GrokScale

Does Curriculum Ordering of Training Data Shift the Grokking Transition Across Transformer Scales and Moduli?

Author: Isaac Shon | Westwood High School, Austin TX | July 2026

Repo: github.com/isaacshon/grokking-curriculum | Target compute: 8x AMD Instinct MI300X (single node), 1–2 weeks

---

1 · Problem

The grokking phenomenon — a sharp delayed transition from memorization to generalization in transformers trained on algorithmic tasks — has been studied exclusively under *random* data ordering since its discovery (Power et al., 2022). No work has asked whether the order in which training examples are presented affects when or whether grokking occurs. This is a meaningful gap: curriculum learning is known to alter training dynamics in supervised settings, and its interaction with the grokking transition is entirely unstudied. A second open question is whether existing predictive signals — weight norm trajectory and centered kernel alignment (CKA) — remain valid under structured curricula, or are artifacts of random ordering.

Preliminary result (Phase 1, already complete on Colab T4): We ran GPT-2 small on modular addition and multiplication mod 97 under five curriculum conditions (random, easy→hard, hard→easy, residue blocks, balanced mini-epoch). The residue-block condition showed <3% validation accuracy at step 16,000, compared to the random baseline grokking at step ~5,250 — a striking early finding that motivates the scaling study below.

The open question is whether this curriculum sensitivity is a small-model artifact that disappears at scale, or a fundamental property of the grokking transition. Answering this requires training runs across model sizes and moduli that far exceed what free Colab T4 GPUs can support.

2 · Approach

We extend the Phase 1 design along three axes, each requiring frontier compute:

Axis 1 — Model scale. Run all five curriculum conditions on GPT-2 small (117M), GPT-2 medium (345M), GPT-2 large (774M), GPT-2 XL (1.5B), and GPT-Neo 2.7B. GPT-2 XL and GPT-Neo 2.7B require >24 GB VRAM per replica, exceeding the T4's 16 GB. The MI300X's 192 GB HBM3 fits all models without gradient checkpointing, keeping training dynamics clean and comparable across scales.

Axis 2 — Modulus scaling. Repeat across six prime moduli: p = 7, 13, 53, 97, 113, 241. Prior work uses p = 97 exclusively. Larger moduli increase the dataset size quadratically (p^2 pairs) and are expected to delay grokking, making the curriculum effect — if real — more pronounced.

Axis 3 — Full CKA analysis. For each completed run, compute centered kernel alignment across all layer pairs at every saved checkpoint. This requires loading full activation tensors (>8 GB for GPT-2 XL at checkpoint density), which is infeasible on T4 but straightforward on MI300X.

The difficulty metric is the wrap-around boundary: pairs where $(a+b) \geq p$ for addition, or $(a \times b) \geq p$ for multiplication, are 'hard.' This gives a mathematically grounded difficulty proxy rather than a heuristic. All hyperparameters are fixed across conditions — only data ordering varies — isolating curriculum as the sole independent variable.

Condition	Description	Difficulty Basis
Random (baseline)	Standard DataLoader shuffle each epoch	N/A
Easy → Hard	No-wrap pairs first, wrap pairs last	Wrap-around boundary
Hard → Easy	Wrap pairs first, no-wrap pairs last	Same, reversed
Residue Blocks	All c=0 pairs, then c=1 ... c=p-1	Structural residue grouping
Balanced Mini-Epoch	Round-robin: one example per residue class per round	Equal class exposure

3 · Dataset

All data is fully synthetic — no download, no licensing, no PHI. For modulus p, the dataset is all p^2 ordered pairs (a, b) with a, b ∈ {0, ..., p-1}, labeled c = (a op b) mod p. A 50/50 random split produces train and validation sets. No preprocessing beyond tokenization into the string format 'a op b = c'.

Property	Value
Moduli	p = 7, 13, 53, 97, 113, 241
Operations	Addition mod p, Multiplication mod p
Dataset sizes	p=7: 49 pairs | p=97: 9,409 | p=241: 58,081
Train / val split	50% / 50% random
External data	None
PHI / licensing	None — fully synthetic

4 · Metrics

Experiment	Metric
Grokking transition detection	First step where val_acc ≥ 95%; DNF if not reached within max steps
Curriculum effect across scales	Grokking step delta relative to random baseline, by model size and modulus; does effect magnitude increase or decrease with scale?
Weight norm robustness	Lag between weight norm peak step and grokking step per condition; does peak still precede grokking under all curricula at all scales?
CKA robustness	Whether inter-layer CKA increases predictably before grokking under non-random curricula; Pearson correlation between CKA slope and grokking step
DNF rate	Fraction of conditions per modulus/scale that never grokk; is residue-block suppression scale-dependent?
Statistical rigor	5 seeds per condition; 95% CI reported for all main results

5 · Compute

Estimates scaled from measured Phase 1 Colab T4 runs (~2.5 h per condition for GPT-2 small). MI300X wall-clock padded ~25% for ROCm bring-up and job scheduling overhead.

Phase	Runs	Hardware	Est. Wall-Clock
Phase 1 (done) — GPT-2 small, mod 97, 10 conditions, 1 seed	10	Colab T4	~30 h (complete)
Phase 2a — GPT-2 medium/large/XL + GPT-Neo 2.7B, mod 97, 5 curricula, 5 seeds	200	8x MI300X	~3–4 days
Phase 2b — GPT-2 small/XL + GPT-Neo 2.7B, 5 moduli x 5 curricula, 5 seeds	150	8x MI300X	~2–3 days
Phase 2c — Full CKA analysis across all Phase 2a/2b checkpoints	N/A (post-hoc)	1–2 MI300X	~1 day
Ablations — no-weight-decay, from-scratch vs fine-tune, integer-step only	50	8x MI300X	~1 day
Total	400+	8x MI300X node	~8–10 days in a 2-week window

GPU memory: GPT-Neo 2.7B at batch size 64 with full activations stored for CKA requires ~40 GB per replica. On 8x MI300X (192 GB HBM3 each) this fits trivially without activation checkpointing, keeping training dynamics unmodified. On an 80 GB H100 it would require checkpointing and tensor parallelism, introducing confounds — the main reason MI300X is preferred.

Storage: Checkpoints at 1,000-step intervals across 400 runs x ~6 GB per XL checkpoint = ~2.4 TB peak. Will prune to final + 5 intermediate checkpoints per run = ~200 GB retained.

Software: ROCm 6.x + official PyTorch ROCm wheel. pip install transformers torch (no CUDA-only kernels — no flash-attn, no xformers, no custom Triton). CKA computed with plain torch.linalg. Bring-up: bash scripts/smoke_mi300x.sh --steps 10.

6 · Deliverables

1. Phase 2 training checkpoints for all conditions, retained on shared storage or HuggingFace Hub.

2. Main result figure: grokking step vs. curriculum condition, faceted by model scale and modulus. Primary claim: curriculum sensitivity is [scale-dependent / scale-invariant] — answered by the data.

3. Weight norm and CKA robustness plots across all conditions — do predictive signals survive curriculum?

4. DNF heatmap: which (curriculum, scale, modulus) combinations never grokk within budget.

5. A short paper (~6 pages, NeurIPS format) with abstract, introduction, methods, results, discussion. Target venue: ICLR 2027 workshop on mechanistic interpretability, or similar.

6. Public GitHub repository (MIT license) with full reproducible codebase, ROCm-specific README section, and per-run JSON metrics. One-command replication: bash scripts/replicate.sh --smoke.

I am not claiming SOTA. The goal is a clean, reproducible test of curriculum-grokking interaction across scales, with weight norm and CKA robustness as secondary contributions.

7 · Risk

Risk	Likelihood	Mitigation
Large models (GPT-Neo 2.7B) don't grokk within 20k steps on any curriculum	Medium	DNF is a result. Extend budget to 50k steps for large-model runs only; Phase 2a can be trimmed to GPT-2 small/medium/large without losing the scaling claim.
Residue-block suppression is a Phase 1 fluke that doesn't replicate	Low-Medium	5-seed Phase 2 design is specifically for this. If it doesn't replicate, the null result (curriculum doesn't matter) is still publishable as a robustness finding for grokking.
ROCm-specific PyTorch gap (e.g. CKA with torch.linalg on MI300X)	Low	CKA op is plain matrix multiply — no custom kernels. CPU fallback available for post-hoc analysis if needed.
Wall-clock overrun on 2-week window	Low-Medium	Phase 2b (modulus scaling) is lower priority than Phase 2a (model scaling). Can drop to 3 moduli and 3 seeds without losing the main claim. train.max_steps cap enforced for all runs.
Checkpoint storage exceeds budget	Low	Automatic pruning script deletes intermediates after CKA pass. Final + 5 checkpoints per run is well under 200 GB.

I will share Phase 2a intermediate results at the 1-week mark so scope can be adjusted. Phase 1 results and codebase available at request for review before allocation decision.