SCALING WIDTH AND SCALING COMPUTE TEACH DIFFERENT ALGORITHMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adding parameters and adding optimisation steps are both called scaling, and they push a network toward different algorithms. We show this on two tasks that share no structure, each solvable either by a cheap shortcut or by a general rule, over a 4×4 grid of width and step budget at fixed batch size and 64 seeds per cell. Compute favours the rule: averaged over widths, the fraction of runs that acquire it climbs monotonically with the budget in every width column of both tasks, from 0.030 to 0.515 on one and from 0.001 to 0.264 on the other. Width favours the shortcut once past a small optimum, falling $0.686 \to 0.520 \to 0.257$ between d=32 and d=128 on the first task and $0.500 \to 0.319$ past its peak on the second. So "more scale" is not one direction, and a study that grows a single compound budget averages two effects of opposite sign. The outcome distribution turns out to depend on the rule rather than on the resources: for parity, an all-or-nothing computation, 92% of $9{,}880$ networks land on one of exactly two solutions, each hitting its predicted accuracy ($1 - \eta$ and 1) to within 0.001 with nothing fitted, and the smooth averaged curve is then the mixing fraction between them, matched by $1 - \eta(1 - f)$ to an RMS of 0.0018, on which no individual model lies. For associative recall, which admits partial credit, the same measurement gives graded outcomes, with 40% of runs in the interior. Discreteness therefore requires a rule that cannot be half-learned; the opposition between width and compute does not.

1 INTRODUCTION

A training set that is consistent with two different rules leaves the choice to the optimiser. Which rule gets learned decides everything about behaviour on inputs the training set never covered, and the training loss barely registers the difference: if the two rules agree on all but a fraction ϵ of the training distribution, any bounded average over that distribution separates them by $O(\epsilon)$ no matter how much data is added.

That networks make such choices is established. They prefer cheap features (Geirhos et al., 2020; Shah et al., 2020), sparse and dense subnetworks compete for the same function (Merrill et al., 2023), and the data distribution can push a shallow transformer between a positional shortcut and an induction head (Kawata et al., 2025). What has been missing is how the two resources we scale act on the choice, and they turn out to disagree.

The central result is that opposition, and it holds on two tasks built on unrelated computations: more optimisation steps favour the general rule, more width favours the shortcut once past a small optimum. Along the way the shape of the outcome distribution separates the tasks rather than uniting them. Where the rule is all-or-nothing, runs land on one of two solutions and never between them, and the smooth curve produced by averaging is the mixing fraction between those solutions, describing no model in the population. Where the rule admits partial credit, the same measurement returns graded outcomes. Discreteness is thus a property of the rule; the resource opposition is not.

Contributions.

- **Discrete outcomes.** 92% of $9{,}880$ networks sit at a pole of a mechanism order parameter, and both branches realise their parameter-free accuracies to 0.001 (Section 4).

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.

- **The averaged curve is a mixing fraction.** Accuracy follows $1 - \eta(1 - f)$ with nothing fitted, RMS 0.0018 over fifteen settings at 64 seeds each (Section 4.1).
- **Opposing resources, on two unrelated tasks.** A 4×4 map of width against optimisation budget at fixed batch size, $12{,}544$ networks in total, separating two effects of opposite sign (Sections 5 and 6).
- **A first-passage account** of why the outcome is two-point, with the acquisition-time distribution measured directly (Section 7).
- **A memorisation control** establishing that the mechanism set is exactly two for parity (Section 4.2), and a second task showing that the two-point structure is a property of all-or-nothing rules rather than of mechanism competition in general (Section 6).

2 Related work

Competing mechanisms. Shortcut learning and simplicity bias (Geirhos et al., 2020; Shah et al., 2020; Ilyas et al., 2019) document the preference for cheap features. Kawata et al. (2025) prove that a shortcut gives way to an induction head as data diversity varies in a one-layer transformer. We measure different objects: how outcomes are distributed across seeds, what that distribution does to the mean curve, and how width and compute act on the choice. Circuit-level studies of particular behaviours (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2023) supply the vocabulary for describing which algorithm a network ended up with.

Abrupt change during training. Grokking (Power et al., 2022; Nanda et al., 2023; Liu et al., 2023) and its reading as a first-order transition (Rubin et al., 2024) concern transitions along the time axis, visible in the loss. Parity is the standard testbed for computational thresholds (Barak et al., 2022; Abbe et al., 2023). Our control parameters are properties of the data and of the resource budget, and the object of study is the distribution of terminal outcomes rather than the path taken to them.

Emergence. Wei et al. (2022) reported abrupt capability gains; Schaeffer et al. (2023) showed that discontinuous metrics manufacture much of the abruptness, and Du et al. (2024) re-plot against pretraining loss. That work is about the shape of a curve against scale. We show that a curve of this kind can be perfectly smooth while describing no member of the population it summarises, which is a separate problem from metric choice and is not fixed by changing the metric.

Variation between runs. Sensitivity to seeds and data order is documented (Dodge et al., 2020; Bouthillier et al., 2021; Picard, 2021; Henderson et al., 2018), and has prompted calls for more careful empirical practice (Lipton & Steinhardt, 2019; Sculley et al., 2018). Those studies report the variance of a score. We report its modality. When the distribution has two points, a mean and an error bar describe nothing that exists, and the mixing fraction is the summary that carries the information.

Scaling. Loss is predictable in model and data size (Kaplan et al., 2020; Hoffmann et al., 2022; Bahri et al., 2024; Sharma & Kaplan, 2022), with refinements for limited data and reduced precision (Muennighoff et al., 2023; Kumar et al., 2024), and the fits themselves are known to be delicate (Besiroglu et al., 2024; Choshen et al., 2024). Those results track a scalar. Section 5 tracks which algorithm is obtained, and finds that two components of scale disagree about it.

3 Task and instruments

Each sequence carries a marker token at position zero followed by eight content tokens (Figure 1). The label is the parity of the high content tokens among the first three positions, which we call the RULE. The marker is a corrupted copy of the label, equal to it except on a fraction η of examples, and reading it is the CUE. Supervision is always the rule, so a pure-cue predictor has population error exactly η while a pure-rule predictor has none. The two solutions are separated by $O(\eta)$ in loss and by 1 on inputs where marker and parity disagree.

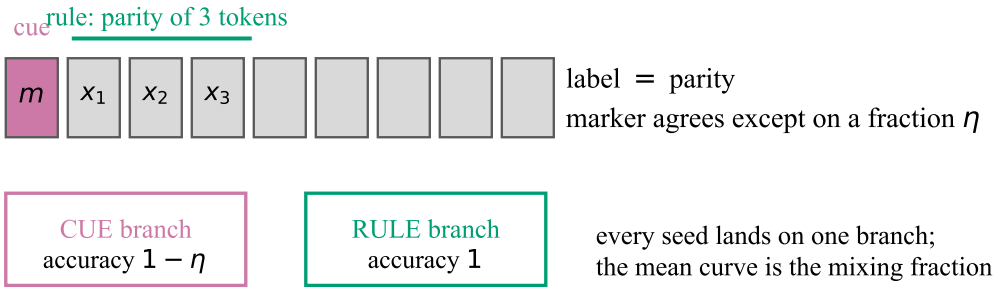


Figure 1: The task has two solutions. The marker agrees with the parity label except on a fraction η. A network using the cue reaches accuracy $1-\eta$; one using the rule reaches 1. Every seed lands on one branch or the other, and averaging over seeds traces the mixing fraction between them.

η	seeds on cue	acc. | cue	predicted $1-\eta$	seeds on rule	acc. | rule
0.02	22	0.9808	0.9800	0	—
0.05	24	0.9508	0.9500	0	—
0.08	23	0.9199	0.9200	0	—
0.14	20	0.8607	0.8600	2	**1.0000**

Table 1: Each run conditioned on the branch it landed on. Both branches match their predicted accuracy to within 0.001, and nothing is fitted.

Order parameter. On held-out inputs where marker and parity disagree by construction, let

$$m = 2\,\Pr[\text{model output} = \text{parity label}] - 1 \in [-1, 1],$$

so $m = +1$ marks the rule and $m = -1$ the cue. We write f for the fraction of seeds with $m > \frac{1}{2}$ and η_c for the disagreement rate at which $f = \frac{1}{2}$.

Models. Two-layer pre-layernorm transformers of width $d \in \{16, 32, 64, 128\}$ with four heads, trained with AdamW (Loshchilov & Hutter, 2019). Every configuration uses 64 independent seeds with 64 independent datasets, batched into one computation whose per-member gradients stay exactly independent (Appendix A).

4 For an all-or-nothing rule, the outcome is discrete

Figure 2(a) shows how m is distributed across seeds at four disagreement rates. The mass sits at the poles everywhere. Across all $9{,}880$ networks 92% satisfy $|m| > 0.95$, and within the valid range $0 < \eta \leq 0.40$ that rises to 94% with only 1.1% in the interior $|m| < 0.5$. A gradual crossover would put mass at intermediate m; this population does not have any.

Table 1 conditions each run on its own branch. Cue-branch runs sit at $1-\eta$ and rule-branch runs at 1, in both cases within 0.001. The branches are the two exact solutions of the task rather than approximate clusters, and m separates them without ambiguity.

4.1 The averaged curve is a mixing fraction

Once every run sits on one of two branches with accuracies $1-\eta$ and 1, the population mean is determined:

$$\mathrm{acc}(\eta) = f \cdot 1 + (1-f)(1-\eta) = 1 - \eta(1-f). \tag{1}$$

Figure 2(b) puts (1) against measurement at fifteen values of η with 64 seeds each. Maximum deviation is 0.0038 and the RMS is 0.0018, with no fitted quantity anywhere.

So the smooth curve is a mixing fraction. It interpolates between two branches, and wherever both are occupied it passes through a region no model occupies. Reading gradual change in behaviour off

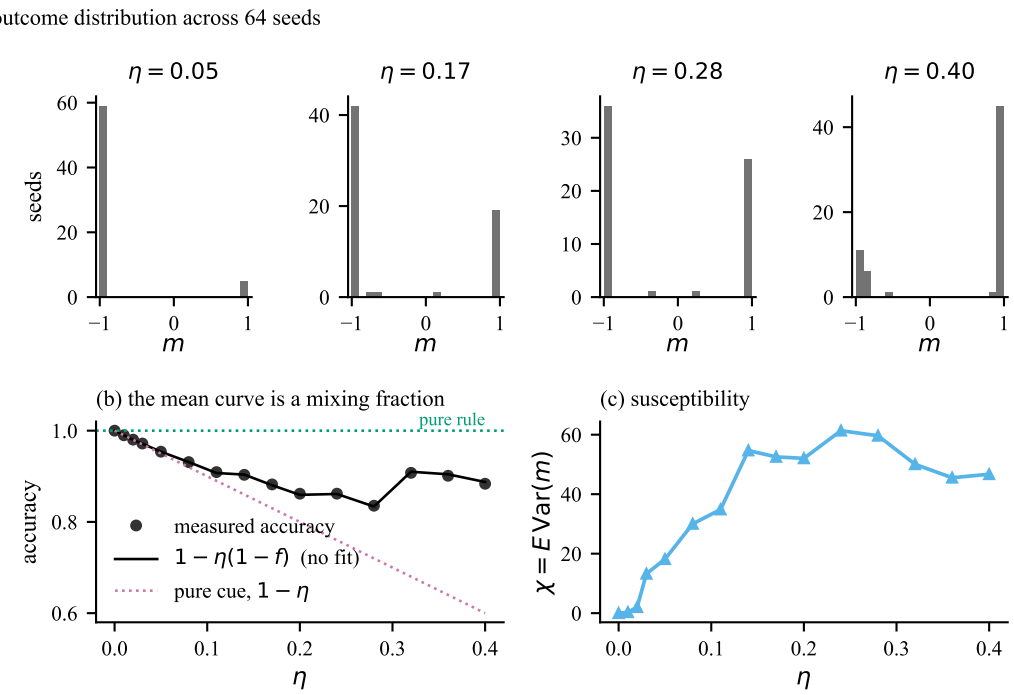


Figure 2: (a) Distribution of the order parameter across 64 seeds at four disagreement rates. (b) Measured accuracy against the parameter-free mixing law (1), with both branch accuracies drawn for reference. (c) Across-seed variance of m.

η	exceptions in training set	train-disagreement acc.	f	difference	memorising
0.02	399	0.011	0.000	+0.011	**0**
0.05	1001	0.001	0.000	+0.001	**0**
0.10	2003	0.126	0.125	+0.001	**0**
0.20	4005	0.168	0.167	+0.001	**0**
0.30	6009	0.335	0.333	+0.002	**0**
0.40	8010	0.388	0.375	+0.013	**0**

Table 2: Training-disagreement accuracy equals f throughout, so it is fully accounted for by the rule-branch networks. None of the 144 networks memorised.

that curve reads something into it that no run exhibits. Figure 2(c) shows the across-seed variance of m, which grows as the population splits.

4.2 The mechanism set is exactly two

A third strategy is available: memorise the ηN exceptions instead of learning the rule. The order parameter cannot see it, because a network that uses the cue and memorises exceptions fails on *held-out* disagreement inputs in the same way a pure-cue network does. Accuracy on *training* disagreement inputs separates them, since memorisation passes there and the cue does not.

Across all six settings train-disagreement accuracy equals f to within 0.013 (Table 2), so the rule-branch networks account for all of it. Not one network memorised, including at $\eta = 0.40$ where 8,010 exceptions were available and capacity was ample. The reachable set is therefore $\{\text{CUE}, \text{RULE}\}$ exactly, which is what makes the two-point description of Section 4 complete. Without this second probe, "uses the cue" and "uses the cue and memorises the exceptions" are indistinguishable, so any claim that a network adopted a shortcut needs it.

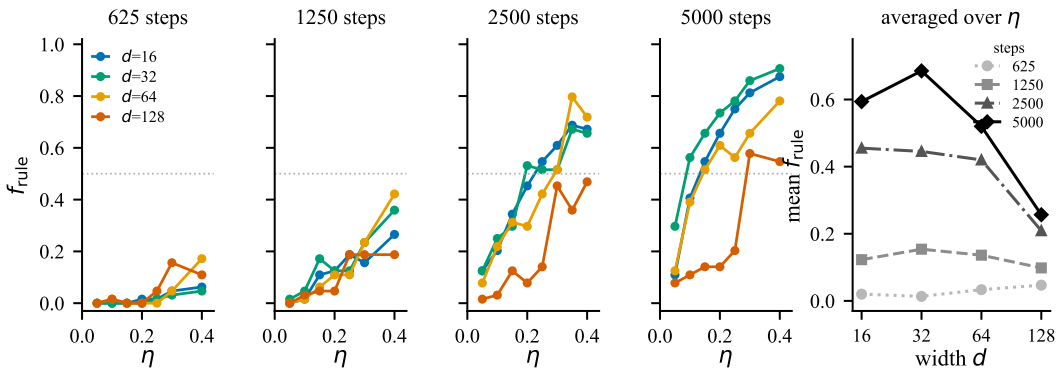


Figure 3: Fraction of seeds acquiring the general rule against disagreement rate, for four widths at four optimisation budgets. Right: the same data averaged over η and plotted against width, one line per budget. Raising the budget lifts every line; moving right along any line past d=32 lowers it.

5 Width and compute act in opposition

We map the choice over a 4×4 grid of width and optimisation budget at seven disagreement rates with 64 seeds per cell, which is 7,424 networks. **Batch size is fixed at** 256 **throughout**, so the budget axis moves only through optimisation steps and cannot be confounded with batch effects.

Compute favours the rule. Averaged over widths, f rises monotonically with the step budget: 0.030, 0.128, 0.347, 0.515 at 625, 1250, 2500 and 5000 steps. The rise appears separately at all four widths.

Width favours the cue. At 5000 steps the width-wise means fall 0.686, 0.520, 0.257 for $d = 32$, 64, 128, and the same ordering holds at 2500 steps (0.414, 0.367, 0.187). A mean over seven disagreement rates carries a standard error of 0.023, which puts the drop from d=32 to d=128 at roughly nineteen standard errors. In terms of the critical rate, at 5000 steps $\eta_c = 0.133, 0.088, 0.144$, 0.290 for $d = 16$ to 128, a 3.3-fold rise from d=32 to d=128, while η_c falls monotonically down every compute column.

A small optimum. f peaks near d=32 and falls on both sides. The narrowest models struggle to represent parity at all, which accounts for the low value at d=16; past the optimum, extra width goes to the cue.

The practical reading is that scale is not one axis. Parameters and optimisation steps push the learned algorithm apart, so a study that grows a single compound budget sees two effects of opposite sign partly cancel, and that is enough to explain why reports of how shortcut reliance varies with model size disagree.

6 A second task: what replicates and what does not

Everything so far rests on one task, so we repeat the resource map on a problem that shares none of its structure. Sequences hold a query token followed by four key–value pairs, the query matches exactly one key, and the target is that key's value. The RULE is associative recall; the SHORTCUT is to return the first value, which is correct whenever the matched pair happens to be first. Training places the match first with probability $1 - \eta$, so the shortcut again carries population error exactly η, over a computation (retrieval) unrelated to parity (arithmetic). The grid, batch size, seed count and order parameter are unchanged: 5,120 further networks.

The resource opposition replicates. Compute raises f_{rule} monotonically in all four width columns (Table 3), and the row means climb $0.001 \to 0.019 \to 0.105 \to 0.264$. Width raises f_{rule} up to an optimum and lowers it beyond, exactly the shape seen for parity: at 5000 steps, 0.500 at d=64 falling

	d=16	d=32	d=64	d=128	row mean
625 steps	0.000	0.000	0.000	0.003	0.001
1250 steps	0.000	0.000	0.047	0.028	0.019
2500 steps	0.000	0.025	0.306	0.091	0.105
5000 steps	0.006	0.231	**0.500**	0.319	0.264

Table 3: Fraction of seeds acquiring associative recall. Compute raises every column; width raises then lowers every row, with the optimum at d=64.

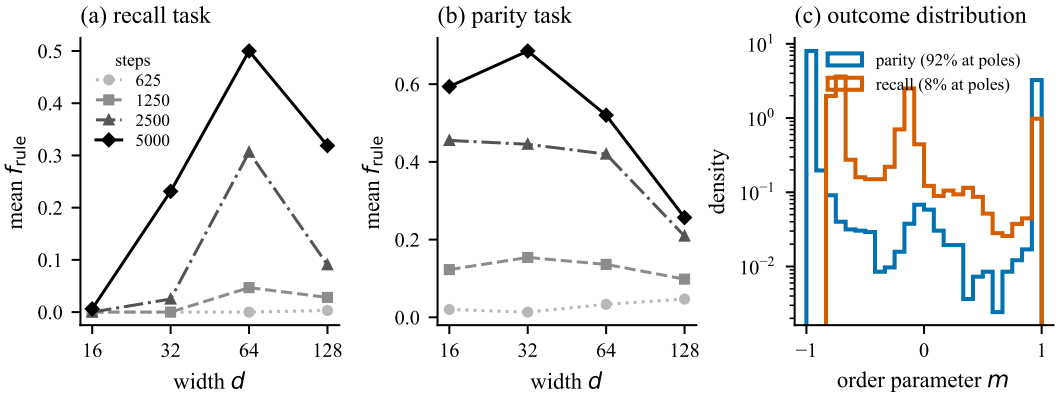


Figure 4: (a) Resource map for associative recall and (b) for parity, plotted against width with one line per step budget. Raising the budget lifts every line on both tasks; moving right past the optimum lowers it on both. (c) Outcome distributions pooled over each grid, log density. Parity concentrates at the poles; recall does not.

to 0.319 at d=128, and the same ordering at 2500 steps. The optimum sits at d=64 here against d=32 for parity, so its location is a property of the task and not a constant; the sign of each resource is what carries over (Figure 4a,b).

Discreteness does not replicate, and the reason is informative. Only 7.9% of the recall networks satisfy $|m| > 0.95$, against 92% for parity, and 40% sit in the interior $|m| < 0.5$ (Figure 4c). The distribution is still bimodal, but one mode sits near $m \approx 0$ rather than at the shortcut pole: those networks retrieve the right value about half the time. Parity cannot be half-computed, so a network either holds the circuit or does not; recall can be, so partial retrieval is a stable intermediate outcome. The two-point structure of Section 4 is therefore a property of all-or-nothing rules, and we state it that way rather than as a general law. The mixing identity (1) inherits the same condition, since it assumes each run realises one of two branch accuracies.

7 Why the outcome is two-point

The structure follows from reading selection as a first-passage problem instead of a comparison of population risks. Here the rule has risk 0 and the cue risk η, so the rule dominates at every $\eta > 0$ and the risk-minimiser never changes. What changes is reachability. The cue is available within a few steps, while the rule needs a search whose completion time τ varies over initialisations and shortens as η grows, since disagreement examples supply the only gradient signal that tells the two apart. Training for T steps yields the rule exactly when $\tau \le T$. Write $f(\eta, T) = \Pr[\tau(\eta) \le T]$.

Proposition 1. *(i) When the rule cannot be partially acquired, the order parameter is two-point on* $\{-1, +1\}$ *with* $\mathbb{E}[m] = 2f - 1$ *and no interior mass; parity satisfies this and associative recall does not, which is what Section 6 measures. (ii)* $\mathrm{acc}(\eta) = 1 - \eta(1 - f)$*, which is* (1)*. (iii)* $\eta_c(T)$ *solves* $\mathrm{median}\, \tau(\eta_c) = T$*, so* η_c *falls as* T *grows.*

Parts (i) and (ii) derive the discreteness and the mixing law, and (iii) fixes the sign of the compute axis. Figure 5 measures τ directly by probing during training. The survival curves are broad and

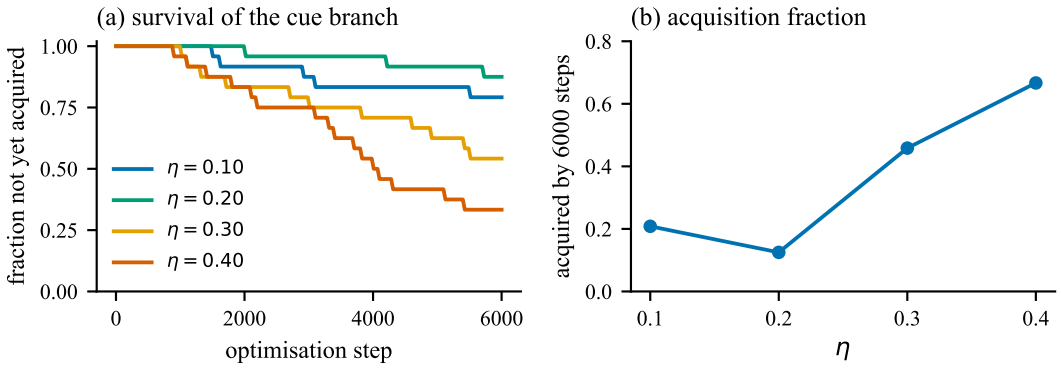


Figure 5: (a) Fraction of seeds that have not yet acquired the rule, against optimisation step, at four disagreement rates. (b) Fraction acquired by 6000 steps.

overlapping, which is what makes the outcome stochastic at fixed configuration, and the acquired fraction at a fixed budget grows with η as (iii) requires. Width enters τ from the other side: a wider network descends the cue direction faster, so the cheap solution consolidates before the parity search finishes. The two axes have opposite signs because each speeds up a different one of two competing clocks.

8 Discussion

Several consequences follow for practice. The two resources scaled most freely act against each other on the learned algorithm, so which algorithm a model ends up with depends on how its compute was divided and not only on the total; a sweep that grows both together will understate each. Where the rule is all-or-nothing, a mean and a standard deviation misdescribe the population, the mixing fraction is the right summary, and it is free to compute from runs already performed. In that regime a smooth curve is not evidence of gradual change, since the smoothest curve here is the one no model realises. Checking which regime a task is in costs one histogram.

The instrument that exposes all of this is cheap: an evaluation set built where the candidate algorithms are constructed to disagree, and enough seeds to see modality instead of variance. Any experiment that already trains a model more than once can afford both.

Reproducibility statement

All numbers come from the code and outputs in the supplementary material. The 4×4 map is produced by `gpu_sweep.py` with the grid recorded in the released job scripts, and the released JSON contains per-seed order parameters and accuracies for every cell, so every table and figure can be regenerated without retraining. Training configurations are in Appendix A. Because the ensemble trainer batches many models into one computation, its independence is a correctness requirement rather than an optimisation: `ens.py` carries a gradient-equivalence test against separately trained models that runs as part of every experiment.

References

Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: Leap complexity and saddle-to-saddle dynamics. In *Conference on Learning Theory*, 2023.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27), 2024.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. In *Advances in Neural Information Processing Systems*, 2022.

Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, et al. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, 2021.

Leshem Choshen, Yang Zhang, and Jacob Andreas. A hitchhiker's guide to scaling law estimation. *arXiv preprint arXiv:2410.11840*, 2024.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective. In *Advances in Neural Information Processing Systems*, 2024.

Nelson Elhage, Neel Nanda, Catherine Olsson, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, 2022.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 2019.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Ryotaro Kawata, Yujin Song, Alberto Bietti, Naoki Nishikawa, Taiji Suzuki, Samuel Vaiter, and Denny Wu. From shortcut to induction head: How data diversity shapes algorithm selection in transformers. In *Advances in Neural Information Processing Systems*, 2025.

Tanishq Kumar, Zachary Ankner, Benjamin F. Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. Scaling laws for precision. *arXiv preprint arXiv:2411.04330*, 2024.

Zachary C. Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship. *Queue*, 17(1), 2019.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems*, 2023.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations*, 2023.

Catherine Olsson, Nelson Elhage, Neel Nanda, et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

David Picard. torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *arXiv preprint arXiv:2109.08203*, 2021.

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.

Noa Rubin, Inbar Seroussi, and Zohar Ringel. Grokking as a first order phase transition in two layer networks. In *International Conference on Learning Representations*, 2024.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems*, 2023.

D. Sculley, Jasper Snoek, Alex Wiltschko, and Ali Rahimi. Winner's curse? on pace, progress, and empirical rigor. *ICLR Workshop Track*, 2018.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, 2020.

Utkarsh Sharma and Jared Kaplan. Scaling laws from the data manifold dimension. *Journal of Machine Learning Research*, 23, 2022.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in gpt-2 small. In *International Conference on Learning Representations*, 2023.

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. *Transactions on Machine Learning Research*, 2022.

A Experimental setup

Sequences have length nine: a marker token at position zero drawn from a two-symbol alphabet disjoint from the content vocabulary, then eight content tokens drawn uniformly from sixteen symbols. The label is the parity of the number of content tokens at positions one through three that fall in the upper half of the content alphabet. Networks are two-layer pre-layernorm transformers with four heads and a fourfold feedforward expansion, trained on $20{,}000$ examples with AdamW (Loshchilov & Hutter, 2019), learning rate 3×10^{-3} under a one-cycle schedule, weight decay 10^{-2}, gradient clipping at norm one, and batch size 256 for the map of Section 5.

Ensembles of E models train as one batched computation with a leading ensemble axis. Two details keep the members independent, and both are easy to get wrong. The objective is the sum over members of each member's mean batch loss, so that no member's gradient is divided by E; and gradient clipping is applied per member, because a global norm couples them. Each member also draws its own dataset. The released test compares batched gradients against separately trained models to floating-point tolerance.

B Probes

The disagreement probe is built by sampling content tokens uniformly, computing the parity label, then setting the marker to the opposite value, so every probe input is one on which the two solutions

differ. The train-disagreement probe of Section 4.2 samples with replacement from each member's own training exceptions, so it tests memorisation of items that member actually saw. Evaluation runs in chunks along the example axis, which keeps evaluation memory independent of probe-set size.

C ACQUISITION TIMES

For Figure 5 the order parameter is evaluated every 100 steps during training at constant learning rate, and τ is the first probe step after which m stays above $\frac{1}{2}$. Seeds that never cross are right-censored and enter the survival curves but not the medians. Constant learning rate is used here rather than one-cycle so that the hazard is not confounded with a decaying step size; the terminal fractions therefore differ slightly from the one-cycle runs elsewhere, and the two schedules are never compared at matched budget.