Pre-registered stakes against a tuned baseline: a one-day ledger of grokking interventions that mostly failed

Aleksei Kudriashov
ORCID 0009-0006-8885-5338

Claude Fable 5 (Anthropic)
autonomous research agent

ChatGPT gpt5.6-sol (OpenAI)
deep research and review

Abstract

We report a pre-registered, git-timestamped ledger of interventions on grokking in a one-layer transformer trained on modular addition ($p = 97$, 30% of pairs). Every claim was written as a numeric prediction with a tolerance before the run; outcomes are "holds" or "violated". Of 40 stakes closed in one day (training repository alone), 25 were violated. The two results we consider useful are negative or corrective: (i) the "proven" recipe we adopted as baseline (AdamW, lr 10^{-3}, wd 1.0; median 7 000 steps to $> 99\%$ test accuracy over 10 seeds) was untuned — lr $4 \cdot 10^{-3}$ groks in 1 250 steps, so a $5.6\times$ "transfer speed-up" we had measured collapsed to $1\times$ against the tuned baseline; (ii) a larger initialisation scale groks *earlier* in our setting, replicated by an independent implementation, but the effect is dominated by the relative step size and adds nothing once lr is tuned. What survived tuning: (a) at a fixed fraction of shown pairs, the structure of the set of differences decides whether grokking happens at all, with a character-sum flatness that predicted the order of three new designs before the runs; (b) pre-training on subtraction makes addition generalise at the first evaluation (250 steps) in both implementations, while a multiplication donor transfers only in the ReLU/no-LayerNorm implementation. We also report a candidate invariant — the MLP weight norm at the grokking step (CV 3–5% across a $16\times$ range of initialisation scale; $\leq 25\%$ drift across $4\times$ lr, $8\times$ wd and $2\times$ modulus) — which replicated in the second implementation only once its final LayerNorm was removed, and which three pre-registered causal tests then demoted from cause to consequence: grokking proceeds unchanged with the norm capped at 18 or floored at 32. We argue that the ledger, not the results, is the contribution: it turned three "findings" into one within hours, at the cost of 40 GPU-minutes.

1 Method: stakes before runs

All experiments use one setting unless stated: a one-layer pre-norm transformer ($d = 128$, 4 heads, MLP width 512, ReLU, no final LayerNorm, learned positions initialised at zero; 224k parameters) trained with AdamW ($\beta = (0.9, 0.98)$, batch 512) on $(a + b) \bmod 97$ with 30% of the 97^2 pairs as training data. "Grokking step" is the first evaluation (every 250 steps) with test accuracy $> 99\%$; runs stop 1 250 steps after it or at 30 000. Every intervention was entered in a JSON ledger *before* the run as a numeric prediction with a tolerance and a named violation condition; outcomes

The experiments, analysis and text were produced by autonomous AI agents under the direction of the author of record, who takes responsibility for the content. AI systems are listed as authors where policy permits; venues that disallow AI authorship should treat the human author as sole author.

Preprint.

are "holds" or "violated", never "partially". The ledger is committed to git with timestamps, so no threshold or metric could be chosen after seeing data. A second implementation, written independently from the description alone (different author, code not shared; final LayerNorm, GELU, positions $0.02\,\mathcal{N}(0,1)$), re-ran the claims that survived. Runs took 30 s each on one L4; the whole ledger cost under one GPU-hour. Agents (Claude) wrote the code, the stakes and ran the sweeps under the human author's direction; the author is responsible for all content.

2 The untuned baseline, twice

We adopted the recipe of the grokking literature (one-layer transformer, lr 10^{-3}, weight decay 1.0) as the "proven" baseline: median 7 000 steps to grokking over 10 seeds (5 750–9 250). Two pre-registered controls then moved the reference twice. First, the same recipe with lr $4 \cdot 10^{-3}$ groks in 1 250 steps (10 seeds: 1 000×3, 1 250×5, 1 500×2; lr $2 \cdot 10^{-3}$: 2 750) — but at that learning rate the first crossing is not sustained: five consecutive evaluations above 99% are first reached at a median of 2 000 (1 000–6 750), while at lr 10^{-3} sustained equals first (7 250 vs 7 000). Second, the architecture was untuned as well: a real-valued network *without attention* — embeddings of the two operands added, one ReLU MLP with a residual connection, linear readout, 156k parameters — groks in 500 steps, sustained, at the same lr, wd and data (10 seeds: 500×5, 750×5, sustained equal, 0/10 un-learn; the independent implementation: 750×10, 0/10 un-learn, vs its transformer 1 250 first / 2 625 sustained with 6/10 un-learning). Everything measured against the untuned baseline was inflated: pre-training on multiplication had "accelerated" addition $5.6\times$ (1 250 vs 7 000); against the lr-tuned baseline the same pre-training *slows* it down (median 4 250). This is the mechanism by which large speed-ups are produced in this field: re-measurements of Grokfast at equal weight decay give $1.11\times$ and $0.91\times$ [2] and $1.01\times$ [3], while two published baselines for the same task differ by $31\times$. We therefore report every intervention against the tuned recipe and, where the metric matters, by the sustained criterion.

3 Initialisation scale and the relative step

Multiplying all parameters at initialisation by $s \in \{0.25, 0.5, 1, 2, 4\}$ (lr 10^{-3}) gives median grokking steps 8 250, 6 750, 6 500, 5 750, 5 250: *larger* starts grok *earlier*, the opposite of the usual recommendation. The independent implementation reproduced the direction (its $s = 0.5$ mostly failed to grok within 30 000 steps; $s = 2, 4$ grokked at 3 000–3 500 vs 4 500 at $s = 1$). Scaling only the transformer block, or only the weight matrices, keeps the effect ($s = 2$: 3 750). Scaling only the readout by α reproduces the known lazy-regime delay ($\alpha = 16$: median 14 000), so the two observations are compatible: they scale different things. The lever is the relative step: lr$\cdot s$ at $s = 4$ groks at 1 250, lr$/s$ at 14 000 — and lr $4 \cdot 10^{-3}$ alone at $s = 1$ also gives 1 250. Once lr is tuned, initialisation scale adds nothing. Stake outcome: "invariant ratio of group norms at grokking" (our first hypothesis) was violated (CV 32–45%); it follows anyway from the AdamW equilibrium analysis [1], which our research stage found after the fact.

4 What transfers between operations

Pre-training the same network for 20 000 steps on another operation over the same residues, then training on addition. The clean evidence is at lr 10^{-3} in the independent implementation, where the donor must beat a random-label donor: after *subtraction*, addition reaches $> 99\%$ test accuracy at the first evaluation (250 steps, 3/3 seeds) against 5 000 after random labels and 4 500 from scratch; after *multiplication* (which had itself generalised) 5 250 — no transfer. The embedding after subtraction carries additive Fourier structure (fraction of spectral energy in the top-5 frequencies 0.33–0.75, median 0.44; random 0.06), after multiplication it does not (0.055–0.067). At the tuned lr the comparison is confounded: any 20 000-step donor phase, random labels included, shrinks the weights under weight decay and yields ≈ 750 steps (750/750/1 000 after random labels in the second implementation; 500–1 500 after subtraction and 500–750 after random labels in ours), so a "$1.7\times$ speed-up from transfer at tuned lr" would again be an artefact of the comparison. What transfers is the representation of the additive group, and it is only measurable where the donor beats the random-label donor. The condition is that the donor has *generalised*: $a + 2b$ does not generalise at 30% data within 30 000 steps (test 0.002; a memorised $a + 2b$ donor gives a partial 3 250), but at

50% data it generalises (8 750–26 500 steps) and then transfers to addition instantly (250/250/250, 3/3 seeds, lr 10^{-3}) — exactly like subtraction. The whole affine family $a + cb$ shares the representation; multiplication does not.

5 What the structure of the training set does

At a fixed fraction of shown pairs, *which* pairs are shown decides whether grokking happens at all. Take as the training set all pairs whose difference $a - b \bmod p$ lies in a set D of 29 residues ($29/97 \approx 30\%$ of the table). For D = 29 consecutive residues no seed groks within 30 000 steps (0/10 at lr 10^{-3}, 0/5 at $4 \cdot 10^{-3}$; second implementation 0/5 at both), while for a random D grokking is 2–2.5× *faster* than for random pairs of the same count (2 750 vs 7 000 at lr 10^{-3}; 750 vs 1 250 at $4 \cdot 10^{-3}$; both implementations). Mirrored pairs (b, a) add nothing (30% plus mirrors, half the table, groks exactly like 30%). The predictor is not the spectral gap of the pair graph (the Paley design, quadratic residues, has the largest gap and is *slower* than a random D in both implementations) but the flatness of the character sums of D, $1 - \max_{k \neq 0} |\sum_{d \in D} e^{2\pi i k d/p}|/|D|$: all failing designs sit at 0.14, Paley at 0.56, random D at 0.60–0.73. A pre-registered test on three new sets built to flatness 0.65/0.45/0.30 predicted the order of grokking times before the runs (2 500/3 750/4 250, no inversions; Paley's 3 500 falls in between). The reading through the Fourier circuit is direct: a large character sum at frequency k gives memorisation a path along that frequency. The qualitative form transfers to language with an honest base: on a character-level nanoGPT (validation drawn from random blocks over the whole text; "scattered" = non-overlapping blocks at the same coverage as the slice; 3 seeds), a contiguous slice is worse than a scattered sample of equal coverage by about 0.06 nats (Shakespeare, 30%) and 0.02 (*War and Peace*, 10%) on a 4-layer model — and about twice that on a 6-layer model (+0.09 to +0.16) — the end slice always worst, and the gap does *not* grow with the number of passes (1k/2k/4k iterations) — a constant cost of low diversity, not memorisation pressure; a 4-gram $\mathrm{KL}(\mathrm{val} \,\|\, \mathrm{sample})$ computed before training orders the four designs on both corpora exactly as the losses do (ordinal only, and only for coarse contrasts: two distant slices beat one where the KL says otherwise), and the effect survives word-level tokenisation with a larger magnitude (+0.06 to +0.21 nats per token for the three slice positions, same ordering). Two earlier versions of this check were invalidated by the stand itself (a positional validation split favours the slice next to it; a sample defined by random window *starts* covers almost the whole text because windows overlap), which we record as methodological points: equal fraction must mean equal coverage, measured before the comparison.

6 A candidate invariant, replicated, and causally demoted

The L_2 norm of the MLP weights at the first grokking step does not depend on the initialisation scale: over $s \in \{0.25, 0.5, 1, 2, 4\}$ with 10 seeds per cell it is 25.4 (lr 10^{-3}; CV across s 0.6%), 23.6 ($2 \cdot 10^{-3}$; 1.6%) and 23.0 ($4 \cdot 10^{-3}$; 2.3%) without a final LayerNorm, and 9.5 (2.5%), 10.3 (1.7%), 13.2 (2.8%) with one — 300 runs, all grokking. The embedding norm at the same step has CV 87% (large starts grok before their embeddings have shrunk), which is why a *ratio* of norms fails. The value is a slowly varying scale of the solution rather than a constant: 28.5 at wd 0.25 and 23.0 at wd 2; 21.9 for $p = 59$, 23.5 for $p = 113$; 19.3 for MLP width 256 and 26.0 for 1024 (5 seeds each) — $\leq 25\%$ drift where the AdamW equilibrium norm $\propto \sqrt{\mathrm{lr}/\mathrm{wd}}$ [1] would predict up to 2.8×. The independent implementation reproduced the numbers once its final LayerNorm was removed (26.2/26.0/25.3/25.2 across s; 1.6%) and with ReLU (24.7); with its LayerNorm the norm was monotone in s, which our own LayerNorm cells do not show, so that discrepancy is not about LayerNorm itself (activation and position initialisation differ; being separated one factor at a time). Three pre-registered causal tests then demoted the invariant from cause to consequence: initialising the MLP at norm 25 gives 1 250 steps (five seeds, as baseline); projecting the MLP norm to ≤ 18 after every step still groks (5/5, at 1 000 steps, norm 19); flooring it at ≥ 32 still groks (5/5, 1 250–1 500); the second implementation reproduced both (cap 16, floor 30). The generalising solution has a characteristic MLP scale, but grokking is not switched by it. The LayerNorm half of the factor transfers to language: on a character-level nanoGPT (two stands, 3 seeds each) removing the final LayerNorm costs about 0.03 nats for both activations, while the ReLU/GELU difference is a budget effect (0.088 at 2k iterations, 0.007 at 10k).

7 Failed interventions (documented)

intervention (all at tuned lr unless noted)	pre-registered prediction	outcome (steps per see
recycle 10% least-active MLP units / 1 000 steps (lr 10^{-3})	$\geq 1.3\times$ slower	no change: 5 750–7 25
recycle 30% least-active	$\leq 2/5$ grok	5/5 grok, 5 250–7 750
recycle 10% *most*-active	$\geq 1.5\times$ slower	6 750–9 250 (+12%)
auxiliary target: unreduced sum $a + b$ (lr 10^{-3})	$\geq 1.5\times$ faster	slower: 7 000–14 000
commutativity+associativity consistency loss, 30% data	≤ 600	1 000–1 250 (no chang
same, 10% data (control: 0/5 grok)	$\geq 3/5$ grok	0/5
consistency loss gated by current error (owner's thesis)	two-sided	1 000×7, 1 250×2, on
curriculum successor→mult→add (lr 10^{-3})	≤ 900	2/5 fail; 750–2 250
"death first": wd 4 phase, then subtraction under wd 4, then addition	≤ 500 (thesis)	0/5 grok; wd-4 phase a
population of 8, worst-2-by-law-consistency die every 500 steps ("core as selection")	$\geq 20\%$ earlier (thesis)	best-of-8 identical to c

8 Limitations, and one correction we owe to an external pass

One task, one architecture, five seeds per cell (ten for baselines); the two implementations differ in LayerNorm and activation; nothing here is claimed beyond $p = 97$. Two measurement issues matter. "At-grok" quantities are not converged [4]: our norm ratio moves from 16.0 at grokking to 21.3 after 1 250 more steps. And our headline metric — the *first* evaluation with test accuracy $> 99\%$ — overstates at the tuned learning rate: within the next 1 250 steps test accuracy falls back below 99% in 3/5 and 4/10 tuned-baseline runs, 5/5 law-loss runs and 9/9 need-gated runs (at lr 10^{-3}: 1/10 and 4/30). An independent multi-agent pass over our own ledger (run by the author with a different assistant) flagged this with a "sustained" metric (five consecutive evaluations $> 99\%$): first 1 250 vs sustained 2 000 (median) on the same baseline. All tuned-lr numbers above are therefore first-crossing numbers; re-measured by the sustained metric (10 seeds, full 30 000 steps): the fraction of seeds losing test accuracy within 2 000 steps of the first crossing is 1/10 at lr 10^{-3}, 6/10 at $2 \cdot 10^{-3}$, 8/10 at $4 \cdot 10^{-3}$; the -20% of need-gated laws vanished (9/10 never sustain). The ledger (released with the code at submission) contains every stake, including the ones we would rather not show.

References

[1] A. Kosson, B. Messmer, M. Jaggi. Rotational equilibrium: how weight decay balances learning across neural networks. ICML 2024 (arXiv:2305.17212).

[2] arXiv:2504.17243, Table 1 (re-measurement of Grokfast at matched weight decay).

[3] arXiv:2506.12284, Table 1 (independent re-measurement, ten initialisations).

[4] arXiv:2607.06639, "At-grok is not converged".

[5] T. Kumar, B. Bordelon, S. Gershman, C. Pehlevan. Grokking as the transition from lazy to rich training dynamics. ICLR 2024 (arXiv:2310.06110).