Grokking Hard Indicators: A Preregistered Evaluation Protocol and a Weak Baseline

Qien Huang*

Abstract

"Hard indicators" for grokking (and regime shifts in learning more broadly) are easy to claim and hard to validate, largely because many evaluation setups become non-evaluable: under common event definitions (e.g., fixed accuracy thresholds), the event may not occur, making predictive metrics ill-defined. We present a preregistered Explore -> Lock -> Evaluate pipeline that enforces temporal separation between indicator development and evaluation. We introduce a jump / regime-shift event definition that yields dense event occurrence in Phase B, enabling meaningful early-warning evaluation. Applying this protocol, we evaluate a baseline composite indicator under explicit false-positive-rate (FPR) control. We find a sharp failure mode: one score orientation achieves higher ranking metrics (e.g., AUC) but exhibits an uncontrollable FPR floor (≈ 0.44), rendering it invalid as an alarm; the alternative orientation admits proper FPR control but provides limited coverage at strict FPR (0.05) and a clear coverage-risk tradeoff. These results establish that ranking metrics alone are insufficient for early-warning validity and motivate treating FPR controllability as a necessary condition for "hard indicators".

Keywords: grokking; early warning; monitorability; calibration; false positive rate; phase transitions; preregistration

1 Introduction

Grokking refers to the phenomenon where a model initially overfits (memorization) and only later "suddenly" generalizes, often after prolonged additional training [1]. Because grokking resembles a regime shift, it is a natural benchmark for early warning: can we raise alarms before the generalization jump?

However, early-warning evaluation is frequently undermined by two methodological problems. First, if the event definition yields zero events in held-out runs, Phase B becomes non-evaluable. Second, if evaluation focuses on ranking metrics (e.g., AUC) without explicit risk constraints, one may report "good indicators" that are unusable as alarms.

This work addresses both problems via a preregistered protocol and explicit alarm validity criteria.

Contributions

This work makes four contributions. First, we introduce an evaluability-first protocol: a preregistered Explore -> Lock -> Evaluate pipeline that prevents retrospective tuning. Second, we propose a dense event definition based on jump / regime-shift detection (E1) that makes Phase B reliably evaluable. Third, we formalize risk-constrained evaluation via explicit FPR control and a validity

*ORCID: 0009-0003-7731-4294. Repository: `https://github.com/qienhuang/F-I-T/`.

gate (FPR controllability) that separates informative scores from deployable alarms. Fourth, we report a negative/weak baseline: one score orientation achieves higher ranking metrics but fails alarm validity due to an FPR floor, while the usable orientation exhibits limited low-FPR coverage and a clear tradeoff curve.

2 Related Work (brief)

Grokking was introduced as a stylized phenomenon in algorithmic/synthetic tasks and has since been used to study optimization dynamics, phase transitions, and delayed generalization [1]. More recent work studies progress measures for grokking from a mechanistic interpretability perspective [2]. Early-warning signals for critical transitions have a long history in complex systems [3, 4]. In ML monitoring and decision systems, ROC/AUC, precision–recall, and calibration are standard tools [5, 6, 7]. This work focuses on protocol and validity, not on proposing a new best indicator.

> **Positioning:** We treat "hard indicators" as a monitoring problem under risk constraints, not merely a ranking problem.

3 Problem Setup and Boundary

3.1 Boundary (held fixed)

We fix a single boundary for this protocol: modular addition $(a + b) \bmod p$ with a small transformer family, synthetic data generation, and checkpoint-level logging of test metrics together with the estimator tuple values.

The protocol is designed to compare indicators under a fixed boundary; new boundaries (other tasks/models) are out of scope for this manuscript.

3.2 Early warning as a risk-constrained decision problem

Early warning is not just "can you rank pre- vs post-event checkpoints." The operational questions are whether an alarm can be triggered at a bounded false positive rate (FPR), how much coverage (fraction of runs warned) can be obtained under that risk budget, and how early the warnings occur (lead time). We therefore evaluate alarms under explicit FPR calibration and report coverage and lead time as primary utility metrics, with ranking metrics (AUC/AP) treated as secondary diagnostics.

4 Protocol: Explore -> Lock -> Evaluate

4.1 Explore (Phase A)

In Phase A (Explore), we develop candidate event definitions and indicators and run diagnostic analyses/ablations to understand their behavior. No claims are made from Phase A results; Phase A is exploratory and is not admissible as evidence.

4.2 Lock

In the Lock phase, we freeze all degrees of freedom that could otherwise be tuned retrospectively: the event definition (E1) and its parameters, indicator definitions and hyperparameters, the thresh-

olding/calibration method and target operating points (e.g., the FPR grid), and the validity checks (notably the FPR controllability criteria).

In this repository, the "lock" is represented by versioned prereg/spec files plus a preregistration note: `experiments/grokking_hard_indicators_v0_2/code/protocol/estimator_spec.v0_2.yaml` (Phase B1 seeds 100–119), `experiments/grokking_hard_indicators_v0_2/code/protocol/estimator_spec.v0_2_1.yaml` (Phase B2 seeds 120–139; fresh held-out), and `experiments/grokking_hard_indicators_v0_2/code/protocol/prereg_v0_2.md` (protocol narrative: what is frozen vs exploratory).

4.3 Evaluate (Phase B)

In Phase B (Evaluate), we run held-out seeds with the locked definitions and report achieved FPR versus target FPR (calibration sanity), coverage versus FPR together with lead time (alarm utility), and ranking metrics (AUC/AP) as secondary diagnostics.

5 Event Definition (E1): Jump / Regime Shift

5.1 Motivation

Fixed accuracy thresholds (e.g., `test_acc >= 0.95`) often lead to zero events in held-out runs, making Phase B non-evaluable. We therefore define an event based on a sustained jump in smoothed performance.

5.2 Definition (high level)

Let $\bar{a}(t)$ denote smoothed test accuracy at checkpoint t. E1 triggers when three conditions hold: (i) $\bar{a}(t)$ increases by at least a fixed Δ over a short window, indicating a sudden improvement rather than gradual drift; (ii) the post-jump level exceeds a floor, to avoid triggering on micro-jumps early in training; and (iii) the trajectory does not immediately revert, enforced by a short hold condition.

Exact parameters are locked in the v0.2 / v0.2.1 protocol specs.

5.3 Evaluability

Across all runs reported here (Phase A seeds 0–4 and Phase B seeds 100–139; 45 runs total), E1 yields 100% event density: every run exhibits a detectable jump. On held-out evaluation runs alone (seeds 100–139; 40 runs), event density is also 100%, making Phase B evaluable.

6 Indicators and Alarm Construction

6.1 Baseline indicator tuple

We evaluate a baseline composite indicator that can be decomposed into two conceptual components: H_spec, a spectral-entropy-like component that captures distributional/structural changes in representations or dynamics, and CorrRate, a correction-rate-like component that captures changes in "self-correction" or error dynamics.

(We treat these as a baseline tuple; the protocol is agnostic to their exact engineering.)

6.2 Score orientation

Because many indicators are ambiguous up to sign, we evaluate two orientations: `score_sign = +1` and `score_sign = -1`. This is not a post-hoc choice; both orientations are evaluated under the same locked protocol.

6.3 Alarm thresholding via target-FPR calibration

Given a target FPR f, we calibrate a threshold θ_f so that, on negative checkpoints (no-event windows),

$$\Pr(s(t) > \theta_f \mid \text{negative}) \approx f.$$

We then apply this threshold to full trajectories to compute alarm triggers and lead times.

7 Metrics

7.1 Ranking metrics (secondary)

We report ROC-AUC and average precision (AP) as secondary diagnostics: they summarize ranking ability but do not determine alarm usability.

7.2 Alarm utility metrics (primary)

The primary utility metrics are achieved FPR (measured on negative checkpoints), coverage (fraction of runs with at least one alarm before the event), and lead time (event time minus first alarm time in steps, conditional on coverage).

7.3 Validity gate: FPR controllability (mandatory)

We treat a detector as **invalid** if it fails either: FPR tracking failure (achieved FPR does not track target within tolerance across multiple target points), or an FPR floor (achieved FPR cannot be driven below a ceiling regardless of threshold, indicative of "always-on" behavior).

This gate is motivated by Phase A2 findings and is enforced before interpreting coverage.

8 Experimental Design

8.1 Seed splits and phases

We report Phase B results on two held-out evaluation seed ranges: **Phase B1** uses seeds 100–119 (20 runs) and **Phase B2** uses seeds 120–139 (20 runs).

(Phase A seeds 0–4 are used only for protocol development / event-density checks; total held-out Phase B evaluations reported here are 40 runs.)

8.2 FPR sweep

We sweep target FPR thresholds:

$$\{0.01, 0.02, 0.05, 0.10, 0.15, 0.20\}.$$

9 Results

9.1 E1 event density and early-warning window

Using the locked jump/regime-shift definition (E1): we observe 100% event density (every evaluated run exhibits an event). Moreover, when alarms trigger successfully, they do so approximately 12–15k steps before the jump, corresponding to roughly 24–30 checkpoints under the current cadence.

This confirms Phase B is evaluable and the early-warning objective is non-trivial.

9.2 Phase A2: FPR-Coverage tradeoff reveals detector degeneracy

We evaluate the baseline score in both orientations under explicit FPR control.

Seed Range	Score Sign	Target FPR Range	Achieved FPR	Coverage (raw)	Mean Lead Time
100–119 (B1)	-1	0.01–0.20	0.4407 (constant)	100%	18775
100–119 (B1)	$+1$	0.01–0.20	tracks target	0% -> 85%	12071–14167
120–139 (B2)	-1	0.01–0.20	0.4442 (constant)	95%	18895
120–139 (B2)	$+1$	0.01–0.20	tracks target	0% -> 75%	7250–15000

Table 1: FPR sweep summary. The `score_sign=-1` orientation exhibits an FPR floor near 0.44, independent of target.

B1 (seeds 100–119): `score_sign = -1` is invalid (FPR floor)

Target FPR	Achieved FPR	Coverage	N Covered	Mean Lead Time
0.010	0.4407	100%	20/20	18775
0.020	0.4407	100%	20/20	18775
0.050	0.4407	100%	20/20	18775
0.100	0.4407	100%	20/20	18775
0.150	0.4407	100%	20/20	18775
0.200	0.4407	100%	20/20	18775

Interpretation: Achieved FPR cannot be reduced below ~ 0.44 regardless of threshold. Raw coverage is therefore meaningless under risk constraints. Under the protocol's validity gate, this configuration is invalid as an alarm.

B1 (seeds 100–119): `score_sign = +1` is valid and exhibits a tradeoff

Target FPR	Achieved FPR	Coverage	N Covered	Mean Lead Time
0.010	0.0100	0%	0/20	N/A
0.020	0.0200	0%	0/20	N/A
0.050	0.0500	35%	7/20	12071
0.100	0.1000	70%	14/20	13071
0.150	0.1500	75%	15/20	14167
0.200	0.2000	85%	17/20	13588

Coverage roughly doubles when moving from FPR=0.05 to FPR=0.10, while lead time remains stable ($\sim$ 12–14k steps).

B2 (seeds 120–139): `score_sign = -1` is invalid (same FPR floor)

Target FPR	Achieved FPR	Coverage	N Covered	Mean Lead Time
0.010	0.4442	95%	19/20	18895
0.020	0.4442	95%	19/20	18895
0.050	0.4442	95%	19/20	18895
0.100	0.4442	95%	19/20	18895
0.150	0.4442	95%	19/20	18895
0.200	0.4442	95%	19/20	18895

Again, achieved FPR is insensitive to target threshold, indicating calibration degeneracy.

B2 (seeds 120–139): `score_sign = +1` remains valid with similar tradeoff

Target FPR	Achieved FPR	Coverage	N Covered	Mean Lead Time
0.010	0.0100	0%	0/20	N/A
0.020	0.0200	10%	2/20	7250
0.050	0.0500	35%	7/20	15357
0.100	0.1000	65%	13/20	11769
0.150	0.1500	70%	14/20	14857
0.200	0.2000	75%	15/20	15000

9.3 Practical operating points

For the only valid orientation (`score_sign = +1`):

Target FPR	Coverage (B1)	Coverage (B2)	Avg. Coverage	Typical Lead Time
0.05	35%	35%	35%	$\sim$ 12–15k steps
0.10	70%	65%	67.5%	$\sim$ 12–14k steps
0.20	85%	75%	80%	$\sim$ 13–15k steps

Table 2: Operating points for the valid orientation (`score_sign=+1`).

9.4 Why AUC is insufficient: ranking-alarm decoupling

The invalid configuration (`score_sign = -1`) can exhibit higher ranking metrics while being unusable as an alarm due to an FPR floor. This provides a concrete counterexample to the common assumption that higher AUC implies better early-warning performance. In early-warning settings, ranking ability (AUC) is insufficient; controllable false-positive behavior is a necessary condition.

9.5 Phase A1 component diagnosis (summary)

Component-level diagnosis indicates that seed dependence is partly explained by which component dominates. In seeds 100–119, the spectral-entropy-like component (H_spec) shows stronger directional association, which can inflate AUC for one orientation. In seeds 120–139, both components weaken substantially, producing near-random discrimination.

However, even when a component provides ranking signal, it can induce calibration failure (FPR floor), making the resulting alarm invalid.

10 Discussion

10.1 What would count as a validated hard indicator?

Under this protocol, a "hard indicator" should satisfy five criteria on held-out evaluation: evaluability (events occur with sufficient density), validity (achieved FPR tracks target with no FPR floor), utility (non-trivial coverage at reasonable FPR such as 0.05–0.10), stability (consistent, non-trivial lead time), and robustness (persistence across seed ranges and ideally boundaries).

Our baseline fails validity in one orientation and is weak under strict low-FPR (0.05) even in the valid orientation.

10.2 Why this negative result is useful

This work clarifies why "hard indicators" are difficult: it is easy to produce scores with seemingly good AUC, but much harder to produce alarms that are usable under risk constraints. The protocol turns this into a measurable, reproducible failure mode.

10.3 Limitations

This manuscript studies a single boundary (modular addition with a small transformer family). Indicator engineering is limited by the baseline tuple; we do not claim optimal features. Some quantities, such as representation-level measures, require heavier instrumentation and are left for future work.

10.4 Next steps (v0.4 direction)

Future protocol versions should develop calibration-first indicators (e.g., autocorrelation and variance-based early-warning proxies), explore multi-gate detectors that aggregate evidence across components to reduce FPR floors, and expand Phase B to additional seed ranges and boundaries to test robustness.

Reproducibility Appendix

Key artifacts

- `experiments/grokking_hard_indicators_v0_2/code/protocol/estimator_spec.v0_2.yaml`
- `experiments/grokking_hard_indicators_v0_2/code/protocol/estimator_spec.v0_2_1.yaml`
- `experiments/grokking_hard_indicators_v0_2/code/protocol/prereg_v0_2.md`
- `experiments/grokking_hard_indicators_v0_2/RESULTS_v0.2_v0.2.1.md`
- `experiments/grokking_hard_indicators_v0_2/results/v0.3_A1_component_diagnosis.md`
- `experiments/grokking_hard_indicators_v0_2/results/v0.3_A2_fpr_tradeoff.md`

Example reproduction commands

```
cd experiments/grokking_hard_indicators_v0_2

# Environment
python -m venv .venv
source .venv/bin/activate
pip install -r code/requirements.txt
pip install -e code

# If you have archived runs, you can evaluate directly (example layout used in this
repo snapshot):
python -m grokking.analysis.fpr_tradeoff_curves --runs_dir runs_v0.2/eval --score_sign=+1
python -m grokking.analysis.fpr_tradeoff_curves --runs_dir runs_v0.2/eval --score_sign=-1

python -m grokking.analysis.fpr_tradeoff_curves --runs_dir runs_v0.2_1/eval --score_sign=+1
python -m grokking.analysis.fpr_tradeoff_curves --runs_dir runs_v0.2_1/eval --score_sign=-1
```

If you do not have the archived `runs_v0.2*` directories, follow `experiments/grokking_hard_indicators_v0_2/README.md` to generate runs via `python -m grokking.runner.sweep`, then point `-runs_dir` at the resulting `runs/\{explore,eval\}` folders.

References

[1] A. Power et al. Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets. *arXiv:2201.02177*, 2022.

[2] N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt. Progress Measures for Grokking via Mechanistic Interpretability. *arXiv:2301.05217*, 2023.

[3] M. Scheffer et al. Early-warning signals for critical transitions. *Nature*, 461:53–59, 2009.

[4] V. Dakos, S. R. Carpenter, E. H. van Nes, and M. Scheffer. Methods for Detecting Early Warnings of Critical Transitions in Time Series Illustrated Using Simulated Ecological Data. *PLoS ONE*, 7(7):e41010, 2012.

[5] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[6] J. Davis and M. Goadrich. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of ICML*, 2006.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. In *ICML*, 2017. *arXiv:1706.04599*.