# Continuous Knowledge Metabolism: Generating Scientific Hypotheses from Evolving Literature

**Jinkai Tao** [1 2 3] **Yubo Wang** [4] **Xiaoyu Liu** [1] **Menglin Yang** [3]

## Abstract

Identifying promising research directions in fast-moving subareas is one of the most cognitively expensive tasks in modern AI research. Existing LLM-driven scientific-discovery systems are typically limited to one-shot prompting on static literature snapshots and validated only against contemporary judges (human reviewers, agent peer review, wet-lab assays, or self-evaluation), leaving open whether they anticipate future trends.

We present *Continuous Knowledge Metabolism* (CKM), an AI workflow for hypothesis generation with three key capabilities: (i) *continuous literature metabolism* via sliding windows that maintain an evolving knowledge state; (ii) *predictive evaluation*, grading hypotheses against papers published *after* the generation window; and (iii) *practitioner-grade failure detection* that diagnoses workflow failure modes from its outputs.

On a 50-topic ML benchmark, CKM-LITE produces at least one validated hypothesis on $72\%$ of topics ($36/50$), more than doubling a one-shot baseline ($30\%$) at $\sim$\$3 per topic and $91\%$ lower token cost. Validated hypotheses precede their matched papers by a mean of $404$ days ($n$=55 hits across 36 topics; median 399, range 66–757).

Broadly, predictive validation against future literature offers a falsifiable, low-cost alternative to contemporary-judge protocols and can be applied wherever a corpus has dated publication records.

[1]TsingyuAI, Beijing, China [2]School of Information, Central University of Finance and Economics, Beijing, China [3]Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China [4]School of Automation, Beijing Institute of Technology, Beijing, China. Correspondence to: Jinkai Tao <jinkaitao.comm@gmail.com>, Menglin Yang <menglin.yang@outlook.com>.

*Proceedings of the $43^{rd}$ International Conference on Machine Learning Workshop* on AI as a Tool for Mathematics, Computer Science, and Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

## 1. Introduction

Picking the next research project is one of the most expensive cognitive acts in modern AI research. Before committing a quarter to a new direction, a researcher needs to know what is under-explored right now: not what last year's top venues said, but what the last six months' papers point at. Doing this by hand means reading 30–60 papers, maintaining cross-session notes, tracking what changed since last quarter, and drafting actionable research questions. A careful pass takes 12–20 hours per topic; a researcher who wants to scope many directions in parallel simply runs out of time.

LLMs have started to compress this loop. A small but rapidly growing literature on LLM-driven scientific discovery now covers idea generation (Wang et al., 2024; Baek et al., 2025), hypothesis generation (Zhou et al., 2024; Gottweis et al., 2026), and end-to-end paper writing (Lu et al., 2026; Lyu et al., 2026) (Appendix A). Across this literature, validation follows a consistent template, in which a contemporary judge (a human reviewer (Si et al., 2025), an agent peer-review loop (Weng et al., 2025), a tournament (Hu et al., 2025), a wet-lab assay (Gottweis et al., 2026), or self-evaluation (Lu et al., 2026)) decides whether the output looks good. What no system in the list tests is whether subsequent literature actually pursued the proposed direction, even though that is arguably the only criterion that matters for a system claiming to anticipate where a field is heading.

We present *Continuous Knowledge Metabolism* (CKM), an AI workflow that turns moving literature into ranked, testable hypotheses and grades each one against papers published *after* the generation window. On a 50-topic ML benchmark, CKM-LITE produces at least one validated hypothesis on $72\%$ of topics ($36/50$), more than doubling a one-shot baseline ($30\%$) at $91\%$ lower token cost; the workflow runs at $\sim$ \$3 per topic on a laptop. We further isolate three regime boundaries (§4), each paired with a user-computable detection signal that turns the trade-off from a hidden failure into a knob.

## 2. Method: The CKM Workflow

We name the workflow *Continuous Knowledge Metabolism* (CKM): it absorbs literature incrementally, integrates each

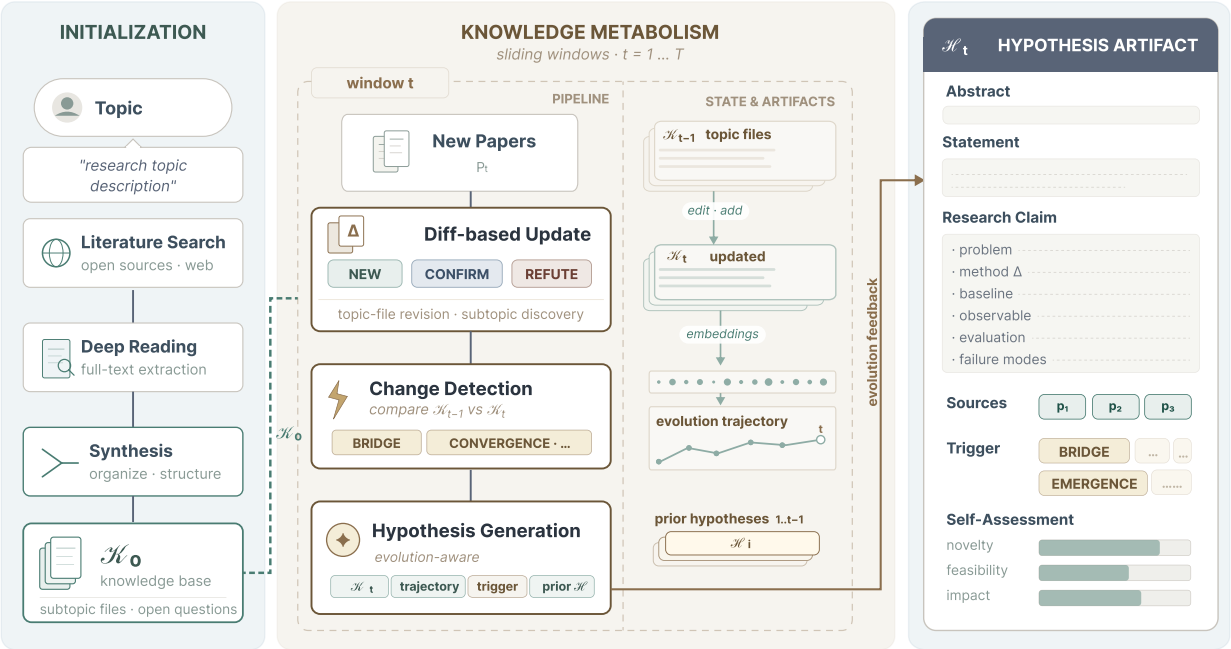


*Figure 1.* The CKM workflow. *Initialization* reads historical papers into a structured knowledge state $\mathcal{K}_0$. Each evolution window $t$ updates $\mathcal{K}_{t-1} \to \mathcal{K}_t$ with new papers and emits hypotheses $\mathcal{H}_t$. CKM-LITE implements the core cycle; CKM-FULL adds an explicit change-detection step (dashed) that we ablate in §4.

finding contrastively with the existing state, and maintains a substrate that anticipates field directions rather than summarizing past work. CKM runs in three stages (Figure 1) as an instance of *tool-augmented reasoning*: deterministic tools (ARXIV_SEARCH, OPENALEX_SEARCH) plus an embedding selector, restricted to arXiv for consistency. The variant CKM-LITE implements the cycle; CKM-FULL adds a change-detection probe (§4).

**Stage 1: Initialization.** We retrieve up to 48 pre-window papers, prompt the LLM to extract structured per-paper notes (core methods, key findings, datasets, open questions), then cluster these into Markdown *topic files* partitioned by sub-theme; the union is the initial knowledge state $\mathcal{K}_0$. The sub-theme partition is itself LLM-induced from the abstracts, not pre-specified, so $\mathcal{K}_0$ adapts to each topic's actual sub-structure rather than imposing a fixed taxonomy.

**Stage 2: Window update.** The evolution period is sliced into sliding windows of width $\Delta t$. The update $\mathcal{K}_t = \mathrm{Update}(\mathcal{K}_{t-1}, \mathcal{P}_t)$ folds in-window papers $\mathcal{P}_t$ into $\mathcal{K}_{t-1}$, appending new claims, strengthening confirming evidence, and surfacing contradictions; the contrast against $\mathcal{K}_{t-1}$ is what distinguishes this from a rolling summary. Mechanically, Update is a single LLM call that edits the topic files in place: new sub-themes are added when warranted, new evidence is attached to confirmed claims, and contradictions are tagged explicitly so they persist across windows rather than being averaged away. The LLM never re-reads the historical corpus, keeping the per-topic budget at ∼0.5M tokens (§3).

**Algorithm 1** CKM workflow over one topic.

**Require:** topic $q$, init range $[t_{\text{init}}, t_0]$, window width $\Delta t$, horizon $T$

$\mathcal{K}_0 \leftarrow \mathrm{Init}(q, \mathcal{P}^{\text{init}})$; $\mathcal{S}_0 \leftarrow \emptyset$; $\mathcal{H} \leftarrow \emptyset$
**for** $t = 1, \ldots, T$ **do**
  $\mathcal{P}_t \leftarrow$ fetch papers in window $t$
  $\mathcal{K}_t \leftarrow \mathrm{Update}(\mathcal{K}_{t-1}, \mathcal{P}_t)$
  $\mathcal{H}_t \leftarrow \mathrm{Gen}(q, \mathcal{K}_t, \mathcal{S}_{1:t-1}, \mathcal{H}_{1:t-1})$
  $\mathcal{S}_t \leftarrow$ summary of $\mathcal{K}_{t-1} \to \mathcal{K}_t$; $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}_t$
**end for**
**return** $\mathcal{H}, \mathcal{K}_T$

**Stage 3: Hypothesis generation.** The generator conditions on three contexts: $\mathcal{K}_t$, a running summary $\mathcal{S}_{1:t-1}$ of cross-window changes, and prior hypotheses $\mathcal{H}_{1:t-1}$ (so the model proposes emerging directions and avoids re-proposing); $\mathcal{S}$ lets the model distinguish a fresh trend from a long-standing direction. Each hypothesis is a structured JSON object (one-sentence claim; research plan with problem, method delta, baseline, expected observable, and evaluation; source citations; self-assessed novelty/feasibility/impact), making downstream verification (§3) tractable. Generation is throttled to ∼3 hypotheses per window under structured-output decoding, so per-topic yield is schedule-determined, not temperature-determined.

| | Predictive performance | | | | Per-hypothesis novelty (1–10) | | | |
|---|---|---|---|---|---|---|---|---|
| **System** | **Yield** | **Hit%** | **Coverage** | **Tokens** | **D1 Orig.** | **D2 Cross-f.** | **D3 Gap** | **D4 Falsif.** |
| CKM-LITE *(this work)* | **17.3** | **5.8** | **36/50** | **0.51M** | 4.20 | 5.78 | 6.21 | **7.86** |
| BATCH | 13.7 | 3.0 | 15/50 | 5.93M | 4.77 | 6.22 | **6.75** | 7.71 |
| CKM-FULL | 17.8 | 1.4 | 11/50 | 5.87M | **5.64** | **6.62** | 6.59 | 7.84 |

*Table 1.* Main results across 50 ML topics. *Left:* predictive performance (coverage = topics with $\geq$1 hit). *Right:* per-hypothesis novelty, scored by an independent LLM judge on a 1–10 scale across four dimensions (full rubric in Appendix H). Bold = best per column; all systems run 53–67 min per topic.

## 3. Experiments

**Benchmark.** We selected 50 ML research topics spanning eight categories (full list in Appendix B), ranging from slow-moving areas (machine translation, information retrieval) to fast-moving LLM topics (chain-of-thought, RLHF, tool-using agents) and applied areas (drug discovery, clinical NLP). Unlike IdeaBench (Guo et al., 2025) and LiveIdeaBench (Ruan et al., 2026), which judge ideas with contemporary scorers, we anchor evaluation in dated future literature, forcing a temporal split. Each topic is split into three temporal phases: *initialization* ($\leq$48 papers from 2019–2024 build the baseline state $\mathcal{K}_0$), *evolution* (six 2-month sliding windows over 2024–2025, $\leq$96 papers per window, generate hypotheses), and *validation* (papers from 2025–2027 serve as future ground truth). A generated hypothesis is judged a *hit* if at least one validation paper aligns with it at score $\geq 6.0$ on a 1–10 LLM-judge scale; judging uses a two-stage cross-provider protocol (GPT-4o-mini pre-filter then GPT-4o re-judgment (OpenAI, 2024), both different from the Gemini-2.5-Flash generator (Gemini Team, 2025); full safeguards in Appendix G).

**Baseline.** The natural baseline is BATCH: dump all in-window papers into a single prompt and ask for hypotheses: what most LLM-using researchers run today. We additionally include CKM-FULL, an instrumented CKM-LITE variant with a change-detection probe (§4).

**Result.** CKM-LITE produces more, better-aligned hypotheses than BATCH at $\sim$11$\times$ lower token cost; CKM-FULL trades coverage for per-hypothesis novelty (Table 1). Per topic across six evolution windows, CKM-LITE yields 17.3 hypotheses with a 5.8% hit rate, against 13.7 and 3.0% for BATCH; the gap is significant under a Wilcoxon paired test on per-topic hit rates over all 50 topics ($p$=0.006, effect size $r = 0.44$, $n_{\text{eff}} = 38$ after dropping ties), with CKM-LITE winning 29 topics, losing 9, and tying 12. Coverage is $36/50 = 72\%$ for CKM-LITE versus $15/50 = 30\%$ for BATCH; per-topic generation tokens drop from 5.93M to 0.51M (91.4% reduction); across CKM-LITE's 64 hit papers, the mean generation-to-publication lead is $\sim$404 days (median 399, range 66–757). The two halves of Table 1 cover orthogonal correctness axes: predictive alignment (left) and per-hypothesis novelty against pre-window literature (right). CKM-FULL's change-detection pipeline produces the highest novelty (D1 originality, D2 cross-field synthesis), which CKM-LITE trades for the broader-tail hit-rate distribution—a high-novelty/low-hit hypothesis may lead the field rather than fail it (Lite vs Full ablation: F1, §4).

**Validity of the result.** Three guards against optimism: post-training-cutoff validation papers (no memorization), cross-provider generation/judging (no self-preference), and re-judging hits by a stronger judge at a higher threshold (full safeguards in Appendix G). A hit at threshold $\geq 6.0$ requires the judge to identify overlapping problem framing, methodological direction, and predicted observable between hypothesis and future paper, not surface-level keyword overlap; the threshold sits at the inflection in the score distribution (Figure 3), separating substantive from incidental matches. The workflow itself is model-agnostic: substituting open-source generators or judges only requires changing two configuration values, not code (Appendix F).

**RLHF Stability** **Trigger: CONVERGENCE** · **Generated Nov 2024** · **Hit ✓ 460-day lead**

*"A novel RLHF framework will integrate adaptive entropy regularization (H-DPO/SEE-DPO style) with theoretically grounded KL-regularization and dynamic uncertainty-aware policy optimization, demonstrating enhanced stability and reduced reward hacking compared to standard DPO or PPO-based RLHF."*

**Matched (Feb 2026)** $\rightarrow$ *SAFE: Stable Alignment Fine-tuning with Entropy-Aware Predictive Control for RLHF* (arXiv:2602.04651). The matched paper's title encodes the same three components named in the hypothesis: entropy-aware control, predictive/uncertainty-aware optimization, and RLHF stability.

*Figure 2.* One CKM-LITE hit and its matched future paper. Generation saw only pre-Nov-2024 papers; the match appeared 15 months later. The hypothesis fused four pre-window papers (H-DPO, SEE-DPO, sharp KL analysis, adversarial uncertainty estimation) that subsequent work converged on. More cases in Appendix J.

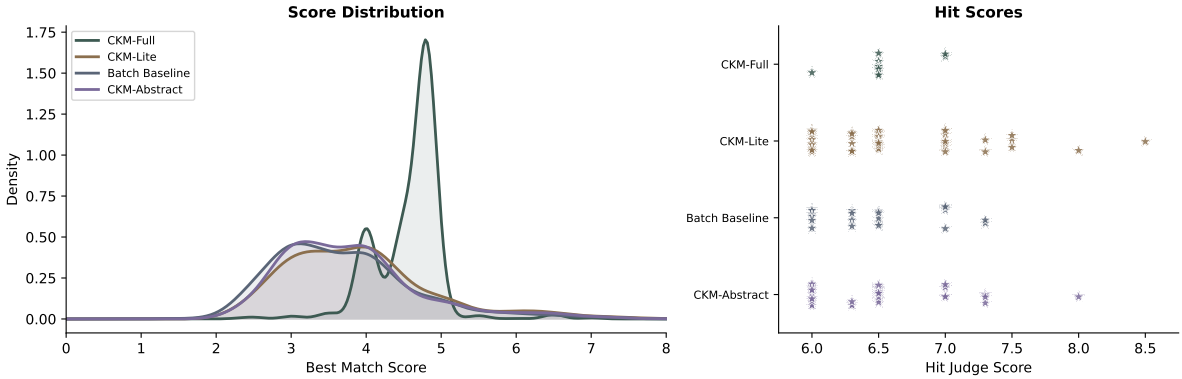


*Figure 3.* The quality-coverage trade-off behind F1, visualized. *Left:* density of best-match alignment scores across all hypotheses for CKM-FULL (instrumented), CKM-LITE (this work), and BATCH. CKM-FULL concentrates in the 4–5 band (high floor, narrow ceiling); CKM-LITE has a broader distribution with a thicker right tail crossing the hit threshold of 6.0. *Right:* individual hit scores for each system. CKM-LITE's coverage advantage comes from the broad-tail population, not from a small number of high-scoring outliers.

## 4. Failure Modes

Three boundary conditions surfaced as we used the workflow at scale; we present each as a discovered trade-off, paired with a concrete *detection signal* a user can compute on their own runs to decide when each regime applies. Figure 3 previews the mechanism behind F1 below.

**F1: A quality-coverage trade-off in change-detection instrumentation.** We tested an instrumented variant (CKM-FULL in Table 1) that adds an explicit change-detection step classifying each window-level update into five canonical types and feeds the labels into the generation prompt. Compared to CKM-LITE, this variant trades coverage for per-hit quality: hit rate moves from $5.8\%$ to $1.4\%$ at $11.5\times$ the token cost, while originality and cross-field synthesis (Table 1, right block) and per-hit alignment (best-match $3.90 \rightarrow 4.58$; lead time $404 \rightarrow 537$ days) all improve. The mechanism is visible in Figure 3: explicit change-tagging concentrates hypotheses around a narrower, higher-quality set, while CKM-LITE's broader distribution carries more right-tail mass across the hit threshold. **Detection signal**: run any new instrumentation in parallel with CKM-LITE on a held-out subset; revert if topic coverage drops by $\geq 2\times$.

**F2: Predictive accuracy varies with topic stability.** Embedding window-level hypothesis centroids and measuring cosine drift between first and last window, 8 of the 10 hit topics fall below the median drift; per-category, the workflow performs strongest on stable areas (multilingual NLP) and is most challenged on fast-shifting LLM applications, where window vocabulary churn (new benchmarks, new model families) keeps the frontier moving faster than the workflow can anticipate. **Detection signal**: compute first-to-last centroid drift per topic; topics above our median (0.27) warrant additional scrutiny of generated hypotheses.

**F3: Convergence-triggered hypotheses are the most predictive.** CONVERGENCE-triggered hypotheses (multiple recent papers reach similar conclusions) hit at $3.8\%$, while CONTRADICTION-triggered ones (new evidence conflicts with prior findings) hit at $0.8\%$ (roughly $5\times$ lower), even though novelty scores are essentially identical across trigger types. The asymmetry reflects intrinsic predictability of the underlying literature signal: convergence narrows the field's next step, whereas contradictions mark genuine open uncertainty rather than a clear trajectory. **Detection signal**: tag each hypothesis with its trigger type at generation; when prioritizing follow-up experiments, sort CONVERGENCE and TREND_CONFIRMED ahead of CONTRADICTION.

## 5. Conclusion

We presented CKM, an AI workflow that processes evolving literature through sliding windows and emits ranked, structured hypotheses scored against papers published after the generation window. On a 50-topic ML benchmark, CKM-LITE attains a $5.8\%$ future-literature hit rate at $\sim\$3$ per topic, $36/50$ topic coverage, and mean temporal lead $\sim 404$ days. Three regime boundaries (§4) qualify these results: instrumentation trades coverage for per-hit quality (F1); high-drift topics resist prediction (F2); convergence signals predict where contradiction signals do not (F3).

We read CKM as one early operationalisation of a broader class of AI-assisted research systems (Lu et al., 2026; Gottweis et al., 2026; Akari Asai & Hajishirzi, 2026) built on incremental ingestion, contrastive integration, and continuous refresh; homeostasis, cross-topic adaptation, and propagation into automated experimentation remain open (Appendix E). More broadly, predictive validation against future literature offers a falsifiable, low-cost alternative to the contemporary-judge protocols dominant in this space, and applies wherever a corpus has dated publication records.

## Impact Statement

This paper presents an AI workflow that helps machine-learning researchers scope research directions from recent literature. Two impact considerations seem worth flagging beyond the standard statement.

First, literature-based hypothesis generation can shape what researchers choose to work on. Because CKM-LITE ranks hypotheses by alignment with where the field is heading, rather than by intrinsic merit, uncritical use could amplify herding around already-popular directions and disadvantage genuinely novel work that has no near-term followers. The failure modes we report (§4) are partly meant to discourage this: in particular, F3 shows that the workflow predicts well exactly when the field is converging, which is also when piling on adds the least marginal value.

Second, the workflow consumes external API capacity and depends on closed commercial models. We have tried to keep per-topic cost low (∼$2–3) so that the workflow remains accessible to researchers without industrial-scale compute, and we report exact run costs so that practitioners can budget honestly. Beyond these considerations, this paper presents work whose goal is to advance the field of Machine Learning, and we do not see specific additional societal consequences that warrant highlighting here.

## References


Akari Asai, Jacqueline He, R. S. W. S. A. S. J. C. C. K. L. L. S. S. F. M. D. D. W. M. L. J. S. J. D. H. V. K. M. T. P. J. S. L. H. T. B. W. Y. X. L. Z. G. N. D. S. W. D. D. W.-t. Y. P. W. K. and Hajishirzi, H. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 650:857–863, February 2026. doi: 10.1038/s41586-025-10072-4. URL https://doi.org/10.1038/s41586-025-10072-4.

Baek, J., Jauhar, S. K., Cucerzan, S., and Hwang, S. J. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6709–6738, 2025.

Cai, B., Xiang, Y., Gao, L., Zhang, H., Li, Y., and Li, J. Temporal knowledge graph completion: A survey. *arXiv preprint arXiv:2201.08236*, 2022.

Cai, L., Mao, X., Zhou, Y., Long, Z., Wu, C., and Lan, M. A survey on temporal knowledge graph: Representation learning and applications. *arXiv preprint arXiv:2403.04782*, 2024.

De Cao, N., Aziz, W., and Titov, I. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, 2021.

Gemini Team, G. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2503.21218.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Sirkovic, P., Myaskovsky, A., Glowaty, G., Weissenberger, F., Orlandi, A., Popovici, D., Palepu, A., Rong, K., Tanno, R., Saab, K., Zhang, F., Blum, J., Carroll, A., Kulkarni, K., Tomašev, N., Zverinski, D., Rendulic, I., Vedadi, E., Hasler, F., Luka Rimanic, M. B., Budiselic, I., Feinstein, B., Bellaiche, M., Sheffer, T., Freyberg, J., Ratcliff, J., Bertolli, O., Chou, K., Hassidim, A., Gokturk, B., Vahdat, A., Guan, Y., Dhillon, V., Vaishnav, E. D., Lee, B., Costa, T. R. D., Penadés, J. R., Peltz, G., Matias, Y., Manyika, J., Hassabis, D., Xu, Y., Kohli, P., Pawlosky, A., Karthikesalingam, A., and Natarajan, V. Accelerating scientific discovery with co-scientist. *Nature*, pp. 1–3, 2026.

Guo, S., Shariatmadari, A. H., Xiong, G., Huang, A., Kim, M., Williams, C. M., Bekiranov, S., and Zhang, A. Ideabench: Benchmarking large language models for research idea generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5888–5899, 2025.

Hu, X., Fu, H., Wang, J., Wang, Y., Li, Z., Xu, R., Lu, Y., Jin, Y., Pan, L., and Lan, Z. Nova: An iterative planning framework for enhancing scientific innovation with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21330–21359, 2025.

Li, L., Xu, W., Guo, J., Zhao, R., Li, X., Yuan, Y., Zhang, B., Jiang, Y., Xin, Y., Dang, R., Rong, Y., Zhao, D., Feng, T., and Bing, L. Chain of ideas: Revolutionizing research via novel idea development with LLM agents. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 8971–9004, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.477. URL https://aclanthology.org/2025.findings-emnlp.477/.

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(8107):914–919, 2026.

Lyu, Y., Zhang, X., Yi, X., Zhao, Y., Shuyu Guo, W. H., Piotrowski, J., Kaliski, J., Urbani, J., Meng, Z., Zhou, L., and Yan, X. Evoscientist: Towards multi-agent evolving

ai scientists for end-to-end scientific discovery. *arXiv preprint arXiv:2603.08127*, 2026.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Fast model editing at scale. In *International Conference on Learning Representations*, 2022.

OpenAI. GPT-4o System Card, 2024. URL https://arxiv.org/abs/2410.21276.

Radensky, M., Shahid, S., Fok, R., Siangliulue, P., Hope, T., and Weld, D. S. Scideator: Human-llm scientific idea generation grounded in research-paper facet recombination. *arXiv preprint arXiv:2409.14634*, 2024.

Ruan, K., Wang, X., Hong, J., Sun, H., Wang, P., and Liu, Y. Evaluating LLMs' divergent thinking capabilities for scientific idea generation with minimal context. *Nature Communications*, mar 2026. doi: 10.1038/s41467-026-70245-1. URL https://doi.org/10.1038/s41467-026-70245-1.

Si, C., Yang, D., and Hashimoto, T. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M23dTGWCZy.

Smalheiser, N. R. and Swanson, D. R. Using arrowsmith: a computer-assisted approach to formulating and assessing scientific hypotheses. *Computer methods and programs in biomedicine*, 57(3):149–153, 1998.

Swanson, D. R. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in Biology and Medicine*, 30(1):7–18, 1986. doi: 10.1353/pbm.1986.0087.

Swanson, D. R. Migraine and magnesium: eleven neglected connections. *Perspectives in biology and medicine*, 31(4): 526–557, 1988.

Wang, Q., Downey, D., Ji, H., and Hope, T. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 279–299, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.18.

Weng, Y., Zhu, M., Bao, G., Zhang, H., Wang, J., Zhang, Y., and Yang, L. Cycleresearcher: Improving automated research via automated review. 2025.

Xiong, G., Xie, E., Shariatmadari, A. H., Guo, S., Bekiranov, S., and Zhang, A. Improving scientific hypothesis generation with knowledge grounded large language models. *arXiv preprint arXiv:2411.02382*, 2024.

Yang, Z., Du, X., Li, J., Zheng, J., Poria, S., and Cambria, E. Large language models for automated open-domain scientific hypotheses discovery. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13545–13565, 2024.

Yang, Z., Liu, W., Gao, B., Liu, Y., Li, W., Xie, T., Bing, L., Ouyang, W., Cambria, E., and Zhou, D. Moose-chem2: Exploring llm limits in fine-grained scientific hypothesis discovery via hierarchical search. *Advances in Neural Information Processing Systems*, 38:89045–89076, 2026.

Yuan, J., Yan, X., Zhang, B., Chen, T., Shi, B., Ouyang, W., Qiao, Y., Bai, L., and Zhou, B. Dolphin: moving towards closed-loop auto-research through thinking, practice, and feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21768–21789, 2025.

Zhou, Y., Liu, H., Srivastava, T., Mei, H., and Tan, C. Hypothesis generation with large language models. In Peled-Cohen, L., Calderon, N., Lissak, S., and Reichart, R. (eds.), *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pp. 117–139, Miami, FL, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nlp4science-1.10.

## A. Related Work

**LLM-driven scientific discovery.** The literature has moved through three rough waves over the last two years, all sharing one structural property: validation against a contemporary judge rather than against the future literature itself (Table 2). The first wave emits standalone research *ideas*: SciMON (Wang et al., 2024) mines neighborhood relations between concepts and validates with automatic scoring plus human review; ResearchAgent (Baek et al., 2025) adds an agent peer-review loop on top of generated ideas and plans; Nova (Hu et al., 2025) runs a Swiss-style tournament; Scideator (Radensky et al., 2024) recombines facets of existing papers in human-in-the-loop fashion; Chain-of-Ideas (Li et al., 2025) chains LLM agents through iterative refinement. The second wave commits to *hypotheses*: AI Co-Scientist (Gottweis et al., 2026) validates with wet-lab experiments (the strongest correctness signal in this set, though limited to a single domain), and MOOSE-Chem2 (Yang et al., 2026) uses hierarchical search against known chemistry targets, both building on the LLM-as-hypothesis-generator paradigm of Zhou et al. (2024) and Yang et al. (2024), grounded in scientific knowledge by Xiong et al. (2024). The third wave generates *full papers*: AI Scientist (Lu et al., 2026) self-evaluates its outputs; Dolphin (Yuan et al., 2025) closes a thinking-practice-feedback loop; CycleResearcher (Weng et al., 2025) pairs generation with simulated peer review; EvoScientist (Lyu et al., 2026) adds multi-agent evolution with human peer review. These systems are more ambitious in scope but inherit the contemporary-judge limitation.

**Foundations and adjacent work.** The argument that anticipation matters in scientific reasoning is not new. Swanson's classical work on bridging disjoint biomedical literatures (Swanson, 1986; 1988; Smalheiser & Swanson, 1998) long ago made the case that connecting evolving findings is itself a form of discovery; modern temporal knowledge-graph methods (Cai et al., 2024; 2022) carry that idea forward at the level of facts and triples; and knowledge-editing work (De Cao et al., 2021; Mitchell et al., 2022) addresses the orthogonal problem of overwriting model facts. Recent benchmarks IdeaBench (Guo et al., 2025) and LiveIdeaBench (Ruan et al., 2026) score generated ideas under minimal context, but again with contemporary judges.

**How CKM differs.** CKM fills the gap left by all of this work along two axes. *On validation*, instead of asking a contemporary judge whether an output looks good, we ask whether subsequent literature actually pursued the predicted direction, turning hypothesis generation into a falsifiable forecasting task. *On generation*, a sliding-window state $\mathcal{K}_t$ is updated incrementally, so each window's hypotheses condition on what just *changed* in the literature rather than on a static snapshot. The empirical consequences of these two choices, both where they help and where they hurt, are the body of the paper.

| System | Output | Validation | Predictive? |
|---|---|---|---|
| SciMON (Wang et al., 2024) | ideas | auto + human | × |
| ResearchAgent (Baek et al., 2025) | ideas+plans | agent peer review | × |
| Nova (Hu et al., 2025) | ideas | Swiss tournament | × |
| Scideator (Radensky et al., 2024) | ideas | human-in-the-loop | × |
| Chain-of-Ideas (Li et al., 2025) | ideas | LLM critique | × |
| AI Co-Scientist (Gottweis et al., 2026) | hypotheses | wet-lab (biomed) | × |
| MOOSE-Chem2 (Yang et al., 2026) | hypotheses | known-target match | × |
| AI Scientist (Lu et al., 2026) | papers | self-evaluation | × |
| Dolphin (Yuan et al., 2025) | papers | benchmark empirical | × |
| CycleResearcher (Weng et al., 2025) | papers | simulated review | × |
| EvoScientist (Lyu et al., 2026) | papers | peer review + experts | × |
| **CKM (ours)** | **hypotheses** | **future-paper alignment** | ✓ |

*Table 2.* Representative LLM-based scientific-discovery systems and their validation methods. All prior systems validate against contemporary judgments (human reviewers, agent peer review, simulated tournaments, lab assays, benchmark targets, or self-assessment) rather than literature that appears *after* the generation window. CKM validates predictively against future literature.

## B. Benchmark Topics

The 50 benchmark topics were chosen to span eight ML categories with deliberately heterogeneous evolution rates, so that the workflow is stress-tested under both stable (mature subfields) and fast-moving (recent LLM topics) conditions. For every topic the underlying arXiv corpus satisfies three temporal-coverage criteria: $\geq 30$ papers in $[2019, 2024]$ for initialization,

$\geq 50$ papers in $[2024, 2025]$ for evolution, and $\geq 30$ papers in $[2025, 2027]$ for validation, with $\geq 100$ papers total per topic. Topics that fail any criterion are excluded from the sampling pool.

**Multilingual & speech (4).** Low-resource machine translation; cross-lingual transfer for low-resource languages; multilingual large language models; low-resource speech recognition.

**LLM core capabilities (10).** Chain-of-thought reasoning in LLMs; long-context understanding in LLMs; factual consistency in LLMs; prompt engineering and in-context learning; instruction tuning for LLMs; complex reasoning for AI agents; tool-using agents with LLMs; accuracy of automated evaluation for LLMs; reinforcement learning from human feedback; code generation with LLMs.

**LLM applications (8).** AI for software engineering; security of code LLMs; document understanding and information extraction; aspect-based sentiment analysis; relation extraction from text; open-domain question answering with LLMs; diversity evaluation in text generation; conversational information retrieval.

**Safety & trust (4).** Explainability and interpretability of neural networks; adversarial robustness of deep learning models; fairness and bias mitigation in machine learning; out-of-distribution detection for language models.

**Multimodal (3).** Vision-language models and multimodal learning; visual question answering with multimodal models; text-to-image generation and diffusion models.

**Domain-specific AI (6).** Clinical natural language processing; machine learning for drug discovery; deep learning for medical image analysis; protein structure prediction with deep learning; surrogate modeling in physics-informed machine learning; AI for hypothesis generation in science.

**Efficiency & data (7).** Efficient fine-tuning of large language models; knowledge distillation for small language models; model compression and pruning for neural networks; mixture of experts routing for language models; data filtering for domain-specific models; synthetic data quality evaluation; data augmentation for natural language processing.

**Other foundations (8).** Continual learning and catastrophic forgetting; domain adaptation and transfer learning; federated learning for language models; knowledge graph completion and reasoning; graph neural networks for natural language processing; active learning for natural language processing; deep learning for recommendation systems; user experience evaluation for large language models.

## C. Trigger-Type Breakdown (extended F3)

F3 in §4 reports a $5\times$ gap between CONVERGENCE and CONTRADICTION triggers, but the change-detection step actually classifies each window-level update into a richer set of types. We list them here, with the per-trigger rationale and the full trigger $\times$ drift breakdown across the $892$ CKM-FULL hypotheses.

**Trigger taxonomy.**

- CONVERGENCE: multiple recent papers independently reach similar conclusions, signalling a coalescing direction.
- TREND_CONFIRMED: a previously emerging trend is corroborated by additional within-window evidence.
- BRIDGE: a finding connects two previously disjoint subareas of the knowledge state.
- NON_OBVIOUS_BRIDGE: same as BRIDGE but across distant subareas (different arXiv categories or vocabularies).
- GAP: a previously identified open question receives partial new evidence without resolution.
- GAP_EXPLOITATION: an open question is directly addressed by an in-window paper, suggesting the gap is now actively being filled.
- CONTRADICTION: a new finding directly conflicts with a claim already in $\mathcal{K}_{t-1}$.

- CROSS_PAPER: a single new paper synthesizes elements from multiple prior $\mathcal{K}_{t-1}$ subareas without producing a new direction.
- INCREMENTAL: routine refinement; included for completeness but rarely fires.

**Trigger × topic-drift hit rates.** Splitting topics at the median first-to-last centroid drift (0.267, used in F2) and crossing with trigger type gives a non-uniform structure (Table 3). The most striking cell is GAP_EXPLOITATION in low-drift topics ($n$=9, 33.3% hit rate); the small sample warrants caution, but the absence of any hits in the corresponding high-drift cell suggests that gap-filling is genuinely a regime-dependent signal rather than a noise artifact. Low-drift topics account for 10 of the 12 total CKM-FULL hits, consistent with F2.

| | Low Drift | | High Drift | |
|---|---|---|---|---|
| **Trigger** | $n$ | Hit% | $n$ | Hit% |
| GAP_EXPLOITATION | 9 | **33.3** | 18 | 0.0 |
| BRIDGE | 104 | 2.9 | 77 | 0.0 |
| GAP | 73 | 2.7 | 72 | 1.4 |
| NON_OBV._BRIDGE | 69 | 1.4 | 103 | 1.0 |
| CONTRADICTION | 81 | 1.2 | 75 | 0.0 |
| CROSS_PAPER | 36 | 0.0 | 54 | 0.0 |

*Table 3.* Hit rate by trigger × topic drift across all 892 CKM-FULL hypotheses. Drift split at the topic-level median (0.267). GAP_EXPLOITATION in low-drift topics achieves the highest hit rate, although the sample is small. CROSS_PAPER and any combination with high drift reach 0%, suggesting that the *interaction* of trigger type with topic stability is what governs predictability, not either alone.

**Practical takeaway.** A practitioner deploying CKM-LITE can use this table as a triage rule: among hypotheses generated in a topic with low semantic drift across windows, prioritize GAP_EXPLOITATION and BRIDGE triggers; skip CROSS_PAPER triggers and any high-drift conditions for time-constrained reading. This is the operational form of F3, generalized beyond the headline convergence-vs-contradiction split in the main body.

## D. Hypothesis Trajectories Across Windows

Figure 4 visualizes the literal motion of CKM's hypothesis centroids across the six 2-month windows for nine representative topics, in PCA-projected embedding space. This is the empirical ground truth behind F2 in §4: low-drift trajectories (top row) tend to produce hits, while high-drift trajectories (bottom row) tend not to. It is also the visual instantiation of the metabolism metaphor introduced in §2: the workflow does not deliver one batch of outputs but produces a moving target whose shape across time is itself diagnostic.

## E. Limitations and Future Work

**Hit rate vs. coverage: which metric matters.** A 5.8% hit rate at 17 hypotheses per topic produces, on average, roughly one validated direction per topic, which in the deployment scenario this work targets (a researcher scoping 4–5 candidate subareas in parallel before committing to one) is precisely the useful unit of output. The substantive question is therefore not what fraction of generated hypotheses hit, but whether the researcher reliably gets at least one valid prediction per topic of interest. Coverage ($36/50 = 72\%$ for CKM-LITE, against $15/50 = 30\%$ for the one-shot baseline) maps directly to that question; the per-hypothesis hit rate is its dual and is mechanically diluted by hypothesis yield, so a low headline rate at high yield does not by itself indicate weak prediction. We recommend that future evaluations of literature-based hypothesis generation report both metrics, and that comparisons of competing systems privilege coverage when deployment to a portfolio of topics is the use case.

**Temporal lead distribution.** Across the 55 validated hits over 36 topics, validated hypotheses precede their matched papers by a mean of 404 days and a median of 399 days (stdev 174, range 66–757). Aggregated to the topic level, the mean per-topic average lead is 411 days (median 424, stdev 160). The headline RLHF case (460 days) sits within this distribution rather than being a tail, supporting the framing that predictive lead is a property of the workflow rather than of a single example.

*Figure 4.* Hypothesis-centroid trajectories across six evolution windows for nine representative topics, PCA-projected from sentence embeddings. Each colored marker is a window's hypothesis-set centroid; the polyline is the temporal sequence. Topics in the top row are low-drift (and produce predictive hits); topics in the bottom row are high-drift (and tend not to). This is the visual evidence behind F2 in §4: 8 of the 10 hit topics fall below the median drift.

**The validation proxy.** We treat "a future paper pursued the same direction" as evidence that a hypothesis was good. This rewards predictions that align with where the field went and penalizes genuinely novel directions the field has not yet visited. Hits should be read as a lower-bound measure of usefulness, not an upper bound on novelty.

**Domain scope.** All 50 topics are within ML. We have not tested whether the workflow transfers to mathematics, theory, or wet-lab biology, where evidence units, citation patterns, and time scales differ; we leave cross-domain transfer to future work and do not extrapolate from these ML results.

**Absence of human evaluation.** We measure usefulness via future-paper alignment, not via researcher judgement of whether a generated hypothesis would change a project plan. A complementary user study (e.g., asking $\sim 10$ ML researchers to score generated hypotheses for novelty, feasibility, and relevance to their current work) would directly probe practical utility; it is the most natural complement to the predictive evaluation we report and the next study we plan to run.

**The workflow does not replace expert judgement.** Every claim in the main body is about ranked output quality on a benchmark. Treating any individual hypothesis as scientifically valid still requires a domain expert reading the cited papers and the proposed experiment.

**Future work.** Three directions follow naturally. *(i) Domain transfer.* Repeat the protocol on math, theory, and biomedicine to test which failure modes generalize and which are ML-specific; a public benchmark on this axis would be a community resource in itself. *(ii) Scale and statistical strength.* Move from $50$ topics to several hundred, which would let F2 (drift sensitivity) be tested at $p < 0.01$ rather than as a qualitative pattern, and would expose lower-frequency trigger types that we currently lack power to characterize. *(iii) From metabolism to homeostasis.* CKM operationalizes one living-system property. The next layer is homeostasis under contradicting evidence (revising prior hypotheses when new conflicting findings arrive) and propagation (feeding promising hypotheses into automated experiment runners). Together these would close the loop from literature monitoring to autonomously-tested predictions.

## F. Reproducibility

Every number in this paper can be regenerated end-to-end on a laptop with API access, in roughly two days of wallclock and at a cost dominated by the API bill rather than the engineering effort. CKM-LITE ships as an installable plugin (`https://github.com/TaoJinkai/ckm-hypogen`), and a complete single-topic run takes three commands: installing the plugin, launching its host runtime, and invoking the workflow's entry-point command with a topic name (e.g. "long-context modeling"). Defaults: 2-month windows, up to 96 papers per window, Gemini-2.5-Flash for generation, GPT-4o for judging. A single-topic run completes in $\sim 50$ minutes wallclock and $\sim$\$2–3 in API cost; the full 50-topic benchmark required $44.4$ hours and $\sim$\$130. Exact prompt templates, the topic list, the per-topic run logs that produced every number above, the judging code, and a one-command verification script are in the code repository (`https://github.com/TaoJinkai/ckm-hypogen`).

**Estimated time savings vs. the manual baseline.** A graduate student doing frontier scoping by hand on one ML subarea (reading 30–60 papers, maintaining cross-session notes, drafting candidate research questions concrete enough to act on) takes 12–20 hours; CKM-LITE produces a comparable ranked list of 17 structured hypotheses in $\sim$50 minutes wallclock at \$2–3 in API spend, a $15\times$–$25\times$ time reduction per topic. The cognitive task that actually motivates this work is doing this for 4–5 candidate directions *in parallel* before committing a quarter to one of them, where the manual baseline runs out of working hours; under continuous-deployment mode (above), CKM-LITE stays within \$10–15/day for a researcher tracking that many topics in the background.

**Installation and minimal API call.** The released repository ships with (i) a Python package installable via `pip install -e benchmark/` (Python $\geq$3.9; dependencies: `openai`, `requests`, `scipy`, `numpy`, `python-dotenv`), (ii) a `.env.example` listing required keys (`OPENAI_API_KEY` for the judge, plus the generator-side key), and (iii) a `python -m ckm_benchmark.recompute --summary results/lite_summary.json` command that reproduces the Table 1 numbers in seconds without any API calls. A minimal API-driven run on one topic uses the same entry-point command as the full benchmark, with the topic name supplied on the command line; the runner, prompt files, and per-stage logging are all contained in the repository under permissive license.

**Topic list.** 50 topics across NLP core (9), LLM methods (10), LLM applications (10), domain-specific AI (6), safety & ethics (5), multilingual NLP (3), multimodal (2), and other (5).

**Per-topic timing.** Median wallclock $47.5$ min; mean generation tokens $0.51$M. Per-topic timing variance is dominated by paper full-text resolution latency, not generation.

**Continuous deployment for daily research tracking.** The 2-month window width used in the benchmark is an evaluation choice that lets us complete six windows over the 2024–2025 evolution period within a single laptop-driven run; it is not a deployment requirement. In production, a user runs CKM-LITE as a daily or weekly background process on a single research topic, ingesting whatever papers appeared since the last run and emitting new hypotheses against an ever-growing knowledge state. At a typical arrival rate of one or two ML papers per day per subarea, the per-day token budget drops below \$0.10, putting literature-driven hypothesis tracking within reach of a graduate student following two or three parallel subareas (the bottleneck §1 identified). Under this deployment mode the metabolism metaphor of §2 translates literally: the workflow becomes a researcher's continuously running daily reading routine, and the knowledge state plus hypothesis log accumulates over months as a personal scientific notebook.

**Model choices and leakage prevention.** Generation uses Gemini-2.5-Flash (Gemini Team, 2025); two-stage judging uses GPT-4o-mini at threshold $5.0$ followed by GPT-4o at threshold $6.0$ (OpenAI, 2024). All three models have training cutoffs in 2024 or earlier (the GPT-4o family has an October 2023 cutoff, and Gemini-2.5-Flash predates Q2 2025), so the bulk of the 2025–2027 validation literature post-dates their training corpora; this is the temporal half of leakage prevention. The cross-provider half (Google for generation, OpenAI for judging, so the two never share weights) and the prompt-level safeguard (generation sees only pre-window literature) are documented in Appendix G. Together these rule out the standard memorization paths: the generator cannot have seen the matched future papers at training time, the judge cannot retrieve them through self-preference, and the prompt itself never contains them.

We deliberately do not use the most capable models available at the time of writing (e.g., GPT-5, Claude Opus 4.x, Gemini 3.x). Two reasons. First, the Flash + 4o pair keeps per-topic cost in the $2–3 range and the full benchmark under $130, which matches the workshop's academic-budget envelope; substituting frontier models would push the per-topic spend an order of magnitude higher without proportional gain on a workflow whose bottleneck is structure rather than raw generation quality. Second, and more importantly, the workflow is intentionally model-agnostic: its structural choices (sliding-window state, contrastive update, structured hypothesis output, embedding-similarity validation) are independent of any particular generator, and any of the three roles (extraction, generation, judging) can be swapped to a stronger LLM without changing the workflow. The numbers in this paper should therefore be read as a *lower bound* on what the same workflow achieves under future deployments using frontier-grade models.

**Reproducing every Table 1 number.** Each per-topic record in the released JSON summaries (one for CKM-LITE, one for CKM-FULL, one for BATCH) contains `duration`, `total_generation_tokens`, `yield`, `hit_rate`, and `best_match_score`. The released code repository includes a single Python script that regenerates every value in Table 1 as well as the Wilcoxon statistic from §3, by reading those JSONs and printing per-condition rows.

**Wilcoxon paired test details.** The hit-rate comparison between CKM-LITE and BATCH is paired by topic ($N = 50$). After dropping $12$ ties, the test runs on $n_{\text{eff}} = 38$ non-zero differences with $W = 182.5$, normal-approximation $|Z| = 2.73$, and $p = 0.006$ (two-sided). The matched-pair effect size $r = |Z|/\sqrt{n_{\text{eff}}} = 0.44$ is in the medium-to-large range; CKM-LITE beats BATCH on $29$ topics, loses on $9$, and ties on $12$. The full computation is reproduced by the same one-command verification script.

## G. Validity Safeguards

(1) Generation prompts see only pre-window literature. (2) Validation papers are drawn from 2025–2027, post-training for the generation model. (3) Generation and judging use different model families. (4) Two-stage judging with the stronger judge at the higher threshold. (5) Embedding pre-filter is model-agnostic and only widens the candidate pool.

## H. Per-hypothesis Novelty Judge

The novelty scores in the right block of Table 1 are produced by a separate, single-stage LLM judge (GPT-4o, distinct from the hit-verification judge of Appendix G) that scores each generated hypothesis on four standalone 1–10 dimensions. The judge sees only the hypothesis text (statement, research claim, source-paper citations) and never the validation pool, so novelty scoring is independent of predictive hit assessment. The four dimensions:

- **D1 Originality.** How original is the proposed method delta relative to the cited source papers? Low scores (1–3) mark hypotheses that essentially restate one cited paper; high scores (8–10) mark hypotheses that propose a non-obvious recombination or extension that no cited source authored alone.
- **D2 Cross-field synthesis.** Does the hypothesis bridge two or more subareas of the knowledge state, or distinct arXiv categories? Low scores mark single-subarea hypotheses; high scores mark genuinely cross-subarea proposals (e.g., applying a technique from one ML community to a problem in another).
- **D3 Gap precision.** How precisely does the hypothesis identify the literature gap it claims to fill? Low scores mark vague gap statements ("X is an open problem"); high scores mark gaps localised to a specific failure mode of a specific prior method, with that method named.
- **D4 Falsifiability.** Does the hypothesis specify an experiment that could refute it? Low scores mark unfalsifiable claims;

high scores mark hypotheses that name a target setting, baseline, expected observable, and evaluation plan concretely enough that an experiment can be staged.

D4 is uniformly high across systems (7.7–7.9) because the structured-output prompt requires every hypothesis to populate the seven-field research-claim schema (Appendix K); falsifiability is therefore driven by the prompt template, not by the generation strategy. D1, D2, D3 differ across systems and are what Table 1 (right block) actually contrasts.

## I. Prompt Templates

We summarize the structural commitments of the three workflow stages (initialization, window update, hypothesis generation) and the two judging stages (GPT-4o-mini pre-filter, GPT-4o re-judgment) below.

**Stage 1 (Initialization) prompt.** The LLM is given a topic query, a list of $\leq 48$ pre-window arXiv papers (title, abstract, full text), and instructions to extract per-paper structured notes (core method, key findings, datasets, open questions) and to organize them into Markdown topic files partitioned by sub-theme.

**Stage 2 (Window update) prompt.** The LLM is given the prior topic files (the current $\mathcal{K}_{t-1}$), the in-window papers $\mathcal{P}_t$, and instructions to produce the updated $\mathcal{K}_t$: append novel claims, mark confirming evidence, and flag contradictions where new findings conflict with $\mathcal{K}_{t-1}$.

**Stage 3 (Hypothesis generation) prompt.** The LLM is given $\mathcal{K}_t$, a running summary of cross-window changes $\mathcal{S}_{1:t-1}$, the cumulative prior hypotheses $\mathcal{H}_{1:t-1}$, and instructions to emit a small batch of new hypotheses in the structured JSON schema documented in Appendix K.

**Judging prompts.** Both judges receive a candidate hypothesis and a candidate future paper (title + abstract + first 2000 characters of body) and return a 1–10 alignment score with a one-paragraph reasoning. The pre-filter (GPT-4o-mini) accepts candidates above $5.0$; the re-judge (GPT-4o) confirms hits above $6.0$.

## J. Additional Hit Examples

The main paper highlights one hit (Figure 2, RLHF stability). We present three additional cases below, sampled to span different research communities and trigger types. In each box the workflow saw only papers published before the listed generation date, and the matched paper appeared after.

**Case A: OOD Detection for LLMs** — **Trigger: CONVERGENCE · Generated May 2024 · Hit ✓ 280-day lead**

> *"Leveraging large language models to generate diverse textual outlier samples for a given in-distribution text classification task, by prompting the LLM to create semantically distinct but plausible OOD examples, will significantly improve OOD detection performance (AUROC, FPR@95) compared to methods relying on synthetic outliers derived solely from ID data, particularly for near-OOD scenarios."*

**Matched (Feb 2025)** → *Out-of-Distribution Detection using Synthetic Data Generation* (arXiv:2502.03323), which independently proposed an LLM-driven synthetic-OOD pipeline targeting the same near-OOD AUROC/FPR@95 metrics named in the hypothesis.

**Case B: Long-context Multimodal Modeling** — **Trigger: CONVERGENCE · Generated March 2024 · Hit ✓ 326-day lead**

> *"A multimodal LLM architecture that combines hierarchical token merging for local segment features with a dynamic, compressed memory bank for long-term historical context will achieve superior fine-grained understanding and temporal consistency in long-term video question answering and captioning, while maintaining efficient GPU memory usage."*

**Matched (Jan 2025)** → *InternVideo2.5: Empowering Video MLLMs with Long and Rich Context Modeling* (arXiv:2501.12386), which uses hierarchical token compression with a memory mechanism for long-video understanding; both structural choices named in the hypothesis.

**Case C: Knowledge Graph Completion** **Trigger: CONVERGENCE · Generated September 2024 · Hit ✓ ∼120-day lead**

> *"A novel inductive KGC framework that integrates LLM-based reasoning with explicit Context Graphs by leveraging latent type constraints and degree-based filtered reasoning paths."*

**Matched (Jan 2025)** → *KG-CF: Knowledge Graph Completion with Context Filtering under the Guidance of LLMs* (arXiv:2501.02711), which combines LLM reasoning with context-graph filtering using type and degree constraints; all three structural elements named in the hypothesis.

**What these cases share.** Across the four hits documented in this paper (RLHF in the main body and Cases A–C here), three patterns recur. First, all four were triggered by CONVERGENCE signals, consistent with F3. Second, the hypothesis text in each case fuses 3–4 pre-window source papers into a single integrated proposal, and the matched future paper realizes essentially that integration: the workflow's value comes less from generating wholly new ideas than from naming the integration the field is about to make. Third, the leads range from ∼120 to ∼680 days, comfortably above the per-topic median (193 days for the long-context topic alone), indicating that the workflow is not merely guessing the next month but anticipating directions on a multi-quarter horizon.

## K. Using a CKM Hypothesis: A Worked Example

A typical run returns ∼17 ranked hypotheses per topic. Each hypothesis is a structured artifact, not a free-form paragraph, and is meant to be *consumed* not merely read. This appendix shows the full schema and walks through how a researcher uses one.

**Hypothesis schema.** Every generated hypothesis follows the structure below. The example fields are filled with the actual content of the RLHF hit highlighted in Figure 2 (hypothesis ID `hyp-2024-11-015`, generated November 2024).

- **Statement** (one sentence): *"A novel RLHF framework will integrate adaptive entropy regularization (H-DPO/SEE-DPO style) with theoretically grounded KL-regularization and dynamic uncertainty-aware policy optimization, demonstrating enhanced stability and reduced reward hacking compared to standard DPO or PPO-based RLHF."*
- **Research Claim** (six fields):
  - PROBLEM: RLHF and DPO methods often struggle with reward overoptimization, mode collapse, and instability, particularly in complex generative tasks.
  - METHOD DELTA: Integrate adaptive entropy regularization with theoretically grounded KL-regularization and dynamic uncertainty-aware policy optimization.
  - TARGET SETTING: LLM alignment (math, code, instruction following) and text-to-image diffusion alignment.
  - BASELINE: Standard DPO or PPO-based RLHF with fixed KL regularization.
  - EXPECTED OBSERVABLE: Smoother reward curves, lower rates of repetitive/nonsensical high-reward outputs, higher pass@k for math/code, better diversity metrics.
  - EVALUATION PLAN: Implement and evaluate on GSM8K, HumanEval, MMLU-Pro, IFEval (LLMs) and Pick-a-Pic-V1 (diffusion); ablate the adaptive regularization components.
  - FAILURE MODE: Interplay between multiple regularizers may introduce hyperparameter tuning challenges.
- **Source Papers** (the workflow's evidence anchors, all pre-window):
  - arXiv:2411.07595 – *Entropy Controllable Direct Preference Optimization* (H-DPO).
  - arXiv:2411.04712 – *SEE-DPO: Self Entropy Enhanced Direct Preference Optimization*.
  - arXiv:2411.04625 – *Sharp Analysis for KL-Regularized Contextual Bandits and RLHF*.
  - arXiv:2403.05171 – *Overcoming Reward Overoptimization via Adversarial Policy Optimization with Lightweight Uncertainty Estimation*.
- **Trigger**: CONVERGENCE; multiple recent papers independently address RLHF stability via adaptive regularization.
- **Self-Assessment** (1–5): novelty 4, feasibility 3, impact 5.

**How a researcher consumes one hypothesis.** We recommend the following five-step protocol, which takes roughly three to five minutes per hypothesis and pairs naturally with the F3 detection signal in §4:

1. **Triage by self-assessment** – drop the hypothesis if feasibility $< 3$ (removes about a third of a typical run).

2. **Triage by trigger** – if the trigger type is CONTRADICTION or CROSS_PAPER, deprioritize unless explicitly hunting high-risk directions (per F3 and Appendix C).
3. **Verify via source papers, not via the hypothesis text** – skim the four cited arXiv abstracts and decide whether the cited papers individually justify the proposed integration. The hypothesis's value lies in naming the integration, not in citing well.
4. **Sanity-check the experiment plan** – the TARGET SETTING, BASELINE, and EXPECTED OBSERVABLE fields should compose into a one-day pilot you would actually run with current infrastructure.
5. **Promote or discard** – two or three hypotheses per topic typically survive this filter; queue them for a 1–2 day pilot and discard the rest.