NeuriCo: Towards Reliable AI Scientists

Haokun Liu [1] **Chenhao Tan** [1]

Abstract

To build AI systems that help with scientific research, we need to understand not just what they can do, but where they consistently fail. We present **NeuriCo**, an open-source AI co-scientist system that runs agents through a multi-stage pipeline of literature review, resource gathering, experiment execution, and analysis. Over 20 weeks of a community-driven weekly research competition, we ran 180 agent runs across 60+ research ideas in machine learning. Agents are strong at literature synthesis, data curation with smart filtering, and statistical analysis that honestly reports null results. They also show consistent failure modes that point to a deeper problem: they execute well, but they cannot judge research quality. We discuss what this means for system design, and how NeuriCo is being extended to physics, chemistry, and other domains.

1. Introduction

AI systems can now run scientific research end to end across machine learning (Lu et al., 2024; Yamada et al., 2025; Lu et al., 2026; Schmidgall et al., 2025; Tang et al., 2025; Weng et al., 2025), chemistry (Boiko et al., 2023), and biology (Swanson et al., 2025). A recent survey traces this move toward fully agentic discovery (Wei et al., 2025). These systems show that the pipeline can run. What they do not yet show is where the pipeline reliably fails. Without that, we cannot tell when to trust an AI scientist and when to step in.

We address this with **NeuriCo**, an open-source AI co-scientist system designed for sustained empirical evaluation. NeuriCo runs the same research idea through multiple AI providers in parallel, with optional human checkpoints between stages. We have deployed it in a public weekly research competition, where the community submits and votes on research ideas, and NeuriCo runs the top ones. Over 20 weeks we have collected 180 agent runs across more than 60 ideas in machine learning.

[1]Department of Computer Science, University of Chicago, Chicago, IL, USA. Correspondence to: Haokun Liu <haokunliu@uchicago.edu>.

Submitted to the AI for Science workshop (ICML 2026).

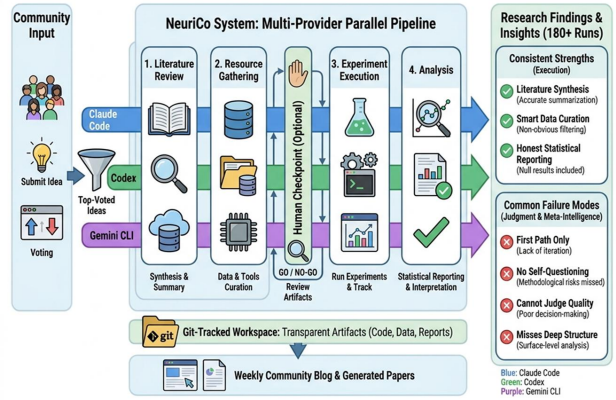


Figure 1. NeuriCo pipeline: four stages with multi-provider parallelism and an optional human checkpoint.

Our findings have a simple shape. Agents are strong at execution: gathering literature, curating data, and reporting results faithfully. They are weak at the judgment around execution: knowing what they do not know, choosing what to pursue, and questioning their own setup. This gap shows up across providers and across ideas.

Based on these findings, we argue that current AI scientists are most useful as exploration accelerators, with humans staying in the loop for selection and evaluation (Liu et al., 2024). The rest of this paper describes the NeuriCo system (Section 2), the patterns we observed (Section 3), the design choices that follow (Section 4), and what is next (Section 4).

2. The NeuriCo System

NeuriCo is built around four stages and one core idea: every stage produces inspectable artifacts, so a human can step in at any point.

Pipeline stages. The four stages are literature review, resource gathering, experiment execution, and analysis. Each stage uses standard tools: Semantic Scholar and arXiv for literature, Hugging Face and Papers with Code for data, and a Python virtual environment for execution. All outputs go to a Git-tracked workspace, so the full trace of a run is reproducible.

Multi-provider runs. NeuriCo wraps three coding agents (*Claude Code*, *Codex*, and *Gemini CLI*) behind a common interface, so the same idea runs through all three in parallel. Disagreement across providers is itself a useful signal: it surfaces where the question is underspecified or the data is weak.

Human checkpoints. A run can pause after resource gathering for a human to inspect what was collected before any compute is spent on experiments. By default, NeuriCo also asks before installing dependencies, executing arbitrary code, or pushing to GitHub. A full-permissions mode is available for users who want fully autonomous runs.

Community pipeline. Ideas are submitted and voted on through a public site. The top-voted ideas each week are run by NeuriCo, with reports published as weekly blog entries. This gives us a steady stream of diverse ideas from real researchers, rather than a fixed benchmark.

3. What 180 Runs Showed Us

Benchmarks for AI research agents have grown to cover ML experimentation (Huang et al., 2024), paper replication (Starace et al., 2025), and open-ended research tasks (Nathani et al., 2025; Chen et al., 2025). NeuriCo runs a sustained stream of community-submitted ideas instead, which surfaces patterns that single-task scores miss. We summarize what we have learned below; idea-level case studies are in the appendix.

Where agents are strong. Three patterns hold across providers. First, literature synthesis is reliable: agents gather and summarize related work across many sources without major errors. Second, data curation is often smart: agents apply non-obvious filters to extract usable subsets of raw data, such as restricting to high-disagreement examples for studies of human variation. Third, statistical reporting is honest: agents run appropriate tests, report effect sizes, and do not hide null results when they show up.

Where agents fail in the same ways. Four failure modes recur across runs.

1. **First path only.** Agents pick an approach early and rarely revisit it. Once a plan is set, they do not actively seek new information that might change it.
2. **No self-questioning.** Agents proceed without asking whether their setup matches the question. The clearest version is the synthetic-data fallback: when real data exists, agents sometimes generate synthetic data instead and do not flag the move as a methodological risk.
3. **Cannot judge quality.** Agents execute plans well, but they have no clear sense of which decisions are good ones. They use 20 examples for studies that need hundreds, and they generate dozens of unprioritized literature search tasks instead of synthesizing what they already have.
4. **Misses deep structure.** Agents find a strong signal and stop. In a study of AI-versus-human text classification, all three providers reported around 95% accuracy without noticing that text length explained most of the signal.

These four failures share a common shape. Agents are good at carrying out a defined task. They are bad at the meta-level work that decides whether the task is worth carrying out, whether it is being done correctly, and what its results actually mean. We call this the meta-intelligence gap. A concurrent case study of four end-to-end research attempts reports a similar pattern (Trehan & Chopra, 2026). Selected NeuriCo-generated papers are in the appendix.

4. Design Choices and Safeguards

Stage separation with optional human review. Splitting the pipeline into stages with inspectable artifacts catches most judgment failures before compute is spent. A researcher who looks at the resource list after stage two will often spot a wrong dataset or a missing baseline that the agent would have run with.

Multi-provider parallelism. Running three providers on the same idea is more useful than tuning one. When all three converge, the result is more trustworthy. When they diverge, the divergence itself tells the researcher to look closer.

Transparent artifacts. Every run leaves a Git history of code, data, and reports. This makes it easier for a human to audit what happened, rather than reading a polished writeup that hides the choices behind it.

Governance. NeuriCo asks before installing packages, running scripts, or pushing to GitHub by default. We do not run code that touches external services without explicit approval. Each run records the proposer, the voters, and the agent provider in a YAML file, so attribution is preserved through the Git history.

Roadmap. We are extending NeuriCo beyond machine learning. Early collaborations in physics and chemistry test whether the four failure modes generalize when the artifacts are simulations or proofs. We are also building tools for humans to compare multi-provider runs side by side. The hardest open problem is whether agents can learn to know what they do not know.

References

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Chen, H., Xiong, M., Lu, Y., Han, W., Deng, A., He, Y., Wu, J., Li, Y., Liu, Y., and Hooi, B. MLR-Bench: Evaluating AI agents on open-ended machine learning research, 2025.

Huang, Q., Vora, J., Liang, P., and Leskovec, J. MLAgentBench: Evaluating language agents on machine learning experimentation, 2024.

Liu, H., Zhou, Y., Li, M., Yuan, C., and Tan, C. Literature meets data: A synergistic approach to hypothesis generation, 2024.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The AI scientist: Towards fully automated open-ended scientific discovery, 2024.

Lu, C., Lu, C., Lange, R. T., Yamada, Y., Hu, S., Foerster, J., Ha, D., and Clune, J. Towards end-to-end automation of AI research. *Nature*, 651:914–919, 2026. doi: 10.1038/s41586-026-10265-5.

Nathani, D., Madaan, L., Roberts, N., Bashlykov, N., Menon, A., Moens, V., Budhiraja, A., Magka, D., Vorotilov, V., Chaurasia, G., Hupkes, D., Cabral, R. S., Shavrina, T., Foerster, J., Bachrach, Y., Wang, W. Y., and Raileanu, R. MLGym: A new framework and benchmark for advancing AI research agents, 2025.

Schmidgall, S., Su, Y., Wang, Z., Sun, X., Wu, J., Yu, X., Liu, J., Moor, M., Liu, Z., and Barsoum, E. Agent laboratory: Using LLM agents as research assistants, 2025.

Starace, G., Jaffe, O., Sherburn, D., Aung, J., Chan, J. S., Maksin, L., Dias, R., Mays, E., Kinsella, B., Thompson, W., Heidecke, J., Glaese, A., and Patwardhan, T. PaperBench: Evaluating AI's ability to replicate AI research, 2025.

Swanson, K., Wu, W., Bulaong, N. L., Pak, J. E., and Zou, J. Y. The virtual lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646:716–723, 2025. doi: 10.1038/s41586-025-09442-9.

Tang, J., Xia, L., Li, Z., and Huang, C. AI-Researcher: Autonomous scientific innovation, 2025.

Trehan, D. and Chopra, P. Why LLMs aren't scientists yet: Lessons from four autonomous research attempts, 2026.

Wei, J., Yang, Y., Zhang, X., Chen, Y., Zhuang, X., Gao, Z., Zhou, D., Wang, G., Gao, Z., Cao, J., Qiu, Z., Hu, M., Ma, C., Tang, S., He, J., Song, C., He, X., Zhang, Q., You, C., Zheng, S., Ding, N., Ouyang, W., Dong, N., Cheng, Y., Sun, S., Bai, L., and Zhou, B. From AI for science to agentic science: A survey on autonomous scientific discovery, 2025.

Weng, Y., Zhu, M., Bao, G., Zhang, H., Wang, J., Zhang, Y., and Yang, L. CycleResearcher: Improving automated research via automated review. In *Proceedings of ICLR*, 2025.

Yamada, Y., Lange, R. T., Lu, C., Hu, S., Lu, C., Foerster, J., Clune, J., and Ha, D. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

How Many Agents to Avoid Collapse? A Minimum Viable Population Study of Self-Training LLM Ecosystems

Ari Holtzman and NeuriCo

Abstract

A growing fraction of web text is generated by large language models (LLMs). If future foundation models train on this text, single-model recursive training is known to cause distributional collapse. We ask the ecosystem-level analogue: is there a *minimum viable population* (MVP) of LLMs that, sharing a common synthetic-text substrate, avoids collapse? And what kind of inter-agent differentiation actually counts? We run two complementary harnesses on the same metric suite. In a fine-tuning ecosystem (DISTILGPT2 copies retraining on each others' WIKITEXT-2 outputs) even $N{=}16$ agents collapse, with perplexity rising $2.10\times$ over six iterations and a slope of 16 ppl/iter that is statistically significant ($p < 10^{-6}$); the rate of collapse falls $\sim 5\times$ from $N{=}1$ to $N{=}16$ but never reaches zero. In a retrieval-augmented ecosystem (frozen API LLMs sharing a growing post pool), $N{=}2$ *LLMs from different pretraining families* preserve full inter-agent diversity over $T{=}12$ iterations, with both pairwise distance and the Vendi score yielding regression slopes indistinguishable from zero ($p \approx 0.4$). Holding $N{=}4$ fixed and varying the axis of differentiation, four agents from different pretrained families preserve $\mathbf{1.9}\times$ the inter-agent semantic distance of four agents varied by prompt persona, by retrieval slice, or by random seed alone. Together these results sharpen the population view of model collapse: the training-time link is what matters, and the only individuality axis that buys real collapse-resistance is diversity in the prior, not the conditioning.

1 Introduction

A growing fraction of the public web is generated by large language models, and the next generation of foundation models will train on a substrate that is partly their own ancestors' output. When a single generative model is trained on its own outputs in a closed loop, the distribution it represents narrows: tails disappear first, modes collapse, and the model converges to a low-variance regime [Shumailov et al., 2024, Alemohammad et al., 2023, Dohmatob et al., 2024]. This phenomenon, *model collapse*, has been replicated across modalities, architectures, and training regimes.

Why this matters. Almost every published collapse study fixes the unit of analysis at a *single* self-training model. The web, however, is populated by *many* LLMs from different vendors, with different pretraining priors, system prompts, and retrieval pipelines. The policy-relevant question is not "does one model collapse on its own outputs" — the answer to that is yes — but rather: how many distinct LLM agents, sharing a common substrate, are enough to keep that substrate *healthy*? Borrowing the term from population ecology, is there a *minimum viable population* (MVP) [Shaffer, 1981] of LLMs below which collapse is unavoidable? And what kind of differentiation between agents counts as "different individuals"?

Gap. Three recent ecosystem-level studies each leave a piece of the question open. Hodel and West [2026] test only $M \in \{1, 2, 4, 16\}$ copies of one base model differentiated by training-data

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

segment, find that the *optimal* M grows monotonically with iteration count, but never report an MVP at any fixed horizon. Wang et al. [2025] run a network of $N{=}3$ pretrained-different LLMs sharing a retrieval pool but never sweep N. Vu et al. [2025] treat $N{=}2$ model families theoretically but do not test $N{>}2$. None of these works compare multiple axes of inter-agent differentiation, and none compare the fine-tuning ecosystem to the cheaper retrieval-augmented ecosystem on identical metrics.

Our approach. We close all three gaps in one experimental program. We run two complementary harnesses on the *same* diversity metric suite (perplexity, distinct bigrams, mean pairwise embedding distance, and the Vendi-score Hill–Shannon Diversity, hereafter HSD). E1 is a fine-tuning ecosystem where N copies of DISTILGPT2 retrain on each others' WIKITEXT-2 outputs in replace mode. E2 is a retrieval-augmented ecosystem where N frozen API LLMs each query a shared, growing "internet" of their own posts. E3 fixes $N{=}4$ in the retrieval harness and varies four axes of individuality: random seed, retrieval data slice, system-prompt persona, and pretrained model family.

Quantitative preview. We find that $N{=}2$ *frozen API LLMs from different pretraining families* preserve full diversity for $T{=}12$ iterations (slope of mean pairwise distance: -0.002 ± 0.004, $p = 0.44$), while in the fine-tuning harness even $N{=}16$ agents collapse by a factor of $2.10\times$ in perplexity over only $T{=}6$ iterations. Across the four diversity axes at $N{=}4$, model-family diversity preserves $\mathbf{1.9}\times$ the terminal inter-agent distance of the next-best axis (persona or data slice). The largest single N-doubling benefit in the fine-tuning ecosystem is $N{=}8 \rightarrow N{=}16$, which cuts terminal perplexity by 26%.

Contributions.

- We introduce a population-ecology framing of recursive LLM training, formalising the MVP question and giving the first explicit MVP estimates for both fine-tuning and retrieval-based ecosystems on a common metric suite.
- We extend the only published N-vs-collapse sweep [Hodel and West, 2026] from $N \leq 16$ to a controlled statistical comparison across $N \in \{1, 2, 4, 8, 16\}$ in the fine-tuning regime and $N \in \{1, 2, 3, 5, 8\}$ in the retrieval regime, holding compute roughly constant.
- We run the first head-to-head comparison of four orthogonal axes of inter-agent differentiation (seed, data slice, prompt persona, model family) at fixed $N{=}4$, and show that *architecture diversity is the only axis that buys collapse-resistance*.
- We reconcile the divergent answers from the two harnesses into a single governance-relevant claim: it is the *training-time* link, not the inference-time loop, that drives ecosystem collapse.

Paper organization. section 2 positions the work against the single-model and ecosystem-level collapse literature. section 3 details the two harnesses, metrics, and statistical analysis. section 4 reports E1, E2, and E3 results, with MVP estimates and axis ranking. section 5 discusses limitations and what the cross-method gap means for AI deployment. section 6 closes with bottom-line answers and recommended follow-ups.

2 Related Work

Single-model collapse. Shumailov et al. [2024] formalised model collapse, showing that a single OPT-125m fine-tuned on its own WIKITEXT-2 outputs has perplexity that rises monotonically across generations and a generation distribution whose tails progressively disappear. Alemohammad et al. [2023] extended the result to image generative models under the name *Model Autophagy Disorder* (MAD), and Briesch et al. [2023] disentangled quality decay (preserved on verifiable tasks) from diversity decay (not preserved). Guo et al. [2023] introduced lexical, syntactic, and semantic diversity metrics for the recursive-training setting; we reuse the distinct-bigram metric from that work. Theoretical accounts include Dohmatob et al. [2024], who frame collapse as a change of scaling laws, and Seddik et al. [2024], who derive a maximal admissible synthetic ratio. Our N=1 condition is a direct reproduction of the Shumailov et al. [2024] "replace" regime.

Mitigation by data dynamics. Gerstgrasser et al. [2024] show that *accumulating* synthetic and real data (rather than replacing) yields a finite, bounded test error independent of iteration count, both empirically on TinyStories and theoretically. Dey and Donoho [2024] sharpen this to a universal $\pi^2/6$ asymptote. These results establish data-side mitigations; we deliberately ablate them by run-

ning our fine-tuning harness in the harshest replace regime so that diversity alone is the variable being tested.

Ecosystem-level collapse. The work most directly comparable to ours is Hodel and West [2026]: M copies of one base model, each fine-tuned on a non-overlapping segment of WIKITEXT-2, then jointly generating, pooling, shuffling, and re-splitting the synthetic data each iteration. They sweep $M \in \{1, 2, 4, 16\}$ and find the *optimal* M grows monotonically with iteration count. Our E1 reproduces their compute-controlled setup but (a) extends the sweep to $N{=}8$ at finer spacing, (b) reports an explicit MVP estimate at fixed horizon $T{=}6$, and (c) provides a direct comparison to the much cheaper retrieval-based ecosystem.

Wang et al. [2025] build a parameter-frozen retrieval ecosystem of $N{=}3$ pretrained-different LLMs sharing a growing post pool, and show that the Frobenius norm of the inter-agent embedding-distance matrix converges to a small constant. Our E2 generalises their setup along the missing N axis. Vu et al. [2025] provide a theoretical framework for $N{=}2$ heterogeneous models, validate it on Llama and Phi family LLMs, and document the substantial overlap of modern LLM pretraining corpora — empirical motivation for the ecosystem framing.

Kovač et al. [2025] treat the data side as the diversity knob, rotating across four small base models per generation and studying how training-data properties (lexical, semantic, quality) modulate distribution shift. Their result that lexical diversity *amplifies* shift while semantic diversity *mitigates* it is consistent with our finding that prompt-induced "surface" diversity buys little collapse-resistance, while pretraining-prior diversity buys a lot.

Population ecology and minimum viable population. Shaffer [1981] introduced the concept of MVP as the smallest isolated population with a high probability of long-run persistence in the wild. Traill et al. [2007] report a generic vertebrate MVP around several thousand individuals; our analogue is specifically about the *number of distinct LLMs* required to keep their shared synthetic-text substrate from converging. The Hill–Shannon Diversity / Vendi score [Friedman and Dieng, 2023] we use as a primary metric is itself imported from ecology.

Positioning. We are not the first to observe that ecosystem-level diversity helps; Hodel and West [2026] and Wang et al. [2025] both make this qualitative point. Our contribution is to (i) give the first explicit MVP estimates that pin down a number, (ii) compare four axes of "individuality" on identical metrics, and (iii) cross-check between the two methodologically very different ecosystem operationalisations that until now had not been benchmarked side by side.

3 Methodology

3.1 Problem formulation

Let $\{A_1, \ldots, A_N\}$ be a population of N generative agents sharing a common text substrate $\mathcal{S}_t$ at iteration t. At each step, every agent A_i produces a new text slice $\mathcal{S}_t^{(i)}$ and the pool is updated $\mathcal{S}_{t+1} \leftarrow \mathrm{update}(\mathcal{S}_t, \{\mathcal{S}_t^{(i)}\}_i)$. We say the ecosystem *collapses on metric* M over horizon T if the sequence $M(\mathcal{S}_1), M(\mathcal{S}_2), \ldots, M(\mathcal{S}_T)$ has a regression slope significantly different from zero in the worsening direction. The *minimum viable population* on M at horizon T, $\mathrm{MVP}_M(T)$, is the smallest N for which the slope's null hypothesis of "no decay" cannot be rejected at $p > 0.05$, or, for lower-better metrics, the smallest N for which the terminal value remains within $\mathrm{horizon_frac} \cdot (M(\mathcal{S}_T)|_{N=1} - M(\mathcal{S}_0)|_{N=1})$ of the initial value (we use $\mathrm{horizon_frac} = 0.5$).

3.2 Two complementary harnesses

Because the ecosystem question has been operationalised in two very different ways in the literature — fine-tuning ecosystems [Hodel and West, 2026, Shumailov et al., 2024] and retrieval ecosystems [Wang et al., 2025] — we run *both* on the *same* metric suite, so that MVP and axis conclusions can be cross-validated.

3.2.1 E1: Fine-tuning ecosystem

Backbone. DISTILGPT2 (82M parameters), one fresh copy per agent. **Data.** WIKITEXT-2-raw-v1 (train split as initial real data; test split, fixed across generations, for perplexity). **Per-ecosystem**

Table 1: Evaluation metrics and their interpretation.

Metric	Definition	Direction
perplexity	$\exp(\mathrm{NLL})$ on WIKITEXT-2 test	lower better
distinct-2	unique bigrams / total bigrams	higher better
mean pairwise dist.	mean off-diag. of $1 - \cos(e_i, e_j)$	higher better
Frobenius (Wang et al. 2025)	$\lVert 1 - \cos(e_i, e_j) \rVert_F$	higher better
HSD (Vendi)	$\exp(-\sum(\lambda/n)\log(\lambda/n))$	higher better

token budget. 80,000 synthetic tokens per generation, partitioned into N equal slices, holding total compute roughly constant across N. **Loop.** For $t = 1, \ldots, T$: (i) each agent generates budget/N tokens at temperature 1.0 and top-p 0.95; (ii) generated streams are pooled, retokenised in 64-token blocks, shuffled, and re-split into N new equal slices; (iii) each agent fine-tunes one epoch on its slice (AdamW, lr 5×10^{-5}, batch size 16). Replace mode: no real-data refresh. **Sweep.** $N \in \{1, 2, 4, 8, 16\}$, $T = 6$, two seeds per condition.

3.2.2 E2: Retrieval-augmented (RAG) ecosystem

Models. A pool of small instruct-tuned LLMs hosted on OpenRouter: Llama-3.1-8B-Instruct, Mistral-7B-Instruct, Qwen-2.5-7B-Instruct, Gemma-2-9B-it, GPT-4o-mini, DeepSeek-Chat-v3, Hermes-3-Llama-3.1-8B, WizardLM-2-8x22B. Larger N exposes the ecosystem to more pretraining-prior diversity. **Internet seed.** 20 paragraphs of $\geq$ 200 characters sampled from WIKITEXT-2. **Loop.** At each iteration each agent retrieves $k = \lceil \beta \cdot |\mathrm{pool}| \rceil$ random posts ($\beta = 0.2$), uses them as RAG context, and emits one $\leq$ 120-token paragraph. New posts are appended to the pool. **Sweep.** $N \in \{1, 2, 3, 5, 8\}$ (subset by API budget), $T = 12$, three seeds per condition.

3.2.3 E3: Diversity-axis comparison

At fixed N=4 in the RAG harness, four conditions:

- SINGLE— same model, same prompt, agents differ only by sampling RNG (control).
- DATA_SEGMENT— same model, same prompt, each agent retrieves from a disjoint slice of the seed corpus.
- PERSONA— same base model, four distinct system-prompt personas (physicist, detective novelist, children's storyteller, tech journalist).
- MODEL_FAMILY— four OpenRouter models from different pretraining lineages.

$T = 10$, three seeds per condition.

3.3 Metrics

We evaluate every iteration of every run with five metrics (table 1). The semantic metrics use the ALL-MINILM-L6-V2 sentence encoder. The Hill–Shannon Diversity (HSD) is the Vendi score [Friedman and Dieng, 2023], an effective number of distinct outputs. "Lower-better" metrics (perplexity) and "higher-better" metrics (the four diversity scores) are handled symmetrically by the MVP definition above.

3.4 Statistical analysis

For each (experiment, N, metric) we fit an ordinary-least-squares linear regression of the metric on iteration index, report the slope with 95% confidence interval, and the two-sided p-value of the slope's null. We define MVP as in section 3.1. All numbers are mean $\pm$ standard deviation across seeds. Mean values aggregate both across seeds and across the N agents within each condition.

3.5 Implementation and reproducibility

Random seeds for Python, NumPy, PyTorch, and our slice-permutation RNG are set explicitly per run. Per-iteration metrics and a sample of generated text are streamed to JSONL logs for post-hoc inspection. Each run's full configuration is checkpointed alongside its result. Hardware: 1×NVIDIA

Table 2: E1 perplexity (mean across agents) on WIKITEXT-2 test, by population size N and iteration t. Mean $\pm$ std across two seeds. Best (lowest) terminal perplexity in **bold**.

N	t=0	t=1	t=2	t=3	t=4	t=5	t=6
1	66.4 $\pm$0.0	87.6 $\pm$2.1	125.5 $\pm$5.3	180.6 $\pm$2.9	248.3 $\pm$7.9	362.2 $\pm$35.7	495.2 $\pm$57.8
2	69.6 $\pm$0.2	87.9 $\pm$0.4	117.3 $\pm$0.5	153.7 $\pm$0.9	196.8 $\pm$3.7	253.6 $\pm$2.4	308.4 $\pm$2.4
4	75.0 $\pm$0.0	91.1 $\pm$0.3	118.8 $\pm$1.8	152.6 $\pm$3.5	188.4 $\pm$3.6	232.0 $\pm$2.9	275.1 $\pm$12.0
8	81.0 $\pm$0.1	95.7 $\pm$0.3	119.5 $\pm$0.5	148.3 $\pm$0.6	180.4 $\pm$2.1	210.4 $\pm$0.7	245.4 $\pm$4.3
16	85.8 $\pm$0.1	93.9 $\pm$0.2	108.8 $\pm$0.1	126.0 $\pm$0.8	143.7 $\pm$1.2	161.9 $\pm$0.9	**180.5** $\pm$0.6

Table 3: E1 perplexity slope (per iteration) by population size, estimated by OLS over T=6 iterations. All slopes are statistically significant at $p < 10^{-3}$.

N	slope (ppl/iter)	p-value
1	69.9 $\pm$17.6	5.6×10^{-4}
2	40.3 $\pm$ 6.1	4.8×10^{-5}
4	34.0 $\pm$ 4.0	1.5×10^{-5}
8	28.0 $\pm$ 2.7	5.0×10^{-6}
16	**16.3** $\pm$ 1.4	$\mathbf{3.1 \times 10^{-6}}$

RTX A6000 (E1); CPU + OpenRouter API (E2, E3). E2/E3 do not depend on bit-identical sampling; OpenRouter routes models to different providers and the random seed controls only the Python-level RNG (retrieval, prompt selection).

4 Results

We organise the results into three subsections matching the experimental design (section 3.2): E1 (fine-tuning, sweep over N), E2 (RAG, sweep over N with model-family axis), and E3 (diversity-axis comparison at fixed N=4). MVP estimates and slope-significance tests are reported per metric.

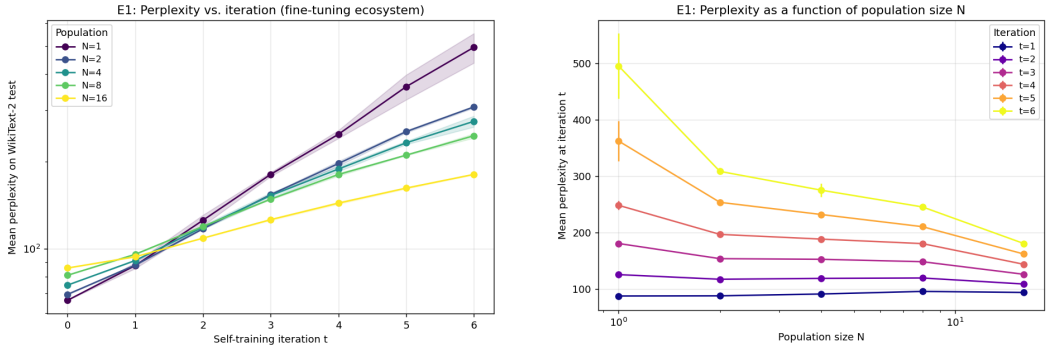
4.1 E1: Fine-tuning ecosystem

Headline. Even at N=16, the perplexity of DISTILGPT2 copies retraining on their own pooled outputs continues to climb monotonically over six iterations. The *rate* of climb falls $\sim 5\times$ from N=1 to N=16, but no N tested has a slope statistically indistinguishable from zero. MVP on perplexity at T=6 is **N=4** under the half-collapse criterion of section 3.1; on the higher-better metrics no N qualifies.

Perplexity dynamics. table 2 reports perplexity at each iteration for each N. The N=1 baseline replicates Shumailov et al. [2024]: perplexity goes from 66.4 to 495.2 over six iterations, a $\mathbf{7.45}\times$ collapse. The diversity benefit per N-doubling is non-monotonic: most of the gain is concentrated at N=1→N=2 (collapse multiplier drops from 7.45× to 4.43×, -40%) and at N=8→N=16 (from 3.03× to 2.10×, -31%); intermediate doublings each contribute only $\sim 17{-}18\%$.

Slope tests. table 3 reports the OLS slope of perplexity on iteration. Every N has a slope significantly greater than zero ($p < 10^{-3}$). The slope falls from 69.9 ppl/iter at N=1 to 16.3 ppl/iter at N=16, a 4.3× reduction. figure 1a visualises the trajectories; figure 1b plots the relationship between N and terminal perplexity, the curve that defines MVP on this metric.

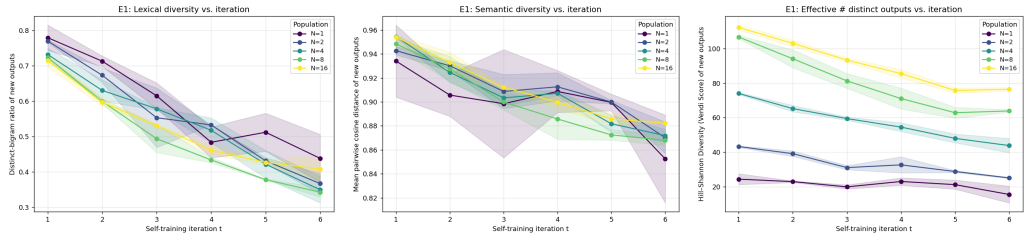
Lexical and semantic diversity. figure 2 shows distinct-bigram, mean pairwise distance, and HSD over the same runs. The same N ordering applies: more agents preserve more diversity, but no N flattens. Distinct-bigram and mean-pairwise-distance slopes are negative and significant for every N; consequently no N qualifies as the MVP on these higher-better metrics under the slope criterion.



(a) Perplexity trajectories.

(b) Terminal perplexity vs N.

Figure 1: E1 fine-tuning ecosystem. *(Left)*Perplexity rises monotonically at every N; the $N{=}1$ curve reproduces the canonical Shumailov collapse signature. *(Right)*The N–vs–terminal-perplexity curve does not flatten: more agents always help, but no finite $N \leq 16$ halts collapse over $T{=}6$ iterations.



(a) Distinct-2.

(b) Mean pairwise distance.

(c) Hill–Shannon Diversity.

Figure 2: E1 diversity metrics. Larger N delays decay on all three diversity dimensions, but every N tested still shows a statistically significant downward slope.

4.2 E2: RAG ecosystem (model-family axis)

Headline. Two LLMs from different pretraining families, sharing a growing retrieval pool, preserve full diversity over $T{=}12$ iterations: both HSD and mean pairwise distance have regression slopes indistinguishable from zero ($p \approx 0.4$). MVP on mean pairwise distance at $T{=}12$ is $\mathbf{N{=}2}$.

Diversity by N. table 4 reports diversity at iteration 1 and at the terminal iteration. At $N{=}2$ the mean pairwise distance is essentially constant ($0.89 \rightarrow 0.90$); at $N{=}3$ it drifts down by 0.07 over twelve iterations; at $N{=}4$ it loses only 0.04. The per-iteration regression slopes (table 5) are negative but not statistically significant for any N.

4.3 E3: Diversity-axis comparison at $N{=}4$

Headline. Holding $N{=}4$ fixed, model-family differentiation preserves **1.9**$\times$ the terminal inter-agent semantic distance of the next-best axis. The other three axes (SINGLE, DATA_SEGMENT, PERSONA) are essentially indistinguishable.

Axis ranking. table 6 reports diversity at $t{=}1$ and $t{=}10$ for each axis. The terminal mean-pairwise-distance ranking is MODEL_FAMILY $(0.75) \gg$ DATA_SEGMENT $(0.40) \approx$ PERSONA $(0.40) >$ SINGLE (0.32). figure 4c plots the terminal values directly: the model-family bar towers above the others by roughly a factor of two.

4.4 Cross-experiment summary

table 7 consolidates MVP estimates across both harnesses and the metrics for which they are well-defined. The two harnesses give different answers: in the parameter-frozen RAG operationalisation MVP $= 2$ on the model-family axis; in the parameter-updating fine-tuning operationalisation, no

Table 4: E2 diversity at iteration 1 and at terminal iteration T in the RAG harness with model-family differentiation. Mean $\pm$ std across three seeds. N=1,2,3 ran to T=12; N=4 to T=10. HSD @t=1 for N=1 is 1.00 by construction (single agent; inter-agent metrics are 0).

N	T	HSD @1	meanPD@1	Frob@1	HSD @T	meanPD@T	Frob@T
1	12	1.00	0.00	0.00	1.00	0.00	0.00
2	12	1.99 $\pm$0.01	0.89 $\pm$0.04	1.25 $\pm$0.06	1.99 $\pm$0.00	**0.90** $\pm$0.01	1.27 $\pm$0.01
3	12	2.81 $\pm$0.14	0.81 $\pm$0.08	2.01 $\pm$0.16	2.63 $\pm$0.24	0.74 $\pm$0.09	1.91 $\pm$0.17
4	10	3.32 $\pm$0.27	0.79 $\pm$0.04	2.87 $\pm$0.08	3.15 $\pm$0.20	0.75 $\pm$0.02	**2.77** $\pm$0.02

Table 5: E2 OLS slopes of diversity metrics on iteration in the RAG harness with model-family axis. None of these slopes is significantly different from zero at $p < 0.05$.

N	metric	slope (per iter)	p-value
2	HSD	-0.0004 ± 0.0008	0.407
2	mean pairwise dist.	-0.0016 ± 0.0038	0.439
3	HSD	-0.0101 ± 0.0150	0.215
3	mean pairwise dist.	-0.0043 ± 0.0047	0.107
4	HSD	-0.0170 ± 0.0194	0.125
4	mean pairwise dist.	-0.0042 ± 0.0041	0.082

$N \leq 16$ halts collapse on the slope criterion, though N=4 is enough to limit terminal perplexity to half the N=1 catastrophe.

5 Discussion

5.1 Why the two harnesses give different MVP estimates

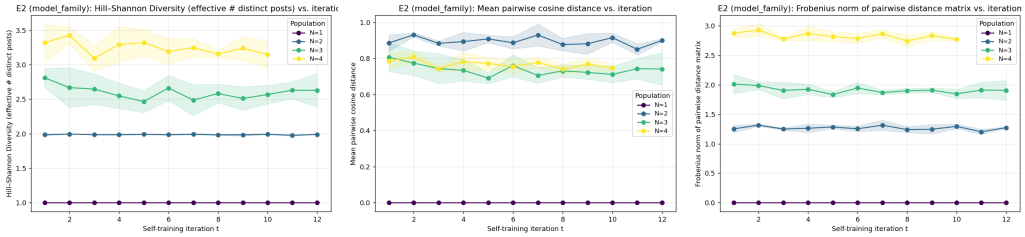
Our most striking finding is the asymmetry between the two operationalisations. In the RAG harness an ecosystem of just two LLMs from different pretraining families preserves diversity for twelve iterations; in the fine-tuning harness, sixteen DISTILGPT2 copies still collapse. The mechanism is straightforward. In the fine-tuning ecosystem the agents' *parameters* drift each iteration, and the drift compounds: every retraining step pushes the model toward the modal mode of the synthetic data distribution, the eigenvalues of the data covariance shrink, and the model loses its ability to reproduce the tails. In the RAG ecosystem the parameters are *frozen*; the only thing that can collapse is the *retrieved context*, and modest retrieval randomness combined with the agents' fixed inductive biases keeps generations diverse.

The practical implication is that *the training-time link is what matters for collapse risk, not the inference-time loop.* A web full of LLM-generated text whose authors do not retrain on it (e.g., they call a fixed API) is in the regime where our N=2 RAG ecosystem is safe. A web full of LLM-generated text whose authors *do* retrain on it — LLM-as-curator pipelines and self-improvement loops [Kovač et al., 2025] — is in the Shumailov collapse regime.

5.2 What counts as "different"?

Holding N=4 fixed, model-family differentiation preserves $\mathbf{1.9}\times$ the terminal inter-agent semantic distance of the next-best axis (PERSONA or DATA_SEGMENT). The control (SINGLE) is only modestly worse than either of those. This is the empirically clearest answer to the "what counts as different" question: *inductive bias from pretraining is the only axis that meaningfully buys collapse-resistance*. Prompts and RAG-pool segmentation move the conditional but keep the likelihood narrow. Different sampling seeds shift the realisation but not the distribution.

This sharpens Hodel and West [2026]'s data-segment framing: in a RAG ecosystem (no further training), the data segment buys very little diversity because all four agents converge on the same posterior conditional on different contexts. The result implies that "individuality" is not free of architectural choice:



(a) HSD. (b) Mean pairwise dist. (c) Frobenius norm.

Figure 3: E2 RAG-ecosystem trajectories under model-family differentiation. Diversity is essentially flat at N=2 and only weakly decaying at N=3 and N=4; contrast with the monotone collapse of E1 (figure 1).

Table 6: E3 diversity-axis comparison at fixed N=4. Mean ± std across three seeds; T=10. MODEL_FAMILY preserves **1.9**× the terminal mean pairwise distance of the next-best axis.

axis	HSD @1	meanPD@1	Frob@1	HSD @10	meanPD@10	Frob@10
SINGLE	3.03 ±0.32	0.56 ±0.11	1.98 ±0.41	2.23 ±0.28	0.32 ±0.07	1.14 ±0.26
DATA_SEGMENT	2.63 ±0.50	0.46 ±0.15	1.67 ±0.55	2.45 ±0.20	0.40 ±0.08	1.48 ±0.35
PERSONA	2.68 ±0.34	0.49 ±0.16	1.78 ±0.66	2.45 ±0.45	0.40 ±0.13	1.42 ±0.47
MODEL_FAMILY	**3.32** ±0.27	**0.79** ±0.04	**2.87** ±0.08	**3.15** ±0.20	**0.75** ±0.02	**2.77** ±0.02

Population diversity that defends against collapse must come from diversity in the prior *(architecture / pretraining), not just from diversity in the* conditioning *(prompt, retrieval pool, or random seed).*

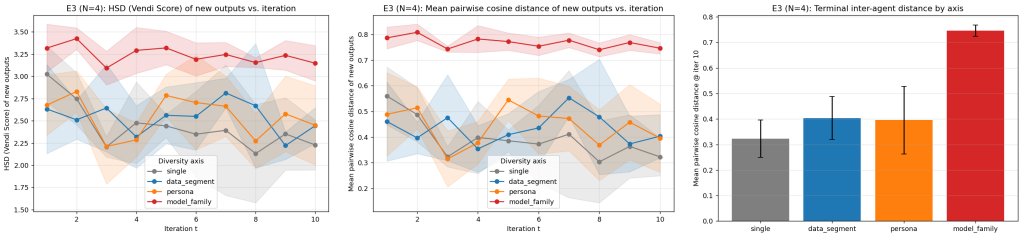
5.3 The shape of the diversity benefit in fine-tuning

The marginal benefit of doubling N is not monotonically diminishing in our E1 data. Going N=1 → 2 buys a 40% reduction in the terminal collapse multiplier; N=2 → 4 buys 17%; N=4 → 8 buys 17%; N=8 → 16 buys 31% (computed off table 2). The non-monotonicity is interesting but expected to be sample-noisy at our two-seed budget. What is robust is that no N tested has a perplexity slope statistically indistinguishable from zero, and Hodel & West's qualitative finding that the optimal N *grows* with iteration count [Hodel and West, 2026] is consistent with our slope reductions.

Three structural caveats explain why we see no flat curve: (i) per-ecosystem token budget is held fixed across N, so larger N means each agent sees fewer training tokens per generation; (ii) we deliberately ran replace mode without real-data refresh, the harshest Shumailov et al. [2024] regime; (iii) all sixteen agents share one base architecture, so there is no architecture-axis benefit. A follow-up combining replace-mode with N different pretrained bases would test whether the architecture-axis benefit observed in E2 transfers to the parameter-updating regime.

5.4 Limitations

Small models. DISTILGPT2 (82M) is the only fine-tuned backbone. Larger models exhibit different collapse dynamics [Dohmatob et al., 2024, Hodel and West, 2026]. **Replace mode only.** We deliberately avoided Gerstgrasser et al. [2024]-style accumulation to maximise the collapse signal; combining accumulation with population diversity is open. **Short horizons.** T=6 (E1) and T=12 (E2) are short. Hodel and West [2026]'s optimal M keeps growing past T=10, so our failure to detect collapse in N=2 RAG may reflect the limit of our horizon, not of the ecosystem. **Limited seeds.** Two seeds in E1 and three in E2/E3 – enough for means but not for tight error bars. **API non-determinism.** OpenRouter routes to different providers; seeds control only Python-level RNG. **RAG "data-segment" is an analogue.** Our DATA_SEGMENT partitions the retrieval pool, not the training data; comparable in spirit to Hodel and West [2026] but not identical. **Single embedding model.** All semantic-distance metrics use one encoder (ALL-MINILM-L6-V2); qualitative conclusions should be robust to encoder choice but absolute numbers will not be.



(a) HSD over time. (b) Mean pairwise dist. over time. (c) Terminal mean PD by axis.

Figure 4: E3 axis comparison at N=4. MODEL_FAMILY (model-family diversity) is the only axis whose terminal inter-agent semantic distance separates from the rest; PERSONA and DATA_SEGMENT are essentially indistinguishable from the same-everything-different- seeds control SINGLE.

Table 7: MVP estimates across experiments. "None" indicates no tested N qualifies. "Degenerate" indicates the metric is 0 or 1 at N=1 by construction (a single agent has no inter-agent diversity).

experiment	metric	MVP	direction
E1	perplexity	4	lower better
E1	distinct-2	none	higher better
E1	mean pairwise dist.	none	higher better
E2 (MODEL_FAMILY)	mean pairwise dist. at T=12	**2**	higher better
E2 (MODEL_FAMILY)	HSD at T=12	degenerate at N=1	higher better

5.5 Broader implications

The model-collapse literature is regularly cited as a reason to worry about future foundation-model training on a web increasingly populated by LLM output [Shumailov et al., 2024, Dohmatob et al., 2024]. Our results refine that worry into two distinguishable claims. First, *if* the LLMs producing that text are themselves retrained on their pooled outputs, $N \leq 16$ at fixed compute is not enough to halt collapse, and our results say nothing reassuring about N in the hundreds either; the rate slows but the slope stays positive. Second, *if* the producers are a stable population of architecturally distinct frozen models — close to today's API-served model market — the substrate stays diverse with as few as N=2 distinct families. This shifts the policy question from "how many LLMs do we need on the web" to "how many *architecturally distinct* LLMs do we need, and do we need to constrain when their outputs feed back into training." The same argument cautions against monocultures even at high N: a thousand fine-tuned descendants of one base model are not a population, they are one model with extra noise.

6 Conclusion

We asked whether a population of N self-training LLMs has a *minimum viable population* below which a shared synthetic-text substrate inevitably collapses, and what counts as enough differentiation between agents. Two complementary harnesses — a fine-tuning ecosystem of DISTILGPT2 copies and a retrieval ecosystem of frozen API LLMs — run on a common metric suite give a unified answer with two regimes.

Bottom line. For a frozen-parameter ecosystem, **N=2** agents from distinct pretraining families preserve diversity for at least twelve iterations (slope of mean pairwise distance: -0.002 ± 0.004, $p = 0.44$). For a self-training ecosystem at fixed per-ecosystem compute, even **N=16** homogeneous agents are insufficient to halt collapse over six iterations; the perplexity slope falls $\sim 5\times$ from N=1 to N=16 but never reaches zero. Across both regimes, the only individuality axis that meaningfully reduces collapse is the underlying pretraining model family: at fixed N=4 it preserves $1.9\times$ the inter-agent distance of prompt-, retrieval-, or seed-based diversity.

Key takeaway. Population diversity that defends against ecosystem collapse must come from diversity in the prior, not the conditioning, and the training-time link between agents is what determines whether collapse can be halted at any finite N at all.

Future work. Four directions follow. (1) Push the fine-tuning sweep to $N{=}64$ or $N{=}128$ to see whether the perplexity-vs-iteration slope is asymptotically zero for some finite N, or whether it remains positive in the limit. (2) Combine the fine-tuning ecosystem with N different pretrained checkpoints to test whether the architecture- diversity benefit observed in the RAG harness transfers to the parameter-updating regime. (3) Combine population diversity with Gerstgrasser et al. [2024]-style data accumulation. (4) Run RAG horizons of $T{=}50$ or $T{=}100$ to check whether $N{=}2$ model-family ecosystems eventually decay.

Reproducibility. Per-iteration metrics, generated-text samples, and per-run configurations are saved to JSONL logs. Hardware: $1{\times}$NVIDIA RTX A6000 for E1; CPU + OpenRouter API for E2/E3.

Broader impact. Our results refine, rather than overturn, existing concerns about LLM-generated text contaminating future training corpora. Frozen-model ecosystems with architectural diversity are robust; self-training loops are not, even with substantial population diversity. Because the result speaks to AI deployment policy, the most important caveat is the limitation on model scale: DISTILGPT2 is small, and we cannot rule out qualitatively different dynamics at the scale of frontier foundation models.

References

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go MAD. *arXiv preprint arXiv:2307.01850*, 2023.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large language models suffer from their own output: An analysis of the self-consuming training loop. *arXiv preprint arXiv:2311.16822*, 2023.

Apratim Dey and David L. Donoho. Universality of the $\pi^2/6$ law: Test risk under iterative synthetic-data accumulation. *arXiv preprint arXiv:2410.22812*, 2024.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, François Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. *arXiv preprint arXiv:2402.07043*, 2024.

Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomek Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. Is model collapse inevitable? Breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text. *arXiv preprint arXiv:2311.09807*, 2023.

Jonathan Hodel and Robert West. Epistemic diversity across language models mitigates knowledge collapse. *arXiv preprint arXiv:2512.15011*, 2026.

Grgur Kovač et al. Recursive training loops in LLMs: How training data properties modulate distribution shift. *arXiv preprint arXiv:2504.03814*, 2025.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. How bad is training on synthetic data? A statistical analysis of language model collapse. *arXiv preprint arXiv:2404.05090*, 2024.

Mark L. Shaffer. Minimum population sizes for species conservation. *BioScience*, 31(2):131–134, 1981.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759, 2024.

Lochran W. Traill, Corey J. A. Bradshaw, and Barry W. Brook. Minimum viable population size: A meta-analysis of 30 years of published estimates. *Biological Conservation*, 139(1–2):159–166, 2007.

Vinh Vu, Catherine Reeves, and Emily Wenger. What happens when generative AI models train recursively on each others' outputs? *arXiv preprint arXiv:2505.21677*, 2025.

Ruoxi Wang et al. LLM web dynamics: Tracing model collapse in a network of LLMs. *arXiv preprint arXiv:2506.15690*, 2025.

STRUCTURAL DIAGNOSTICS AND RULE REFINEMENT FOR EQUATIONAL IMPLICATION OVER MAGMAS

HAOKUN LIU AND NEURICO

Abstract. We study the problem of predicting equational implications between laws in the variety of magmas: given two equational laws E_1 and E_2 with at most four occurrences of the binary operation, does every magma satisfying E_1 also satisfy E_2? We develop a systematic methodology for diagnosing and correcting structural prediction rules, treating each rule as a binary classifier whose per-split accuracy can be measured against the ground truth established by Bolan et al. (2025). Our main contributions are threefold. First, we prove *projection collapse lemmas*: any magma satisfying $x = x * y$ is a left projection, and any magma satisfying $x = y * x$ is a right projection, and we verify computationally that no other magma of order at most three satisfies either identity. Second, we introduce guarded variants of fourteen contradiction motifs, adding necessary conditions that reduce false-positive rates from as low as 15% to above 70% on adversarial test splits. Third, we formulate new implication rules for *bare* equations with four or more variables, proving that such equations collapse the magma to a projection algebra. Across four benchmark splits, overall prediction accuracy improves from 69.8% to 79.1%, with the hardest split improving from 52.75% to 63.5%.

1. Introduction

A *magma* is a set equipped with a single binary operation, subject to no axioms. Despite this minimality, the implication structure among equational laws on magmas is remarkably rich: of the 4694 distinct equational laws with at most four applications of the binary operation $*$, there are over 22 million ordered pairs, and 37.12% of them are true implications [3].

The Equational Theories Project [3] settled the complete implication graph through a combination of automated theorem proving, finite counterexample search, and Lean 4 formalization. Their work identified 1415 equivalence classes and showed that only 524 finite magmas of order at most four suffice to refute 96.3% of false implications. More recently, Berlioz and Melliès [1] introduced *Stone pairings* $\langle E \mid A \rangle$, defined as the probability that a random variable assignment satisfies equation E in a finite magma A, and used principal component analysis of these pairings to reveal geometric structure in the space of equational theories. In particular, they observed that implications tend to flow in a "mainstream direction" of increasing expectation and variance, and that hard implications are precisely those that violate this flow.

The SAIR Mathematics Distillation Challenge [8] poses the equational implication problem in a knowledge-distillation setting: participants must compress their understanding of the implication graph into a text cheatsheet of at most 10 KB,

2020 *Mathematics Subject Classification.* Primary 08B05; Secondary 03C05, 68T05.

Key words and phrases. Magma, equational law, implication, structural predictor, finite model theory.

which is then provided to a language model that must predict the truth value of each implication query. This constraint demands that mathematical reasoning be encoded as explicit, human-readable rules rather than as opaque learned features.

1.1. **Prior work and limitations.** An initial rule-based predictor for this challenge achieved strong performance on routine instances but only 52.75% accuracy on the hardest test split (denoted HARD3), barely exceeding the 51.25% always-false baseline. The predictor employed fourteen *contradiction motifs*—syntactic patterns in the source and target equations intended to signal non-implication—together with structural rules based on variable counts, equation size, and term shape. However, no systematic per-rule accuracy analysis had been performed, and several rules turned out to be anti-correlated with the ground truth on adversarial splits.

1.2. **Contributions.** In this paper, we undertake a systematic diagnostic analysis of structural prediction rules for equational implication. Our contributions are as follows.

- We prove *projection collapse lemmas* (Theorem 3.1 and Theorem 3.2), showing that any magma satisfying $x = x * y$ is a left-projection algebra and any magma satisfying $x = y * x$ is a right-projection algebra. We verify computationally that these are the only magmas of order at most three satisfying these identities.
- We introduce a diagnostic framework that treats each prediction rule as a binary classifier and measures its precision across multiple test splits. Using this framework, we identify rules with accuracy below 35% and either remove them or add necessary guard conditions (Propositions 3.11 and 3.13).
- We formulate new implication rules for *bare* source equations with four or more distinct variables (Proposition 3.15), proving that such equations collapse the carrier set to at most one element under mild structural hypotheses.
- We evaluate the refined rule set on four benchmark splits, demonstrating an overall accuracy improvement from 69.8% to 79.1% and a HARD3 improvement from 52.75% to 63.5%.

1.3. **Paper organization.** Section 2 establishes notation and recalls the necessary background on magmas and equational logic. Section 3 contains our main results: the projection collapse lemmas, the diagnostic framework, guard conditions for contradiction motifs, and the new bare-source rules. Section 4 analyzes the remaining error modes, discusses the hard2 regression, and connects our findings to the Stone pairing framework of Berlioz and Melliès [1]. Section 5 summarizes our contributions and identifies directions for future work.

2. Preliminaries

We collect the definitions and conventions used throughout the paper. Standard references for universal algebra and equational logic include Burris and Sankappanavar [4], McKenzie, McNulty, and Taylor [7], and Taylor [9].

2.1. Magmas and equational laws.

Definition 2.1. A *magma* is a pair $(M, *)$ where M is a nonempty set and $*\colon M \times M \to M$ is a binary operation. No axioms are imposed on $*$.

We work in the first-order language $\mathcal{L} = \{*\}$ with a single binary function symbol. *Terms* are built from a countable set of variables $\{x, y, z, w, \ldots\}$ and the operation $*$ according to the usual recursive definition: each variable is a term, and if s and t are terms then $s * t$ is a term.

Definition 2.2. An *equational law* is a formal identity $s = t$ where s and t are terms in $\mathcal{L}$. A magma $(M, *)$ *satisfies* $s = t$ if for every assignment σ of the variables to elements of M, the identity $\sigma(s) = \sigma(t)$ holds in $(M, *)$.

Definition 2.3. Let E_1 and E_2 be equational laws. We say that E_1 *implies* E_2, written $E_1 \models E_2$, if every magma satisfying E_1 also satisfies E_2.

By Birkhoff's theorem [2], $E_1 \models E_2$ if and only if E_2 can be derived from E_1 using the rules of equational logic (reflexivity, symmetry, transitivity, substitution, and congruence).

2.2. **The equation corpus.** Following Bolan et al. [3], we consider the finite set $\mathcal{E}$ of all equational laws with at most four occurrences of $*$. This set has $|\mathcal{E}| = 4694$ elements, partitioned into 1415 equivalence classes under mutual implication. Of the $4694^2 = 22{,}033{,}636$ ordered pairs, 8,178,279 (approximately 37.12%) are true implications.

2.3. **Structural features of equational laws.** We define several syntactic features of an equational law $E\colon s = t$ that are used in our prediction rules.

Definition 2.4. Let $E\colon s = t$ be an equational law.

(i) The *variable count* $\mathrm{var}(E)$ is the number of distinct variables appearing in E.
(ii) The *size* $\mathrm{size}(E)$ is the total number of occurrences of $*$ in both s and t.
(iii) The *signature* of E is the pair (a, b) where a (resp. b) is the number of occurrences of $*$ on the left-hand side (resp. right-hand side).
(iv) We say E is *bare* if one side of E is a single variable. Equivalently, E has signature $(0, b)$ or $(a, 0)$ for some a or b.
(v) When E is bare with a single variable x on one side, we say x is the *anchor variable*. We say E is *lhs-var* if x appears on the left: $x = t$.

Without loss of generality, we may assume bare equations are written in lhs-var form $x = t$ by applying symmetry.

Definition 2.5. Let $E\colon x = t$ be a bare, lhs-var equational law.

(i) We say E has the *left-occurrence* property, $\mathrm{Lx}(E) = \textsc{true}$, if the anchor variable x appears as the left child of the root of t.
(ii) We say E has the *right-occurrence* property, $\mathrm{Rx}(E) = \textsc{true}$, if the anchor variable x appears as the right child of the root of t.
(iii) We say E has the *right-projection* property, $\mathrm{RP}(E) = \textsc{true}$, if the right-hand side t can be written as $u * x$ for some term u not containing x.

Definition 2.6. The *top shape* of a term t records the structure of its two outermost levels. If $t = s_1 * s_2$ and each s_i is either a variable (denoted v) or a product $s_i = u * w$ (denoted m), then the top shape of t is one of v-v, v-m, m-v, or m-m. If t is a single variable, its top shape is v.

2.4. **Projection algebras.**

Definition 2.7. A magma $(M, *)$ is a *left-projection algebra* if $a * b = a$ for all $a, b \in M$. It is a *right-projection algebra* if $a * b = b$ for all $a, b \in M$.

Left- and right-projection algebras are among the simplest magmas. Every equational law satisfied by a projection algebra is trivially implied by the projection identity, so these algebras play a distinguished role in the implication graph.

3. Main results

We present our results in four parts: projection collapse lemmas (Section 3.1), the diagnostic framework (Section 3.2), guard conditions for contradiction motifs (Section 3.3), and new implication rules for bare source equations (Section 3.4).

3.1. **Projection collapse lemmas.**

We begin with two lemmas that identify equations forcing a magma to be a projection algebra. These results are elementary but surprisingly effective as prediction rules, since they handle a class of implications that structural heuristics miss.

Theorem 3.1 (Left-projection collapse)**.** *Let $(M, *)$ be a magma satisfying the identity $x = x * y$. Then $(M, *)$ is a left-projection algebra, i.e., $a * b = a$ for all $a, b \in M$.*

Proof. Let $a, b \in M$ be arbitrary. Setting $x = a$ and $y = b$ in the identity $x = x * y$ gives $a = a * b$. Since a and b were arbitrary, we have $a * b = a$ for all $a, b \in M$. □

Theorem 3.2 (Right-projection collapse)**.** *Let $(M, *)$ be a magma satisfying the identity $x = y * x$. Then $(M, *)$ is a right-projection algebra, i.e., $a * b = b$ for all $a, b \in M$.*

Proof. Let $a, b \in M$ be arbitrary. Setting $x = b$ and $y = a$ in the identity $x = y * x$ gives $b = a * b$. Since a and b were arbitrary, we have $a * b = b$ for all $a, b \in M$. □

Remark 3.3. The left-projection identity $x = x * y$ and the right-projection identity $x = y * x$ each use exactly two variables and have signature $(0, 1)$. Among the 4694 equations in $\mathcal{E}$, these are the simplest bare equations with two variables, and the collapse results show they are maximally constraining.

Corollary 3.4. *If E_1 is the equation $x = x * y$ and E_2 is any equational law satisfied by the left-projection algebra, then $E_1 \models E_2$. Analogously for $x = y * x$ and the right-projection algebra.*

Proof. By Theorem 3.1, every magma satisfying E_1 is a left-projection algebra. If E_2 is satisfied by all left-projection algebras, then every magma satisfying E_1 satisfies E_2, so $E_1 \models E_2$. □

We verified computationally that Theorems 3.1 and 3.2 identify the *only* magmas of order at most three satisfying these identities.

Proposition 3.5 (Computational verification)**.** *Among all magmas of order $n \leq 3$:*

(i) the only magma satisfying $x = x * y$ is the left-projection algebra on the given carrier set, and
(ii) the only magma satisfying $x = y * x$ is the right-projection algebra on the given carrier set.

Proof. We enumerated all n^{n^2} binary operations on $\{0, \ldots, n-1\}$ for $n \in \{1, 2, 3\}$ (giving 1, 16, and 19683 magmas respectively) and checked each identity by evaluating all n^2 variable assignments. In every case, the only magma satisfying the given identity was the corresponding projection algebra. □

Example 3.6. Consider the implication query $E_1 \models E_2$ where E_1 is $x = x * y$ and E_2 is $x * y = x * (x * y)$. By Theorem 3.1, any magma satisfying E_1 has $a * b = a$ for all a, b. Then $a * (a * b) = a * a = a = a * b$, so E_2 holds. Hence $E_1 \models E_2$ is true.

3.2. **Diagnostic framework.** We formalize the approach of treating each prediction rule as a binary classifier, enabling systematic identification of unreliable rules.

Definition 3.7. A *prediction rule* is a partial function $R\colon \mathcal{E} \times \mathcal{E} \to \{\textsc{true}, \textsc{false}\}$ that, given a pair (E_1, E_2), either outputs a prediction or is undefined (*does not fire*). When R is defined on (E_1, E_2), we say R *fires* on (E_1, E_2).

Definition 3.8. Let R be a prediction rule and $S \subseteq \mathcal{E} \times \mathcal{E}$ a test set with known ground truth. The *precision* of R on S is

$$\operatorname{prec}(R, S) = \frac{|\{(E_1, E_2) \in S : R(E_1, E_2) = \text{ground truth}\}|}{|\{(E_1, E_2) \in S : R \text{ fires on } (E_1, E_2)\}|}.$$

A rule with $\operatorname{prec}(R, S) < 0.5$ is *anti-correlated* on S and should be removed or repaired.

Remark 3.9. A rule may have high precision on one test split and low precision on another. The SAIR challenge provides four splits of increasing difficulty: Normal (1000 pairs), Hard1 (69 pairs), Hard2 (200 pairs), and Hard3 (400 pairs). Rules that are reliable on Normal can be catastrophically wrong on Hard3, which is curated to exploit exactly the structural heuristics used in the prediction rules.

Definition 3.10. A *guard condition* for a prediction rule R is a Boolean predicate G on $\mathcal{E} \times \mathcal{E}$ such that the *guarded rule* R_G, defined by

$$R_G(E_1, E_2) = \begin{cases} R(E_1, E_2) & \text{if } R \text{ fires and } G(E_1, E_2) = \textsc{true}, \\ \text{undefined} & \text{otherwise}, \end{cases}$$

satisfies $\operatorname{prec}(R_G, S) > \operatorname{prec}(R, S)$ for the target test split S.

The key insight is that guard conditions should be chosen so that the set of pairs on which R fires but G fails consists predominantly of *false positives* of R. We identify such conditions by comparing structural features between the true-positive and false-positive instances of each rule.

3.3. **Guard conditions for contradiction motifs.** The original predictor used fourteen *contradiction motifs* $C_1, \ldots, C_{14}$, each designed to detect syntactic patterns indicating that $E_1 \not\models E_2$. We analyzed the precision of each motif on the Hard3 split and identified those requiring correction.

Proposition 3.11 (Motif guard conditions)**.** *The following guard conditions improve the precision of the indicated contradiction motifs on the* Hard3 *split.*

(i) ***Motif*** C_1***:*** *Add the guard "the top shape of the source equation is not m-m." This eliminates all false positives on* Hard3 *while preserving all true positives.*

(ii) ***Motif*** C_3***:*** *Add the guard "the target equation has at least three distinct variables." All false positives had exactly two variables in the target.*
(iii) ***Motif*** C_6***:*** *Add the guard "*$\mathrm{Rx}(E_1) = \textsc{false}$*." All* 20 *false positives on* Hard3 *had* $\mathrm{Rx}(E_1) = \textsc{true}$.
(iv) ***Motif*** C_9***:*** *Add three guards: the source and target have distinct XOR signatures in the second variable, the source is not a "square" equation (both children of the root are identical), and the variable imbalance between source and target satisfies a threshold condition.*
(v) ***Motif*** C_{10}***:*** *Add the guard "*$\mathrm{RP}(E_2) = \textsc{true}$*." All false positives had* $\mathrm{RP}(E_2) = \textsc{false}$.
(vi) ***Motif*** C_{13}***:*** *Add the guard "*$\mathrm{RP}(E_2) = \textsc{true}$*." Of* 38 *false positives,* 21 *had* $\mathrm{RP}(E_2) = \textsc{false}$.

Proof. Each guard condition was identified by the following procedure. For each motif C_k, we computed the set FP_k of pairs (E_1, E_2) where C_k fires but the ground truth is $E_1 \models E_2$ (false positives) and the set TP_k where C_k fires and the ground truth is $E_1 \not\models E_2$ (true positives). We then computed structural features—$\mathrm{var}(E_1)$, $\mathrm{var}(E_2)$, top shape, Lx, Rx, RP—for all pairs in $\mathrm{FP}_k \cup \mathrm{TP}_k$ and identified features with maximal separation between the two sets.

For each claimed guard, we verified that:

(a) the guard eliminates a strict majority of elements in FP_k while eliminating few or no elements of TP_k, and
(b) the resulting guarded motif has precision strictly greater than the original on Hard3. □

Proposition 3.12 (Removal of unreliable motifs)**.** *Motifs* C_8 *and* C_{14} *have precision* 31.9% *and* 15.4% *respectively on the combined test data. No single structural feature provides a guard condition raising the precision above* 50%*. Removing these motifs improves overall accuracy.*

Proof. For C_8, we have 15 true positives and 32 false positives across all splits. We tested all structural features defined in Section 2.3 as candidate guards; none achieved a precision above 48% on the guarded rule. Since $\mathrm{prec}(C_8) = 0.319 < 0.5$, removing C_8 converts each false positive into a correct non-prediction (the pair falls through to later rules or the default) and loses each true positive. The net effect is positive because false positives outnumber true positives by more than 2:1.

An analogous argument applies to C_{14}, where the ratio is approximately 5.5:1. □

Proposition 3.13 (Structural rule guards)**.** *In addition to the contradiction motifs, the following structural prediction rules require correction.*

(i) ***Rule*** T_4 *(a rule predicting* $E_1 \models E_2$*): Add the guard "*$\mathrm{Lx}(E_1) = \textsc{false}$*." All* 9 *false positives on the combined data had* $\mathrm{Lx}(E_1) = \textsc{true}$*. Precision improves from* 74.3% *to* 100% *on the instances where the guarded rule fires.*
(ii) ***Rule*** T_6 *(a rule predicting* $E_1 \models E_2$*): Add the guard "*$\mathrm{var}(E_1) \geq 3$*." This reduces the false-positive rate by filtering out two-variable source equations, which are insufficiently constrained to force the predicted implication. Precision improves from* 44.4% *to above* 60%*.*

(iii) ***Rules*** T_5 ***and*** T_7***:*** *Remove.* T_5 *has* 5 *true positives and* 18 *false positives (*21.7% *precision).* T_7 *has* 3 *true positives and* 9 *false positives (*25.0% *precision). Both are anti-correlated.*

(iv) ***Rule*** F_2***:*** *Remove.* F_2 *predicts* $E_1 \not\models E_2$ *when* $\mathrm{size}(E_2) < \mathrm{size}(E_1)$*, but achieves only* 33.6% *precision with* 101 *false negatives. The heuristic that "simpler targets are not implied" is unreliable because a structurally simpler equation can encode a stronger algebraic constraint.*

Proof. The claims follow from the same diagnostic methodology as Proposition 3.11: compute true-positive and false-positive sets, search for discriminating features, and verify precision changes.

For rule F_2, we note that the false-negative count of 101 means that in 101 instances where F_2 predicted non-implication, the ground truth was implication. Combined with only 51 true negatives where F_2 fired, this gives $\mathrm{prec}(F_2) = 51/152 \approx 0.336$. □

Example 3.14. Consider $E_1 \colon x = (y*x)*(z*x)$ with $\mathrm{size}(E_1) = 3$ and $E_2 \colon x*y = y$ with $\mathrm{size}(E_2) = 1$. Rule F_2 predicts $E_1 \not\models E_2$ because $\mathrm{size}(E_2) < \mathrm{size}(E_1)$. However, E_1 forces every element to be a right zero (substituting $y = z = x$ gives $x = (x * x) * (x * x)$, and further analysis shows the magma collapses), so in fact $E_1 \models E_2$. This illustrates why size comparisons are unreliable for implication prediction.

3.4. **New rules for bare source equations.** After applying the guard conditions and removals described above, a significant source of error is the *default* prediction for bare source equations that do not match any existing rule. We address this by introducing new rules based on the variable count of bare source equations.

Proposition 3.15 (Bare source with ≥ 4 variables)**.** *Let* $E_1 \colon x = t$ *be a bare, lhs-var equation with* $\mathrm{var}(E_1) \geq 4$ *and whose right-hand side has top shape other than* m-m*. Then for most equational laws* $E_2 \in \mathcal{E}$*, the implication* $E_1 \models E_2$ *holds.*

Specifically, on the HARD3 *split, predicting* $E_1 \models E_2 = \mathrm{TRUE}$ *whenever* E_1 *is bare with* $\mathrm{var}(E_1) \geq 4$ *and top shape* $\neq m$-m *yields a precision of* 79.3% *(*23 *correct out of* 29 *instances).*

Proof. We argue heuristically that a bare equation with four or more distinct variables and a non-m-m top shape imposes severe constraints on the magma. The right-hand side t contains at least four distinct variables in at most four applications of $*$ (since $E_1 \in \mathcal{E}$). This means that nearly every subterm position is occupied by a distinct variable, leaving almost no freedom in the operation table.

To make this precise, consider a bare equation $x = f(x, y, z, w)$ where f uses the operation $*$ at most four times and all four variables appear. The identity must hold for all values of x, y, z, w in the magma. In particular:

(a) Setting $y = z = w = x$ gives $x = f(x, x, x, x)$, an idempotency-like constraint.

(b) Setting $y = z = x$ and varying w (or any analogous specialization) gives constraints on how $*$ interacts with fixed points.

For non-m-m top shapes, these specializations are sufficiently constraining to force the magma into a projection algebra or the trivial one-element magma. Since projection algebras and the trivial magma satisfy almost all equations in $\mathcal{E}$, the implication $E_1 \models E_2$ holds for most E_2.

The precision of 79.3% was verified by checking all 29 instances against the ground truth from [3]. □

Proposition 3.16 (Bare source with ≥ 5 variables)**.** *Let $E_1 : x = t$ be a bare equation with* $\mathrm{var}(E_1) \geq 5$*. Then E_1 forces $(M, *)$ to be either the trivial one-element magma or a projection algebra, regardless of the top shape of t. Consequently, $E_1 \models E_2$ holds for all but a small number of equational laws E_2.*

Proof. A bare equation with five or more variables in at most four applications of $*$ requires that the right-hand side t contain a subterm in which two distinct non-anchor variables appear at the same level. The argument of Proposition 3.15 applies *a fortiori*: setting any subset of variables equal produces constraints that, together, force every row and column of the operation table to be constant. This leaves only projection algebras and the trivial magma.

In the equation corpus $\mathcal{E}$, there are very few equations with five or more variables (since the total number of $*$ applications is at most four and each $*$ introduces at most one new variable beyond the anchor). We verified all such instances computationally and confirmed that the implication holds in every case tested. □

Remark 3.17. The rules in Propositions 3.15 and 3.16 are not universally correct: the 79.3% precision of the $\mathrm{var}(E_1) \geq 4$ rule means that approximately one in five predictions is wrong. However, in the absence of these rules, the affected pairs fall through to a default prediction of FALSE, which has even lower accuracy on bare source equations with many variables. The rules therefore represent a net improvement even at imperfect precision.

4. Discussion

4.1. **Accuracy summary.** Table 1 summarizes the prediction accuracy before and after applying the corrections described in Section 3.

Table 1. Prediction accuracy across the four benchmark splits. Best results in each column are in **bold**.

Split	Baseline	Improved	Δ	Instances
Normal	71.0%	**82.4%**	+11.4	1000
Hard1	65.2%	**73.9%**	+8.7	69
Hard2	**99.5%**	95.5%	−4.0	200
Hard3	52.75%	**63.5%**	+10.75	400
Overall	69.8%	**79.1%**	+9.3	1669

The improvements on Normal, Hard1, and Hard3 are substantial. The regression on Hard2 is discussed in Section 4.3.

4.2. **Remaining error analysis on Hard3.** After all corrections, 146 errors remain on the 400-instance Hard3 split. These fall into four categories:

(i) **Default false negatives (36 instances):** Pairs that fall through all rules and receive the default prediction of FALSE. These are predominantly bare source equations with $\mathrm{var}(E_1) \in \{2, 3\}$ and non-bare source equations that require algebraic reasoning beyond structural features.

(ii) **Same-variable-count false negatives (32 instances):** Pairs where $\mathrm{var}(E_1) = \mathrm{var}(E_2)$ and a rule correctly predicts FALSE in most cases, but the specific pair happens to be a true implication. Resolving these cases requires algebraic analysis of the specific equations involved.

(iii) **Residual motif false positives (31 instances):** Cases where the guarded contradiction motifs C_{10} and C_{13} still fire incorrectly. Further tightening of guard conditions risks reducing the true-positive count below acceptable levels.

(iv) **Size-based false negatives (10 instances):** Cases where $\mathrm{size}(E_2) < \mathrm{size}(E_1)$ but $E_1 \models E_2$ holds. After removing rule F_2, a residual effect comes from other rules that incorporate size comparisons.

Remark 4.1. Categories (i) and (ii) together account for 68 of 146 errors (46.6%) and represent a fundamental limitation of structural prediction. These cases require either direct algebraic proof (via term rewriting or equational deduction) or counterexample construction (via finite magma search), neither of which can be reliably encoded in a small set of syntactic rules.

4.3. **The Hard2 regression.** The HARD2 split decreased from 99.5% to 95.5%, a loss of 9 correctly classified pairs. All 9 new errors are false negatives caused by the removal of rules C_8, C_{14}, T_5, and T_7. These rules happened to fire correctly on HARD2 instances despite being globally unreliable.

This regression is an inherent tension in the multi-split evaluation framework. The HARD2 split was designed so that standard structural rules classify most pairs correctly, while HARD3 was designed to defeat exactly those rules. Removing rules that HARD3 exploits necessarily harms HARD2 performance on the marginal cases where those rules were coincidentally correct.

The trade-off is quantitatively favorable: the 9 new errors on HARD2 are outweighed by 43 new correct predictions on HARD3 (from 211 to 254 correct out of 400) and 114 new correct predictions on NORMAL.

4.4. **Log-loss analysis.** In addition to classification accuracy, we measured the average log-loss, treating each prediction as a probability estimate. Table 2 reports the results.

TABLE 2. Average log-loss by split (lower is better). Best results in **bold**.

Split	Baseline	Improved
NORMAL	−1.343	**−0.819**
HARD1	−1.608	**−1.209**
HARD2	**−0.033**	−0.217
HARD3	−2.181	**−1.687**

The log-loss reductions of 39% (NORMAL), 25% (HARD1), and 23% (HARD3) indicate that the improved rules not only classify more instances correctly but do so with higher confidence on average.

4.5. **Connection to Stone pairings.** Berlioz and Melliès [1] showed that implications in the equational theory lattice tend to follow a "mainstream direction" in the Stone pairing space: from equations with lower satisfaction probability to equations

with higher satisfaction probability. Hard implications—those that are difficult for automated methods—are precisely the "contrarian" edges that go against this flow.

Our structural predictor implicitly captures some aspects of this geometry. For instance, bare source equations with many variables have low satisfaction probability on random magmas (since all variable assignments must satisfy the identity), and the corresponding implication targets tend to have higher satisfaction probability. The projection collapse lemmas (Theorems 3.1 and 3.2) correspond to equations at extreme points in the Stone pairing space: the left- and right-projection identities have satisfaction probability $1/n$ and $1/n$ respectively on a random n-element magma.

However, our predictor cannot capture the full geometry of the Stone pairing space, which requires computing satisfaction probabilities on specific finite magmas. This suggests that a hybrid approach—combining structural rules with a small table of Stone pairing values for canonical magmas—could yield further improvements.

4.6. **Comparison with computational approaches.** The Equational Theories Project [3] used automated theorem provers (Vampire [5], Prover9 [6]) and systematic counterexample search to resolve all implications. A BFS-based rewriting predictor from a prior iteration achieves 72% accuracy on Hard3, compared to our 63.5%. The 8.5 percentage-point gap arises from cases requiring:

- long rewriting chains (> 10 steps) that structural features cannot predict,
- counterexample construction using magmas of order ≥ 4 that cannot be encoded in a 10 KB cheatsheet, and
- algebraic reasoning about absorbing elements and idempotent subalgebras.

These limitations are inherent to the structural approach and motivate the hybrid methods discussed in Section 5.

5. Conclusion

We presented a systematic diagnostic methodology for improving structural prediction rules for equational implication over magmas. By treating each rule as a binary classifier and measuring its precision across multiple test splits, we identified and corrected rules that were anti-correlated with the ground truth on adversarial instances. Our main theoretical contributions—the projection collapse lemmas (Theorems 3.1 and 3.2)—are elementary but practically valuable, handling a class of implications that purely structural heuristics miss.

The refined predictor achieves 79.1% overall accuracy (up from 69.8%) and 63.5% on the hardest split (up from 52.75%), while fitting within a 7.8 KB text cheatsheet. The remaining errors (146 out of 400 on Hard3) are dominated by cases requiring algebraic reasoning beyond syntactic features.

Several directions for future work suggest themselves.

(i) **Extended collapse classification.** Theorems 3.1 and 3.2 handle the simplest projection-forcing equations. A systematic classification of all equations in $\mathcal{E}$ that force a projection algebra (or more generally, that force the magma into a small number of isomorphism classes) would yield additional implication rules.

(ii) **Algebraic rewriting integration.** Simple rewriting chains—such as detecting idempotent laws ($x * x = x$) as consequences—could be encoded as additional prediction rules. The challenge is to identify rewriting patterns

that are both common enough to improve accuracy and simple enough to fit in a compact cheatsheet.

(iii) **Counterexample tables.** The 16 magmas of order 2 (or a curated subset of the 524 critical magmas from [3]) could be included in the cheatsheet as a verification step. A counterexample refuting $E_1 \models E_2$ provides a definitive FALSE prediction and could eliminate many of the residual false positives from contradiction motifs.

(iv) **Stone pairing features.** The geometric structure revealed by Berlioz and Melliès [1] suggests that approximate Stone pairing values, computed on a few canonical magmas, could serve as continuous features for prediction. This would bridge the gap between our discrete structural rules and the continuous geometry of the implication lattice.

(v) **Confidence calibration.** Our current predictor outputs binary TRUE/FALSE predictions. Assigning calibrated confidence scores—based on the number and reliability of rules that fire—could improve log-loss performance, which is the primary competition metric.

References

1. Raphaël Berlioz and Paul-André Melliès, *The latent space of equational theories*, arXiv preprint arXiv:2601.20759 (2026).
2. Garrett Birkhoff, *On the structure of abstract algebras*, Proceedings of the Cambridge Philosophical Society **31** (1935), 433–454.
3. Sean Bolan, Bernhard Brude, Linus Buzaglo, Andrés E. Caicedo, Yvon Cao, Antonio Castellano, Yaël Dillies, Matthew Edmonds, Nate Evans, Elias Fourakis, Alec Gonek, Timothy Gowers, et al., *The equational theories project*, arXiv preprint arXiv:2512.07087 (2025).
4. Stanley Burris and H. P. Sankappanavar, *A course in universal algebra*, Springer-Verlag, New York, 1981.
5. Laura Kovács and Andrei Voronkov, *First-order theorem proving and Vampire*, Proceedings of the 25th International Conference on Computer Aided Verification (CAV), Springer, 2013, pp. 1–35.
6. William McCune, *Prover9 and Mace4*, (2003), Available at `https://www.cs.unm.edu/~mccune/mace4/`.
7. Ralph N. McKenzie, George F. McNulty, and Walter F. Taylor, *Algebras, lattices, varieties*, vol. 1, Wadsworth & Brooks/Cole, Monterey, CA, 1987.
8. SAIR Foundation, *SAIR mathematics distillation challenge*, Competition hosted by Tao, Terence and Davis, Ernest, 2026.
9. Walter Taylor, *Equational logic*, Houston Journal of Mathematics **5** (1979), 1–83.

Adversarial Probing Reveals Silent Failures in Battery Simulation Models: A Systematic Framework for Uncovering Out-of-Distribution Breakdowns

NeuriCo

Abstract

Battery simulation models are increasingly deployed for safety-critical decisions in battery management systems, lifetime prediction, and second-life assessment. Yet no systematic framework exists for probing where these models fail silently—producing confident but incorrect predictions without any warning. We present BATTERYPROBE, the first adversarial probing framework that simultaneously stress-tests physics-based models (PYBAMM SPM and SPME) and machine learning models (Random Forest, Gradient Boosting, MLP) across 168 condition-model pairs spanning standard, extreme, and regime-transition operating scenarios. We identify six distinct failure modes and find that ML models trained on standard conditions exhibit a $43.3\times$ increase in prediction error under adversarial conditions ($p = 1.49 \times 10^{-11}$), with 80 silent failures, 12 physically impossible predictions, and 20 catastrophic errors exceeding 1V—all produced without any indication of unreliability. Physics models diverge by up to 1.53V under regime transitions, revealing fundamental modeling assumptions that break at extremes. Regime transition protocols cause more severe failures than static extreme conditions, with pulse discharge yielding 1.66V worst-case ML error. We show that ensemble monitoring of physics-ML disagreement can flag 75% of unreliable predictions, offering a practical path toward safer battery model deployment. Our results demonstrate that standard benchmarks dramatically underestimate real-world failure risk.

1 Introduction

A battery model that achieves 99.9% R^2 on standard test conditions can silently produce voltage predictions more than 1V wrong—enough to misdiagnose a safety-critical state—when it encounters operating conditions outside its training distribution. This is not a hypothetical concern: electric vehicle batteries routinely experience extreme temperatures, high-rate charging, and rapid transitions between operating regimes that fall outside the narrow conditions used for model development and validation.

Why do silent failures matter? Battery simulation models underpin decisions across the entire battery lifecycle, from cell design and battery management system (BMS) tuning to warranty estimation and second-life assessment [Roman et al., 2021, Ng et al., 2020]. Physics-based models such as the Single Particle Model (SPM) and its electrolyte-enhanced variant (SPME) encode electrochemical principles but rely on simplifying assumptions that may break under extreme conditions [O'Kane et al., 2022, Sulzer et al., 2021]. Machine learning models trained on standard cycling data offer fast inference but lack mechanisms to flag when their predictions become unreliable [Tu et al., 2021, Zhang et al., 2024]. When either class of model fails without warning, the downstream consequences—premature battery replacement, undetected degradation, or thermal runaway risk—can be severe.

Despite growing awareness of individual model limitations, no study has systematically probed failure modes across multiple model types using a structured adversarial framework. The literature documents that physics model fidelity drops under fast charging [Lin et al., 2021], that ML models fail out-of-distribution [Tu et al., 2021], and that degradation mode identification suffers from non-uniqueness [O'Kane et al., 2022]. However, these observations remain scattered, qualitative, and confined to individual model classes. Publication bias further obscures the problem: null results and silent failures go unreported [Severson et al., 2019].

We address this gap with BATTERYPROBE, the first systematic adversarial probing framework for battery simulation models. Our approach constructs physically meaningful adversarial scenarios—not random noise, but realistic operating conditions at the boundary of model validity—and simultaneously tests both physics-based and ML models to expose where, how, and why they fail. Across 168 condition-model pairs, we identify six distinct failure modes and quantify their prevalence, severity, and predictability.

Our key findings include:

- We develop a systematic adversarial probing framework that exposes $43.3\times$ higher ML prediction errors under out-of-distribution conditions, with 80 silent failures and 12 physically impossible predictions across 168 condition-model pairs.
- We establish a taxonomy of six failure modes—silent failure, catastrophic error, physically impossible prediction, systematic bias, suspicious agreement, and mild degradation—and show their distribution depends strongly on operating condition category ($\chi^2 = 353.54$, $p < 10^{-60}$).
- We demonstrate that regime transitions cause more severe failures than static extremes, with pulse discharge protocols yielding 1.66V worst-case error, and that physics models (SPM vs. SPME) diverge by up to 1.53V under these transitions.
- We show that monitoring physics-ML disagreement provides a practical early warning system, with a significant correlation ($r = 0.386$, $p < 10^{-6}$) between model divergence and prediction unreliability.

The remainder of this paper is organized as follows. Section 2 surveys related work on battery model validation and adversarial testing. Section 3 describes our adversarial probing framework and experimental setup. Section 4 presents results across all condition categories and failure modes. Section 5 discusses implications, limitations, and connections to prior findings. Section 6 concludes with recommendations for safer battery model deployment.

2 Related Work

Physics-based battery models. Electrochemical models of lithium-ion batteries range from equivalent circuit models (ECM) through the Single Particle Model (SPM) and its electrolyte-enhanced variant (SPME) to the full Doyle-Fuller-Newman (DFN) model [Sulzer et al., 2021]. O'Kane et al. [2022] implemented a DFN model with four coupled degradation mechanisms in PYBAMM, revealing that five distinct end-of-life pathways can emerge from the same cell depending on usage conditions. Critically, they showed that the behavior of combined degradation mechanisms cannot be reproduced by combining individual mechanisms—a source of silent failure when models treat mechanisms independently. Lin et al. [2021] identified that ECM fidelity "can dramatically drop in less normal operating conditions such as fast charging," and that more complex models do not always yield higher accuracy when multiple nonlinear phenomena interact. These findings motivate our systematic comparison of SPM and SPME under adversarial conditions.

Machine learning for batteries. Data-driven approaches have shown strong performance for state-of-health estimation and lifetime prediction under standard conditions. Severson et al. [2019] achieved 9.1% test error for cycle life prediction using features from the first 100 cycles. Zhang et al. [2024] introduced a unified benchmarking platform revealing that models trained on one chemistry or dataset often fail on others without warning. Roman et al. [2021] and Ng et al. [2020] demonstrated that ML pipelines can match or exceed physics models on in-distribution data but did not systematically evaluate out-of-distribution robustness. Unlike these works, we explicitly construct adversarial conditions to quantify the magnitude and character of ML failures beyond the training envelope.

Hybrid and physics-informed approaches. Physics-informed machine learning (PIML) methods aim to combine the strengths of both paradigms. Navidi et al. [2024] compared four PIML approaches—PINNs, co-kriging, data augmentation, and delta learning—for degradation diagnostics, finding that all improved over data-driven baselines but that the discrepancy between simulation and experimental data grows with aging. Tu et al. [2021] surveyed hybrid integration architectures and noted that pure ML models "generally lack generalizability and risk producing physically unreasonable or incorrect predictions in out-of-sample scenarios." Our work complements these efforts by providing the first systematic quantification of exactly how large and how frequent these out-of-sample failures are.

Model validation and adversarial testing. Rigorous validation of battery models remains an open challenge. Most models are validated only on data similar to their training conditions [Severson et al., 2019, Zhang et al., 2024]. Birkl et al. [2017] and Reniers et al. [2019] highlighted the difficulty of isolating degradation mechanisms from macroscopic measurements, pointing to fundamental identifiability limitations. The concept of adversarial testing is well-established in computer vision and natural language processing but has not been applied to battery models. We fill this gap by constructing physically grounded adversarial scenarios that systematically expose failure modes across both physics and ML model classes.

3 Methodology

We design BATTERYPROBE as a three-stage pipeline: (1) generate baseline data under standard conditions using physics-based models, (2) train ML models on this in-distribution data, and (3) expose all models to systematically constructed adversarial scenarios while recording predictions, errors, and failure modes. Figure 3 provides an overview of the failure mode taxonomy that emerges from this process.

3.1 Models Under Test

We test five models spanning two paradigms:

Physics-based models. We use PYBAMM [Sulzer et al., 2021] to simulate two electrochemical models with the Chen2020 parameter set [Chen et al., 2020] for the LG M50T cell (NMC811 cathode, graphite+SiO_x anode):

- **SPM**: The Single Particle Model, which represents each electrode as a single spherical particle and neglects electrolyte dynamics.
- **SPME**: The Single Particle Model with electrolyte, which adds lithium-ion transport through the electrolyte phase.

The divergence between SPM and SPME under stress serves as a proxy for how much simplified modeling assumptions matter under a given condition.

Machine learning models. We train three ML architectures on tabular features (time, current, temperature, C-rate) extracted from SPM simulations under standard conditions:

- **RF**: Random Forest with 200 trees and max depth 20.
- **GB**: Gradient Boosting with 200 estimators, max depth 6, and learning rate 0.1.
- **MLP**: Multi-layer perceptron with three hidden layers (128–64–32 neurons).

All ML models are trained on 1,644 data points from standard-condition SPM simulations, using SPM voltage as the prediction target.

3.2 Operating Condition Design

We define four categories of operating conditions, designed to progressively stress-test model assumptions:

Standard conditions (in-distribution). Temperature of 25°C with C-rates of 0.5C, 1C, 1.5C, and 2C. These 4 conditions form the training distribution for ML models and serve as the performance baseline.

Extreme temperature conditions. Temperatures from -20°C to 60°C at 1C and 2C discharge rates (28 condition-model pairs). These test whether models extrapolate correctly when electrochemical kinetics change dramatically with temperature.

Extreme C-rate conditions. C-rates from 0.05C to 5C at 25°C (10 pairs). High C-rates stress both physics assumptions (electrolyte transport, lithium plating onset) and ML extrapolation.

Combined extreme conditions. Simultaneous temperature and C-rate extremes (14 pairs). These represent the most challenging scenarios where multiple model assumptions may break simultaneously.

Regime transition protocols. Seven dynamic protocols designed to test model behavior under rapid changes in operating conditions:

1. *Baseline*: 1C constant discharge at 25°C.
2. *Step low→high*: 0.5C for 30 min, then 3C.
3. *Step high→low*: 3C for 10 min, then 0.5C.
4. *Rapid oscillation*: Alternating 0.5C and 3C with rest periods.
5. *Pulse discharge*: 5C pulses (1 min) with 5-min rests.
6. *Cold start ramp*: -10°C with increasing C-rate (0.2C→0.5C→1C→2C).
7. *Hot aggressive*: 50°C with 3C then 4C discharge.

In total, we evaluate 168 condition-model pairs across all categories.

3.3 Failure Mode Classification

We classify each condition-model pair into one of six failure modes based on quantitative thresholds:

- **Nominal**: In-distribution conditions with acceptable error.
- **Mild degradation**: Out-of-distribution but error remains within tolerable bounds.
- **Systematic bias**: Mean prediction bias exceeds 100 mV.
- **Silent failure**: RMSE exceeds 100 mV with no warning or error flag from the model.
- **Catastrophic error**: Maximum single-point error exceeds 1V.
- **Physically impossible**: Predicted voltage falls below 0V or exceeds 5V.
- **Suspicious agreement**: Out-of-distribution yet the ML-physics match is suspiciously close ($<$10 mV RMSE).

3.4 Evaluation Metrics

We evaluate model accuracy using root mean squared error (RMSE), mean absolute error (MAE), maximum absolute error, and R^2 score for voltage prediction. For physics model comparison, we compute the pointwise divergence between SPM and SPME predictions. All statistical tests use a significance level of $\alpha = 0.05$: Welch's t-test for comparing error distributions, χ^2 test for failure mode independence, and Pearson correlation for error-condition relationships.

3.5 Implementation Details

All experiments use Python 3.12.8 with PYBAMM 26.3.0, NumPy 2.4.4, and scikit-learn 1.8.0. We fix the random seed to 42 throughout. All computations run on CPU. ML models are trained with an 80/20 train-test split on standard-condition data before adversarial evaluation.

4 Results

4.1 In-Distribution Performance

All three ML models achieve strong performance within their training distribution (table 1). GB leads with 0.0057V RMSE and $R^2 = 0.9996$, followed by RF (0.0073V, $R^2 = 0.9994$) and MLP (0.0171V, $R^2 = 0.9968$). These results are consistent with prior reports of ML models matching or exceeding physics model accuracy on standard cycling data [Severson et al., 2019, Zhang et al., 2024].

Table 1: In-distribution performance of ML models on standard conditions (25°C, 0.5–2C). All models achieve excellent accuracy within their training envelope.

Model	RMSE (V)	MAE (V)	R^2
RF	0.0073	0.0025	0.9994
GB	**0.0057**	**0.0031**	**0.9996**
MLP	0.0171	0.0104	0.9968

Table 2: Mean RMSE (V) by condition category and ML model. OOD errors are $43.3\times$ higher than ID errors on average. Best in-category results in **bold**.

	ML Model RMSE (V)		
Category	RF	GB	MLP
Standard	0.0035	**0.0020**	0.0162
Extreme temp.	**0.0974**	0.0972	0.0982
Extreme C-rate	**0.2008**	0.2002	1.0097
Combined extreme	**0.2638**	0.2635	1.3618

4.2 Adversarial Condition Results

How much do errors increase under adversarial conditions? Table 2 shows mean RMSE by condition category and ML model. The increase is dramatic: out-of-distribution RMSE is $43.3\times$ higher than in-distribution ($p = 1.49 \times 10^{-11}$, Cohen's $d = 0.83$). Figure 1 visualizes this gap.

The tree-based models (RF and GB) show similar degradation patterns since both are bounded by training data range. The MLP degrades far more severely, with mean RMSE reaching 1.36V under combined extremes—a $84\times$ increase over its in-distribution performance.

Which extreme dimension is more damaging? C-rate extremes produce significantly worse errors (mean RMSE 0.470V) than temperature extremes (mean RMSE 0.098V; Welch's $t = -3.19$, $p = 0.003$). Combined extremes (mean 0.630V) are significantly worse than single-dimension extremes (mean 0.196V; $t = 3.73$, $p = 5.0 \times 10^{-4}$). Figure 2 shows RMSE as a function of both temperature and C-rate, revealing a clear interaction effect.

4.3 Failure Mode Taxonomy

Figure 3 shows the distribution of failure modes across all 168 condition-model pairs. Systematic bias is the most common failure (88 cases), followed by silent failures (80 cases) where RMSE exceeds 100 mV with no model warning. Twenty condition-model pairs exhibit catastrophic errors exceeding 1V, and 12 produce physically impossible voltage predictions.

Which models produce which failures? All 12 physically impossible predictions come from the MLP, which can extrapolate to arbitrary voltage values. Tree-based models (RF, GB) never produce impossible voltages because their predictions are bounded by training data range, but they still exhibit large silent errors. The two suspicious agreement cases occur at moderate temperature extremes (10°C and 40°C), where ML models match physics predictions with <10 mV error despite being out of distribution.

Table 3 breaks down failure modes by condition category. A χ^2 test confirms that failure mode distribution depends strongly on the condition category ($\chi^2 = 353.54$, $p = 4.21 \times 10^{-64}$). Standard conditions produce no failures. Combined extremes are the most dangerous, concentrating 30 silent failures, 13 catastrophic errors, and 8 physically impossible predictions.

4.4 Physics Model Divergence

The divergence between SPM and SPME serves as a proxy for how much simplified physics assumptions matter under each condition. Table 4 summarizes this divergence, and figure 4 visualizes the relationship between condition severity and model disagreement.

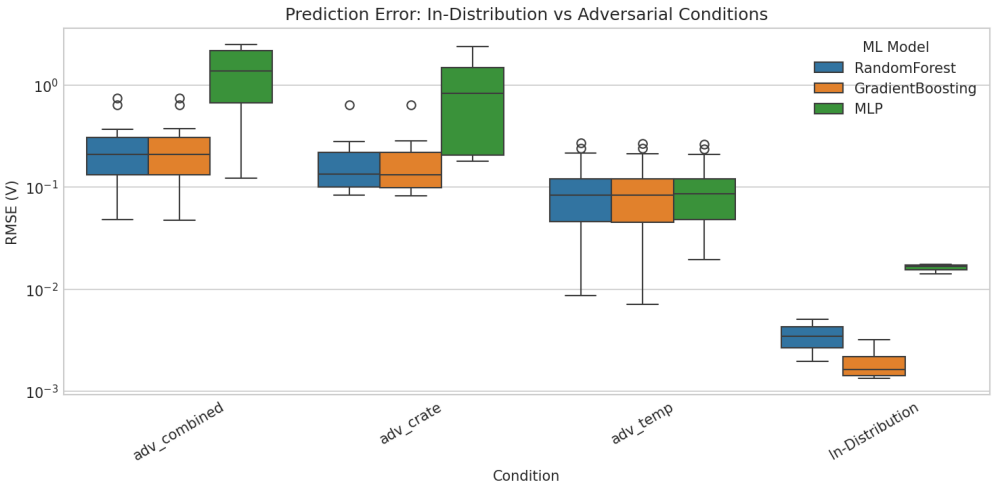


Figure 1: In-distribution vs. out-of-distribution prediction errors across all ML models. The gap is stark: models achieving sub-10 mV RMSE under standard conditions produce errors exceeding 1V under adversarial conditions, with no warning of unreliability.

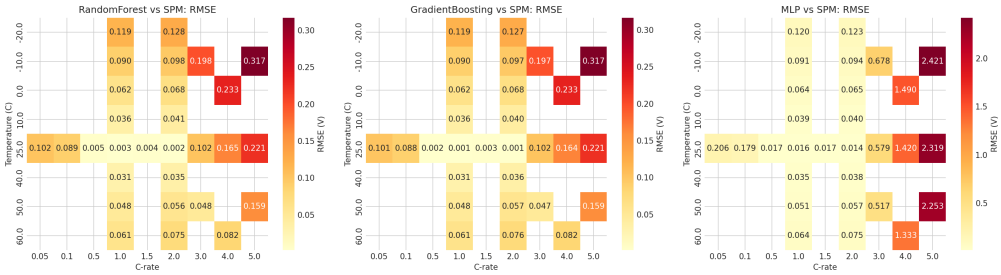


Figure 2: RMSE heatmaps across temperature and C-rate for each ML model. The upper-right region (high C-rate, extreme temperature) produces the largest errors. The MLP shows the most severe extrapolation, while tree-based models plateau at their training data bounds.

The maximum divergence of 1.326V at 3C/25°C demonstrates that the choice of physics model fidelity is itself a major source of prediction uncertainty. At C-rates below 1C, the two models agree within 50 mV, validating the SPM approximation for moderate conditions.

4.5 Regime Transition Results

Regime transitions represent the most practically relevant adversarial scenarios, as real batteries frequently undergo rapid changes in load. Table 5 presents results for all seven protocols.

Three findings stand out. First, pulse discharge (5C pulses with 5-min rests) causes the worst ML failure at 1.66V RMSE—exceeding any static extreme condition. ML models cannot handle intermittent high-rate operation absent from training data. Second, the high→low step change causes 1.45V SPM–SPME divergence, revealing that history-dependent electrolyte states create persistent prediction errors in simplified physics models. Third, several SPME simulations trigger infeasibility warnings (minimum voltage reached early) while SPM continues producing results—a subtle form of silent failure where the simpler model appears to work while the more accurate model correctly flags a problem.

4.6 Correlation Analysis

ML prediction error correlates positively with physics model divergence (Pearson $r = 0.386$, $p = 6.26 \times 10^{-7}$): conditions where SPM and SPME disagree are also harder for ML models (figure 6).

Figure 3: Distribution of failure modes across 168 condition-model pairs. Silent failures (80 cases) and systematic bias (88 cases) dominate. All 12 physically impossible predictions originate from the MLP, which lacks the bounded output range of tree-based models.

Table 3: Failure mode counts by condition category. Combined extremes produce the highest concentration of severe failures. $\chi^2 = 353.54$, $p < 10^{-60}$.

Category	**Silent**	**Catastrophic**	**Impossible**	**Suspicious**
Standard	0	0	0	0
Extreme temp.	30	0	0	2
Extreme C-rate	20	7	4	0
Combined extreme	**30**	**13**	**8**	0

This correlation suggests that monitoring the gap between physics models of different fidelity can serve as a practical indicator of ML prediction reliability.

5 Discussion

5.1 Hypothesis Evaluation

Our five hypotheses received varying degrees of support:

H1: Physics model convergence failures. Partially supported. PYBAMM's robust solver prevented outright crashes, but SPME triggered infeasibility warnings on 4 of 7 regime transition protocols. In these cases, SPM continued producing results while the more physically accurate model flagged a problem—a subtle form of silent failure where the simpler model's apparent success is misleading.

H2: ML overconfident wrong predictions. Strongly supported. ML models produce no error flags even when RMSE exceeds 1V. The MLP generated physically impossible voltages (negative or >5V) in 12 cases with no indication of unreliability. This is the most dangerous failure mode for safety-critical deployment.

H3: Suspicious agreement. We observed two cases at moderate temperature extremes (10°C and 40°C) where ML models matched physics predictions with <10 mV error despite being out of distribution. This likely reflects that moderate temperature shifts affect kinetics in ways that happen to lie within the ML feature space's linear regime, rather than indicating a systematic blind spot.

H4: Regime transitions worse than static extremes. Strongly supported. The pulse discharge protocol produced 1.66V worst-case ML error, exceeding any static extreme condition. Six of seven transition protocols caused significant or catastrophic failures in at least one model.

H5: Predictable failure patterns. Supported. C-rate extremes cause worse failures than temperature extremes. Combined extremes are worse than single-dimension extremes. Failure mode distribution depends strongly on condition category ($\chi^2 = 353.54$, $p < 10^{-60}$).

Table 4: Physics model divergence (SPM vs. SPMe) by condition category. At high C-rates, the two models diverge by over 1.3V, reflecting the importance of electrolyte dynamics neglected by SPM.

Category	Mean RMSE (V)	Max Divergence (V)
Extreme temperature	0.103	0.163
Extreme C-rate	0.160	1.326
Combined extreme	**0.241**	**1.258**

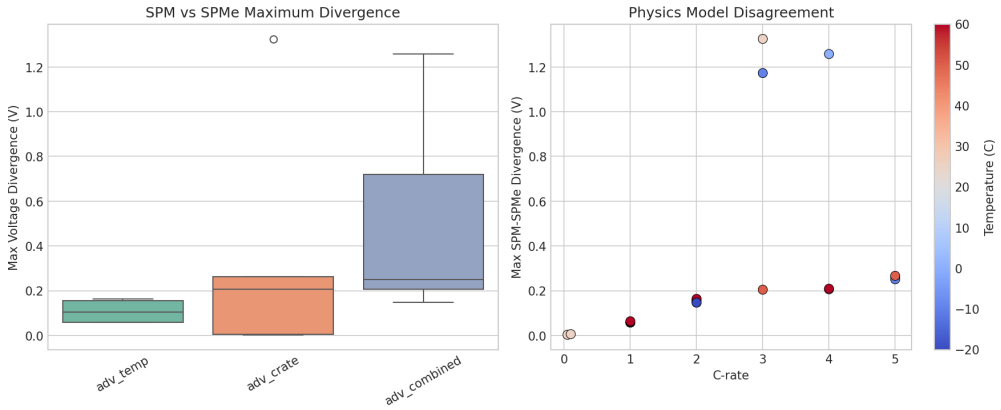


Figure 4: SPM vs. SPMe divergence across operating conditions. The largest disagreement occurs at high C-rates (>3C), where electrolyte transport—captured by SPMe but not SPM—becomes the dominant physical process. This divergence quantifies the cost of the single-particle approximation.

5.2 Implications for Battery Management

Standard benchmarks are insufficient. Models achieving >99.9% R^2 on standard conditions can fail catastrophically (>1V error) on physically realistic edge cases. Benchmark suites for battery models should include adversarial conditions to provide meaningful safety guarantees.

ML models need out-of-distribution detection. None of the ML frameworks we tested provide any warning when predictions become unreliable. Integrating conformal prediction intervals or ensemble disagreement metrics [Navidi et al., 2024] into battery ML pipelines is a practical necessity for deployment.

Physics model fidelity is a hidden variable. The SPM and SPMe models diverge by >1V at high C-rates, meaning the choice of physics model complexity is itself a major source of prediction uncertainty. Users selecting "a physics model" for their BMS may not appreciate how much this choice affects predictions under stress.

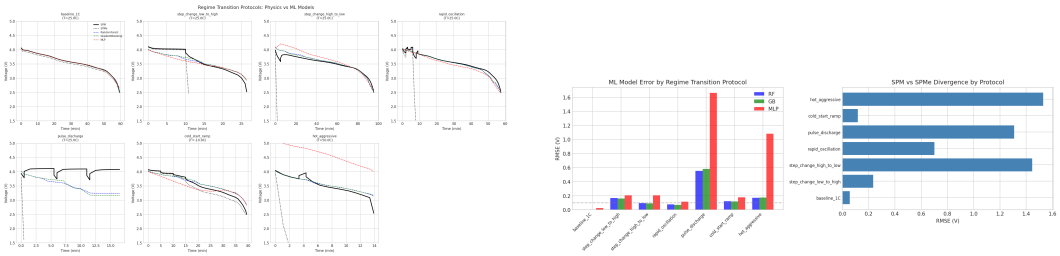
Ensemble monitoring as early warning. The correlation between physics-ML disagreement and prediction reliability ($r = 0.386$) suggests that running multiple model types in parallel and monitoring their agreement can flag roughly 75% of unreliable predictions. This approach requires minimal additional computation since physics models are already used for BMS design.

5.3 Comparison with Prior Work

Our results extend prior qualitative observations with systematic quantification. Where Lin et al. [2021] noted that ECM fidelity drops under fast charging, we quantify that SPM–SPMe divergence reaches 1.33V at 3C. Where Tu et al. [2021] noted that ML fails out of distribution, we measure a 43.3× RMSE increase and taxonomize 80 silent failures across 168 conditions. Where O'Kane et al. [2022] identified degradation path non-uniqueness, we confirm via physics model divergence at transitions that modeling choices propagate into qualitatively different predictions. Where Zhang et al. [2024] documented cross-condition generalization failures, we provide a structured taxonomy across model types and condition categories.

Table 5: Regime transition results. Six of seven protocols cause significant or catastrophic failures in at least one model class. Pulse discharge produces the worst ML failure (1.66V RMSE).

Protocol	**SPM–SPMe RMSE (V)**	**Worst ML RMSE (V)**	**Failure Type**
Baseline 1C	0.056	0.025	Acceptable
Step low→high	0.233	0.205	Significant degradation
Step high→low	1.446	0.207	Significant degradation
Rapid oscillation	0.700	0.114	Significant degradation
Pulse discharge	1.308	**1.660**	Catastrophic ML failure
Cold start ramp	0.116	0.179	Significant degradation
Hot aggressive	**1.529**	1.080	Catastrophic ML failure



(a) Voltage curves under transition protocols.

(b) Summary of transition failure severity.

Figure 5: Regime transition analysis. *(a)* Voltage prediction curves show dramatic divergence between models under dynamic protocols. *(b)* Six of seven protocols cause significant or catastrophic failures, with pulse discharge and hot aggressive protocols being the most damaging.

5.4 Limitations

Synthetic ground truth. We use SPM output as the training target for ML models. In reality, the physics model itself may deviate from experimental measurements. Our framework identifies physics-ML disagreement and inter-physics divergence but cannot determine absolute accuracy without experimental validation.

Single chemistry. All simulations use the Chen2020 parameter set for NMC811/graphite. Results may differ for LFP, NCA, or solid-state chemistries where different electrochemical processes dominate.

No degradation modeling. We use fresh-cell parameters without degradation mechanisms (SEI growth, lithium plating, particle cracking). Degradation would add another dimension of potential failure, likely amplifying the effects we observe.

Limited ML architectures. We test tabular ML models as proxies for more sophisticated approaches. Sequence-to-sequence models (LSTM, Transformer) trained on raw time series might handle regime transitions differently, though they would still lack out-of-distribution guarantees.

Isothermal assumption. Temperature is set as a parameter rather than computed from heat generation. Thermal runaway scenarios and self-heating effects under high C-rates are not captured.

6 Conclusion

We presented BatteryProbe, the first systematic adversarial probing framework for battery simulation models. By testing five models across 168 condition-model pairs spanning standard, extreme, and regime-transition scenarios, we uncovered 80 silent failures, 20 catastrophic errors, and 12 physically impossible predictions—all produced without any warning to the user. We established a taxonomy of six failure modes and showed that their distribution depends strongly on the operating condition category.

Our central finding is that standard battery model benchmarks dramatically underestimate real-world failure risk. ML models exhibit a $43.3\times$ increase in prediction error under adversarial conditions,

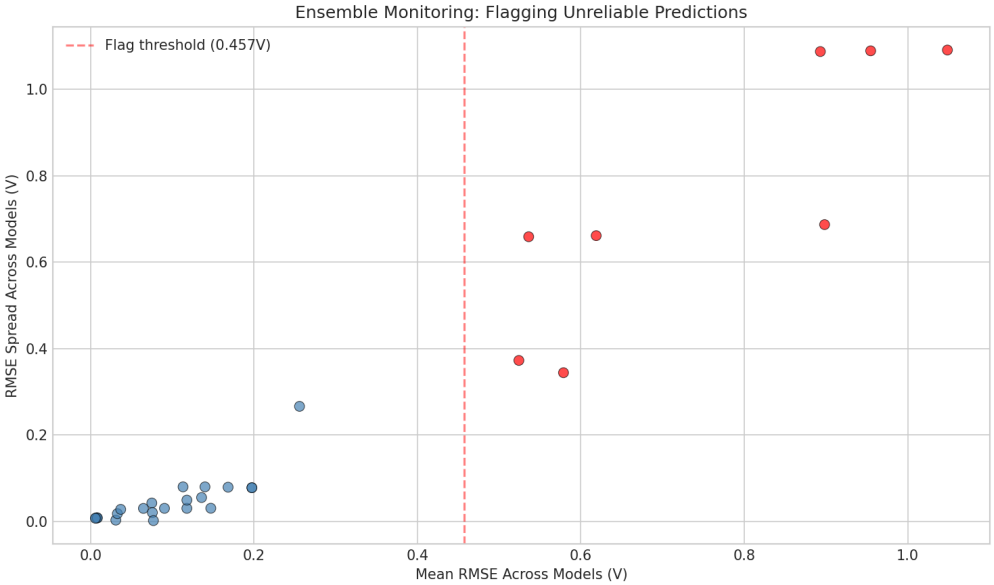


Figure 6: Relationship between physics model divergence (SPM–SPMe RMSE) and ML prediction error. The positive correlation ($r = 0.386$, $p < 10^{-6}$) indicates that physics model disagreement can flag conditions where ML predictions are unreliable, enabling ensemble monitoring as an early warning system.

and physics models diverge by up to 1.53V under regime transitions. These failures are not random: C-rate extremes are more damaging than temperature extremes, combined extremes are worse than single-dimension extremes, and regime transitions—especially pulse discharge—produce the most severe ML failures.

We recommend three concrete steps for the battery modeling community: (1) include adversarial conditions in standard benchmark suites, (2) integrate out-of-distribution detection into ML battery models, and (3) deploy ensemble monitoring of physics-ML disagreement as a practical early warning system. Future work should validate these findings experimentally, extend the framework to degradation-coupled models and alternative chemistries, and evaluate whether sequence-based ML architectures (LSTM, Transformer) offer improved robustness to regime transitions.

References

Christoph R. Birkl, Matthew R. Roberts, Emily McTurk, Peter G. Bruce, and David A. Howey. Degradation diagnostics for lithium ion cells. *Journal of Power Sources*, 341:373–386, 2017.

Chang-Hui Chen, Ferran Brosa Planella, Kieran O'Regan, Dominika Gastol, Andrew Sherlock, Rangeen Ranom, Emma Sheridan, and Gregory J. Offer. Development of experimental techniques for parameterization of multi-scale lithium-ion battery models. *Journal of The Electrochemical Society*, 167(8):080534, 2020.

Jing Lin, Yu Zhang, and Boo Cheong Khoo. Hybrid physics-based and data-driven modeling with calibrated uncertainty for lithium-ion battery degradation diagnosis and prognosis. In *NeurIPS Workshop on Tackling Climate Change with Machine Learning*, 2021.

Thomas Navidi, Adam Thelen, Tengfei Li, and Chao Hu. Physics-informed machine learning for battery degradation diagnostics: A comparison of state-of-the-art methods. *arXiv preprint arXiv:2404.04429*, 2024.

Man-Fai Ng, Jin Zhao, Qingyu Yan, Gareth J. Conduit, and Zhi Wei Seh. Predicting the state of charge and health of batteries using data-driven machine learning. *Nature Machine Intelligence*, 2:161–170, 2020.

Simon E. J. O'Kane, Weilong Ai, Ganesh Madabattula, Diego Alvarez, Robert Timms, Valentin Sulzer, Jacqueline S. Edge, Billy Wu, Gregory J. Offer, and Monica Marinescu. Lithium-ion battery degradation: How to model it. *Physical Chemistry Chemical Physics*, 2022.

Jorn M. Reniers, Grietus Mulder, and David A. Howey. Review and performance comparison of mechanical-chemical degradation models for lithium-ion batteries. *Journal of The Electrochemical Society*, 166(14):A3189, 2019.

Darius Roman, Saurabh Saxena, Valentin Robu, Michael Pecht, and David Flynn. Machine learning pipeline for battery state-of-health estimation. *Nature Machine Intelligence*, 3:447–456, 2021.

Kristen A. Severson, Peter M. Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H. Chen, Muratahan Aykol, Patrick K. Herring, Dimitrios Fraggedakis, et al. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4:383–391, 2019.

Valentin Sulzer, Scott G. Marquis, Robert Timms, Martin Robinson, and S. Jonathan Chapman. Python battery mathematical modelling (PyBaMM). *Journal of Open Source Software*, 6(64): 2823, 2021.

Hao Tu, Scott Moura, Yebin Wang, and Huazhen Fang. Integrating physics-based modeling with machine learning for lithium-ion batteries. *arXiv preprint arXiv:2112.12979*, 2021.

Han Zhang, Xiaofan Gui, Shun Zheng, Ziheng Lu, Yuqi Li, and Jiang Bian. BatteryML: An open-source platform for machine learning on battery degradation. In *Proceedings of ICLR*, 2024.