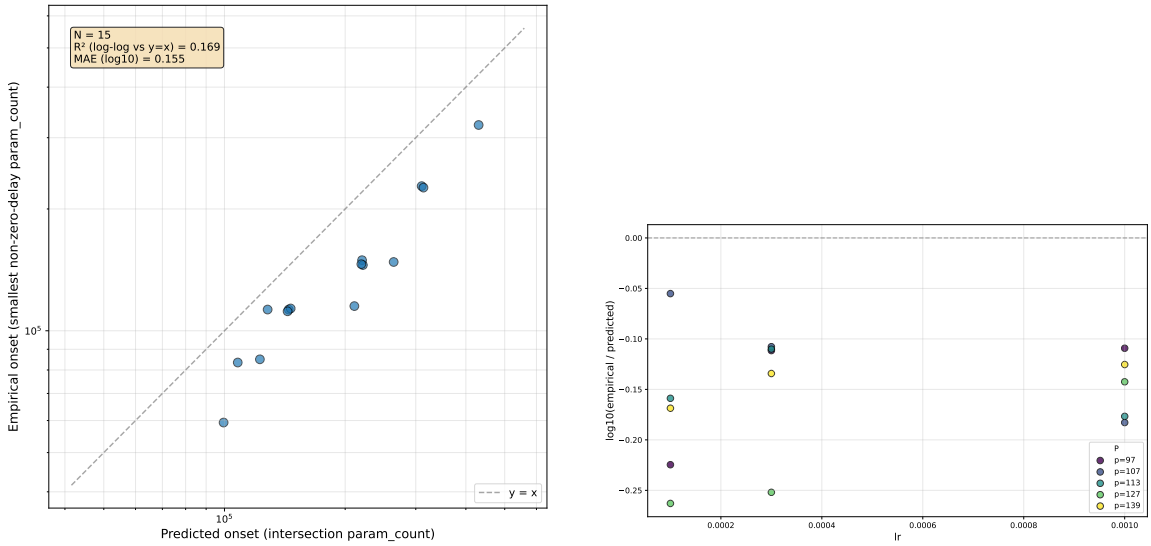
Sub-claim / Test	Statistic	p-value
Spearman ρ	0.839	$p_{\text{perm}} = 10^{-4}$
Kendall τ	0.676	2.0×10^{-4}
Lin's CCC (95% CI)	0.622 [0.28, 0.71]	—
Slope $\hat{b}$ / Intercept $\hat{a}$	1.583 / −3.290	—
Joint F-test for $(a = 0, b = 1)$	$F = 5.29$	0.021
Wilcoxon on log-residuals	median −0.154	0.022
M_1 vs. M_0	$F = 30.7$	9.5×10^{-5}
M_3 vs. M_2	$F = 52.0$	1.1×10^{-5}
M_3 vs. M_1	$F = 14.3$	2.6×10^{-3}
`init_scale` (residual slope)	−0.236/unit	0.0031

Appendix F. Architectural invariance

F.1. Depth scaling

Scope. $L_{\text{depth}} \in \{2, 4, 6, 8, 10\}$ at fixed H=1, matched by parameter count rather than d: target counts $\{5{\times}10^3, 10^4, 2{\times}10^4, 5{\times}10^4, 10^5, 2{\times}10^5, 5{\times}10^5\}$ realised by selecting per depth the dim closest to each target. Five primes per cell, 4 seeds per cell. Capacity re-measured at $L_{\text{depth}} \in \{2, 6, 10\}$, pinned at $C_{\text{model}} = 2.16$ otherwise.

Verdict. At L_{depth}=2 the predictor reproduces its baseline behaviour ($\sigma_{\log} = 0.024$). As depth increases, the constant offset becomes more negative (-0.061 dex/unit, p=0.012), and deeper models ($L_{\text{depth}} \geq 6$) routinely fail to exhibit a speed crossover within the param-count range we sweep ($\leq 5{\times}10^5$): $T_{\text{mem}}(P)$ stays above $T_{\text{gen}}(P)$ throughout, so $\widehat{P}_{\text{cross}}$ is



(*a*) Predicted vs. empirical onset. (*b*) Per-cell log-residual vs. η.

Figure 8: Learning-rate sweep.

Table 7: Learning-rate sweep: pooled hypothesis tests (15 cells).

Sub-claim / Test	Statistic	p-value
Spearman ρ	0.972	$p_{\text{perm}} = 10^{-4}$
Kendall τ	0.900	3.2×10^{-6}
Lin's CCC (95% CI)	0.699 [0.45, 0.81]	—
Slope $\hat{b}$ / Intercept $\hat{a}$	0.942 / 0.150	—
Joint F-test for $(a = 0, b = 1)$	$F = 51.8$	6.4×10^{-7}
Wilcoxon on log-residuals	median -0.142	6.1×10^{-5}
M_1 vs. M_0	$F = 127$	4.4×10^{-8}
M_3 vs. M_2	$F = 86.3$	7.9×10^{-7}
M_3 vs. M_1	$F = 0.98$	0.342
η (residual slope)	$+20.4$/unit	0.62

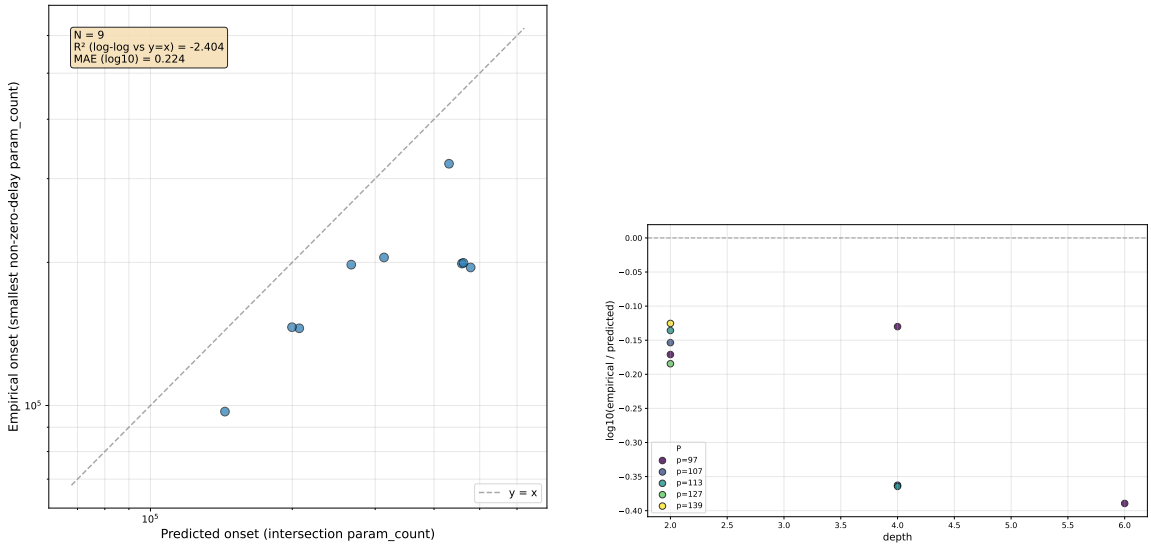
undefined while empirical onsets still occur. Extending the param-count range to recover well-defined intersections at $L_{\text{depth}} \geq 6$ is left for future work.

Appendix G. Task invariance

G.1. Modular addition

Scope. $\circ \in \{+, /\}$ at the same five primes; for $\circ = +$ the dataset enumerates all p^2 pairs (rather than $p(p-1)$ for $/$), so K_{mem} and n_{equiv} are recomputed accordingly. The $\circ = /$ cells reuse the central-sweep speed and grokking runs.

Verdict. Within each operation, log-residuals are tight ($\sigma_{\log} \leq 0.04$): the predictor ranks the five primes correctly and incurs only a constant multiplicative offset within each task.



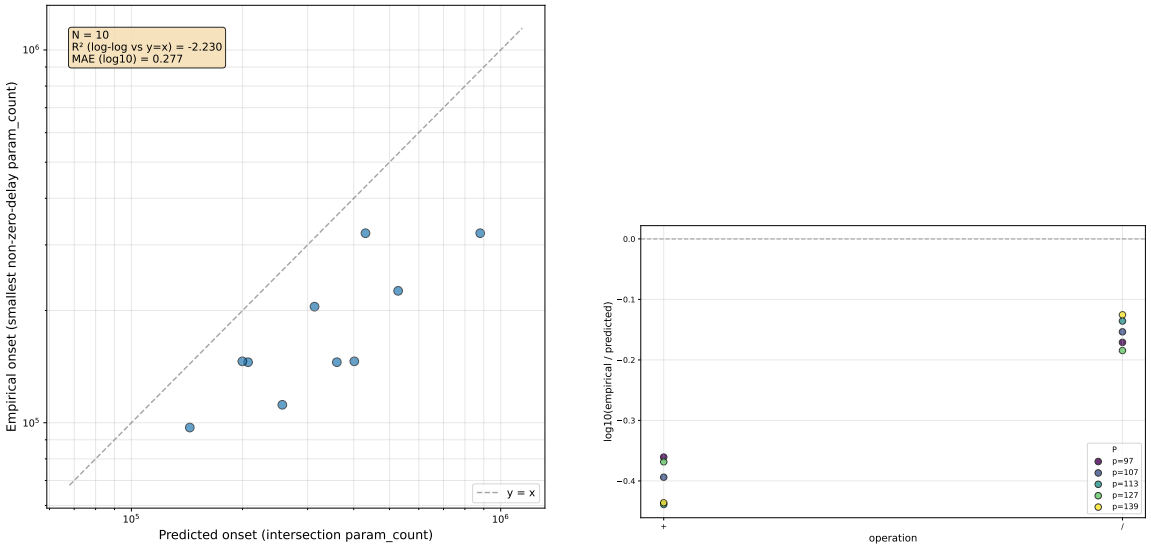
(*a*) Predicted vs. empirical onset. (*b*) Per-cell log-residual vs. L_{depth}.

Figure 9: Depth-scaling sweep.

Table 8: Depth-scaling sweep: pooled hypothesis tests (9 valid cells out of 25; 16 cells at $L_{\text{depth}} \geq 4$ have no recorded intersection within the swept param-count range).

Sub-claim / Test	Statistic	p-value
Spearman ρ	0.600	$p_{\text{perm}} = 0.10$
Kendall τ	0.444	0.119
Lin's CCC (95% CI)	0.392 [0.07, 0.64]	—
Slope $\hat{b}$ / Intercept $\hat{a}$	0.603 / 1.955	—
Joint F-test for $(a = 0, b = 1)$	$F = 31.6$	3.1×10^{-4}
Wilcoxon on log-residuals	median -0.171	3.9×10^{-3}
M_1 vs. M_0	$F = 13.7$	7.7×10^{-3}
M_3 vs. M_2	$F = 21.8$	3.4×10^{-3}
M_3 vs. M_1	$F = 3.93$	0.095
depth (residual slope)	-0.061/unit	0.012

Across operations, the predictor over-shoots empirical onset by $\approx 2.5\times$ for $+$ versus $\approx 1.4\times$ for $/$. We interpret this as expected task-specific calibration — T_{mem} on random labels of equivalent complexity is operation-agnostic by construction, but T_{gen} for $+$ is consistently faster than for $/$ — not a within-task failure of the framework.



(*a*) Predicted vs. empirical onset. (*b*) Per-cell log-residual vs. operation.

Figure 10: Task-addition sweep.

Table 9: Task-addition sweep: pooled hypothesis tests (10 cells).

Sub-claim / Test	Statistic	p-value
Spearman ρ	0.801	$p_{\text{perm}} = 7.5 \times 10^{-3}$
Kendall τ	0.644	0.011
Lin's CCC (95% CI)	0.395 [0.12, 0.58]	—
Slope $\hat{b}$ / Intercept $\hat{a}$	0.634 / 1.740	—
Joint F-test for $(a = 0, b = 1)$	$F = 35.0$	1.1×10^{-4}
Wilcoxon on log-residuals	median -0.273	2.0×10^{-3}
M_1 vs. M_0	$F = 16.1$	3.9×10^{-3}
M_3 vs. M_2	$F = 263$	8.3×10^{-7}
M_3 vs. M_1	$F = 82.8$	4.0×10^{-5}
operation (categorical)	$F = 155$	1.6×10^{-6}