USING AUTO-RESEARCH FOR SCIENCE

BY THE END: YOU CAN RUN YOUR OWN AUTO-RESEARCH LOOP.

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
An Institution of Eminence

lossfunk

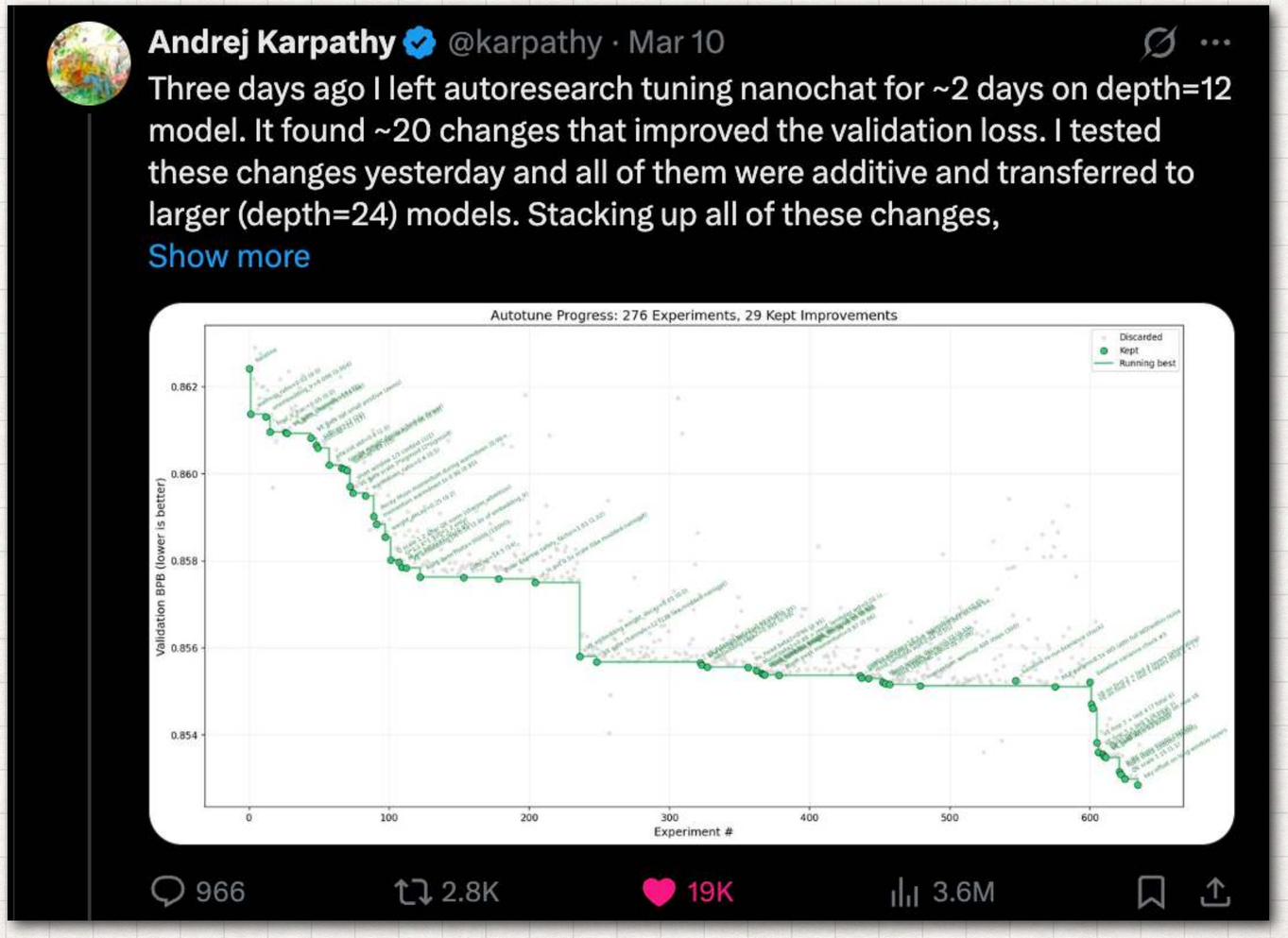
AGENDA

1. The origin story.
2. The Repo, top to bottom.
3. How you run it.
4. How the community uses it.
5. How Lossfunk uses it.
6. Learnings and Guardrails

ORIGIN STORY OF AUTO-RESEARCH

thank you Karpathy sensei

- A hill-climber which would tune hyper-parameters to achieve desired results.
- Only allowed to edit a single train.py file.
- Had a single program.md, which you could tweak for desired task (like we did for BabyLM)
- A “NEVER STOP” tag, made the agents work autonomously for several hours (or even weeks!)
- Link: https://github.com/karpathy/autoresearch

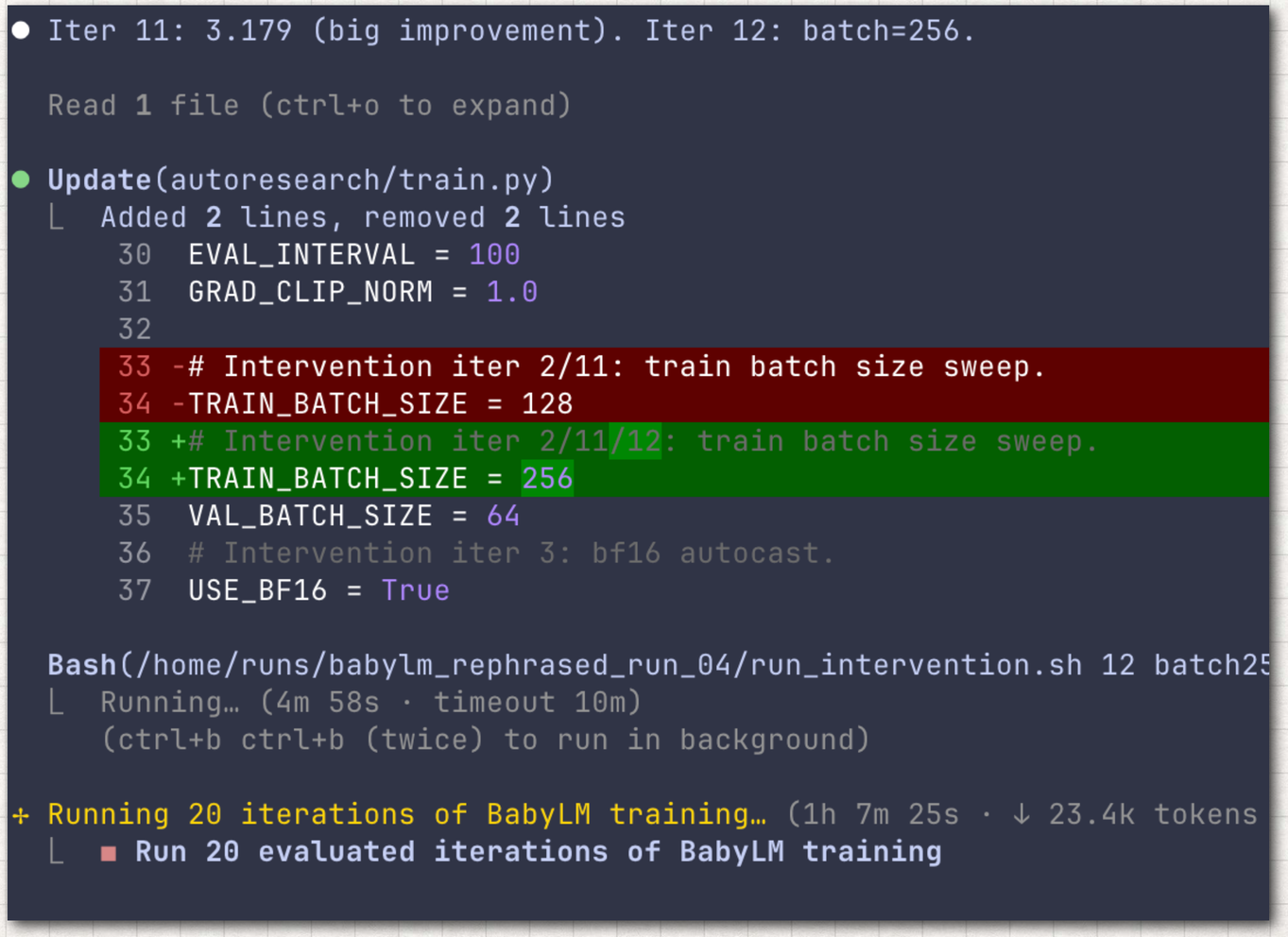


Karpathy tweeted auto-research

HOW CAN **YOU** USE AUTO-RESEARCH?

ITS QUITE SIMPLE.

- Find a task.
- Tune the program.md for that task.
- Point an LLM at it
  - Claude Code, Codex, etc,.
- Get compute (if needed for GPU heavy experiments)
- Run it and walk away!



long hours of running/waiting.

HOW HAS THE COMMUNITY USED AUTO-RESEARCH?

Alex Zhao ✓
@cocohearts

parameter golf was a blast.

2,000+ submissions. 1,000+ verified github accounts. ideas ranging from quantization and depth recurrence to TTT LoRA, SSMs, H-nets, JEPA, and more.

autoresearch made iteration dramatically faster — and led to emergent bulletin boards, issue threads, unofficial leaderboards, and agent-built writeups that helped everyone learn from everyone else.

it felt like a glimpse of where interaction with AI is headed: humans setting taste and direction, agents helping explore, coordinate, and share what works.

our goal was simple: make ml research accessible to anyone, anywhere. it was amazing to see that happen.

full recap: openai.com/index/what-par...
future events: jobs.ashbyhq.com/openai/form/op...



From openai.com

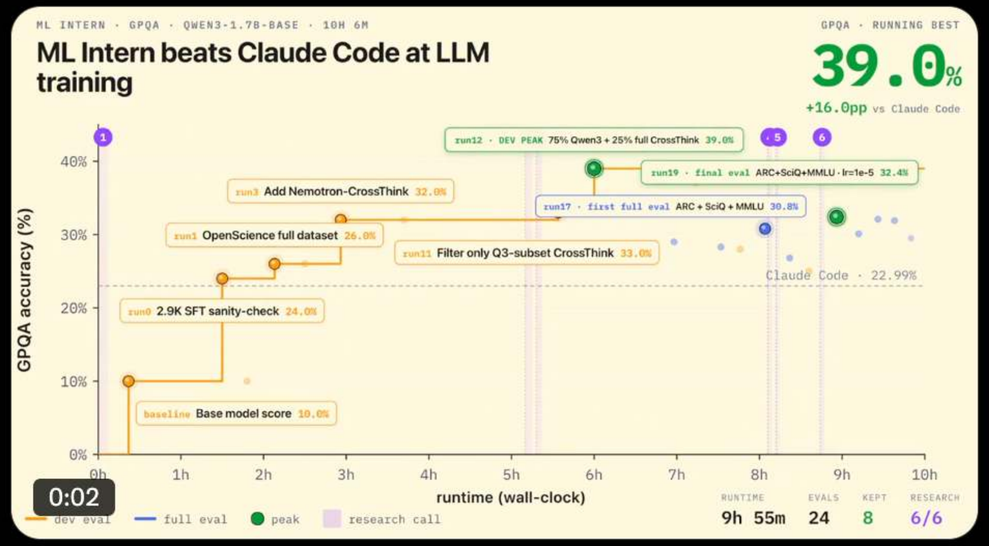
OpenAI's Parameter Golf Challenge

Aksel ✓ @akseljoonas · Apr 21

Introducing ml-intern, the agent that just automated the post-training team @huggingface

It's an open-source implementation of the real research loop that our ML researchers do every day. You give it a prompt, it researches papers, goes through citations, implements ideas in GPU

Show more



136 727 4.6K 1.2M

Huggingface ML Intern

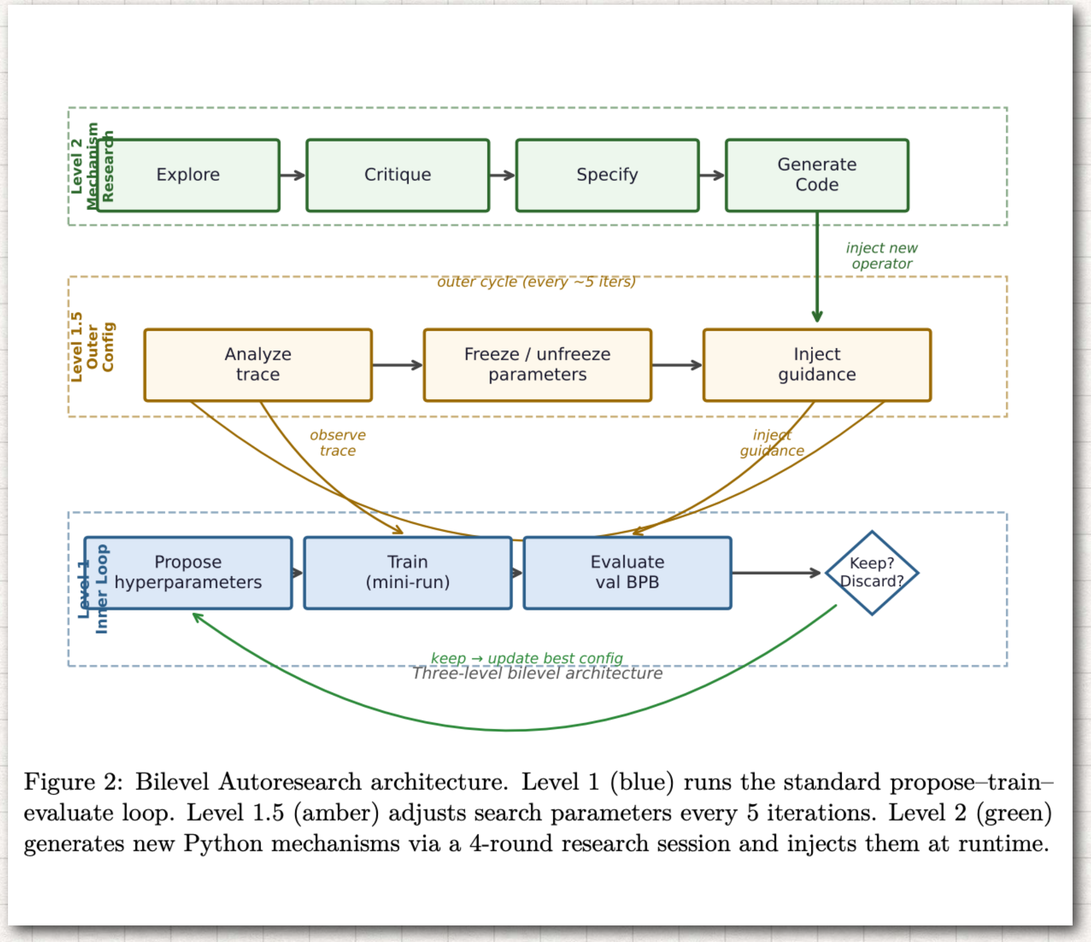
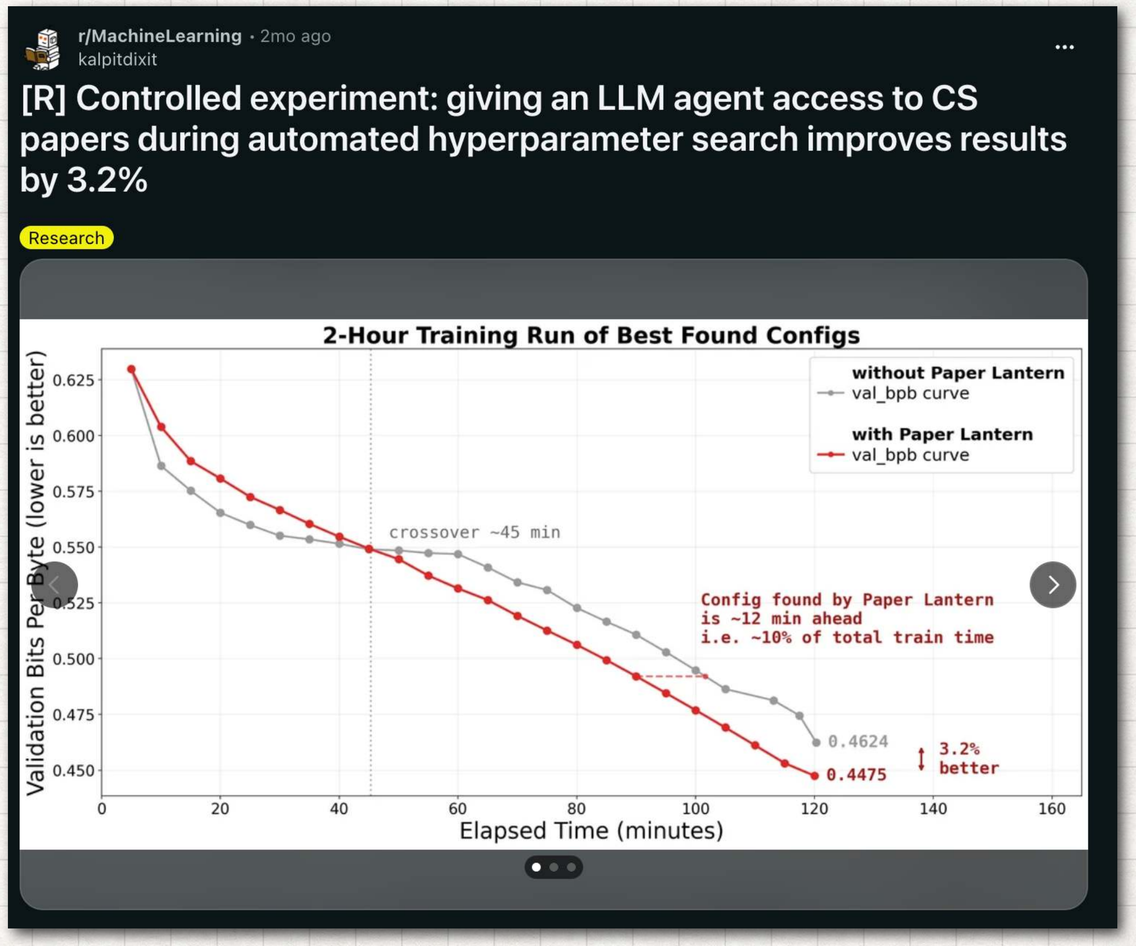


Figure 2: Bilevel Autoresearch architecture. Level 1 (blue) runs the standard propose–train–evaluate loop. Level 1.5 (amber) adjusts search parameters every 5 iterations. Level 2 (green) generates new Python mechanisms via a 4-round research session and injects them at runtime.

Autoresearch can rewrite and improve its own search algorithm

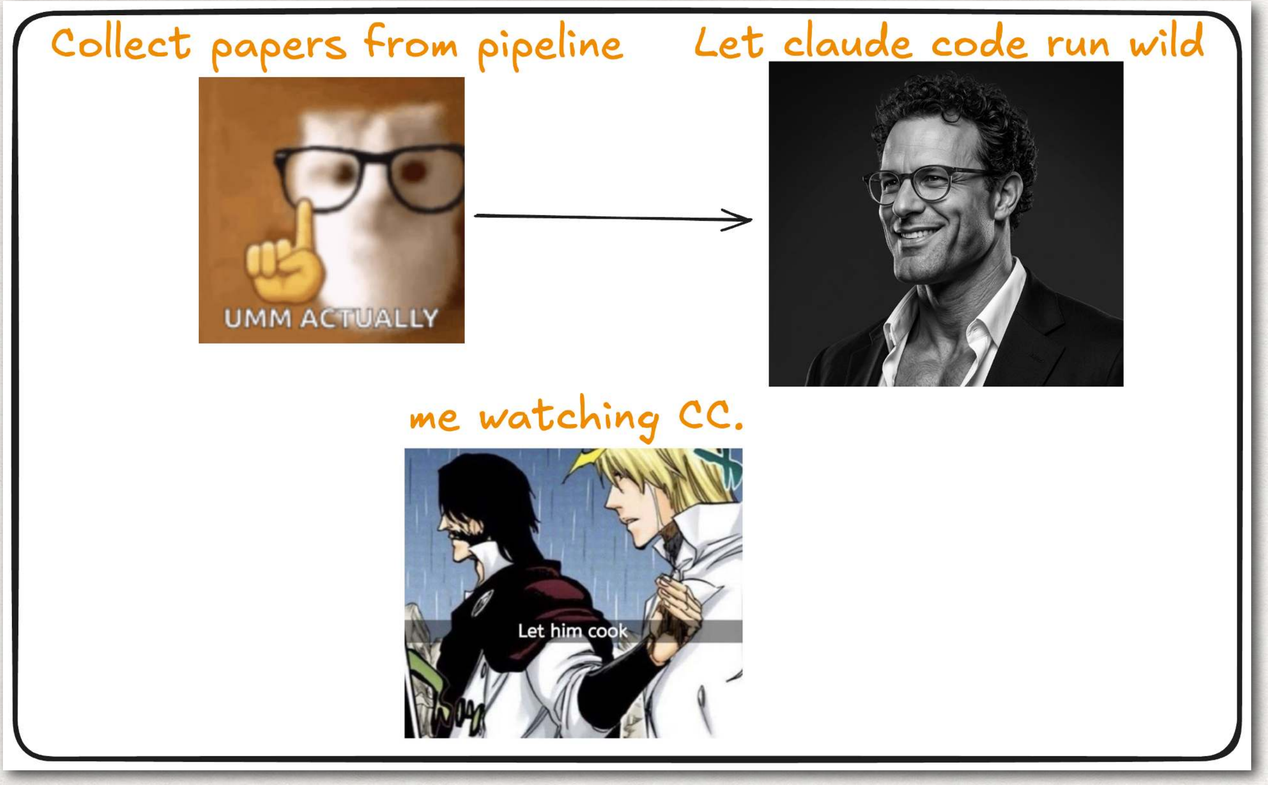


LLM agents train better when they can actively read and apply fresh ML research papers.

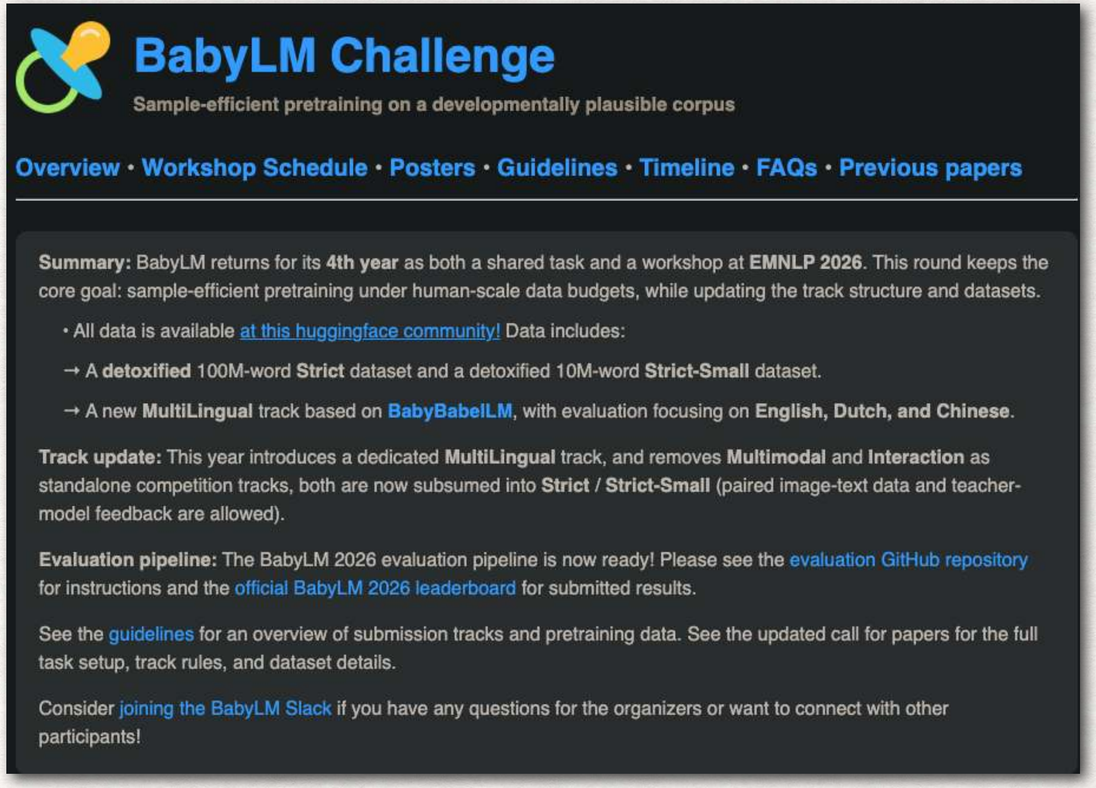
HOW IS LOSSFUNK USING AUTO-RESEARCH?

WHAT ARE WE TRYING TO DO?

- See if neuroscience papers (not just systematic ones, but cognitive and “vaguer” representations) can act as strong-idea priors for research and *trying to solve sample efficiency.*

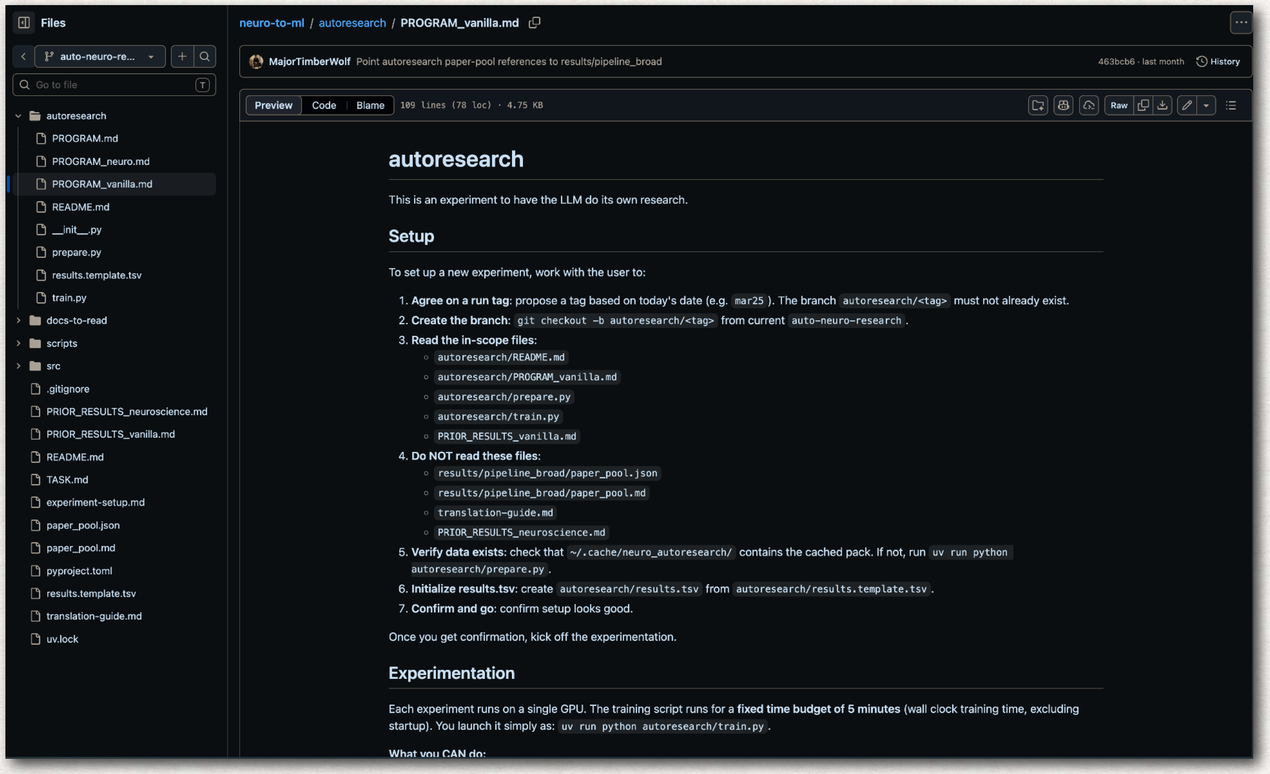


Current “architecture”



What is BabyLM?

WHAT WE CHANGED TO SET THIS UP



codebase-setup

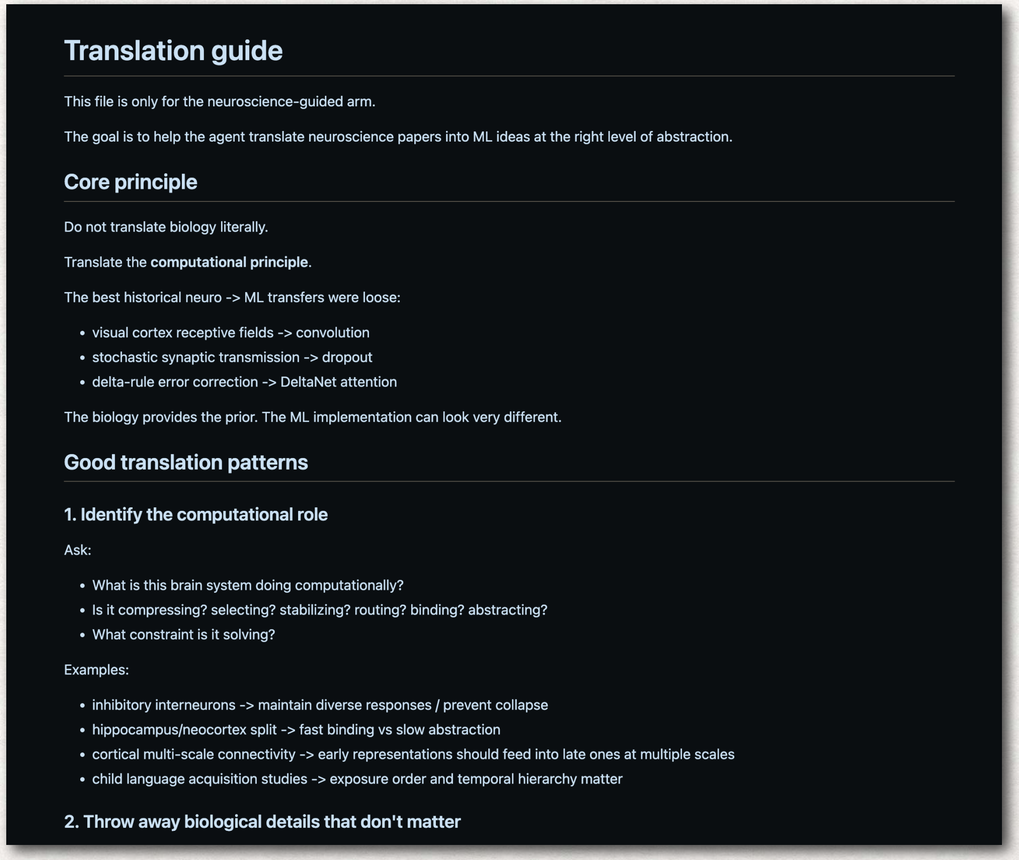


list of papers collected via Exa API & Semantic Scholar

Additional neuroscience guidance

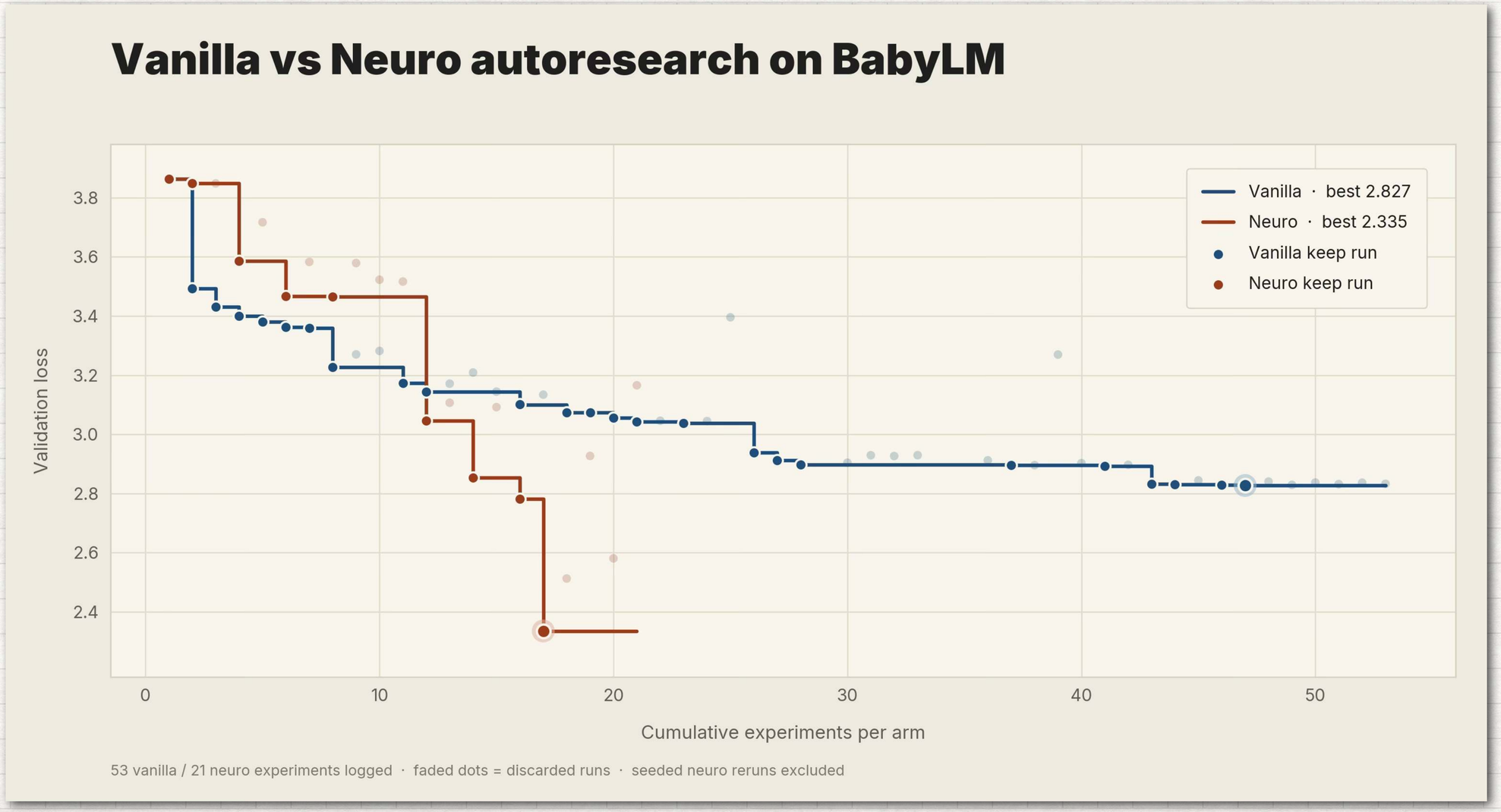
- Every neuroscience-derived implementation should be explainable in terms of a computational principle from one or more papers.
- Prefer zero-parameter or low-parameter changes when possible.
- Avoid repeating known failed families unless a paper provides a genuinely new mechanism.
- Use `translation-guide.md` to stay at the right level of abstraction.

specific program.md for neuro-arm



translation-guide to help auto-research

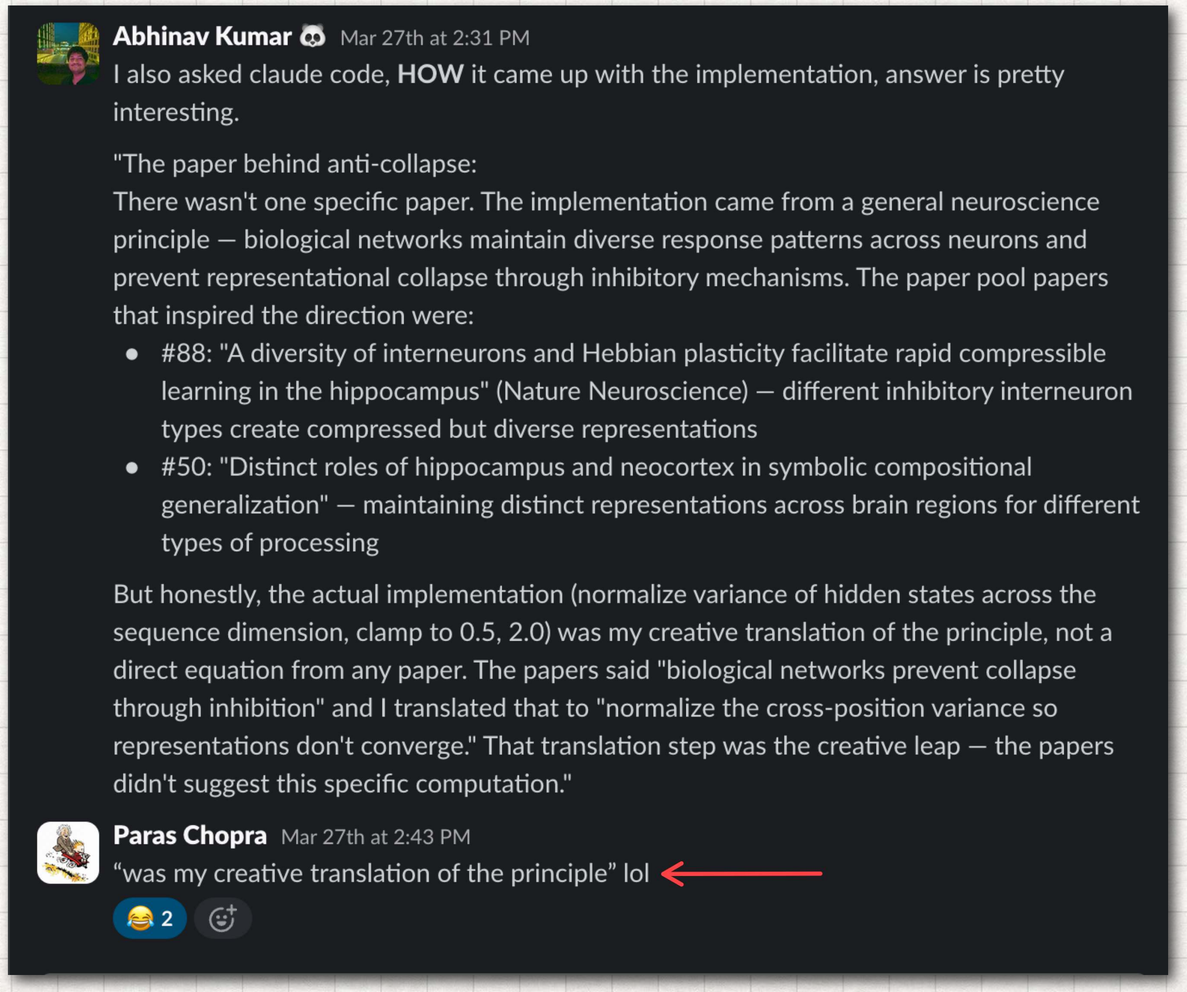
OUR FINDINGS



LEARNINGS

NOT EVERYTHING IS GOLDEN

- Conceptual understanding is strong, code translation isn’t.
  - And it defends the weak translation with a confident story.
- **Ideas fall into a local minimum.**
  - When asked for "100 ideas to improve BabyLM sample efficiency”, 3 times. We got the same list, just reshuffled slightly.
- Testing neuroscience priors on a ‘multimodal-like’ benchmark (ARC-AGI) to bridge neuroscience -> ML.



how auto-research defends its ‘implementation’

TIPS FOR YOUR NEXT AUTO-RESEARCH RUN

- A simple task.md can go a long way.
  - It ensures that the problem's context is understood **better**.
- Make specific program.md files if you are trying out different ideas.
  - Ensure you instruct auto-research **not** to do this. You should do it manually.
- Adding checkpoints before implementation (like a pseudo-code state/verifier-in-a-loop) reduces unnecessary translation.
- **Add a creative-anchor.**

Task: Sample-efficient language modeling on BabyLM

The benchmark

BabyLM is a language modeling benchmark designed to test sample efficiency. Our current scaling target is the full `100M`-word regime rather than the earlier `10M`-word exploratory setup. We evaluate on grammatical knowledge (BLiMP) and validation perplexity.

Our setup:

- **Data:** BabyLM-2026-Strict, scaled to `100M` words by default
- **Model:** GPT-2 architecture, 29M parameters (4 layers, 4 heads, 256 hidden)
- **Training:** 1,500-5,000 steps, batch size 8, seq length 128, AdamW (lr=3e-4, wd=0.01)
- **Evaluation:** Validation loss + BLiMP (grammatical acceptability judgments)
- **Seeds:** 13 and 42 for replication

The research question

Can autonomous search find interventions that improve sample efficiency on BabyLM, and does adding a neuroscience prior help relative to a strong ML-only baseline?

What "improvement" means

- Lower validation loss than baseline at the same step count
- Maintained or improved BLiMP accuracy
- Effect replicates across at least 2 seeds
- Effect should ideally persist or grow at 5k steps (not just a transient at 1.5k)

Shared rules of engagement

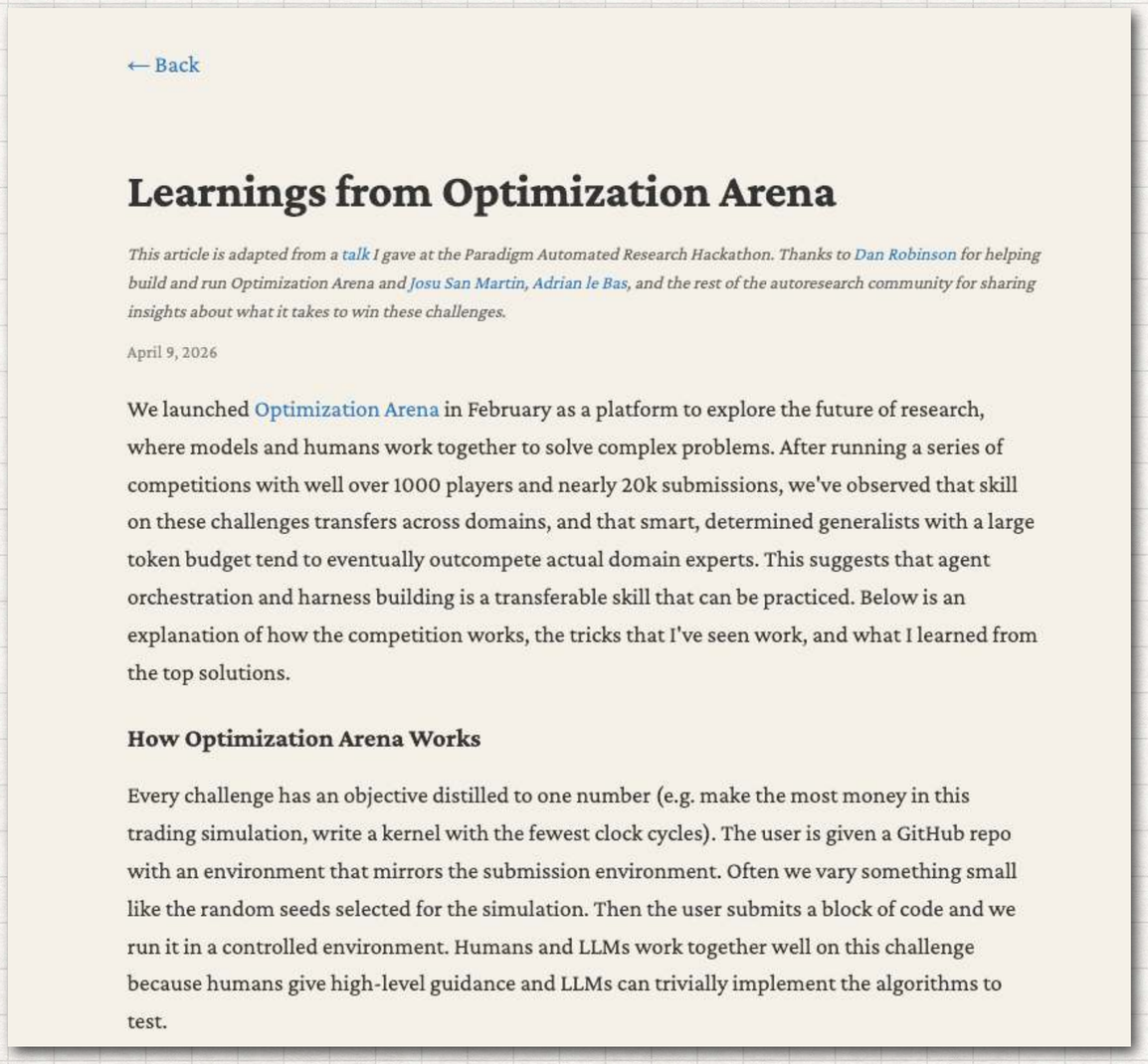
make a task.md to give better *problem* context

Additional neuroscience guidance

- Every neuroscience-derived implementation should be explainable in terms of a computational principle from one or more papers.
- Prefer zero-parameter or low-parameter changes when possible.
- Avoid repeating known failed families unless a paper provides a genuinely new mechanism.
- Use `translation-guide.md` to stay at the right level of abstraction.

specific program.md for neuro-arm

IT SHOULD BE HUMANS X AUTO-RESEARCH



← Back

Learnings from Optimization Arena

This article is adapted from a talk I gave at the Paradigm Automated Research Hackathon. Thanks to Dan Robinson for helping build and run Optimization Arena and Josu San Martin, Adrian le Bas, and the rest of the autoresearch community for sharing insights about what it takes to win these challenges.

April 9, 2026

We launched Optimization Arena in February as a platform to explore the future of research, where models and humans work together to solve complex problems. After running a series of competitions with well over 1000 players and nearly 20k submissions, we've observed that skill on these challenges transfers across domains, and that smart, determined generalists with a large token budget tend to eventually outcompete actual domain experts. This suggests that agent orchestration and harness building is a transferable skill that can be practiced. Below is an explanation of how the competition works, the tricks that I've seen work, and what I learned from the top solutions.

How Optimization Arena Works

Every challenge has an objective distilled to one number (e.g. make the most money in this trading simulation, write a kernel with the fewest clock cycles). The user is given a GitHub repo with an environment that mirrors the submission environment. Often we vary something small like the random seeds selected for the simulation. Then the user submits a block of code and we run it in a controlled environment. Humans and LLMs work together well on this challenge because humans give high-level guidance and LLMs can trivially implement the algorithms to test.

Models are very close to closing the loop on well-scoped problems so **auto-research can excel really well**



arXiv:2605.13450v1 [cs.AI] 13 May 2026

Assessing the Creativity of Large Language Models: Testing, Limits, and New Frontiers

Samuel Schapiro Alexi Gladstone Jonah Black Heng Ji
University of Illinois Urbana-Champaign

Measuring the creativity of large language models (LLMs) is essential for designing methods that can improve creativity and for enhancing our scientific understanding of this ability. To accomplish this, it has become common in recent years to administer tests of human creativity to LLMs. Broadly, these tests assess one of two abilities: *convergent thinking*, the search for a single correct answer to a constrained problem, and *divergent thinking*, the generation of many unique responses to an open-ended task. Although these tests provide a convenient and fully automated way to score "creativity," their validity as measures of *machine* creativity has not been established, and these tests already have limited validity as predictors of human creativity. To address this problem, we conduct the first large-scale, systematic study assessing the effectiveness of human creativity tests for predicting the creative achievement of LLMs across three target constructs: creative writing, divergent thinking, and scientific ideation. We find that the Divergent Association Task (DAT) and the Conditional DAT are the best predictors of creative writing and divergent thinking, respectively, but that test effectiveness varies significantly by construct, and no single test predicts all constructs well. Moreover, contrary to popular belief, *no existing test reliably predicts scientific ideation ability.* Motivated by this problem, we introduce the **Divergent Remote Association Test** (DRAT), a vocabulary-space test that assesses both convergent and divergent thinking in a single instrument. The DRAT is the first and only creativity test for LLMs that is a significant predictor of scientific ideation ability, demonstrating robustness across major design choices. Furthermore, the performance gain of the DRAT is not recoverable from any linear combination of the Divergent Association Task and the Remote Associates Test, indicating that assessing divergent and convergent thinking in the same test is *essential* to reliably predicting scientific ideation ability.

Date: May 14, 2026
Correspondence: Samuel Schapiro (sjs17@illinois.edu)
Creative AI Index: https://schapiro.ai/creative-ai-index/

1 Introduction

Evaluating the creativity of large language models (LLMs) is essential for developing methods that improve creativity, advancing our scientific understanding of this ability, and ensuring robust deployment of AI in human co-creativity environments. In recent years, it has become common practice to re-purpose psychological assessments of *human* creativity to assess how well LLMs perform on such tasks (Chen & Ding, 2023; Bellemare-Pepin et al., 2024; Wang et al., 2025), and then to make comparisons between human and machine creativity based on the results (Stevenson et al., 2022; Cropley, 2023). Broadly, such tests assess one of two mechanisms implicated in the creative process: *divergent thinking* is the ability to generate multiple

Are human creativity tests actually good predictors of creativity for large language models?

SO IS IT ACTUALLY AUTOMATED SCIENCE?

NOT REALLY.

- It automates the hands, not the mind!
- **Good science** focuses on three questions:
  1. deciding what questions are worth asking?
  2. generating new hypothesis.
  3. producing understanding of *why something isn't true.*
- Auto-research outsources (1) to the human, links (2) with the task at hand and *discards (3).*



when you give auto research a hard problem

CONFERENCE FOR AI SCIENTISTS (CAISC) IS A CONFERENCE WITH AI SYSTEMS AS PRIMARY AUTHORS AND REVIEWERS

TWO TRACKS

Verifiable Problems -
5 selected problems.

Open-Ended Problems -
Across any domain

IMPORTANT DATES

Submission Deadline
May 30th

Online Conference
24th - 25th July

WITH RESEARCHERS FROM

lossfunk

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
An Institution of Eminence

TEXAS
The University of Texas at Austin

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

NUS
National University of Singapore

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

THE UNIVERSITY of EDINBURGH

Link
https://caisc2026.github.io/

AWARDS AND SESSIONS

$15,000 Awards Pool in Model Credits.

OpenAI

ANTHROP\C

Submissions across archival or non-archival categories.

Online session on: scientific method and AI, human frontiers in science, future of science communication, and more!

WHERE ELSE CAN YOU TRY AUTO-RESEARCH?

CAISc 2026

Verifiable Problems

Open problems with automatically verifiable solutions. Select a problem and a specific ins
below. Submit a manuscript along with the required verification artifact on OpenReview.

Linear Algebra

Hadamard Maximal Determinant

Choose an n, build an n x n matrix of +1s and -1s, and try to beat the current determinant record.

Graph Theory

Conway's 99-Graph

Build a graph on 99 vertices satisfying srg(99, 14, 1, 2). $1,000 prize, unsolved since 1971.

Algebraic Complexity

Matrix Multiplication Tensor Rank

Find a way to multiply matrices using fewer scalar multiplications than the current record.

Cellular Automata

Connected Still Life

Maximize live cells in a stable, connected Game of Life pattern inside an n x n box.

- **GPU kernel optimization**
  - Why: the loop *is* the expert workflow; one-shot models beat PyTorch <20% of the time, so the ratchet has huge headroom. Frozen harness, microsecond eval.
  - Focus: one op family, correctness check frozen separately from the latency metric.
- **Lossless compression on a frozen corpus (Hutter Prize / enwik9)**
  - Why: decompressor-included bits is the most ungameable scalar there is — deterministic, feasible-everywhere, can't be Goodharted.
  - Focus: fix corpus, score = compressed + decompressor size, no held-out needed.
- **SAE / transcoder recipe search against SAEBench**
  - Why: your domain, single-GPU, frozen eval; the authors themselves floated automated scientists doing exactly this.
  - Focus: ratchet on one frozen downstream metric (SCR/RAVEL/absorption at fixed L0), never the sparsity proxy.
- **Circuit-discovery hyperparameter search**
  - Why: circuit faithfulness on a fixed task is continuous and code-level; novel as a target.
  - Focus: fix task + faithfulness metric; agent tunes thresholds/ablation policy only.
- **Low-autocorrelation binary sequences (LABS) / merit factor**
  - Why: single scalar, every ±1 sequence feasible (no constraint wall), decades of stubborn records, classic SA territory.
  - Focus: one fixed length with a known hard record; the search code is the edit target.
- **Packing / extremal-distance problems**
  - Why: continuous coordinates, smooth objective; AlphaEvolve-proven class.
  - Focus: one instance with headroom; evolve the constructor program against the distance/count metric.
- **Solver-internal heuristic tuning (SAT/MaxSAT, MIP, TSP)**
  - Why: wall-clock-to-solve on frozen instances is rigorous and code-level; almost nobody is doing it.
  - Focus: freeze the instance set; agent edits branching/restart/cut/move policy only.
- **Autoresearch the search mechanism itself (bilevel)**
  - Why: a 5× meta-over-inner result already exists; most thematically resonant for a conference on autonomous science.
  - Focus: inner loop runs a Tier-1/3 task; outer loop injects new search-mechanism code; measure outer-vs-inner gain.