AN IMPORTANT BOTTLENECK IN LLM OPTIMIZERS: PRINCIPLE-LEVEL LOCAL OPTIMA

Anonymous authors
Paper under double-blind review

ABSTRACT

Feedback-driven LLM optimizers are increasingly used for evolutionary search, Bayesian optimization, heuristic design, and scientific discovery. Yet rapid initial gains often give way to prolonged plateaus despite substantial remaining evaluation budgets. We investigate *principle-level search collapse*, where an optimizer's search becomes confined to a narrow set of design principles, even though it may continue to generate distinct candidates and achieve small improvements. We assemble a diagnostic benchmark spanning molecular optimization, network sentinel-node selection, and GPU-kernel optimization, and introduce a decision-tree protocol combining principle narrowing, prolonged performance plateaus, and gaps to attainable reference solutions. Across 161 assessable runs on the three problem domains, 78.3% receive support for collapse, including 55.3% with strong support. Candidate-level variation can nevertheless conceal restricted search at the principle level. We then test *diversity injection*, a common response to search stagnation, such as higher t, multiple LLMs, and web search, and show that it provides limited and inconsistent benefits: among 65 molecular runs, still 73.8% and 53.8% receive support and strong support respectively. We finally test reference-principle interventions and find that they improve some runs, but their benefits vary across models and tasks: access to a better principle does not guarantee its effective use. Our benchmark and diagnostic protocol provide a basis for evaluating future optimizers on principle-level collapse. Our findings highlight the need for online collapse detection and better mechanisms for generating and maintaining alternative principles during optimization.

1 INTRODUCTION

Large language models are increasingly used as iterative optimizers: they propose candidates, receive evaluator feedback, and use that feedback to guide the next proposals. This pattern appears in evolutionary and Bayesian search, heuristic design, code optimization and open-ended scientific discovery (Ran et al., 2025b; Agarwal et al., 2025; Ye et al., 2024; Novikov et al., 2025; Agarwal et al., 2026; Qu et al., 2026). However, the ability to improve from feedback raises a deeper question: when an LLM optimizer stops improving, has it exhausted the search space, or has it become trapped by the *principles* that made its early improvements possible? Recent work has begun to report versions of this behavior: LLM optimizers often exploit early successes, converge quickly,and then plateau in ways that are difficult to escape (Redko et al., 2026; Wiedemann et al., 2026; Li et al., 2026; Ran et al., 2025b), but what exactly has collapsed, and at which level, remains undiagnosed.

We found that this collapse often happens at the level of design principles rather than at the level of surface candidates. We define a *principle* as a design assumption behind a family of candidates, such as "modifying a known molecular scaffold", "selecting nodes with a particular network role", or "split the KV range into parallel blocks and merge partial results". This failure can be hard to see because candidate-level diversity can remain healthy: molecules may remain diverse under Tanimoto-based measures, code may not repeat verbatim, and node sets may continue to change. Yet these different candidates can still instantiate a narrow set of underlying principles, and these principles stops changing, causing the search to become stuck in a local optimum. We call this failure **principle-level search collapse**; Section 3.1 makes the two levels precise and Section 3.2 states how we diagnose it.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The usual response to a plateau is *diversity injection*: we define *diversity injection* as methods that try to broaden the candidate pool through more random search or external source, such as higher temperature, external search, multiple LLMs, or other sources of variation (Wang et al., 2026; Ran et al., 2025a; Qu et al., 2026); in our results, these methods can increase candidate diversity but do not change which principle dominates the retained solutions. Because they are still exploring the upper bound of the existing prior, and their benefits are limited, especially for open-ended scientific discovery problems where good answers cannot be found in existing sources. Methods that revise principles in response to anomalies face a different problem: a poorly performing search direction can produce feedback consistent with predictions, so stagnation need not trigger revision (Pu et al., 2026). Furthermore, we show that even supplying a better principle is a non-trivial operation because it needs to consider the current population and generalizability across different LLMs. These failures point to a different requirement: an optimizer should detect principle-level locking during its own run and should have a mechanism for proposing and maintaining alternative principles.

Contributions: (i) **A diagnostic framework.** We characterize principle-level search collapse and introduce a decision-tree protocol that combines principle narrowing, prolonged performance plateaus, and gaps to attainable reference solutions. This framework distinguishes evidence of harmful restriction from benign convergence without treating candidate diversity alone as evidence of continued exploration (§3). (ii) **A cross-domain diagnostic benchmark.** We assemble a reusable benchmark spanning molecular optimization, network sentinel-node selection, and GPU-kernel optimization, covering multiple optimizer designs and LLM backbones. The results reveal that principle-level restriction can persist beneath continued candidate variation, providing a basis for evaluating whether future optimizers alleviate this failure (§4). We provide open-source code for the benchmark and analysis.[1] (iii) **Evidence on how restricted search persists.** We examine how weak first attempts are associated with less subsequent investment in kernel search, while PiEvo's prediction-error-based updates can leave poorly performing but predictable directions unrevised (§4.1). (iv) **An evaluation of current remedies.** We systematically test diversity injection and reference-principle interventions. Diversity injection offers limited and inconsistent benefits, while access to a better principle does not guarantee its effective use. These findings motivate online collapse detection and better mechanisms for generating and maintaining principles (§4.2, §4.3).

2 Related Work

Exploration, Prior Dependence, and Diversity Collapse. Exploration collapse in self-evolving agents (Li et al., 2026), semantic diversity collapse in multi-agent ideation (Chen et al., 2026), prior-driven rather than feedback-driven behaviour in code optimization (Redko et al., 2026), and mode collapse in GPU-kernel evolution (Wiedemann et al., 2026) all report that LLM search narrows. These works point to a recurring narrowing behavior in LLM search. Although KernelFoundry and K-Search (Cao et al., 2026) show that preserving routes can be useful, our focus is different. We treat narrowing as a failure mode to be diagnosed rather than only as a behavior to be mitigated architecturally. In particular, we ask which level has collapsed: the surface candidates, the principles being proposed, or the principles that survive selection and condition future search. A run can continue generating diverse candidates while surviving principles have already narrowed. We also separate benign convergence from local optima by comparing the narrowed trajectory to a third-party reference solution. This points to a concrete research direction: optimizers should detect principle-level locking online and should have mechanisms for proposing and maintaining alternative principles during the run.

LLMs as feedback-driven optimizers. LLM-in-the-loop search appears as evolutionary optimization with LLM operators (Ran et al., 2025b;a; Novikov et al., 2025; Ye et al., 2024), LLM-assisted Bayesian and hierarchical optimization (Agarwal et al., 2025; Schwanke et al., 2026), and open-ended discovery driven by surprise (Agarwal et al., 2026). These works propose methods; they do not diagnose trajectory-level failure. We study a failure mode of the structure they share, and we cover dominant families: LLM-as-optimizer in evolutionary loop and Bayesian Optimization, and with explicit principle space modeling. Recently, there are principle-aware methods try to address the this object, but is incomplete. They still depend on the coverage and revision of the principle

[1] https://anonymous.4open.science/r/LLM-Principle-Collapse-FBEA/

space: a predefined or early principle set that misses a high-ceiling direction can itself become the search bottleneck, and an anomaly-based revision rule may not expand it when poor performance remains consistent with the model's predictions (Pu et al., 2025; 2026). Our PiEvo experiments in §4.1 evaluate this boundary empirically.

Quality-diversity and behavioural descriptors. Our descriptor is a behaviour descriptor in the sense of quality-diversity (QD) search (Mouret & Clune, 2015; Pugh et al., 2016), a concept established in the evolutionary computation literature, and our candidate-versus-principle distinction is the genotype-versus-behaviour distinction of Mouret & Doncieux (2012). More broadly, this concentration has a classical analogue in evolutionary computation: Holland's schema theorem describes how above-average schemata can be amplified under selection (Holland, 1992). The self-reinforcement of a low-ceiling principle is the deception problem that motivated novelty search (Lehman & Stanley, 2011). Descriptors need not be hand-specified: AURORA learns them (Cully, 2019), QDAIF and ELM let a language model define or apply them (Bradley et al., 2024; Lehman et al., 2022). A QD archive is built to stop selection from erasing already discovered behavioural niches: once a cell is filled, a better candidate from another cell cannot overwrite it. That protection applies only to the descriptor used to define the grid, and only to niches the search has already reached. Mainstream LLM optimizers typically apply selection directly to surviving candidates and keep no archive at the principle level, so concentration at that level is the expected outcome.

3 Measuring Principle-Level Collapse

This section makes the failure measurable. Section 3.1 defines principles and candidates. Section 3.2 states the three conditions under which we diagnose principle-level collapse and how they combine into a verdict. Section 3.3 describes how we observe the principles an optimizer keeps on each task, and Table 1 gives the task-specific values.

3.1 Principles and candidates

A *candidate* is one concrete solution the optimizer emits and the evaluator scores: a kernel source file, a molecule, a set of sentinel nodes. A *principle* is the design idea a family of candidates shares. On GPU kernels, "pad the keys and values into one batched matrix multiply" and "split the KV range into parallel blocks and merge the partial results" are two principles; the dozens of kernels that vary tile sizes under either are candidates of the same principle. On molecules an example principle is the scaffold kept while substituents change; on sentinel-node selection it is the pool of nodes a run keeps drawing from. We model the loop as sampling candidates from a mixture over principles,

$$x_{t+1} \sim q(x \mid H_t) = \sum_{p \in \mathcal{P}} q(x \mid p, H_t)\, P(p \mid H_t), \tag{1}$$

where H_t is the history available at step t. A candidate-level local optimum means the candidates stop changing; principle-level collapse means the weights $P(p \mid H_t)$ have concentrated on a few principles and stopped moving while candidates within them can still change.

3.2 Diagnosing principle-level collapse

We diagnose principle-level collapse using three complementary conditions: **(1)A substantial reference gap.** The run remains well below an attainable reference solution, indicating scope for improvement rather than convergence to a known optimum. **(2)Principle narrowing.** Search concentrates on a restricted set of principles, measured through task-specific structural descriptors or, when available, an explicit model of the principles used during search. **(3)A long performance plateau.** Best-so-far performance makes little subsequent progress despite substantial remaining evaluation budget.

No single condition establishes collapse. Let $F_T = \max_{t \le T} f(x_t)$ and $\Delta_T = f(x_{\mathrm{ref}}) - F_T$. We denote a substantial reference gap by $\mathrm{Headroom} = [\Delta_T \ge \gamma]$ (2) and write Narrow for structural evidence of principle narrowing. Narrowing is measured on the diagnostic set $\mathcal{A}(b)$, the candidates the optimizer keeps at budget b (Table 1). A post-hoc descriptor assigns each valid candidate one

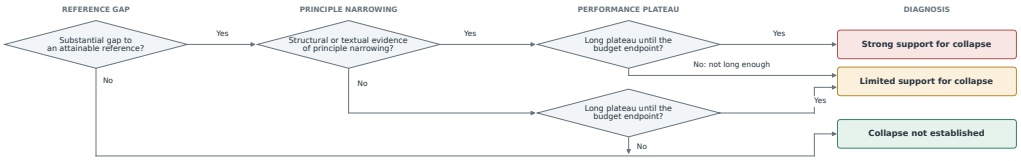


Figure 1: Overview of the collapse diagnosis. Narrowing may be supported by structural metrics or an eligible audit of principle texts. A long plateau leaves at least 30% of planned budget. The lower branch summarizes limited narrowing evidence. Text-audit eligibility and exceptions are specified in the full protocol (Appendix C, Figure 6).

structural label, $\phi : \mathcal{X}^{\text{valid}} \to \mathcal{P}$. Its empirical distribution in the diagnostic set $\mathcal{A}(b)$ is

$$\hat{\pi}_b(p) = \frac{|\{x \in \mathcal{A}(b) : \phi(x) = p\}|}{|\mathcal{A}(b)|}. \tag{3}$$

Effective count $N_{\text{eff}} = \exp(-\sum_p \hat{\pi}_b(p) \ln \hat{\pi}_b(p))$ measures effective scaffolds or routes. This metric accounts for both the number of unique scaffolds and their frequency: for example, if two scaffolds occur among 50 candidates but one appears 49 times and the other only once, N_{eff} remains close to 1 rather than 2, reflecting the strong concentration on a single scaffold. Narrow holds when the final N_{eff} is at most the task cutoff in Table 1. $\mathrm{Plateau}$ holds when the run's best result stops improving while a substantial share of the budget is still unspent (cutoffs in Table 1). Strong support requires all three conditions:

$$\mathrm{Strong} = \mathrm{Headroom} \wedge \mathrm{Plateau} \wedge \mathrm{Narrow}\ . \tag{4}$$

If $\mathrm{Headroom}$ fails, collapse is not established: the run may have converged to a competitive solution. If $\mathrm{Headroom}$ holds but only one of Narrow and $\mathrm{Plateau}$ does, the run receives limited support. For an optimizer that keeps explicit principle texts (PiEvo here), an audit of those texts can also establish Narrow (Appendix B). Incomplete evidence yields limited support or collapse not established, instead of a claim that search remains diverse. Figure 1 summarizes the rule. Because the latent principle distribution $P(p \mid H_t)$ is not directly observable, we use these structural measurements as observable proxies for principle diversity and concentration.

3.3 Instantiation on the three tasks

Table 1: Instantiation of the framework on the three tasks. Task-specific structural descriptors and references complement the graded decision tree in Figure 1.

	GPU kernel	**Molecular optimization**	**Sentinel-node selection**
Candidate x	one PyTorch/Triton source file	SMILES string	node subset of size r (4 or 5)
Utility f	$-\ln$(geometric-mean latency)	scalarized five-objective score	$-\ln$(approximation error)
Descriptor ϕ	executed route, blind LLM annotation against a per-problem rubric (6–9 leaf routes + families)	Bemis–Murcko scaffold	node incidence in selected sets
Diagnostic set $\mathcal{A}$	fastest 10 valid kernels	top 50 distinct molecules	top 50 distinct node sets
Candidate-level diversity	5-gram Jaccard distance between source files	fingerprint diversity, uniqueness	set uniqueness
Reference x_{ref}	FlashInfer library kernel, same GPU	best known molecule, $F = 4.6985$	exhaustive optimum (Dolphin); published optimized set (residence hall)
Headroom threshold γ	$\ln 2$ (best latency $\geq 2\times$ expert)	top-1 below 4.3	$\ln 2$ (error $\geq 2\times$ reference)
Narrowing cutoff (final N_{eff})	≤ 2 routes	≤ 3 scaffolds	$\leq 4\times$ the distinct-set lower bound (31.4 / 31.9 nodes)
Plateau tolerance (later gain allowed)	$\leq 10\%$ of the remaining gap to the expert	≤ 0.01 in Top-1 fitness	$\leq 1\%$ relative error reduction
Negative control	RMSNorm	—	Dolphin (exhaustive reference)
Intervention	L1/L2/L3 $\times$ {onset, start}	L2 $\times$ {40%, start}; temperature	L2 $\times$ {40%}

Table 1 instantiates the framework across tasks. We measure search concentration through task-specific structural descriptors: molecular scaffolds, node incidence in sentinel sets, and executed kernel routes. For molecules and networks, we reconstruct a diagnostic set $\mathcal{A}_{50}(b)$ of top 50 distinct candidates from the archive at evaluation budget b. Kernels use the top 10 candidates because of smaller budget and population size. Networks instead normalize node-occurrence counts across

top 50 unordered sets, since each candidate contains several nodes instead of a single label; applying the same entropy-based measure to node frequencies gives effective nodes, not a literal number of principles, so the network cutoff is stated relative to the lower bound that 50 distinct sets impose. References are the best molecule reported across 100 runs by ExLLM, the authors' published residence-hall set, and the expert library kernel timed on the same GPU; Dolphin, whose optimum is known by enumeration, and RMSNorm, where every run beats the expert, are negative controls. PiEvo keeps explicit principle texts, so its narrowing is additionally audited on its principle texts. We additionally report molecular scaffold richness, dominant share and Morgan-fingerprint diversity; network normalized top-five coverage and pair/triple reuse; and kernel dominant-route share and source-code 5-gram Jaccard diversity. Appendix L.1 gives the formulas, bounds and plotting conventions. We use task-specific working thresholds, including at least 30% remaining budget for a long plateau. Detailed thresholds and textual-audit rules are in Appendix B; the full decision tree and classifications are in Appendix C. Threshold sensitivity is discussed in Appendix I.

4 RESULTS

Tasks. GPU kernel optimization uses four operators from DeepSeek-V3 and Qwen3 inference on FlashInfer-Bench (Lin et al., 2026) (GQA paged decode, MLA paged decode, MLA paged prefill with causal mask, FP8 block-scale MoE) plus RMSNorm as a negative control. A candidate is one self-contained PyTorch/Triton file; the official harness compiles it, checks correctness on every workload and reports geometric-mean latency on a data-center GPU with locked clocks. The library is not installed in the evaluation container, so candidates cannot reach it by delegation. We also adopt a five-objective **Molecular Optimization** from prior LLM optimizers (Ran et al., 2025b;a; Wang et al., 2025): maximize QED and JNK3 activity while minimizing SA, DRD2 and GSK3β; candidates are SMILES strings. Reference molecules have scores $>$4.6 and the best one achieves 4.699 from ExLLM. **Sentinel-node selection** follows MacLaren et al. (2025): choose a small node subset whose mean state tracks the mean state of a networked dynamical system, on the Dolphin social network ($N = 62$, $r = 4$, double-well dynamics) and a residence-hall network ($N = 217$, $r = 5$, gene-regulatory dynamics). The Dolphin instance is small enough to enumerate, so its reference is the exhaustive optimum (4.041×10^{-3}); for the residence hall, with 3.8×10^9 subsets, the reference is the authors' published optimized set (2.3818×10^{-7}). Except for Dolphin and RMSNorm, we study these tasks as open-ended scientific optimization problems without known global optima. We therefore use independently obtained reference solutions to assess the remaining scope for improvement. For molecular optimization and Residence Hall, the references are the best solutions reported across 100 runs in the original studies; for the remaining kernel tasks, they are expert implementations reported in the corresponding papers. These references are not guaranteed to be globally optimal. They demonstrate performance achieved with finite computational or human effort under a fixed evaluation budget. We use the reference gaps to contextualize search stagnation and principle interventions. Evaluation budgets are 200 for each kernel task, 5,000 for molecules, 3,100 for Dolphin, and 10,850 for residence hall following the corresponding studies.

Optimizers. We select optimizers to cover LLM-as-optimizers in evolutionary search, reflective optimization, and Bayesian optimization. ExLLM (Ran et al., 2025b) combines evolutionary search with a compact, evolving experience mechanism and reports state-of-the-art results across multiple domains. ReEvo (Ye et al., 2024) represents reflective evolutionary optimization, while MCCE (Ran et al., 2025a) combines multi-LLM collaboration with RL-based adaptation of a trainable optimizer. PiEvo (Pu et al., 2026), introduced at ICML 2026, is also a LLM optimizer performing Bayesian optimization over an evolving principle space and revising principles in response to detected anomalies. For ExLLM, we use GPT-5.4, Claude Sonnet 5, Qwen3.5-35B-A3B, and DeepSeek V4 Flash: two proprietary and two open-weight backbones, to assess whether the findings persist across different LLMs. Together, these settings allow us to test whether principle-level collapse is broadly observed across different LLM-optimizer algorithms, LLM backbones, and problem domains. We additionally evaluate GPT-4o on molecules and kernels to compare older and newer backbones. This comparison is motivated by the substantially higher molecular uniqueness reported for GPT-4o in ExLLM; the molecular findings are discussed in Appendix F, with kernel results in Appendix I.

Intervention protocol. All tasks include an **L2 intervention**, which provides a specific design principle underlying reference solutions without revealing the candidates themselves. For kernel

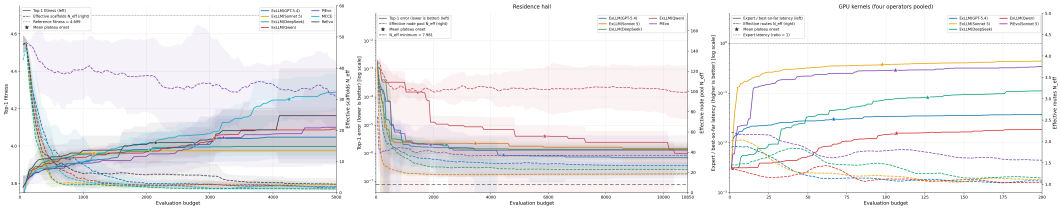


Figure 2: Best performance and effective count: molecules (left), residence hall (middle) and GPU kernels (right; the four operators pooled). Solid/dashed lines use the left/right axes. Stars mark mean plateau onset. The molecular reference is 4.699; the dashed line in the network is the lower bound of N_{eff}; the kernel panel shows expert latency divided by best-so-far latency (expert $= 1$) as the geometric mean across runs, and N_{eff} in the diagnostic set of the ten fastest kernels; per-operator curves are in Figure 5.

optimization, we additionally test two intervention strengths. **L1** provides only a general warning that the search may have settled prematurely on one decomposition and encourages exploring a fundamentally different one, without naming a specific alternative. **L3** supplements L2 with an implementation outline, including grid organization, tiling, and data layout, but provides no code.

4.1 Diagnosing Collapse and Its Severity

We ask three questions in the result section: (1) how common and severe is principle-level collapse, (2) can diversity injection resolve it, and (3) does supplying a better principle directly improve performance?

Principle-level collapse We report diagnostic counts as strong support / limited support / not established (S/L/N). Including GPT-4o, the molecular comparison covers five ExLLM backbones, PiEvo, MCCE and ReEvo, giving $23/13/4$ across 40 runs. Residence hall gives $13/6/6$ and Dolphin $0/0/25$, each across 25 runs. On residence hall, the four ExLLM backbones have mean best errors 2.94–6.36 times the published reference. Sonnet, DeepSeek and GPT-5.4 reduce their effective node pools from 72–119 to 19–27. Their saved experiences contain fixed pools or anchors, such as *keep 143 fixed* (Appendix K). Qwen retains a broad effective node pool of about 99 despite its performance gap. Its S/L/N count is $0/4/1$: four runs provide limited evidence of harmful stagnation. The molecular comparison additionally includes MCCE and ReEvo. For the four primary ExLLM backbones, all 20 molecular runs finish below 4.3; their mean plateau onsets fall between 23% and 62% of budget. Their mean effective scaffold counts fall from 39–43 to 1.24–2.58. MCCE and ReEvo also concentrate strongly, ending at 1.66 and 2.74 effective scaffolds on average. Their performance qualifications differ: MCCE's mean plateau begins at 85% of budget, and one ReEvo seed reaches 4.579. Limited support reflects either a late plateau or incomplete evidence of narrowing.

Kernel search shows the same restriction at the level of executed decompositions (Figure 2, right, and Figure 5). We measure route frequencies among the top 10 kernels found so far. Across the four main ExLLM backbones and four different kernels, 64/80 runs finish with at most two effective routes while remaining at least twice as slow as the expert. Of these, 48 also plateau with at least 30% of their observed evaluation attempts remaining: 5/20 on GQA decode, 9/20 on MLA decode, 19/20 on MLA prefill and 15/20 on FP8 MoE. Here a plateau allows subsequent gains of at most 10% of the remaining gap to the expert; task-specific definitions and sensitivity tests are in Appendix I. The other 16 concentrated, large-gap runs plateau too late to meet this joint criterion. Concentration is widespread, but its performance cost and duration vary by task and backbone.

Together, these results identify principle-level collapse as a recurring limitation across tasks, LLM backbones and optimizer designs, warranting more attention to this bottleneck.

Candidate variation may conceal principle restriction. Figure 3 shows marked declines in uniqueness, scaffold richness and effective scaffold count, alongside increasing dominant-scaffold share. Mean molecular fingerprint diversity remains 0.46–0.59 across the four main ExLLM backbones, even as dominant share reaches 0.66–0.94. Candidate diversity also decreases, but its still-moderate values can obscure how few scaffolds account for most selected molecules. Network

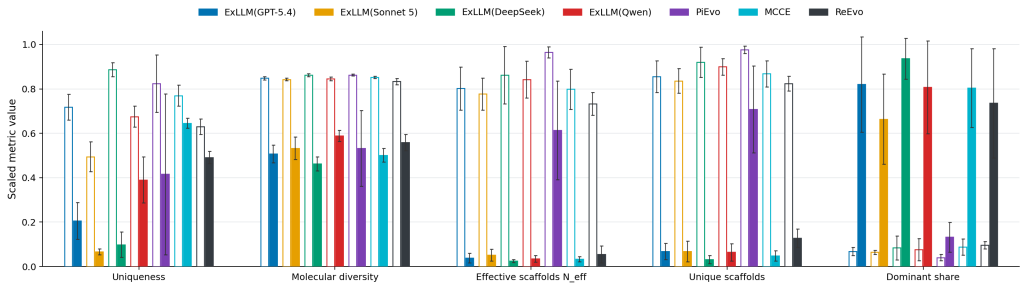


Figure 3: Molecular early/late diagnostics. Hollow/filled bars are early/final values. Uniqueness uses the first/last 1,000 generated candidates. Other metrics use reconstructed 50-candidate sets. Effective and unique scaffold counts are divided by 50.

diagnostics measure normalized coverage of the five most-used nodes and reuse of node pairs or triples; higher values indicate concentration. All are computed within seeds, not from overlap between seeds. Supplementary trajectories and network bars show that repeated structures and node combinations become more prevalent, although the extent differs across backbones (Appendix D, Figures 7 and 8). For kernels, the final effective route count averages only 1.03–1.14 across the four backbones, with dominant-route shares of 0.96–0.99. Yet mean code diversity, measured by pairwise token 5-gram Jaccard distance, remains 0.55–0.72 for GPT-5.4, DeepSeek and Qwen. Sonnet's code diversity falls more sharply, to 0.31. Thus changing source code need not change the computational decomposition. Appendix I shows the early/late comparisons

First-attempt quality and subsequent development. A route that first appears with a weak implementation tends to receive little further development. For every route that enters a kernel run after the run's first route, we compare its first kernel with the run's best latency at that moment and count the kernels the run later spends on it (Figure 23). Routes whose first kernel already beats the incumbent are developed for dozens of further kernels; routes that start behind receive a handful, and those that start far behind are often never tried again. Restricting to routes that appear with ample budget left does not change this ordering, so late arrival does not explain it. The expert's route is no exception: it almost always enters slower than the incumbent, and whether the run then keeps investing in it decides whether it ever reaches expert speed. These are associations between first-attempt quality and later investment, not a measured causal effect of selection (Appendix J).

Concentration is not always failure. Dolphin provides a reference with a different status: exhaustive enumeration establishes its optimum. PiEvo's five best errors are within 0–4.4% of it, including one exact optimum; the ExLLM mean errors are also within 5%. Restriction here need not imply a costly missed direction. RMSNorm plays the analogous role for kernel search, although beating an expert implementation does not prove global optimality. All 25 ExLLM RMSNorm runs, including the GPT-4o control, finish at 0.437–0.480 times the expert latency; 24 also have at most two effective routes. None meets the large-gap collapse criterion. These two controls distinguish benign convergence toward competitive solutions from harmful narrowing that leaves substantial attainable gains unexplored (Appendix D, Figure 9).

PiEvo: coverage without continued development. PiEvo's final molecular top-50 retains 30.69 effective scaffolds on average. Yet broader structures can implement the same strategy. In seed 42, search plateaus at budget 1,295 with Top-1 3.898, leaving a 0.800 reference gap and performing worse than the other methods' mean endpoints. A macrocycle-oriented strategy accounts for 90.4% of the last 1,000 generated attempts; candidate inspection links explicit macrocycle instructions to molecules containing 8–15-member rings. Thus broad scaffold coverage can coexist with restricted strategy use and substantial performance headroom. Across molecular seeds, mean principle-generation/update counts fall from 3.8 in the first 1,000 generated attempts (after the ten forced principle-generation warm-up rounds) to 1.0 in the last 1,000 attempts. The late windows also average only one anomaly-triggered round. Combining structural evidence with the eligible textual audit gives PiEvo S/L/N counts of $2/3/0$ for molecules and $4/1/0$ for residence hall.

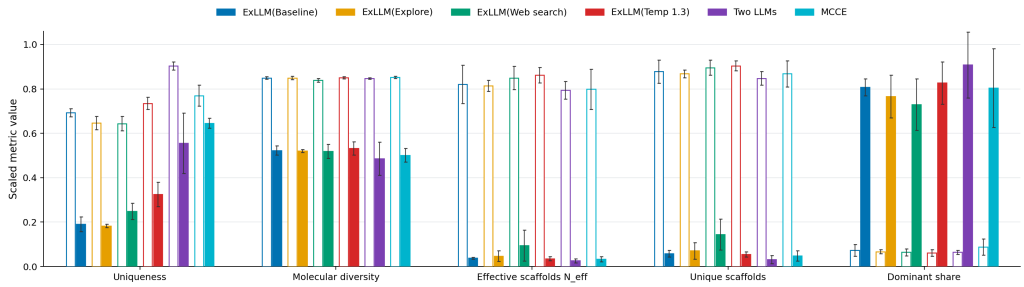


Figure 4: Diversity-injection diagnostics, with the definitions and scaling of Figure 3. ExLLM variants average backbones within each seed. Two LLMs and MCCE are independent five-seed comparisons. Broader generation does not reliably preserve a broad selected scaffold distribution.

PiEvo's anomaly detector measures GP prediction error. A poorly performing direction can therefore produce feedback consistent with the model's predictions and persist without triggering revision. This is a limitation of using surprise to guide principle updates. The saved-record audit also identifies rounds with too few scored candidates to satisfy the trigger condition. Together, these results show that PiEvo can preserve structural diversity without reliably sustaining the development of alternative principles. Maintaining several directions is insufficient if they remain unproductive and no better direction is developed. Appendix G reports the text, candidate and detector evidence, including the additional $\sigma = 0.9$ PiEvo experiment for comparison.

On kernels, PiEvo(Sonnet) reduces its mean effective route count from 2.44 to 1.51. Three of its 12 runs meet the joint collapse criterion; seven have concentrated routes but a latency ratio below two, one has a large gap but plateaus late, and one remains unconcentrated. PiEvo(Gemini) has fewer than 20 valid kernels in 10/12 runs, so those runs cannot support a reliable concentration verdict. Explicit principle modeling therefore neither guarantees escape nor implies failure in every run; valid-candidate coverage must also be reported.

4.2 Can Diversity Injection Resolve Collapse?

A natural response to prolonged stagnation is diversity injection: broadening candidate generation in the hope of finding a more productive direction. We test whether such changes prevent or reduce principle-level collapse. On molecules, we test higher temperature (1.3), parent-independent proposals (crossover:mutation:explore $= 7 : 2 : 1$), external literature/database search, and joint proposing by two LLMs. MCCE adds a comparison combining multi-LLM collaboration and reinforcement-based adaptation. Figure 4 averages the four ExLLM backbones within each seed to summarize the common concentration trend; temperature uses three because Sonnet does not support this setting. Per-backbone plots and five-seed tables are in Appendix E. Across 65 runs, the S/L/N counts are $35/13/17$. Among the 17 not-established runs, nine reach Top-1 ≥ 4.3; eight remain below 4.3 but lack both structural concentration and a long plateau. The nine higher-scoring runs are scattered across methods and backbones, with no method–backbone combination reaching this threshold in all five seeds. Only four of the 65 runs exceed 4.6. These gains are occasional rather than a reliable resolution of collapse; detailed counts are in Appendix C.

Mean Top-1 rises from 4.028 for pooled ExLLM to 4.048 with explore and 4.108 with web search. Temperature reaches 4.086 on its three-backbone subset. Two-LLM proposing and MCCE reach 4.165 and 4.288, with global uniqueness 0.636 and 0.656, but retain concentrated final scaffold distributions. MCCE delays its plateau: its S/L/N count is $0/3/2$, reflecting two endpoints above the large-gap cutoff and three late plateaus rather than consistently successful search. Overall, diversity injection introduces additional search variation and can yield occasional substantial gains, but its benefits are limited and inconsistent across seeds and backbones. This supports our hypothesis that broadening candidate generation alone does not reliably overcome principle-level collapse. These open-ended tasks have no known globally optimal answer to retrieve. Diversity injection broadens sampling from model priors and external knowledge, but does not itself ensure that the optimizer recognizes an unproductive principle or develops an alternative. The results are consistent with

this limitation. Because the variants operate from the start, they test prevention or reduction of collapse, not recovery from an already diagnosed collapse. We next test whether supplying a specific reference-informed direction helps.

4.3 Reference Principles Help, but Their Use Is Nontrivial

A better direction can improve an established trajectory. Using ExLLM with the four main backbones, we first supply L2 at 40% of the molecular and network budgets; kernel interventions branch at detected narrowing onset. Hints name a search direction without giving a molecule, node set or kernel implementation (Appendix M). On residence hall, mean best error decreases by 52.7% for Qwen, 32.6% for DeepSeek and 27.0% for GPT-5.4; Sonnet does not improve on average. Dolphin improves little near its optimum. On molecules, delayed L2 raises mean Top-1 from 3.975 to 4.485 for Sonnet and from 3.998 to 4.699 for DeepSeek, but not for GPT-5.4 or Qwen. Success means improvement beyond the previous plateau. Full comparisons of metrics are in Figure 12 and numbers are in Appendix H. For kernels, onset-L2 improves 13/41 matched runs, worsens nine and leaves 19 within a $\pm 5\%$ band of paired gain (baseline latency divided by hinted latency). The geometric-mean gain is 1.010, close to neutral. These are the four main backbones, restricted to trajectories for which the onset detector produced a branch; they are not a random subset of all baseline runs. Figure 15 separates adoption of the reference route from performance, and Appendix I reports all hint conditions.

Earlier injection does not remove model dependence. A delayed hint may arrive after earlier candidates and experience have restricted search. Providing L2 from the first molecular generation tests whether this timing alone explains poor outcomes. DeepSeek again reaches mean Top-1 4.699, and Sonnet improves further to 4.617. GPT-5.4 improves to 4.264 but remains below the reference region; Qwen falls from 4.091 to 4.020. Thus failures cannot be explained solely by delayed access: a hint from the start also does not ensure effective use. Kernel L2 from the start likewise gives mixed outcomes: 14/46 usable matched runs improve, 21 worsen and 11 stay within the gain band; geometric-mean gain is 0.833. Thus late access cannot be the sole explanation for ineffective hints. This is not an isolated causal effect of timing: onset arms exist only for detector-triggered baselines, whereas start arms cover all four non-control tasks.

More specific instructions need not help more. Kernels additionally compare a general redirection warning (L1), a named reference strategy (L2), and a strategy plus implementation outline (L3) as explained above. From the start, L1 has the largest pooled geometric-mean gain (1.169), compared with 0.833 for L2 and 1.067 for L3. L1 improves 28/48 matched runs but worsens 12; it is not a universal remedy. The ordering also depends on the statistic: at onset L3 improves more runs than L1 (16 versus 12 of 41), yet its geometric-mean gain is lower (0.986 versus 1.110). Naming a route and supplying an implementation outline can increase adoption without producing a fast implementation. The route-conditioned results in Appendix I expose this remaining implementation gap. These experiments separate receiving a promising direction from developing a competitive candidate. Effective principle generation and maintenance must support both.

5 Conclusion

Across molecular, network and GPU-kernel optimization, our results identify principle-level collapse as a serious and easily overlooked bottleneck in feedback-driven LLM optimization. Distinct candidates and occasional small gains can coexist with restricted design principles, prolonged plateaus and substantial gaps to attainable references. Diversity injection offers limited and inconsistent benefits, while explicit principle modeling and access to better principles do not guarantee effective exploration. The challenge is not simply to generate more candidates, but to recognize when current principles no longer support meaningful progress. We call for research on online collapse detection and better mechanisms for generating, developing and maintaining alternative principles in scientific discovery tasks. We encourage future optimizers to use our benchmark to demonstrate reduced evidence of collapse alongside improved performance and smaller reference gaps. Progress should also be evaluated across LLM backbones and task domains to establish whether these mechanisms generalize beyond individual settings.

6 AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks requiring mandatory disclosure. In particular, generative AI was not used to develop theoretical or conceptual frameworks, formulate mathematical claims, propose or refine research hypotheses, design the research methodology or experiments, implement the proposed methods, analyze data, or interpret results.

We used generative AI tools for several tasks for which disclosure is recommended. These included assisting with the organization and structuring of the manuscript, including drafting and refining the paper outline; editing the manuscript to improve grammar, clarity, and readability; and assisting with non-core software development tasks. In particular, generative AI was used to help write auxiliary helper functions for data processing and visualization, review code for potential errors and completeness, and improve code organization and readability. These uses did not involve the design or implementation of the proposed methods or the scientific interpretation of experimental results.

All AI-assisted text and code were reviewed by the authors. AI-assisted code was manually inspected and tested for correctness before being used in the reported experiments or analysis.

We take full responsibility for the final content of this work, including all text, claims, code, figures, and other artifacts produced with the aid of generative AI.

REFERENCES

Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with llms via bayesian optimization. In *International Conference on Learning Representations*, volume 2025, pp. 67180–67201, 2025.

Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson, Megha Chakravorty, Satvika Reddy Gavireddy, Aditya Parashar, Harshit Surana, Bhavana Dalvi Mishra, Andrew McCallum, Ashish Sabharwal, et al. Autodiscovery: Open-ended scientific discovery via bayesian surprise. *Advances in Neural Information Processing Systems*, 38:25181–25219, 2026.

Herbie Bradley, Andrew Dai, Hannah Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Grégory Schott, and Joel Lehman. Quality-diversity through AI feedback. In *International Conference on Learning Representations (ICLR)*, 2024.

Shiyi Cao, Ziming Mao, Joseph E Gonzalez, and Ion Stoica. K-search: Llm kernel generation via co-evolving intrinsic world model. *arXiv preprint arXiv:2602.19128*, 2026.

Nuo Chen, Yicheng Tong, Yuzhe Yang, Yufei He, Xueyi Zhang, Zou Qingyun, Qian Wang, and Bingsheng He. Diversity collapse in multi-agent llm systems: Structural coupling and collective failure in open-ended idea generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 251–306, 2026.

Antoine Cully. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2019.

John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The MIT press, 1992.

Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley. Evolution through large models. *arXiv preprint arXiv:2206.08896*, 2022.

Yibo Li, Jiashuo Yang, Zhi Zheng, Zhiyuan Hu, Yuan Sui, Shizun Wang, Yufei He, and Bryan Hooi. Apex: Autonomous policy exploration for self-evolving llm agents. *arXiv preprint arXiv:2605.21240*, 2026.

Edward Lin, Sahil Modi, Siva Kumar Sastry Hari, Qijing Huang, Zhifan Ye, Nestor Qin, Fengzhe Zhou, Yuan Zhang, Jingquan Wang, Sana Damani, et al. Sol-execbench: Speed-of-light benchmarking for real-world gpu kernels against hardware limits. *arXiv preprint arXiv:2603.19173*, 2026.

Neil G MacLaren, Baruch Barzel, and Naoki Masuda. Observing network dynamics through sentinel nodes. *Nature Communications*, 16(1):10211, 2025.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Jean-Baptiste Mouret and Stéphane Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary Computation*, 20(1):91–133, 2012.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Yingming Pu, Tao Lin, and Hongyu Chen. Piflow: Principle-aware scientific discovery with multi-agent collaboration. *arXiv preprint arXiv:2505.15047*, 2025.

Yingming Pu, Tao Lin, and Hongyu Chen. Principle-evolvable scientific discovery via uncertainty minimization. *arXiv preprint arXiv:2602.06448*, 2026.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.

Ao Qu, Han Zheng, Zijian Zhou, Yihao Yan, Yihong Tang, Shao Yong Ong, Fenglu Hong, Kaichen Zhou, Chonghe Jiang, Minwei Kong, et al. Coral: Towards autonomous multi-agent evolution for open-ended discovery. *arXiv preprint arXiv:2604.01658*, 2026.

Nian Ran, Zhongzheng Li, Yue Wang, Qingsong Ran, Xiaoyuan Zhang, Shikun Feng, Richard Allmendinger, and Xiaoguang Zhao. Mcce: A framework for multi-llm collaborative co-evolution. *arXiv preprint arXiv:2510.06270*, 2025a.

Nian Ran, Yue Wang, Xiaoyuan Zhang, Zhongzheng Li, Qingsong Ran, Wenhao Li, and Richard Allmendinger. Exllm: Experience-enhanced llm optimization for molecular design and beyond. *arXiv preprint arXiv:2502.12845*, 2025b.

Dmitry Redko, Albert Fazlyev, Konstantin Sozykin, Maria Ivanova, Evgeny Burnaev, and Egor Shvetsov. Prior knowledge or search? a study of llm agents in hardware-aware code optimization. *arXiv preprint arXiv:2605.19782*, 2026.

Andrej Schwanke, Lyubomir Ivanov, David Salinas, Fabio Ferreira, Aaron Klein, Frank Hutter, and Arber Zela. Improving llm-based global optimization with search space partitioning. In *International Conference on Learning Representations*, volume 2026, pp. 76314–76344, 2026.

Haorui Wang, Marta Skreta, Cher-Tian Ser, Wenhao Gao, Lingkai Kong, Felix Strieth-Kalthoff, Chenru Duan, Yuchen Zhuang, Yue Yu, Yanqiao Zhu, et al. Efficient evolutionary search over chemical space with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 51694–51727, 2025.

Haorui Wang, Parshin Shojaee, Kazem Meidani, Kunyang Sun, José Miguel Hernández-Lobato, Teresa Head-Gordon, Jiajun He, Chandan K Reddy, Chao Zhang, and Yuanqi Du. Towards diverse scientific hypothesis search with large language models. *arXiv preprint arXiv:2606.10587*, 2026.

Nina Wiedemann, Quentin Leboutet, Michael Paulitsch, Diana Wofk, and Benjamin Ummenhofer. Kernelfoundry: Hardware-aware evolutionary gpu kernel optimization. In *International Conference on Machine Learning (ICML)*, 2026.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. Reevo: Large language models as hyper-heuristics with reflective evolution. *Advances in neural information processing systems*, 37:43571–43608, 2024.

APPENDIX

A LIMITATIONS

The latent principle distribution is not directly computable or observable, so our diagnosis relies on proxies that introduce measurement uncertainty. Each task requires its own principle descriptors, metrics and thresholds, which limits direct comparison across domains. Applying the protocol to a new problem therefore may require identifying suitable descriptors and validating how well they capture meaningful differences in search direction, as well as calibrating the diagnostic thresholds.

REPRODUCIBILITY STATEMENT

Our code is publicly available at `https://anonymous.4open.science/r/LLM-Principle-Collapse-FBEA/`. Experimental settings and diagnostic procedures are detailed in the main text and appendices to facilitate reproduction of our results.

B DIAGNOSTIC THRESHOLDS AND TEXTUAL AUDIT

The reference-gap indicator in Eq. equation 2 uses task-specific boundaries: molecular Top-1 below 4.3 (strictly), and network error at least twice the reference. These are working diagnostic thresholds, not universal definitions of failure. A feasible reference need not be globally optimal; it establishes attainable improvement.

Structural concentration, denoted Narrow, uses the final selected-50 set:

$$\mathrm{Narrow} = \begin{cases} [N_{\mathrm{eff}} \le 3], & \text{molecules,} \\ [N_{\mathrm{eff}} \le 4N_{\mathrm{eff,min}}], & \text{networks.} \end{cases} \tag{5}$$

The distinct-set lower bounds are approximately 7.837 for Dolphin and 7.981 for residence hall, giving cutoffs 31.35 and 31.92. Node concentration is a structural proxy, not a literal count of principles. Kernel diagnostics instead use at most two effective routes among the fastest ten valid kernels; their current export and audit limitations are described in Appendix I.

Let $L = [(B - b_{\mathrm{plat}})/B \ge 0.30]$, where B is planned budget. The retrospective plateau begins when all subsequent Top-1 gains total at most 0.01 on molecules, or the subsequent best-error reduction is at most 1% on networks. This measures remaining budget, not observed evaluations wasted after early stopping.

If structural concentration is absent, textual auditing is permitted only when an explicit principle-space model covers the algorithm's actual search directions. PiEvo qualifies in this comparison; ExLLM, MCCE and ReEvo experience summaries do not represent all directions and are not audited as if they did. Coverage here does not mean that the model contains every possible useful principle. Over the last 1,000 generated attempts, including duplicates and invalid candidates, we merge wording variants and test:

- T_1: one semantic strategy accounts for at least 50% of generated attempts, weighted by actual use rather than number of texts.
- T_2: at most five minor, same-direction revisions, with no major direction change.
- T_3: no new principle direction is proposed, including directions that are never adopted.

For eligible runs, the textual branch of Eq. equation 4 is

$$\text{Principle narrowing in text} = T_1 \wedge T_2 \wedge T_3. \tag{6}$$

This combines concentrated strategy use with little revision and no newly proposed direction. For other optimizers, the textual branch cannot establish strong support; it is inapplicable, not evidence that their principles remain diverse. Figure 1 summarizes the diagnosis in four steps: reference gap, principle narrowing, plateau duration and support level. Structural and textual evidence are two ways of assessing the same narrowing condition, not separate failure types. The simplified figure

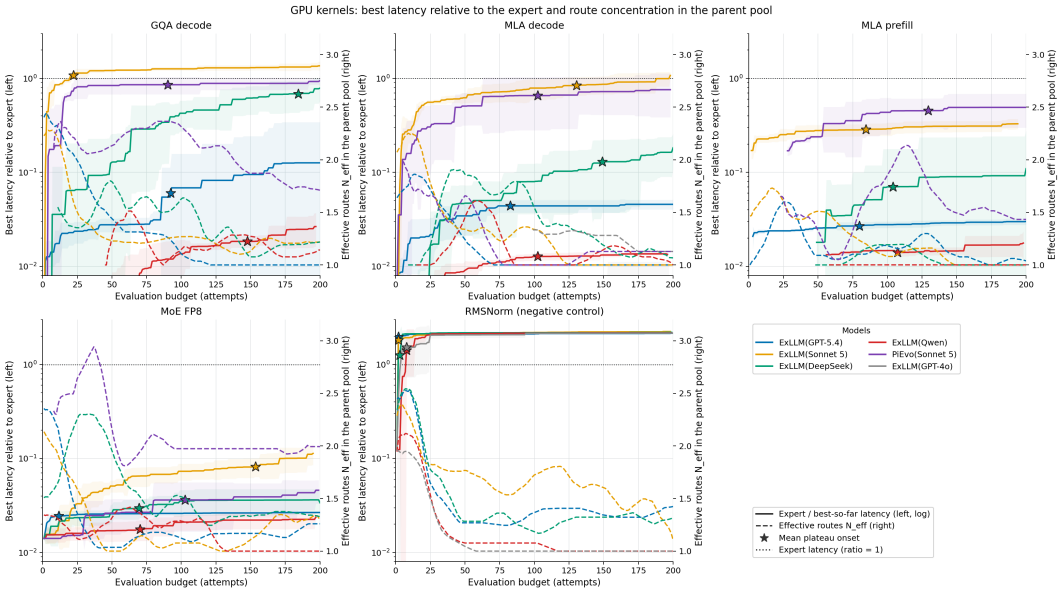


Figure 5: Kernel performance and route concentration. Solid lines show expert latency divided by best-so-far latency (left, log scale; higher is better); dashed lines show effective route count in the top 10 (right). Stars mark mean plateau onset. RMSNorm is a negative control.

omits text-audit eligibility and exceptions; the full protocol in Appendix C, Figure 6, defines the exact classifications. In particular, not every failed or unresolved textual audit qualifies for limited support merely because a plateau is long. With a large gap, concentration but a shorter plateau gives limited support. Without structural concentration, a long plateau also gives limited support if no covering principle model exists, or if the eligible audit satisfies $\neg T_1 \wedge T_2 \wedge T_3$. Other cases are classified as collapse not established. Missing audit evidence never passes a textual gate. Limited support based only on performance does not establish principle concentration; not established is not proof of continuing exploration. Full branch counts and audit qualifications appear in Appendix C.

C Decision-Tree Counts and Text Audit

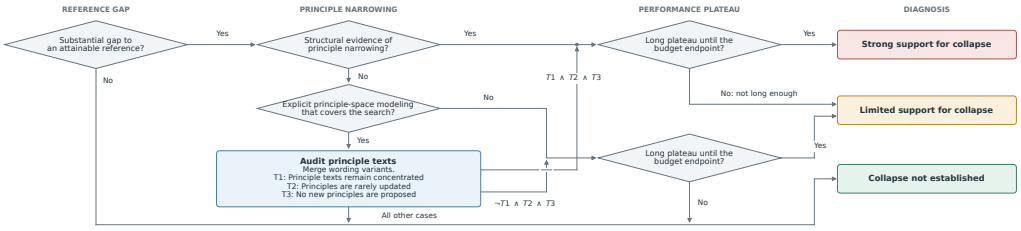


Figure 6: Full diagnostic protocol. Structural evidence, eligibility for textual auditing, and the text audit are all parts of assessing principle narrowing. Text auditing requires explicit principle-space modeling covering the actual search directions. In the final 1,000 generated attempts, T_1 requires dominant strategy share $\geq 50\%$, T_2 allows at most five minor revisions without a major change, and T_3 requires no new proposed direction.

Figure 1 is a main-text overview; Figure 6 specifies the full branching rules used for these counts. In particular, a long plateau without established narrowing provides limited support only when the text audit is inapplicable, or when $\neg T_1 \wedge T_2 \wedge T_3$ holds. An eligible audit that fails other conditions or remains unresolved does not automatically receive limited support. These tables apply the full protocol to saved molecular and network runs. Principle interventions are excluded. Counts are

individual runs, not classifications of the mean trajectory. MCCE is shared between the method comparison and diversity-injection analysis and must not be counted twice as independent evidence. The three outcomes are strong support (S), limited support (L), and collapse not established (N), always reported in the order S/L/N. The full molecular main comparison includes five ExLLM backbones, original PiEvo, MCCE and ReEvo: 23/13/4 over 40 runs. The two networks each include four ExLLM backbones and original PiEvo: residence hall 13/6/6, Dolphin 0/0/25. A large gap without structural concentration enters textual auditing only for an explicit principle-space model that covers the actual search directions. Ordinary experience summaries are ineligible; with a long plateau they provide limited performance evidence, not evidence of semantic concentration.

Table 2: Molecules: current non-intervention diagnostic counts.

Method	Condition	S	L	N
PiEvo ($\sigma = 0.9$)	Sigma 0.9	1	1	3
PiEvo	Original	2	3	0
ExLLM(Sonnet 5)	Explore (7:2:1)	5	0	0
ExLLM(Sonnet 5)	Original	3	2	0
ExLLM(Sonnet 5)	Web search	3	2	0
ExLLM(Sonnet 5 Thinking)	Original	5	0	0
ExLLM(GPT-4o)	Original	2	3	0
ExLLM(DeepSeek)	Explore (7:2:1)	4	0	1
ExLLM(DeepSeek)	Original	5	0	0
ExLLM(DeepSeek)	Temperature 1.3	5	0	0
ExLLM(DeepSeek)	Web search	3	0	2
ExLLM(Qwen)	Explore (7:2:1)	1	2	2
ExLLM(Qwen)	Original	4	1	0
ExLLM(Qwen)	Temperature 1.3	0	3	2
ExLLM(Qwen)	Web search	1	2	2
ExLLM(GPT-5.4)	Explore (7:2:1)	4	1	0
ExLLM(GPT-5.4)	Original	4	0	1
ExLLM(GPT-5.4)	Temperature 1.3	3	0	2
ExLLM(GPT-5.4)	Web search	2	0	3
MCCE	MCCE	0	3	2
ReEvo	Original	3	1	1
ExLLM(GPT-4o + Gemini 2.5 Flash)	Two LLMs	4	0	1

Table 3: Non-intervention diagnostic counts on both networks. S/L/N denote strong support, limited support, and collapse not established. All rows use the original optimizer setting.

	Dolphin			Residence hall		
Method	S	L	N	S	L	N
PiEvo	0	0	5	4	1	0
ExLLM(Sonnet 5)	0	0	5	3	0	2
ExLLM(DeepSeek)	0	0	5	5	0	0
ExLLM(Qwen)	0	0	5	0	4	1
ExLLM(GPT-5.4)	0	0	5	1	1	3

Why a run is not strong. Among the 40 main molecular runs, ten receive limited support and four are not established. MCCE contributes 0/3/2 and ReEvo 3/1/1 to the combined S/L/N counts. Borderline cases include Sonnet seed 44 ($N_{\text{eff}} = 3.010$) and PiEvo seed 43 (29.5% budget remaining). Such boundaries motivate sensitivity analysis rather than distinct mechanistic claims. On residence hall, five main runs fail the twofold-gap gate, one Qwen run lacks both structural concentration and a long plateau, and six runs receive limited support. All 25 main Dolphin runs fail the large-gap gate; this does not mean every run reaches the exhaustive optimum.

PiEvo audit. Final-1,000-attempt windows are reconstructed from IDS proposal batches and checkpoint generation counters, including retries. Strategy labels merge wording variants using rule-assisted semantic coding with retained original texts and explicit review of revisions; this is not an independent double-annotator validation. Unknown labels are not automatically merged. In original molecular seed 45, five reviewed minor revisions satisfy T_2, but only 4.7% of budget remains at plateau onset, giving limited support. Original molecular counts are 2/3/0; the $\sigma = 0.9$ sensitivity

run gives 1/1/3. For residence-hall PiEvo, seeds 42 and 43 pass the structural route. Seeds 44 and 46 pass the textual route with dominant shares 59.6% and 62.4%, no late revisions or new directions, and long plateaus. Seed 45 has dominant share 40.0%, no revisions or new directions, and a long plateau: limited support, not established concentration. These statistics use actual generated attempts rather than distinct principle strings.

D Detailed Performance and Concentration

The tables report final-checkpoint performance and selected-50 concentration statistics. Each ordinary method row summarizes five seeds. Molecular performance and concentration are separated for readability. ReEvo and MCCE are presently molecular comparisons only; missing network or kernel coverage is not assumed.

Table 4: Molecular baseline performance. HV and diversity use the reconstructed 50-candidate set. Remaining budget is planned budget after the retrospective plateau onset, not measured wasted evaluations.

Method / condition	Top-1	Top-10	HV (50)	U	D	b_{plat}	Remaining (%)
ExLLM(GPT-5.4)	4.047 ± 0.118	4.027 ± 0.110	0.551 ± 0.199	0.326 ± 0.067	0.507 ± 0.040	2762.800 ± 829.500	44.744 ± 16.590
ExLLM(Sonnet 5)	3.975 ± 0.008	3.962 ± 0.012	0.374 ± 0.022	0.122 ± 0.011	0.533 ± 0.050	1172.400 ± 459.202	76.552 ± 9.184
ExLLM(DeepSeek)	3.998 ± 0.049	3.986 ± 0.043	0.419 ± 0.093	0.312 ± 0.072	0.463 ± 0.032	2064.000 ± 802.579	58.720 ± 16.052
ExLLM(Qwen)	4.091 ± 0.109	4.047 ± 0.077	0.659 ± 0.220	0.338 ± 0.094	0.589 ± 0.025	3079.200 ± 1415.091	38.416 ± 28.302
PiEvo	4.100 ± 0.200	4.057 ± 0.189	0.610 ± 0.363	0.559 ± 0.303	0.532 ± 0.170	3214.800 ± 1509.945	35.704 ± 30.199
MCCE	4.288 ± 0.103	4.220 ± 0.059	0.860 ± 0.080	0.656 ± 0.020	0.501 ± 0.030	4255.600 ± 534.429	14.888 ± 10.689
ReEvo	4.162 ± 0.253	4.116 ± 0.201	0.684 ± 0.301	0.533 ± 0.039	0.559 ± 0.036	2146.200 ± 1502.429	57.076 ± 30.049

Table 5: Molecular concentration. Early/late denote first-offspring/final reconstructed sets; uniqueness instead uses first/last 1,000 generated candidates. These are top-50 measurements, not the earlier top-10 window-based verdicts.

Method / condition	U_e	U_l	D_e	D_l	$N_{\mathrm{eff},e}$	$N_{\mathrm{eff},l}$	S_e	S_l	m_e	m_l
ExLLM(GPT-5.4)	0.717 ± 0.058	0.206 ± 0.082	0.849 ± 0.008	0.507 ± 0.040	40.117 ± 4.846	1.908 ± 1.116	42.800 ± 3.564	3.400 ± 1.817	0.068 ± 0.018	0.820 ± 0.215
ExLLM(Sonnet 5)	0.495 ± 0.067	0.067 ± 0.013	0.842 ± 0.006	0.533 ± 0.050	38.842 ± 3.583	2.577 ± 1.312	41.800 ± 2.775	3.400 ± 2.302	0.064 ± 0.009	0.664 ± 0.204
ExLLM(DeepSeek)	0.886 ± 0.032	0.098 ± 0.058	0.862 ± 0.007	0.463 ± 0.032	43.106 ± 6.482	1.241 ± 0.332	46.000 ± 3.391	1.600 ± 0.894	0.084 ± 0.054	0.936 ± 0.092
ExLLM(Qwen)	0.675 ± 0.047	0.390 ± 0.103	0.845 ± 0.008	0.589 ± 0.025	42.094 ± 4.131	1.755 ± 0.730	45.000 ± 1.871	3.200 ± 1.924	0.076 ± 0.050	0.808 ± 0.209
PiEvo	0.824 ± 0.129	0.415 ± 0.363	0.863 ± 0.005	0.532 ± 0.170	48.276 ± 1.257	30.685 ± 11.115	48.800 ± 0.837	35.400 ± 9.813	0.040 ± 0.014	0.132 ± 0.067
MCCE	0.770 ± 0.047	0.646 ± 0.022	0.852 ± 0.004	0.501 ± 0.030	39.926 ± 4.504	1.659 ± 0.547	43.400 ± 2.966	2.400 ± 1.140	0.088 ± 0.036	0.804 ± 0.177
ReEvo	0.631 ± 0.035	0.491 ± 0.028	0.834 ± 0.014	0.559 ± 0.036	36.663 ± 2.549	2.742 ± 1.885	41.200 ± 1.643	6.400 ± 2.074	0.096 ± 0.017	0.736 ± 0.246

Table 6: Network baseline performance and effective node pool. Effective nodes summarize node usage, not semantic principles; lower errors are better.

Method / condition	Top-1 error	Top-10 error	U	$N_{\mathrm{eff},e}$	$N_{\mathrm{eff},l}$	b_{plat}	Remaining (%)
Dolphin							
ExLLM(GPT-5.4)	$4.071 \times 10^{-3} \pm 2.406 \times 10^{-5}$	$4.147 \times 10^{-3} \pm 2.617 \times 10^{-5}$	0.228 ± 0.028	45.780 ± 2.351	13.971 ± 1.419	852.400 ± 276.991	72.503 ± 8.935
ExLLM(Sonnet 5)	$4.115 \times 10^{-3} \pm 7.582 \times 10^{-5}$	$4.189 \times 10^{-3} \pm 8.557 \times 10^{-5}$	0.420 ± 0.035	44.787 ± 4.002	13.990 ± 0.630	929.000 ± 570.675	70.032 ± 18.409
ExLLM(DeepSeek)	$4.216 \times 10^{-3} \pm 3.604 \times 10^{-5}$	$4.320 \times 10^{-3} \pm 3.408 \times 10^{-5}$	0.271 ± 0.061	48.251 ± 2.670	14.214 ± 1.602	657.400 ± 309.967	78.794 ± 9.999
ExLLM(Qwen)	$4.234 \times 10^{-3} \pm 1.073 \times 10^{-4}$	$4.373 \times 10^{-3} \pm 1.501 \times 10^{-4}$	0.482 ± 0.115	50.305 ± 1.465	21.705 ± 7.702	1291.200 ± 334.970	58.348 ± 10.805
PiEvo	$4.098 \times 10^{-3} \pm 7.122 \times 10^{-5}$	$4.176 \times 10^{-3} \pm 1.034 \times 10^{-4}$	0.099 ± 0.052	33.441 ± 12.692	14.108 ± 1.878	962.600 ± 453.699	68.948 ± 14.635
Residence hall							
ExLLM(GPT-5.4)	$7.005 \times 10^{-7} \pm 5.689 \times 10^{-7}$	$1.352 \times 10^{-6} \pm 6.547 \times 10^{-7}$	0.378 ± 0.109	101.948 ± 6.277	26.752 ± 3.587	4440.800 ± 2574.126	59.071 ± 23.725
ExLLM(Sonnet 5)	$1.514 \times 10^{-6} \pm 1.084 \times 10^{-6}$	$4.434 \times 10^{-6} \pm 2.171 \times 10^{-6}$	0.303 ± 0.109	72.039 ± 9.282	18.709 ± 1.172	3462.400 ± 3369.235	68.088 ± 31.053
ExLLM(DeepSeek)	$1.402 \times 10^{-6} \pm 8.028 \times 10^{-7}$	$3.846 \times 10^{-6} \pm 1.887 \times 10^{-6}$	0.255 ± 0.049	118.957 ± 5.068	23.065 ± 5.774	2454.600 ± 1514.687	77.377 ± 13.960
ExLLM(Qwen)	$9.961 \times 10^{-7} \pm 6.875 \times 10^{-7}$	$4.993 \times 10^{-5} \pm 6.598 \times 10^{-5}$	0.731 ± 0.087	128.440 ± 3.310	99.446 ± 26.186	5875.800 ± 3029.916	45.845 ± 27.925
PiEvo	$1.299 \times 10^{-6} \pm 5.886 \times 10^{-7}$	$4.436 \times 10^{-6} \pm 1.781 \times 10^{-6}$	0.315 ± 0.064	75.339 ± 39.760	36.953 ± 19.704	1906.400 ± 986.742	82.429 ± 9.094

E Diversity-Injection Details

Baseline, explore and web search pool four backbones; temperature pools three. This differs from the dispersion of 15 or 20 independent runs. Two LLMs uses the updated five-seed GPT-4o plus Gemini batch, not the earlier two-seed batch. MCCE is retained in both main and diversity-injection comparisons, using the same five runs rather than independent evidence. Full table is in table 8, table 9, table 10. Per-backbone figures are in figure 10 and figure 11.

Table 7: Network node concentration. C_5 is normalized top-five coverage; R_2, R_3 are maximum pair/triple reuse. Higher values indicate concentration; e/l denote early/late.

Method / condition	U_e	U_l	$C_{5,e}$	$C_{5,l}$	$R_{2,e}$	$R_{2,l}$	$R_{3,e}$	$R_{3,l}$
Dolphin								
ExLLM(GPT-5.4)	0.482 ± 0.021	0.169 ± 0.041	0.270 ± 0.021	0.742 ± 0.080	0.088 ± 0.011	0.324 ± 0.134	0.052 ± 0.011	0.128 ± 0.056
ExLLM(Sonnet 5)	0.652 ± 0.132	0.282 ± 0.047	0.294 ± 0.024	0.745 ± 0.041	0.084 ± 0.009	0.372 ± 0.070	0.056 ± 0.009	0.156 ± 0.038
ExLLM(DeepSeek)	0.546 ± 0.079	0.154 ± 0.027	0.244 ± 0.017	0.703 ± 0.114	0.072 ± 0.011	0.360 ± 0.235	0.044 ± 0.009	0.108 ± 0.048
ExLLM(Qwen)	0.658 ± 0.040	0.389 ± 0.172	0.248 ± 0.019	0.617 ± 0.178	0.064 ± 0.009	0.360 ± 0.258	0.036 ± 0.009	0.132 ± 0.097
PiEvo	0.447 ± 0.129	0.039 ± 0.029	0.404 ± 0.126	0.693 ± 0.050	0.216 ± 0.109	0.328 ± 0.076	0.080 ± 0.037	0.124 ± 0.022
Residence hall								
ExLLM(GPT-5.4)	0.715 ± 0.073	0.255 ± 0.129	0.152 ± 0.011	0.443 ± 0.069	0.080 ± 0.020	0.236 ± 0.067	0.068 ± 0.011	0.104 ± 0.009
ExLLM(Sonnet 5)	0.749 ± 0.030	0.171 ± 0.098	0.285 ± 0.033	0.604 ± 0.019	0.148 ± 0.033	0.428 ± 0.088	0.072 ± 0.018	0.196 ± 0.033
ExLLM(DeepSeek)	0.665 ± 0.043	0.149 ± 0.044	0.114 ± 0.013	0.504 ± 0.089	0.072 ± 0.011	0.324 ± 0.071	0.064 ± 0.009	0.160 ± 0.024
ExLLM(Qwen)	0.903 ± 0.065	0.624 ± 0.138	0.110 ± 0.010	0.178 ± 0.096	0.044 ± 0.009	0.080 ± 0.032	0.032 ± 0.011	0.064 ± 0.017
PiEvo	0.599 ± 0.084	0.217 ± 0.088	0.270 ± 0.200	0.390 ± 0.146	0.208 ± 0.210	0.216 ± 0.124	0.140 ± 0.097	0.108 ± 0.046

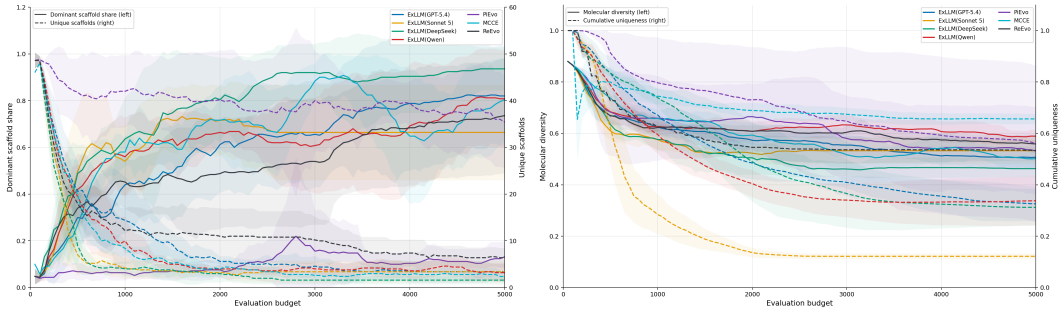


Figure 7: Additional molecular trajectories: dominant scaffold share and unique scaffold count (left), fingerprint diversity and cumulative uniqueness (right). Final values are carried forward to budget 5,000.

F Older-Model and Thinking Controls

GPT-4o's global uniqueness is 0.833, compared with 0.326 for GPT-5.4 and 0.122–0.338 across the four main ExLLM backbones. Yet all five GPT-4o runs end with a single scaffold in their top-10, and their mean top-50 effective count is 1.057. Distinct molecule generation therefore does not imply exploration of distinct core designs.

Sonnet thinking raises global uniqueness from 0.122 to 0.203, but mean Top-1 changes from 3.975 to 3.972 and mean final effective count from 2.577 to 1.665. In this comparison, more generation-level variation does not translate into better optimization. One hypothesis is that differences in post-training make some newer models more conservative in repeated proposing. These experiments do not identify that cause: model scale, pretraining, decoding and provider behavior also differ, and post-training is not independently manipulated. We therefore report an older/newer-model contrast, not a demonstrated adverse effect of additional post-training.

G PiEvo: Text, Candidate and Update Evidence

Our PiEvo is adapted from official code, but evaluates all proposals rather than only the top-ranked one because we have much more budget than experiments in original PiEvo.

Counting texts is not semantic annotation. Effective used-text count is $\exp(-\sum_j q_j \ln q_j)$, where q_j is the fraction of generated attempts using exact normalized principle text j. Early/late windows are the first/last 1,000 generated attempts after initialization. Usage and posterior summaries are attempt-weighted, with partial boundary batches included proportionally. Updates are counted at batch start and anomaly events at batch end; event counts are not fractionally weighted. Different wording may describe the same strategy, and one strategy can produce multiple scaffolds.

In original molecular seed 42, the dominant late instruction emphasizes 8–15-member macrocycles, with peripheral changes intended to tune properties. The post-plateau operational-route effective count is 1.149, with share 0.975 and no first use of a new route after the plateau. In the structural

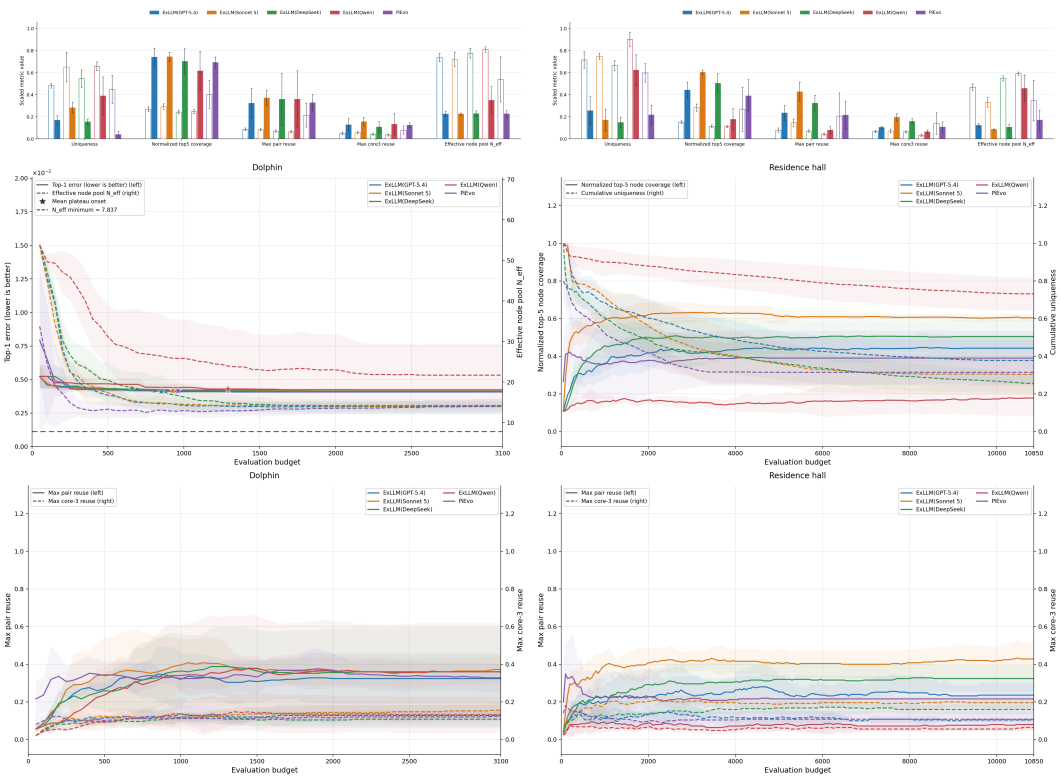


Figure 8: Network early/late bars and supplementary trajectories. Dolphin's concentration is interpreted against its exhaustive optimum, not as failure on its own. Pair/triple reuse measures repeated node combinations; effective-node count measures spread of node usage.

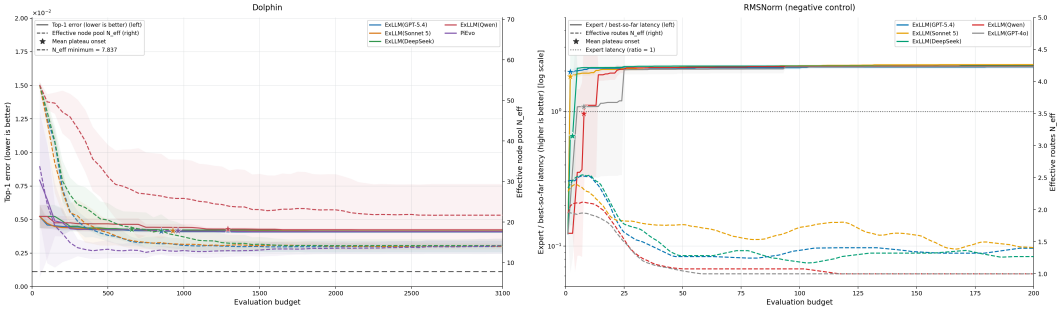


Figure 9: Negative controls in the layout of Figure 2: Dolphin (left; the reference is the exhaustive optimum) and RMSNorm (right; the reference is the expert kernel, ratio $= 1$). Solid/dashed lines use the left/right axes; stars mark mean plateau onset. Both tasks plateau early with concentrated pools, but near or beyond the reference, so the Headroom condition fails and collapse is not established. The RMSNorm curve is the geometric mean across seeds; N_{eff} uses the diagnostic set of the ten fastest kernels, smoothed as in Figure 5; the GPT-4o control is included.

follow-through audit, 661 of 734 newly evaluated late molecules under the explicit macrocycle instruction contain an 8–15-member ring. Its high scaffold count therefore does not refute a narrower instruction-level search.

Performance qualifies the interpretation. That seed remains 0.800 below the attainable molecular reference, with 74.1% of planned budget left at plateau onset. Other PiEvo molecular seeds show different timing: some plateau only near the budget limit, so one case does not establish universal long-term collapse. On residence hall, all five best errors are 3.42–8.84 times the reference, with

Table 8: Diversity injection, average backbones within each seed, then report mean of five seed averages. Temperature uses three backbones; baseline/explore/web use four. Two LLMs and MCCE each use five independent runs.

Method / condition	Top-1	Top-10	HV (50)	U	D	b_{plat}	Remaining (%)
ExLLM(Baseline)	4.028 ± 0.011	4.005 ± 0.007	0.501 ± 0.039	0.274 ± 0.044	0.523 ± 0.021	2269.600 ± 439.577	54.608 ± 8.792
ExLLM(Explore)	4.048 ± 0.079	4.021 ± 0.070	0.503 ± 0.115	0.263 ± 0.025	0.520 ± 0.007	2262.700 ± 554.565	54.746 ± 11.091
ExLLM(Web search)	4.108 ± 0.121	4.067 ± 0.091	0.627 ± 0.174	0.301 ± 0.005	0.519 ± 0.031	2591.850 ± 594.545	48.163 ± 11.891
ExLLM(Temp 1.3)	4.086 ± 0.116	4.046 ± 0.087	0.577 ± 0.148	0.400 ± 0.031	0.532 ± 0.029	2425.200 ± 1042.968	51.496 ± 20.859
Two LLMs	4.165 ± 0.304	4.141 ± 0.274	0.667 ± 0.356	0.636 ± 0.106	0.485 ± 0.075	2032.200 ± 949.502	59.356 ± 18.990
MCCE	4.288 ± 0.103	4.220 ± 0.059	0.860 ± 0.080	0.656 ± 0.020	0.501 ± 0.030	4255.600 ± 534.429	14.888 ± 10.689

Table 9: Diversity-injection concentration under the same aggregation as Table 8.

Method / condition	U_e	U_l	D_e	D_l	$N_{\mathrm{eff},e}$	$N_{\mathrm{eff},l}$	S_e	S_l	m_e	m_l
ExLLM(Baseline)	0.693 ± 0.018	0.190 ± 0.033	0.850 ± 0.006	0.523 ± 0.021	41.040 ± 4.320	1.871 ± 0.230	43.900 ± 2.626	2.900 ± 0.720	0.073 ± 0.026	0.807 ± 0.038
ExLLM(Explore)	0.646 ± 0.030	0.183 ± 0.007	0.849 ± 0.008	0.520 ± 0.007	40.740 ± 1.241	2.366 ± 1.209	43.400 ± 0.894	3.550 ± 1.866	0.067 ± 0.009	0.766 ± 0.096
ExLLM(Web search)	0.644 ± 0.032	0.248 ± 0.036	0.839 ± 0.008	0.519 ± 0.031	42.475 ± 2.613	4.727 ± 3.496	44.800 ± 1.681	7.200 ± 3.475	0.064 ± 0.016	0.730 ± 0.116
ExLLM(Temp 1.3)	0.735 ± 0.027	0.326 ± 0.055	0.851 ± 0.006	0.532 ± 0.029	43.091 ± 1.725	1.774 ± 0.442	45.200 ± 1.121	2.733 ± 0.596	0.061 ± 0.014	0.827 ± 0.096
Two LLMs	0.903 ± 0.019	0.556 ± 0.136	0.848 ± 0.003	0.485 ± 0.075	39.731 ± 1.984	1.298 ± 0.461	42.400 ± 1.517	1.600 ± 0.894	0.064 ± 0.009	0.908 ± 0.148
MCCE	0.770 ± 0.047	0.646 ± 0.022	0.852 ± 0.004	0.501 ± 0.030	39.926 ± 4.504	1.659 ± 0.547	43.400 ± 2.966	2.400 ± 1.140	0.088 ± 0.036	0.804 ± 0.177

75.5–97.8% of planned budget remaining at onset. Under the current decision tree, the S/L/N counts are $4/1/0$. The last-1,000-attempt audit is summarized in Appendix C; the earlier post-plateau summaries use different windows. Dolphin's five endpoints are within 4.4% of the exhaustive optimum, including one exact optimum.

G.1 Noise sensitivity

Increasing σ from 0.01 to 0.9 is an additional molecular comparison, not a main baseline. It can soften posterior concentration, with mean early/late exact-text effective counts of $8.213/9.053$, compared with $2.675/2.229$ at the original setting. The higher-noise setting averages only 0.4 late principle updates versus 1.0 originally (Table 12); increasing observation noise does not supply a stagnation detector.

H Principle-Intervention Details

These tables report performance rather than exact reference hits. Molecular L2 is supplied either from the start or after 40% of budget; networks use only the latter. These are targeted interventions, not guarantees that the proposed direction is fully implemented.

I Kernel Results and Sensitivity

Diagnostic set and concentration. At each evaluation, the kernel diagnostic set contains the ten fastest valid candidates found so far. Applying the route descriptor ϕ gives frequencies $\hat{\pi}_j^{\mathrm{top10}}(p)$ as in Eq. 3 and

$$N_{\mathrm{eff},j}^{\mathrm{top10}} = \exp\left[-\sum_p \hat{\pi}_j^{\mathrm{top10}}(p) \ln \hat{\pi}_j^{\mathrm{top10}}(p)\right].$$

Its range is $[1, 10]$ for a full set; the dominant share is $\max_p \hat{\pi}_j^{\mathrm{top10}}(p)$. Concentration means final $N_{\mathrm{eff}} \leq 2$. We assess a run only after at least 20 valid kernels.

Plateau and joint criterion. Let $G(t)$ be the best-so-far latency divided by expert latency, T_{obs} the final observed evaluation attempt, and B the planned evaluation budget. For a run ending above the reference, define

$$\tau_{\mathrm{ker}} = \min\left\{t : G(t) - G(T_{\mathrm{obs}}) \leq \epsilon_{\mathrm{ker}}[G(t) - 1]\right\}, \qquad \epsilon_{\mathrm{ker}} = 0.10.$$

The fraction of the planned budget remaining at plateau onset is

$$R_{\mathrm{budget}} = \frac{B - \tau_{\mathrm{ker}}}{B}.$$

Table 10: Per-backbone molecular diversity-injection comparisons. Sonnet has no temperature-1.3 arm.

Method / condition	Top-1	Top-10	HV (50)	U	D	b_{plat}	Remaining (%)
ExLLM(GPT-5.4)							
ExLLM(GPT-5.4) / Explore (7:2:1)	3.993 ± 0.117	3.978 ± 0.106	0.429 ± 0.216	0.367 ± 0.038	0.532 ± 0.020	2094.600 ± 1632.767	58.108 ± 32.655
ExLLM(GPT-5.4)	4.047 ± 0.118	4.027 ± 0.110	0.551 ± 0.199	0.326 ± 0.067	0.507 ± 0.040	2762.800 ± 829.500	44.744 ± 16.590
ExLLM(GPT-5.4) / Temperature 1.3	4.204 ± 0.366	4.140 ± 0.295	0.656 ± 0.476	0.342 ± 0.048	0.482 ± 0.028	2880.600 ± 1771.310	42.388 ± 35.426
ExLLM(GPT-5.4) / Web search	4.062 ± 0.271	4.018 ± 0.210	0.559 ± 0.356	0.310 ± 0.061	0.523 ± 0.058	2781.000 ± 1498.193	44.380 ± 29.964
ExLLM(Sonnet 5)							
ExLLM(Sonnet 5) / Explore (7:2:1)	4.028 ± 0.062	4.009 ± 0.067	0.468 ± 0.109	0.176 ± 0.020	0.491 ± 0.040	1485.800 ± 345.575	70.284 ± 6.912
ExLLM(Sonnet 5)	3.975 ± 0.008	3.962 ± 0.012	0.374 ± 0.022	0.122 ± 0.011	0.533 ± 0.050	1172.400 ± 459.202	76.552 ± 9.184
ExLLM(Sonnet 5) / Web search	4.048 ± 0.101	4.027 ± 0.100	0.513 ± 0.182	0.139 ± 0.028	0.511 ± 0.056	1722.400 ± 690.889	65.552 ± 13.818
ExLLM(DeepSeek)							
ExLLM(DeepSeek) / Explore (7:2:1)	4.116 ± 0.325	4.082 ± 0.270	0.531 ± 0.435	0.177 ± 0.013	0.439 ± 0.031	2168.600 ± 1623.028	56.628 ± 32.461
ExLLM(DeepSeek)	3.998 ± 0.049	3.986 ± 0.043	0.419 ± 0.093	0.312 ± 0.072	0.463 ± 0.032	2064.000 ± 802.579	58.720 ± 16.052
ExLLM(DeepSeek) / Temperature 1.3	4.004 ± 0.075	3.995 ± 0.071	0.412 ± 0.127	0.327 ± 0.078	0.480 ± 0.070	1062.600 ± 442.082	78.748 ± 8.842
ExLLM(DeepSeek) / Web search	4.192 ± 0.329	4.138 ± 0.281	0.660 ± 0.466	0.215 ± 0.053	0.452 ± 0.035	2171.600 ± 1648.882	56.568 ± 32.978
ExLLM(Qwen)							
ExLLM(Qwen) / Explore (7:2:1)	4.057 ± 0.134	4.017 ± 0.102	0.583 ± 0.269	0.333 ± 0.049	0.620 ± 0.052	3301.800 ± 1852.730	33.964 ± 37.055
ExLLM(Qwen)	4.091 ± 0.109	4.047 ± 0.077	0.659 ± 0.220	0.338 ± 0.094	0.589 ± 0.025	3079.200 ± 1415.091	38.416 ± 28.302
ExLLM(Qwen) / Temperature 1.3	4.050 ± 0.075	4.002 ± 0.066	0.664 ± 0.087	0.532 ± 0.084	0.634 ± 0.040	3332.400 ± 1777.876	33.352 ± 35.558
ExLLM(Qwen) / Web search	4.132 ± 0.064	4.085 ± 0.052	0.777 ± 0.082	0.540 ± 0.060	0.592 ± 0.054	3692.400 ± 755.305	26.152 ± 15.106

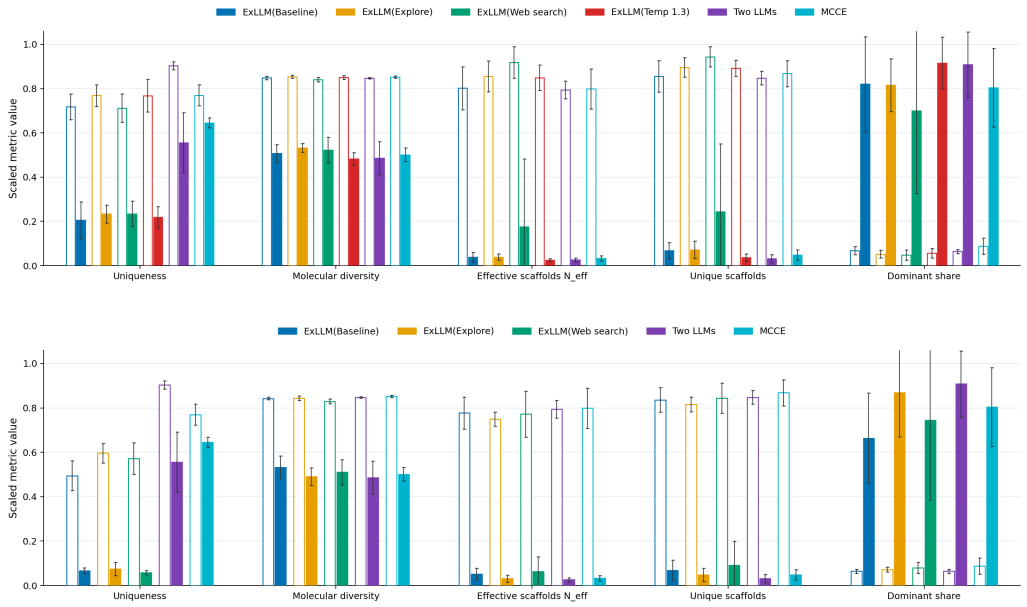


Figure 10: Per-backbone diversity injection: GPT-5.4 and Sonnet. Sonnet has no temperature-1.3 arm. Two LLMs and MCCE are fixed independent comparators, not variants of the named backbone.

The joint kernel criterion requires final $N_{\text{eff}} \leq 2$, $R_{\text{budget}} \geq 0.30$, and $G(T_{\text{obs}}) \geq 2$. All three must hold. This is a descriptive criterion, not a proof of a mathematical local optimum. For runs reaching $G \leq 1$, the plotting code marks the first reference-reaching point if no earlier point satisfies its plateau test; this control marker is not evidence that subsequent gains cease.

Threshold sensitivity. The supplied sensitivity grid includes GPT-4o: 82 of 100 non-control baseline runs are assessable, versus 80/80 for the four main backbones. At the default settings it gives 49/82 joint cases (48/80 without GPT-4o). Holding the other settings fixed, changing the effective-count ceiling from 1.5 to 2 to 3 gives 47, 49 and 50 cases; changing the remaining-fraction threshold to 0.2/0.3/0.4/0.5 gives 55/49/43/36. Plateau tolerances 0.05/0.10/0.20 give 43/49/53 cases. The recent-20-valid-candidate diagnostic gives 46 cases at the default thresholds. Thus the exact count is threshold-dependent, but concentration with substantial performance headroom is not confined to one setting.

Intervention coverage and comparison. Table 20 uses only the four main backbones. Each start condition has 48 planned runs (four tasks, four backbones, three seeds); L2 and L3 each lack a usable

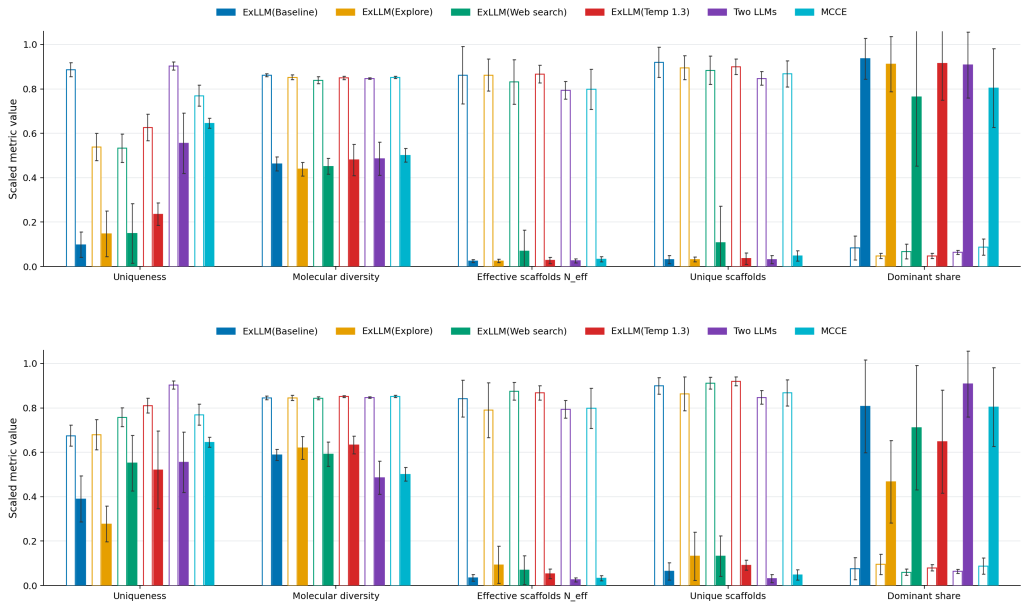


Figure 11: Per-backbone diversity injection: DeepSeek and Qwen. These panels expose heterogeneity that a pooled mean can conceal; they do not establish a uniform effect size across backbones.

Table 11: Older-model and thinking controls. Model training differences are not experimentally isolated.

Method / condition	Top-1	Top-10	U	U_e	U_l	$N_{\text{eff},l}$	Top-10 scaffolds
ExLLM(GPT-5.4)	4.047 ± 0.118	4.027 ± 0.110	0.326 ± 0.067	0.717 ± 0.058	0.206 ± 0.082	1.908 ± 1.116	1.800 ± 0.447
ExLLM(Sonnet 5)	3.975 ± 0.008	3.962 ± 0.012	0.122 ± 0.011	0.495 ± 0.067	0.067 ± 0.013	2.577 ± 1.312	2.400 ± 1.342
ExLLM(DeepSeek)	3.998 ± 0.049	3.986 ± 0.043	0.312 ± 0.072	0.886 ± 0.032	0.098 ± 0.058	1.241 ± 0.332	1.200 ± 0.447
ExLLM(Qwen)	4.091 ± 0.109	4.047 ± 0.077	0.338 ± 0.094	0.675 ± 0.047	0.390 ± 0.103	1.755 ± 0.730	1.800 ± 1.304
ExLLM(GPT-4o)	4.190 ± 0.007	4.170 ± 0.006	0.833 ± 0.061	0.988 ± 0.007	0.682 ± 0.075	1.057 ± 0.083	1.000 ± 0.000
ExLLM(Sonnet thinking)	3.972 ± 0.004	3.963 ± 0.011	0.203 ± 0.038	0.575 ± 0.062	0.094 ± 0.018	1.665 ± 0.770	1.600 ± 0.894

final latency in two runs. Each onset condition has 41 matched runs, selected by the experiment's onset detector, not retrospectively by the final joint criterion above. Gain is baseline latency divided by hinted latency. Better means gain > 1.05, worse means gain < 0.95, and the remaining cases fall within that band. These cutoffs are on gain, not exactly a symmetric 5% change in latency.

Onset branch point. Onset arms branch from the baseline run once its search has narrowed. The branch point was fixed online, before any hinted run, as the first time the run's recent proposals stayed concentrated on one route for two consecutive windows of 20 valid kernels; it is found in 41 of the 48 baseline runs of the four main backbones with seeds 42–44, and the other seven have no onset arm. Checked against the conditions of Section 3.2, every branch point already satisfies Narrow: the diagnostic set of the ten fastest kernels has $N_{\text{eff}} \leq 2$ at that moment (median 1.0). In 36 of the 41 runs Headroom also holds, with best latency at least twice the expert's; the other five are Sonnet and DeepSeek runs already within twice the expert latency. About half of the branch points fall after the run's plateau onset in Figure 5 and half shortly before it. By the end of the budget, 28 of the 41 baseline runs meet the joint criterion, 7 are concentrated but still improving, and 6 converge near the expert.

J First-Attempt Quality and Subsequent Development

Events are the first appearances of a route after the run's first route in the 80 ExLLM baseline runs of the four main backbones (four operators, five seeds): 186 events in 70 runs; the other ten runs used a single route throughout. r and N_{after} follow Eq. (20); the control keeps the events with at least 50 attempts left when the route appeared. The expert's route accounts for 18 events, 16 from

Table 12: PiEvo on molecules. Early/late are the first/last 1,000 generated attempts after initialization; updates include warm-up. Text usage and posterior summaries are attempt-weighted; events are counted at their recorded batch boundaries.

Method / condition	Updates early	Updates late	Anomalies all	Anomalies early	Anomalies late	$T_{\mathrm{eff},e}$	$T_{\mathrm{eff},l}$
PiEvo	13.600 ± 2.966	1.000 ± 2.236	19.600 ± 13.903	3.600 ± 2.966	1.000 ± 2.236	2.675 ± 0.951	2.229 ± 1.191
PiEvo / Sigma 0.9	11.000 ± 0.707	0.400 ± 0.894	3.800 ± 2.168	1.000 ± 0.707	0.400 ± 0.894	8.213 ± 0.843	9.053 ± 1.411

Table 13: PiEvo on residence hall. Early/late are the first/last 1,000 generated attempts after initialization; updates include warm-up. Text usage and posterior summaries are attempt-weighted; events are counted at their recorded batch boundaries.

Method / problem	Updates early	Updates late	Anomalies all	Anomalies early	Anomalies late	$T_{\mathrm{eff},e}$	$T_{\mathrm{eff},l}$
PiEvo /dolphin	11.600 ± 1.342	0.000 ± 0.000	4.200 ± 2.950	2.600 ± 1.817	0.000 ± 0.000	5.623 ± 3.048	2.826 ± 1.487
PiEvo /residence hall	11.800 ± 1.304	0.000 ± 0.000	6.400 ± 2.702	2.200 ± 1.095	0.000 ± 0.000	7.429 ± 1.589	6.382 ± 2.348

Sonnet and 2 from DeepSeek; 16 start slower than the incumbent and 11 later produce the run's fastest kernel (Table 22).

K SAVED EXPERIENCE SUMMARIES

The following excerpts illustrate restrictive rules in the final Residence Hall experience; they are examples rather than an exhaustive coding of all runs. Claude records *Trusted core pool (zero error):* $\{$*205, 184, 113, 25, 60, 169, 104, 30, 97*$\}$ and a list of banned nodes. The phrase *zero error* is the model's wording, not an evaluator-verified claim. DeepSeek states that candidates must be five distinct integers from *whitelist* $\{$*42, 87, 118, 139, 143, 168, 196, 211*$\}$. GPT-5.4 recommends *keep 143 fixed* and *conservative one-node, role-preserving edits around stable cores*. Qwen records *strict arithmetic progressions with uniform spacing*, a rule with no explicit network-role justification.

Hinted runs need not faithfully restate the supplied principle to benefit from it. The saved-summary analysis reports no clear reference-principle alignment in Qwen's five Residence Hall summaries, and memorized narrow rules in four, despite lower mean error. GPT-5.4 also retains restrictive core-editing language. These observations distinguish the content of the stored summary from the actual proposals, scores, and potential influence of a separately prepended hint.

K.1 PERFORMANCE AND BUDGET

Molecular fitness: sum of normalized objectives.
Let $a_d(x)$ be raw objective d, with fixed bounds L_d, U_d, and $\widetilde{a}_d(x) = (a_d(x) - L_d)/(U_d - L_d)$. Orient each objective toward maximization:

$$v_d(x) = \begin{cases} \widetilde{a}_d(x), & d \text{ is maximized}, \\ 1 - \widetilde{a}_d(x), & d \text{ is minimized}, \end{cases} \qquad f(x) = \sum_{d=1}^{M} v_d(x), \qquad z_d(x) = 1 - v_d(x). \tag{7}$$

The evaluator stores $\mathbf{z}(x)$ as a minimization vector. For the five molecular objectives,

$$\begin{aligned} f(x) = 1 - \frac{\mathrm{SA}(x) - 1}{9} + (1 - \mathrm{DRD2}(x)) + \mathrm{QED}(x) \\ + (1 - \mathrm{GSK3}\beta(x)) + \mathrm{JNK3}(x). \end{aligned} \tag{8}$$

SA has bounds $[1, 10]$ and the other objectives $[0, 1]$, so $f \in [0, 5]$, with larger better. Normalization follows previous work (Ran et al., 2025b).

Top-1 and top-10 fitness.
Let $\mathcal{X}_t$ be the archive of distinct valid candidates at budget t, ordered as $x_{(1)}, x_{(2)}, \ldots$ by decreasing f, and $k_t = \min(K, |\mathcal{X}_t|)$. Then

$$F_K(t) = \frac{1}{k_t} \sum_{i=1}^{k_t} f(x_{(i)}), \qquad F_1(t) = \max_{x \in \mathcal{X}_t} f(x). \tag{9}$$

Top-10 F means F_{10}, the average over ten distinct candidates, not ten generations. At the final budget, $F_1(T) = F_T$ of Eq. (2). Molecular $F_K \in [0, 5]$, with larger better.

Table 14: PiEvo molecular noise sensitivity. Belief entropies are attempt-weighted over the first/last 1,000 generated attempts. They measure posterior uncertainty, not scaffold coverage.

Method / condition	Top-1	Top-10	H_e^{belief}	H_l^{belief}	Scaffold $N_{\text{eff},l}$	b_{plat}
PiEvo	4.100 ± 0.200	4.057 ± 0.189	0.010 ± 0.007	0.000 ± 0.000	30.685 ± 11.115	3214.800 ± 1509.945
PiEvo / Sigma 0.9	4.068 ± 0.156	4.040 ± 0.152	0.115 ± 0.026	0.008 ± 0.004	18.972 ± 9.671	3595.200 ± 970.716

Table 15: Molecular L2 timing comparison. Performance improvements, not exact reference identity, define intervention benefit.

Method / condition	Top-1	Top-10	U	$N_{\text{eff},l}$	m_l
ExLLM(GPT-5.4) / Hint from 40%	4.021 ± 0.090	3.992 ± 0.095	0.396 ± 0.038	3.717 ± 1.083	0.544 ± 0.190
ExLLM(GPT-5.4) / Hint from start	4.264 ± 0.098	4.201 ± 0.088	0.315 ± 0.040	1.541 ± 0.511	0.856 ± 0.172
ExLLM(GPT-5.4)	4.047 ± 0.118	4.027 ± 0.110	0.326 ± 0.067	1.908 ± 1.116	0.820 ± 0.215
ExLLM(Sonnet 5) / Hint from 40%	4.485 ± 0.298	4.394 ± 0.254	0.151 ± 0.016	2.440 ± 1.292	0.636 ± 0.234
ExLLM(Sonnet 5) / Hint from start	4.617 ± 0.183	4.513 ± 0.163	0.170 ± 0.033	2.162 ± 0.314	0.572 ± 0.064
ExLLM(Sonnet 5)	3.975 ± 0.008	3.962 ± 0.012	0.122 ± 0.011	2.577 ± 1.312	0.664 ± 0.204
ExLLM(DeepSeek) / Hint from 40%	4.699 ± 0.000	4.597 ± 0.015	0.424 ± 0.085	2.392 ± 0.245	0.564 ± 0.079
ExLLM(DeepSeek) / Hint from start	4.699 ± 0.000	4.584 ± 0.019	0.350 ± 0.025	2.154 ± 0.436	0.656 ± 0.155
ExLLM(DeepSeek)	3.998 ± 0.049	3.986 ± 0.043	0.312 ± 0.072	1.241 ± 0.332	0.936 ± 0.092
ExLLM(Qwen) / Hint from 40%	3.994 ± 0.098	3.963 ± 0.079	0.470 ± 0.061	2.859 ± 1.080	0.644 ± 0.177
ExLLM(Qwen) / Hint from start	4.020 ± 0.051	3.960 ± 0.055	0.550 ± 0.044	3.990 ± 2.027	0.592 ± 0.217
ExLLM(Qwen)	4.091 ± 0.109	4.047 ± 0.077	0.338 ± 0.094	1.755 ± 0.730	0.808 ± 0.209

Sentinel approximation error and top-10 error.
For network state y_{cv} at condition c and node v, write $\bar{y}_c = N^{-1} \sum_v y_{cv}$ and $\widehat{y}_c(x) = r^{-1} \sum_{v \in x} y_{cv}$ for a subset of size r. The unweighted evaluator returns

$$e(x) = \frac{\sum_c (\widehat{y}_c(x) - \bar{y}_c)^2}{\sum_c \bar{y}_c}, \qquad \text{Err}_K(t) = \frac{1}{k_t} \sum_{i=1}^{k_t} e(x_{[i]}), \tag{10}$$

where $x_{[i]}$ are ordered by increasing error. The denominator is positive on both datasets; error is non-negative with no universal upper bound, and smaller is better. This is not variance-normalized MSE. The theoretical utility $f(x) = -\ln e(x)$ in Table 1 gives the same ranking as the implementation's score $-e(x)$; tables average raw errors, not log errors.

Kernel latency and reference gap.
For workload latencies $\ell_w(x) > 0$, $w = 1, \ldots, W$, geometric-mean latency and the reference ratio are

$$\ell(x) = \exp\left(\frac{1}{W} \sum_{w=1}^{W} \ln \ell_w(x) \right), \qquad G(t) = \frac{\min_{x \in \mathcal{X}_t} \ell(x)}{\ell(x_{\text{ref}})}. \tag{11}$$

Smaller is better: $G = 1$ matches the expert and $G < 1$ beats it. With $f = -\ln \ell$, $\Delta_T = \ln G(T)$. Replacing latency by error gives the analogous network error ratio and log gap.

Hypervolume (HV).
Using the normalized minimization objectives $\mathbf{z}(x)$ from Eq. equation 7, let $\mathcal{Z}_{50}^{\text{ND}}(b)$ be the non-dominated objective vectors in $\mathcal{A}_{50}(b)$. With reference vector $\mathbf{z}_{\text{ref}} = (1.1, \ldots, 1.1)$, we compute

$$\text{HV}_{50}(b) = \lambda_M \left(\bigcup_{\mathbf{z} \in \mathcal{Z}_{50}^{\text{ND}}(b)} \prod_{d=1}^{M} [z_d, 1.1] \right), \tag{12}$$

where λ_M denotes M-dimensional volume; a vector outside the reference box contributes no volume. Higher HV indicates better joint objective coverage. For normalized objectives in $[0, 1]$, $0 \le \text{HV}_{50} \le 1.1^M$; with five objectives the upper bound is approximately 1.611, not 1. The reported values are not rescaled to $[0, 1]$.

Reference-region hits.
For run s, $h_s = \mathbf{1}[F_1^{(s)}(T) > 4.6]$. The reported count is $\sum_{s=1}^{n} h_s$ out of n, e.g., $3/5$. It measures whether any evaluated molecule enters the high-score region, not whether it exactly reproduces the reference structure. Larger is better.

Table 16: Dolphin delayed L2 intervention. This is a continuation comparison, not a test of reproducing the reference node set.

Method / condition	Top-1 error	Top-10 error	U	$N_{\mathrm{eff},l}$
ExLLM(GPT-5.4) / Hint from 40%	$4.077 \times 10^{-3} \pm 3.422 \times 10^{-5}$	$4.144 \times 10^{-3} \pm 2.249 \times 10^{-5}$	0.213 ± 0.031	13.947 ± 1.283
ExLLM(GPT-5.4)	$4.071 \times 10^{-3} \pm 2.406 \times 10^{-5}$	$4.147 \times 10^{-3} \pm 2.617 \times 10^{-5}$	0.228 ± 0.028	13.971 ± 1.419
ExLLM(Sonnet 5) / Hint from 40%	$4.115 \times 10^{-3} \pm 7.580 \times 10^{-5}$	$4.196 \times 10^{-3} \pm 7.556 \times 10^{-5}$	0.391 ± 0.061	14.292 ± 0.928
ExLLM(Sonnet 5)	$4.115 \times 10^{-3} \pm 7.582 \times 10^{-5}$	$4.189 \times 10^{-3} \pm 8.557 \times 10^{-5}$	0.420 ± 0.035	13.990 ± 0.630
ExLLM(DeepSeek) / Hint from 40%	$4.089 \times 10^{-3} \pm 5.906 \times 10^{-5}$	$4.172 \times 10^{-3} \pm 4.753 \times 10^{-5}$	0.336 ± 0.079	15.383 ± 1.991
ExLLM(DeepSeek)	$4.216 \times 10^{-3} \pm 3.604 \times 10^{-5}$	$4.320 \times 10^{-3} \pm 3.408 \times 10^{-5}$	0.271 ± 0.061	14.214 ± 1.602
ExLLM(Qwen) / Hint from 40%	$4.174 \times 10^{-3} \pm 8.443 \times 10^{-5}$	$4.255 \times 10^{-3} \pm 7.479 \times 10^{-5}$	0.554 ± 0.098	15.200 ± 2.006
ExLLM(Qwen)	$4.234 \times 10^{-3} \pm 1.073 \times 10^{-4}$	$4.373 \times 10^{-3} \pm 1.501 \times 10^{-4}$	0.482 ± 0.115	21.705 ± 7.702

Table 17: Residence hall delayed L2 intervention. This is a continuation comparison, not a test of reproducing the reference node set.

Method / condition	Top-1 error	Top-10 error	U	$N_{\mathrm{eff},l}$
ExLLM(GPT-5.4) / Hint from 40%	$5.112 \times 10^{-7} \pm 3.635 \times 10^{-7}$	$1.152 \times 10^{-6} \pm 2.242 \times 10^{-7}$	0.395 ± 0.065	27.815 ± 2.855
ExLLM(GPT-5.4)	$7.005 \times 10^{-7} \pm 5.689 \times 10^{-7}$	$1.352 \times 10^{-6} \pm 6.547 \times 10^{-7}$	0.378 ± 0.109	26.752 ± 3.587
ExLLM(Sonnet 5) / Hint from 40%	$1.579 \times 10^{-6} \pm 1.759 \times 10^{-6}$	$3.600 \times 10^{-6} \pm 2.861 \times 10^{-6}$	0.341 ± 0.111	20.962 ± 1.078
ExLLM(Sonnet 5)	$1.514 \times 10^{-6} \pm 1.084 \times 10^{-6}$	$4.434 \times 10^{-6} \pm 2.171 \times 10^{-6}$	0.303 ± 0.109	18.709 ± 1.172
ExLLM(DeepSeek) / Hint from 40%	$9.447 \times 10^{-7} \pm 5.885 \times 10^{-7}$	$2.660 \times 10^{-6} \pm 1.405 \times 10^{-6}$	0.360 ± 0.087	25.469 ± 5.719
ExLLM(DeepSeek)	$1.402 \times 10^{-6} \pm 8.028 \times 10^{-7}$	$3.846 \times 10^{-6} \pm 1.887 \times 10^{-6}$	0.255 ± 0.049	23.065 ± 5.774
ExLLM(Qwen) / Hint from 40%	$4.716 \times 10^{-7} \pm 5.194 \times 10^{-8}$	$2.481 \times 10^{-5} \pm 2.963 \times 10^{-5}$	0.846 ± 0.048	91.715 ± 18.594
ExLLM(Qwen)	$9.961 \times 10^{-7} \pm 6.875 \times 10^{-7}$	$4.993 \times 10^{-5} \pm 6.598 \times 10^{-5}$	0.731 ± 0.087	99.446 ± 26.186

L METRIC DEFINITIONS

We retain the notation of Section 3: x is a candidate, $f(x)$ is utility (larger is better), $\phi(x) \in \mathcal{P}$ is its post-hoc structural label, H_t is optimization history, and $\mathcal{A}_{50}(b)$ is the reconstructed diagnostic set at evaluation budget b. Logarithms are natural unless specified otherwise. Scalar entropy is used to define effective count.

L.1 SELECTED-POPULATION DIAGNOSTICS

The following metrics use the same selected-50 reconstruction as the molecular and network budget plots.

Reconstructed sets and timing.
Let $\mathcal{X}_b$ be the distinct valid candidates observed by evaluation budget b, and $\mathcal{A}_{50}(b)$ the diagnostic selection of 50 candidates. Duplicate molecules are canonicalized; network candidates are deduplicated by unordered node set. The top-10 comparison selects the ten best scalar fitness values separately.

Early molecular snapshots follow the first complete offspring generation. PiEvo networks have no initial population; their early snapshot is the first complete round containing at least 50 distinct candidates. Late snapshots use the current final checkpoint.

Budget curves sample every 50 evaluations, from 0 to 5,000 for molecules, 3,100 for Dolphin and 10,850 for residence hall. Stars mark mean per-seed plateau onset on the mean Top-1 curve. Residence-hall error is displayed on a log axis, but means and SD are calculated in raw error units.

Scaffold effective count, richness and dominant share.
For canonical Bemis–Murcko scaffold frequency $p_s(b) = |\{x \in \mathcal{A}_{50}(b) : \phi(x) = s\}|/50$,

$$N_{\mathrm{eff}}^{\mathrm{scaf}}(b) = \exp\left[-\sum_s p_s(b) \ln p_s(b)\right], \qquad S_{50}(b) = |\{s : p_s(b) > 0\}|, \qquad m_{50}(b) = \max_s p_s(b). \tag{13}$$

The ranges are $1 \le N_{\mathrm{eff}}^{\mathrm{scaf}} \le S_{50} \le 50$ and $1/50 \le m_{50} \le 1$. Smaller effective count and richness, or larger share, indicate concentration, not necessarily failure. Similar but nonidentical scaffolds can still implement the same strategy. Molecular diversity uses Eq. 31 on the same $\mathcal{A}_{50}(b)$. Bars divide effective and unique counts by 50; curves retain raw counts.

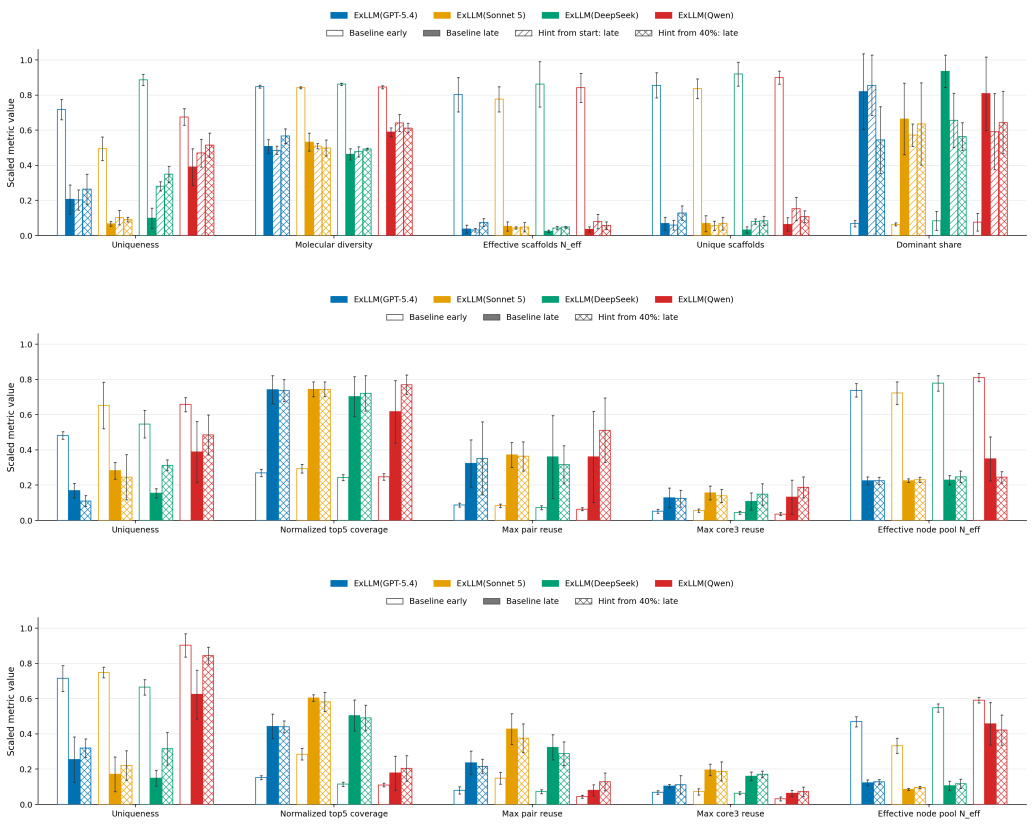


Figure 12: Intervention concentration diagnostics. Within each backbone: baseline early, baseline late, from-start late when available, and delayed-hint late. Improved performance does not require an increase in principle diversity; a run may move to a better concentrated region.

Table 18: Joint kernel diagnostics for the four main ExLLM backbones, five seeds each. "Late plateau" means concentrated with $G \geq 2$ but less than 30% of observed attempts remaining. The $G < 2$ column does not imply matching the expert.

Task	n	Collapse	Late plateau	$G < 2$, concentrated	Not concentrated
GQA decode	20	5	5	10	0
MLA decode	20	9	6	5	0
MLA prefill	20	19	1	0	0
MoE FP8	20	15	4	0	1
RMSNorm	20	0	0	19	1

Node effective count and reuse.
For $q = 50$ distinct sets of r nodes, let $c_v = \sum_{x \in \mathcal{A}_{50}} \mathbf{1}[v \in x]$ and $p_v = c_v/(qr)$. Then

$$N_{\text{eff}}^{\text{node}} = \exp\left(-\sum_v p_v \ln p_v\right),$$
$$R_2 = \max_{u<v} \frac{\sum_{x \in \mathcal{A}_{50}} \mathbf{1}[\{u,v\} \subseteq x]}{q}, \tag{14}$$
$$R_3 = \max_{u<v<w} \frac{\sum_{x \in \mathcal{A}_{50}} \mathbf{1}[\{u,v,w\} \subseteq x]}{q}.$$

Table 19: Kernel performance and final route concentration. ExLLM uses five seeds per task; PiEvo uses three. Late effective counts and collapse denominators include only runs with at least 20 valid kernels; latency summaries include runs with a recorded latency even when they fail that coverage requirement. $0/0$ means unassessable, not no collapse. GPT-4o is an additional control. No PiEvo RMSNorm results are supplied.

Method	Task	n	G (latency / expert)	Late N_{eff}	Collapse / assessed
ExLLM(GPT-5.4)	GQA decode	5	28.015 ± 15.670	1.000 ± 0.000	3/5
ExLLM(GPT-5.4)	MLA decode	5	22.140 ± 2.674	1.000 ± 0.000	4/5
ExLLM(GPT-5.4)	MLA prefill	5	33.396 ± 2.389	1.000 ± 0.000	5/5
ExLLM(GPT-5.4)	MoE FP8	5	37.548 ± 1.011	1.260 ± 0.355	5/5
ExLLM(GPT-5.4)	RMSNorm	5	0.467 ± 0.013	1.399 ± 0.298	0/5
ExLLM(Claude Sonnet 5)	GQA decode	5	0.762 ± 0.062	1.207 ± 0.298	0/5
ExLLM(Claude Sonnet 5)	MLA decode	5	1.151 ± 0.317	1.000 ± 0.000	0/5
ExLLM(Claude Sonnet 5)	MLA prefill	5	3.234 ± 0.455	1.000 ± 0.000	5/5
ExLLM(Claude Sonnet 5)	MoE FP8	5	10.434 ± 3.505	1.168 ± 0.377	1/5
ExLLM(Claude Sonnet 5)	RMSNorm	5	0.449 ± 0.009	1.360 ± 0.805	0/5
ExLLM(DeepSeek V4 Flash)	GQA decode	5	1.528 ± 0.529	1.168 ± 0.377	0/5
ExLLM(DeepSeek V4 Flash)	MLA decode	5	7.054 ± 2.786	1.077 ± 0.172	1/5
ExLLM(DeepSeek V4 Flash)	MLA prefill	5	34.779 ± 22.097	1.000 ± 0.000	5/5
ExLLM(DeepSeek V4 Flash)	MoE FP8	5	29.543 ± 8.427	1.317 ± 0.709	4/5
ExLLM(DeepSeek V4 Flash)	RMSNorm	5	0.457 ± 0.010	1.269 ± 0.421	0/5
ExLLM(Qwen3.5-35B)	GQA decode	5	44.126 ± 14.952	1.130 ± 0.290	2/5
ExLLM(Qwen3.5-35B)	MLA decode	5	73.652 ± 2.103	1.000 ± 0.000	4/5
ExLLM(Qwen3.5-35B)	MLA prefill	5	61.453 ± 12.202	1.000 ± 0.000	4/5
ExLLM(Qwen3.5-35B)	MoE FP8	5	46.419 ± 12.678	1.000 ± 0.000	5/5
ExLLM(Qwen3.5-35B)	RMSNorm	5	0.461 ± 0.007	1.000 ± 0.000	0/5
ExLLM(GPT-4o)	GQA decode	5	2758.283 ± 44.046	–	0/0
ExLLM(GPT-4o)	MLA decode	5	152.824 ± 37.781	1.000 ± 0.000	1/2
ExLLM(GPT-4o)	MLA prefill	5	–	–	0/0
ExLLM(GPT-4o)	MoE FP8	5	70.345 ± 0.936	–	0/0
ExLLM(GPT-4o)	RMSNorm	5	0.465 ± 0.002	1.000 ± 0.000	0/5
PiEvo(Claude Sonnet 5)	GQA decode	3	1.065 ± 0.112	1.678 ± 0.309	0/3
PiEvo(Claude Sonnet 5)	MLA decode	3	1.534 ± 0.674	1.128 ± 0.222	1/3
PiEvo(Claude Sonnet 5)	MLA prefill	3	2.303 ± 1.044	1.433 ± 0.375	1/3
PiEvo(Claude Sonnet 5)	MoE FP8	3	22.785 ± 5.584	1.816 ± 0.912	1/3
PiEvo(Gemini 2.5 Flash)	GQA decode	3	1887.464 ± 1634.618	–	0/0
PiEvo(Gemini 2.5 Flash)	MLA decode	3	173.297 ± 0.958	–	0/0
PiEvo(Gemini 2.5 Flash)	MLA prefill	3	63.976 ± 0.000	–	0/0
PiEvo(Gemini 2.5 Flash)	MoE FP8	3	57.502 ± 22.985	1.976 ± 0.838	1/2

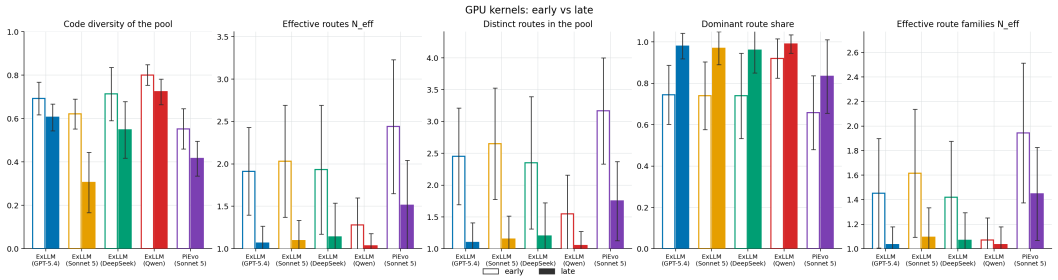


Figure 13: Kernel early/late diagnostics in the fastest-ten sets. Empty/filled bars denote initial/final sets. Code diversity is mean pairwise token 5-gram Jaccard distance, not the fraction of distinct candidates. Fine-grained routes and their broader families are both shown.

$R_2, R_3 \in [1/50, 1]$ for 50 nonempty sets of size at least three; high values indicate a repeatedly selected core. If V_5 contains the five most-used nodes, normalized top-five coverage is

$$C_5 = \frac{\sum_{v \in V_5} c_v/(qr)}{C_5^{\max}(N, r, q)}. \tag{15}$$

For 50 distinct sets, $C_5^{\max} = 0.775$ on Dolphin and 0.804 on residence hall, so $C_5 \in [0, 1]$. These are graph-independent combinatorial maxima given N, r, q, not fitted to a run. The corresponding minimum effective counts are 7.837 and 7.981, respectively: the minimum is not simply r, because

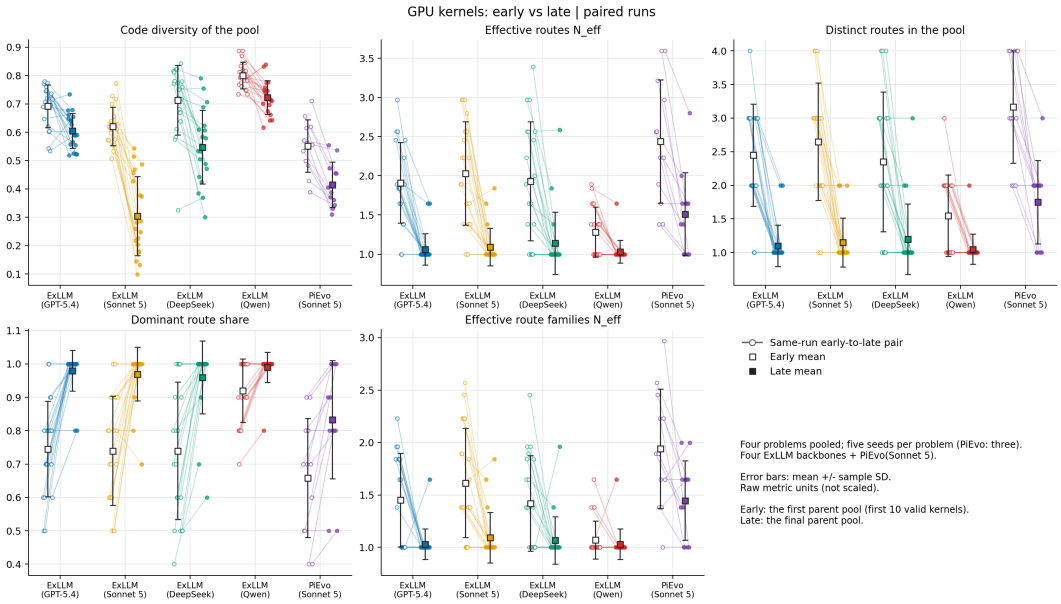


Figure 14: The same kernel early/late diagnostics with paired individual runs. Differences across problems and seeds remain visible rather than being inferred solely from pooled means.

Table 20: Paired kernel intervention outcomes across the four main backbones. Geometric-mean gain above one favors the hint. Missing final latencies are excluded, not assigned neutral gains.

Timing	Hint	Pairs / planned	Better	Worse	Within band	Geom. gain
Start	L1	48/48	28	12	8	1.169
Start	L2	46/48	14	21	11	0.833
Start	L3	46/48	20	15	11	1.067
Onset	L1	41/41	12	10	19	1.110
Onset	L2	41/41	13	9	19	1.010
Onset	L3	41/41	16	8	17	0.986

all 50 sets must differ. Upper bounds are $\min(N, qr)$. Bars divide node effective count by N; trajectories retain raw counts. These metrics describe node identities and shared subsets; low effective node count alone does not imply poor performance.

Early and late uniqueness.
Let G count generated candidates and $R(G)$ count proposals repeated relative to prior history. For a generation-count window $[a, b]$,

$$U_{[a,b]} = 1 - \frac{R(b) - R(a)}{b - a}. \tag{16}$$

We use the first and last 1,000 generated candidates after initialization, not 1,000 evaluations. Counters at unsaved boundaries are linearly interpolated within the boundary batches, except for ReEvo: its generated count is incremented for a whole batch before duplicate counts are updated candidate by candidate. ReEvo's first/last-1,000 windows are therefore reconstructed exactly from ordered saved proposals, with canonical-SMILES deduplication and all final counts checked against the logs. Global uniqueness and cumulative-uniqueness curves retain the logged values. Thus a high unique-candidate rate and a concentrated scaffold distribution are compatible.

Retrospective plateau onset and reference headroom.
For observed endpoint T_{obs} and best molecular fitness $F_1(b)$, define

$$b_{\text{plat}} = \min\{b : F_1(T_{\text{obs}}) - F_1(b) \le \epsilon_F\}, \qquad \epsilon_F = 0.01. \tag{17}$$

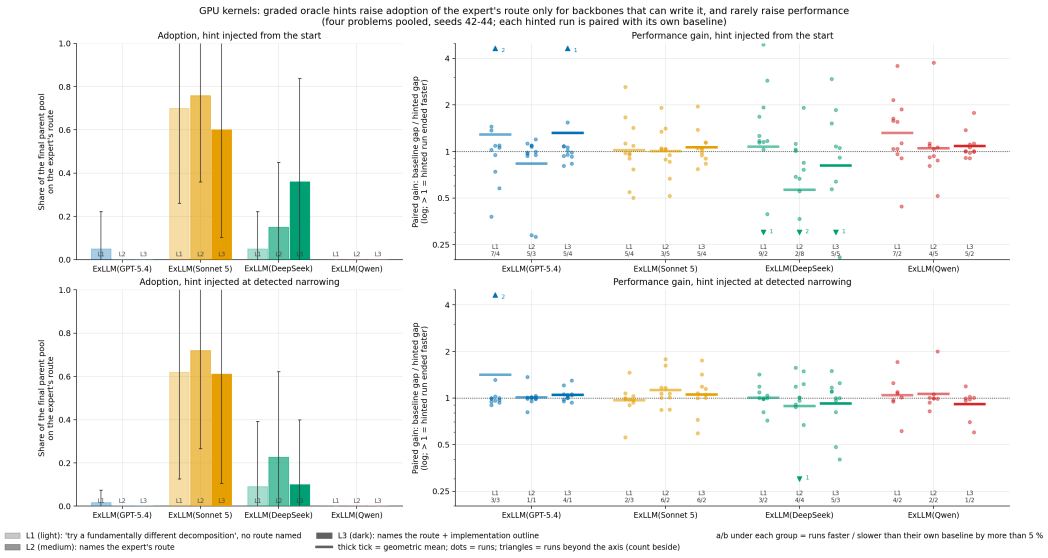


Figure 15: Kernel principle interventions, paired with each run's baseline on seeds 42–44. Left: reference-route share in the final fastest-ten set (runs with at least 20 valid kernels). Right: baseline/hinted latency, with geometric means and individual runs; values above one indicate benefit. Top/bottom rows inject from the start/at detected narrowing. Adoption and performance need not improve together. Exact coverage and gain-band definitions are in Table 20.

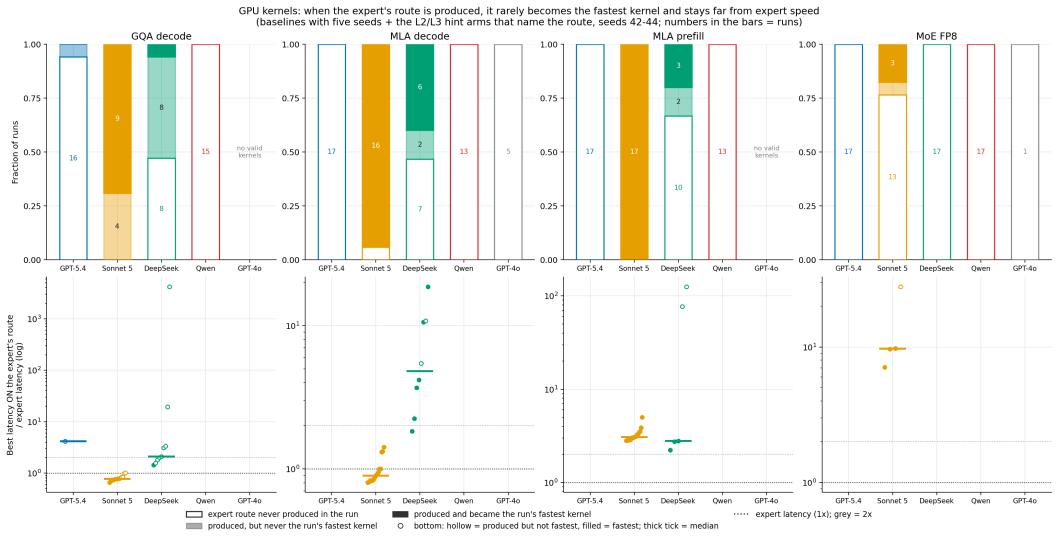


Figure 16: Best achieved latency by executed route in baselines and L2/L3 interventions. Route membership does not ensure an expert-quality implementation.

For best network error $E_1(b)$ use

$$b_{\text{plat}} = \min\left\{ b : \frac{E_1(b) - E_1(T_{\text{obs}})}{E_1(b)} \le 0.01 \right\}, \qquad R_{\text{budget}} = \frac{B - b_{\text{plat}}}{B}, \tag{18}$$

where B is the planned budget. A working long-plateau condition is $R_{\text{budget}} \ge 0.3$. This is a retrospective description, not an online detector or a causal estimate of wasted evaluations. Molecular headroom uses $F_1 < 4.3$, a gap greater than 0.3985 to the reference. Both networks use $E_1/E_{\text{ref}} \ge 2$, with the exhaustive Dolphin reference and $E_{\text{ref}} = 2.3817544421 \times 10^{-7}$ for residence hall. A gap does not establish restriction, and a plateau near a good reference is not classified as harmful simply because it is long.

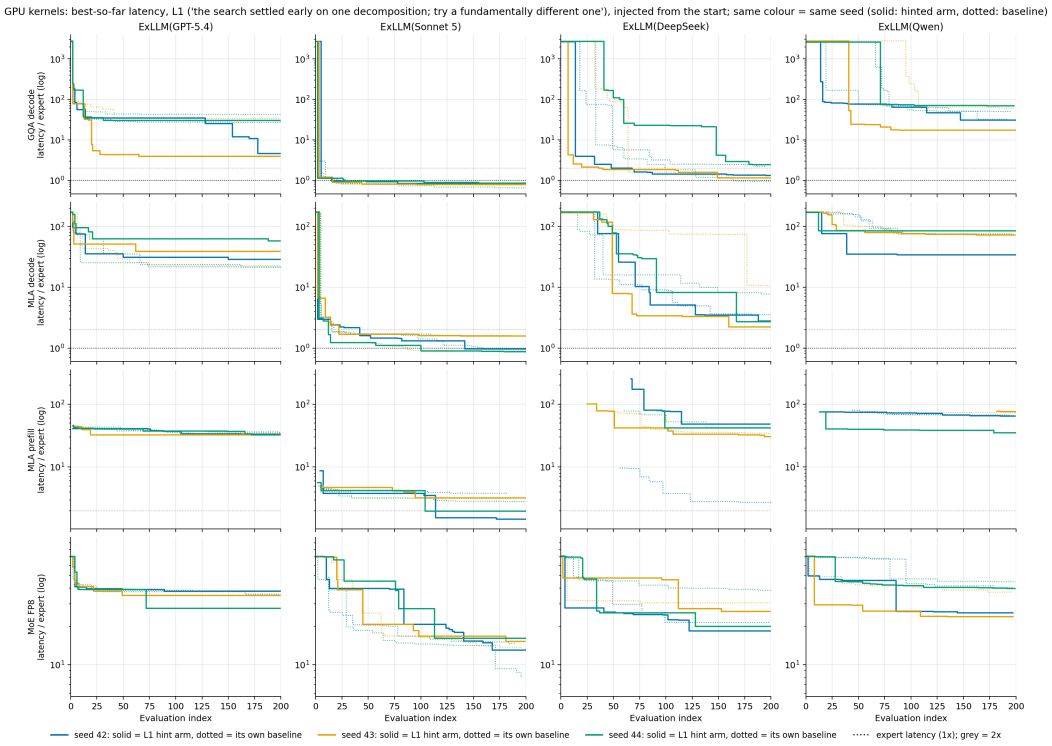


Figure 17: L1 from the start: hinted runs (solid) and their baselines (dotted), matched by seed and color. Latency is divided by expert latency; lower is better.

Table 21: Further development of a route by the quality of its first kernel (186 events, 80 baseline runs, five seeds). IQR is the interquartile range of N_{after}; the last two columns restrict to events with at least 50 attempts left.

First kernel of the route vs. the run's best so far	events	never developed again	median N_{after}	IQR	events with ≥ 50 attempts left	their median N_{after}
First kernel faster than the incumbent	73	3%	59	15–96	71	64
Up to 2x slower	69	23%	5	1–28	67	5
More than 2x slower	44	36%	1	0–4	36	1.5
All events	186	18%	9.5	1–78.5	174	14

The graded verdict is defined in Section 3.2, with structural concentration thresholds in Appendix B. Endpoint concentration does not assert that the same restriction persisted throughout the entire plateau; the temporal curves provide complementary evidence.

Diagnostic support rates.
Let n_S, n_L, n_N count strong support, limited support, and collapse not established in the specified comparison, and let $n_{\text{assess}} = n_S + n_L + n_N$. Then

$$P_{\text{strong}} = \frac{n_S}{n_{\text{assess}}}, \qquad P_{\text{support}} = \frac{n_S + n_L}{n_{\text{assess}}}. \tag{19}$$

Both lie in $[0, 1]$ and are descriptive diagnostic frequencies, not probabilities that collapse is true. Higher values indicate more evidence of the failure mode, not better optimization. Runs with insufficient valid candidates to assess the kernel structural diagnostic are excluded from this denominator and reported separately. Aggregate non-control rates exclude Dolphin and RMSNorm. Limited support is not confirmed collapse, and not established is not proof of its absence. Table entries headed Remaining (%) are $100R_{\text{budget}}$.

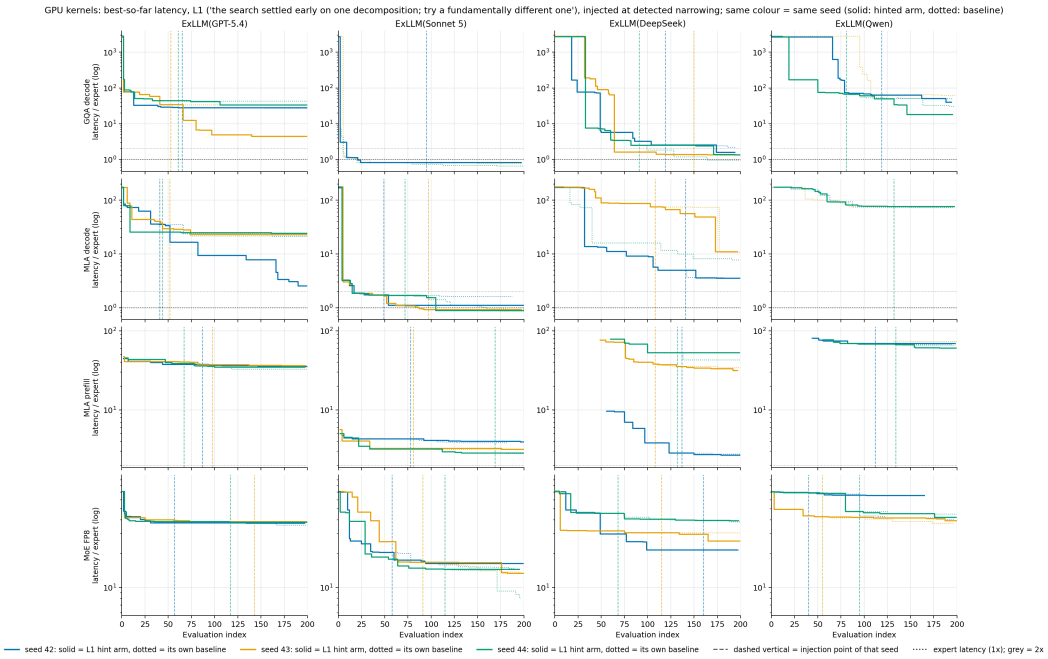


Figure 18: L1 at detected narrowing. Dashed vertical lines mark injection; other conventions match the from-start comparison.

Table 22: The 18 first appearances of the expert's route in the baseline runs: evaluation attempt of the first kernel, its ratio r to the run's best latency at that moment, further kernels on the route, and the best latency reached on the route relative to the expert ($\star$: the route produced the run's fastest kernel).

Problem	Backbone	seed	attempt	r	N_{after}	best on the route / expert
GQA decode	Sonnet 5	42	22	1.35	14	0.98
GQA decode	Sonnet 5	43	15	3.39	37	0.77 $\star$
GQA decode	Sonnet 5	44	14	3.95	63	0.66 $\star$
GQA decode	Sonnet 5	45	19	2.99	13	1.01
GQA decode	Sonnet 5	46	13	1.55	18	0.79 $\star$
MLA decode	Sonnet 5	43	16	3.27	151	0.86 $\star$
MLA decode	Sonnet 5	44	12	2.45	99	0.95 $\star$
MLA decode	Sonnet 5	45	12	1.02	160	1.00 $\star$
MLA decode	Sonnet 5	46	5	2.39	105	1.33 $\star$
MLA decode	DeepSeek	42	42	1.93	1	10.79
MLA prefill	Sonnet 5	43	5	1.06	6	4.16
MLA prefill	Sonnet 5	43	127	2.32	0	7.35
MLA prefill	Sonnet 5	44	5	1.28	73	3.20
MLA prefill	Sonnet 5	44	76	1.18	70	2.83 $\star$
MLA prefill	Sonnet 5	45	7	0.96	131	2.85 $\star$
MLA prefill	Sonnet 5	45	58	1.70	0	5.46
MLA prefill	Sonnet 5	46	28	0.91	113	3.55 $\star$
MLA prefill	DeepSeek	42	74	1.60	52	2.78 $\star$

L.2 First-shot and intervention metrics

First-shot quality and subsequent investment.
For a kernel route p, let $t_{\text{first}}(p) = \min\{i : \phi(x_i) = p\}$. Its first-shot ratio and subsequent investment are

$$r(p) = \frac{\ell(x_{t_{\text{first}}(p)})}{\min_{i<t_{\text{first}}(p)} \ell(x_i)}, \qquad N_{\text{after}}(p) = \sum_{i>t_{\text{first}}(p)} \mathbf{1}[\phi(x_i) = p]. \tag{20}$$

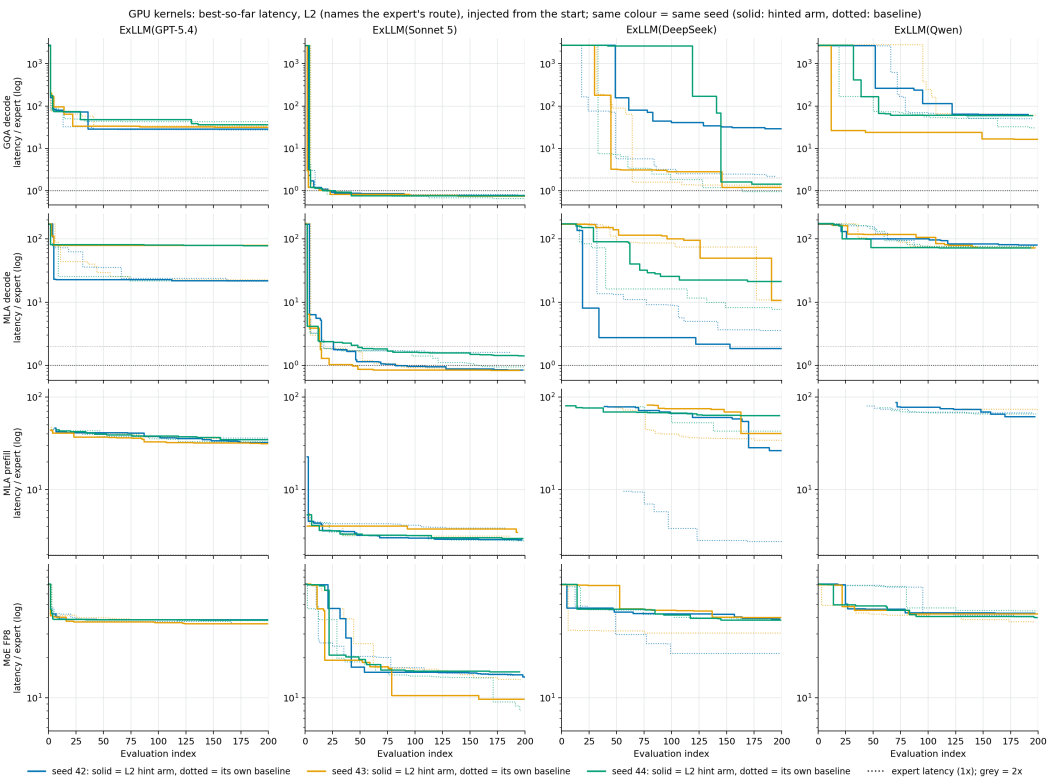


Figure 19: L2 from the start, paired with baseline trajectories.

The first route is excluded because it has no prior comparator. Smaller r means a stronger first attempt: $r < 1$ beats the prior best, whereas $r > 1$ is slower. The displayed groups are $r < 1$, $1 \le r < 2$, and $r \ge 2$. Investment is a nonnegative count bounded by remaining attempts; it measures development effort, not performance. The analysis contains 186 route-first-appearance events. The budget control retains events with at least 50 evaluation attempts left. For a group of events $\mathcal{E}$, the fraction never developed again is

$$P_{\text{no development}} = \frac{1}{|\mathcal{E}|} \sum_{p \in \mathcal{E}} \mathbf{1}[N_{\text{after}}(p) = 0]. \tag{21}$$

This lies in $[0, 1]$; larger values indicate less follow-up, not necessarily a worse final solution.

Relative benefit and paired kernel gain.
Relative reduction in network group-mean error and paired kernel gain are

$$I_{\text{error}} = 1 - \frac{\overline{\text{Err}}_{1,\text{hint}}}{\overline{\text{Err}}_{1,\text{base}}}, \qquad g_s = \frac{\ell^{\star}_{\text{base},s}}{\ell^{\star}_{\text{hint},s}}. \tag{22}$$

Positive I_{error} means lower mean error; it is not a mean of per-seed percentage reductions. Kernel comparisons match task, backbone, and seed, using whole runs for from-start arms and post-onset candidates over the remaining budget for onset arms. The summary counts $g_s > 1.05$ as better, $g_s < 0.95$ as worse, and the rest as unchanged. $g_s = 1$ means no gain; larger is better.

Kernel route concentration and reference-route adoption.
For the diagnostic set $\mathcal{A}_{10}(b)$ of up to ten fastest valid kernels, let $n_b = |\mathcal{A}_{10}(b)|$ and use the label frequencies in Eq. equation 3. We report

$$S_{\text{route}} = |\{p : \hat{\pi}_b(p) > 0\}|, \quad N^{\text{route}}_{\text{eff}} = \exp\left[-\sum_p \hat{\pi}_b(p) \ln \hat{\pi}_b(p)\right], \quad m_{\text{route}} = \max_p \hat{\pi}_b(p). \tag{23}$$

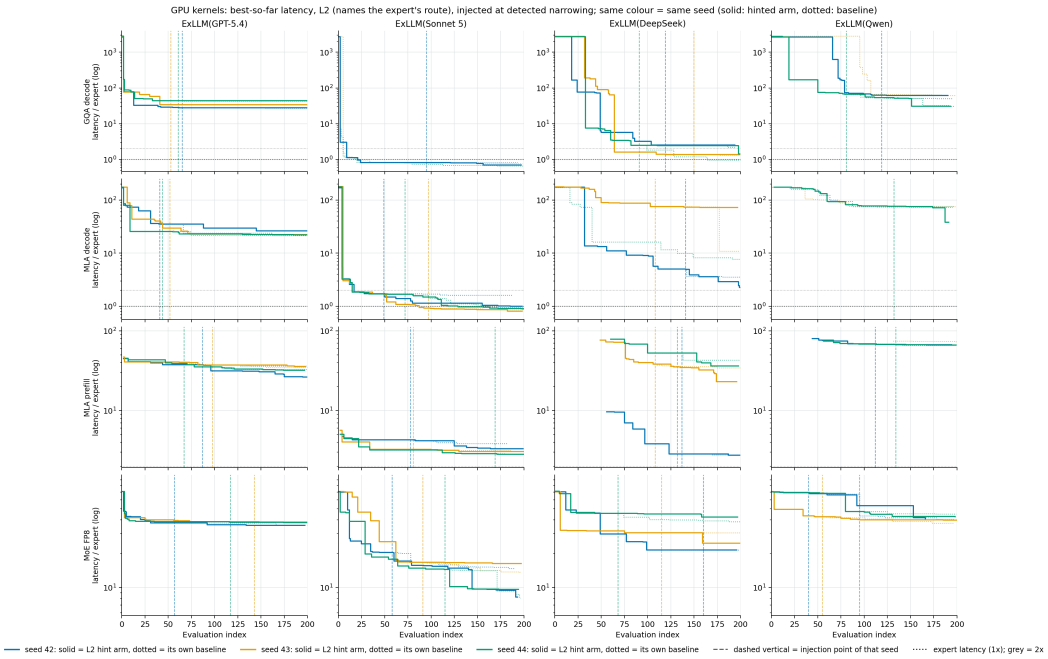


Figure 20: L2 at detected narrowing, paired with baseline trajectories.

For $n_b > 0$, $1 \le N_{\text{eff}}^{\text{route}} \le S_{\text{route}} \le n_b$ and $1/n_b \le m_{\text{route}} \le 1$. Route-family effective count uses the same formula after merging leaf routes by the annotation rubric. Lower effective count or higher dominant share indicates concentration, not necessarily failure. Reference-route adoption is

$$q_{\text{ref}}(b) = \frac{1}{n_b} \sum_{x \in \mathcal{A}_{10}(b)} \mathbf{1}[\phi(x) = p_{\text{ref}}], \tag{24}$$

where p_{ref} is the expert's route. It lies in $[0, 1]$; higher values mean greater adoption, not necessarily a faster implementation. Intervention figures use this final fastest-ten share, not a last-20-proposal retention threshold. Route-conditioned performance is the best observed latency on a route divided by expert latency; smaller is better and it is undefined if that route was never evaluated. The kernel performance curves show $1/G$ (Eq. equation 11), so larger is better and one matches the expert.

L.3 PiEvo and statistical summaries

Belief-posterior entropy.
Let $b_t(p)$ be PiEvo's recorded posterior over J_t active textual principles, distinct from the structural descriptor distribution $\hat{\pi}_b$. Then

$$H_t^{\text{belief}} = -\sum_{p=1}^{J_t} b_t(p) \ln b_t(p), \qquad \overline{H}^{\text{belief}} = \frac{\sum_{t \in \mathcal{B}} w_t H_t^{\text{belief}}}{\sum_{t \in \mathcal{B}} w_t}, \tag{25}$$

where $\mathcal{B}$ indexes recorded belief updates in the first or last 1,000 generated attempts, and w_t is the number of attempts from that round inside the window. Entropy lies in $[0, \ln J_t]$ nats. Zero means all posterior mass is on one principle, or only one exists. Higher means less concentrated belief, not necessarily more diverse molecules or better search. Tables report attempt-weighted averages within runs, then mean±std across seeds; missing beliefs are omitted rather than assigned zero entropy.

Effective used-text count and semantic strategy share.
In a generated-attempt window $\mathcal{W}$, let w_t be the number of attempts from round t inside the window

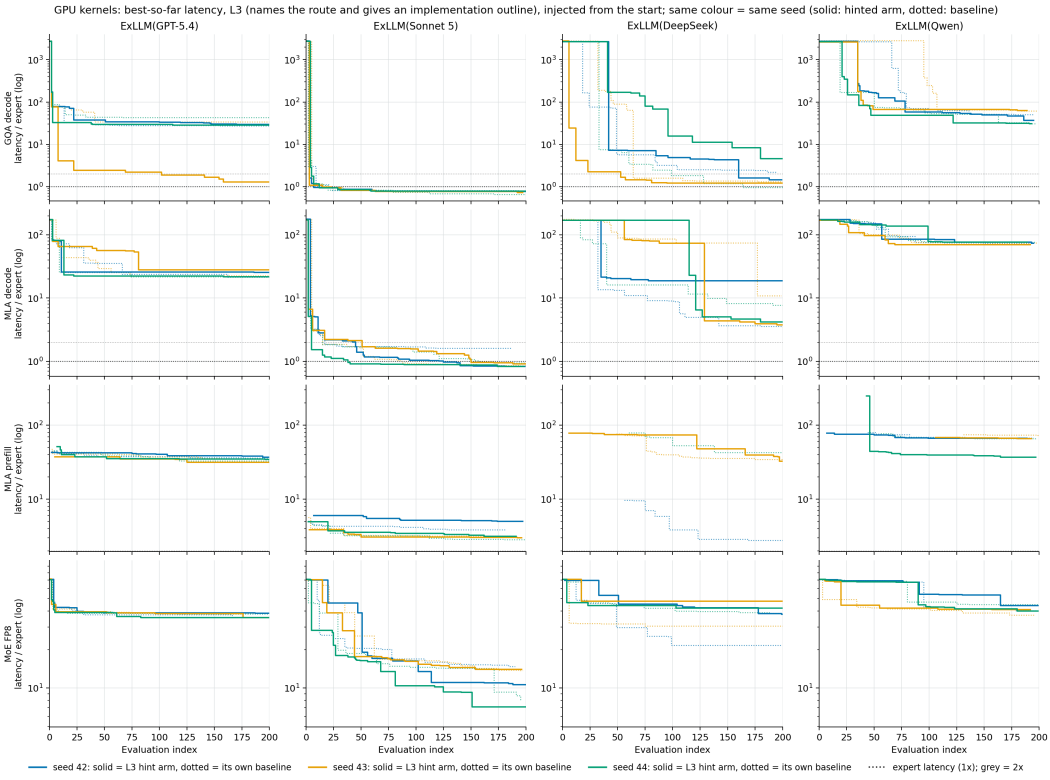


Figure 21: L3 from the start, paired with baseline trajectories.

and j_t its actually used principle text, normalized only for whitespace. With $W = \sum_t w_t$,

$$q_j = \frac{\sum_t w_t \mathbf{1}[j_t = j]}{W}, \qquad T_{\text{eff}} = \exp\left[-\sum_j q_j \ln q_j\right]. \tag{26}$$

This is the $T_{\text{eff},e/l}$ reported in the PiEvo tables. Its range is $[1, K_{\mathcal{W}}]$, where $K_{\mathcal{W}}$ is the number of distinct used texts. Larger means more varied text usage, not necessarily distinct strategies. For semantic-route statistics, merge wording variants using the recorded operational labels and apply the same effective-count formula to the resulting frequencies. Dominant strategy share is their maximum, in $[1/K_{\text{route}}, 1]$; larger means more concentrated use. The annotations and their limitations are described in Appendix G.

Anomaly counts and principle-update counts.
Let $\mathcal{R}_{\mathcal{W}}$ contain rounds whose batch ends fall within $\mathcal{W}$, and $\mathcal{Q}_{\mathcal{W}}$ contain principle decisions at batch starts within $\mathcal{W}$. From the complete saved records,

$$A_{\text{count}}(\mathcal{W}) = \sum_{t \in \mathcal{R}_{\mathcal{W}}} \mathbf{1}[\text{triggered}_t], \quad A_{\text{rate}}(\mathcal{W}) = \frac{A_{\text{count}}(\mathcal{W})}{|\mathcal{R}_{\mathcal{W}}|}, \quad U_{\text{principle}}(\mathcal{W}) = \sum_{t \in \mathcal{Q}_{\mathcal{W}}} \mathbf{1}[g_t \neq \texttt{PAUSE_PRINCIPLES}]. \tag{27}$$

Counts are integers bounded by the number of eligible records; the rate lies in $[0, 1]$ and is undefined for an empty window. Updates count generation/refinement calls, not proven new semantic directions; higher counts do not necessarily mean better search. Events are not fractionally weighted at window boundaries. Unless stated otherwise, early/late PiEvo tables use the first/last 1,000 attempts after initial-candidate loading and include forced principle warm-up in the early window. The main-text update comparison instead uses the 1,000 attempts immediately after the ten warm-up rounds: 3.800 ± 2.775 updates versus 1.000 ± 2.236 late for original PiEvo, and 1.000 ± 0.707 versus 0.400 ± 0.894 for $\sigma = 0.9$. These shifted update windows do not change the early posterior

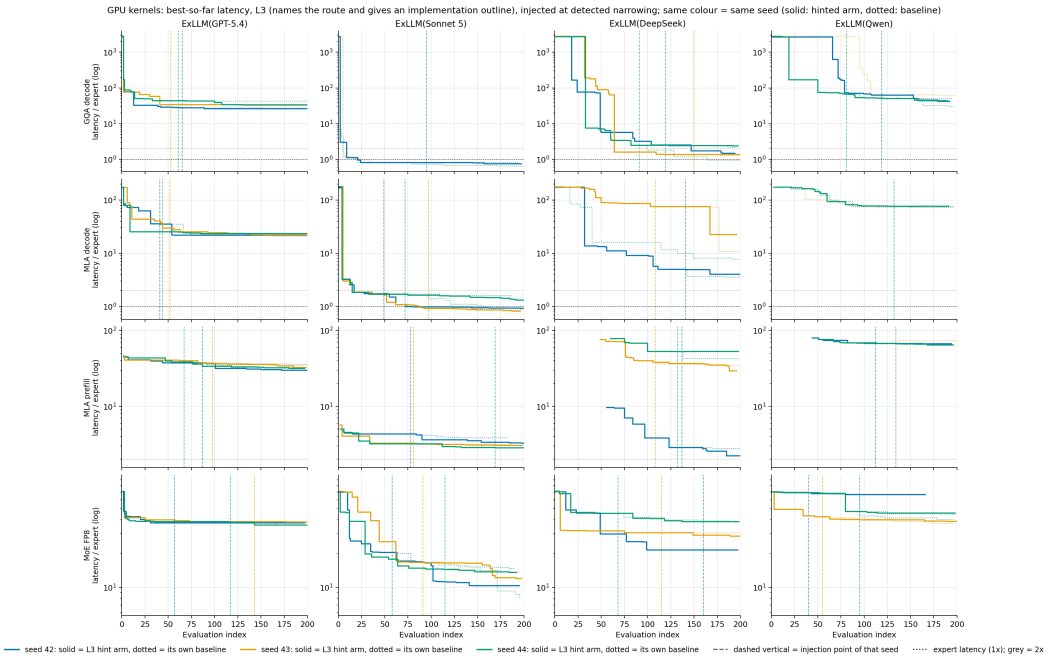


Figure 22: L3 at detected narrowing, paired with baseline trajectories.

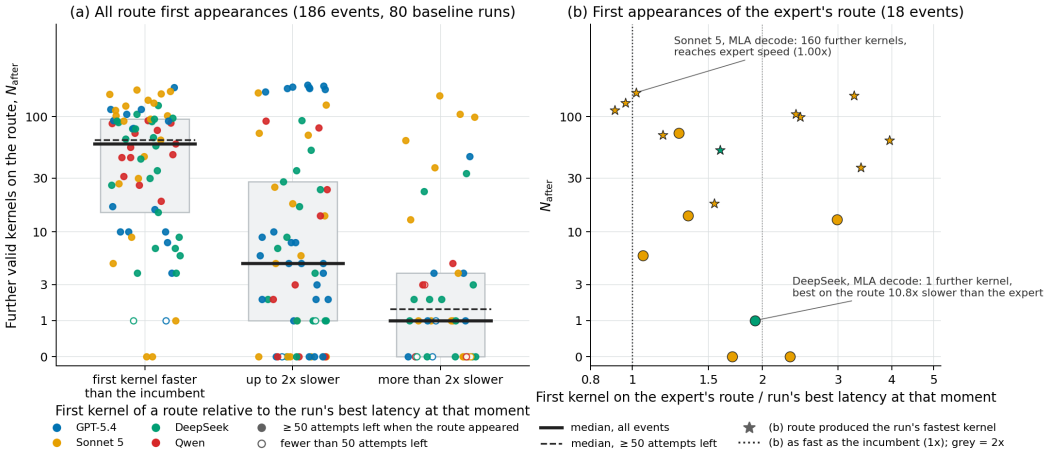


Figure 23: Kernel first-attempt quality versus subsequent development. (a) All 186 events, grouped by the ratio of the route's first kernel to the run's best latency at that moment. Dots are events coloured by backbone (filled: at least 50 attempts left, hollow: fewer); boxes span the interquartile range; the solid line is the median and the dashed line the median of the control. The vertical axis is logarithmic in $1 + N_{\text{after}}$ so that zero can be shown. (b) The 18 first appearances of the expert's route; stars mark routes that later produced the run's fastest kernel.

or text-usage windows in the tables. Trigger scarcity alone does not show that small improvements suppress anomalies.

Statistical summaries.
For a run-level metric z_s on n observed seeds,

$$\bar{z} = \frac{1}{n}\sum_{s=1}^{n} z_s, \qquad s_z = \sqrt{\frac{1}{n-1}\sum_{s=1}^{n}(z_s - \bar{z})^2}. \tag{28}$$

Tables and convergence bands use mean±sample standard deviation, not standard error. Nonlinear metrics are calculated within runs before aggregation. First-attempt plots summarize events by the median and interquartile range: the table reports $[Q_{0.25}, Q_{0.75}]$, while the IQR width is $Q_{0.75} - Q_{0.25}$. These describe event variation, not confidence intervals. For positive paired kernel gains, the geometric mean is $\exp(n^{-1}\sum_s \ln g_s)$; pooled kernel speed curves likewise average the log ratios before exponentiating. Diversity-injection summaries first average backbones within each seed, then compute over the five seed averages.

Token usage and estimated price.
For token counts $T_{\text{in}}, T_{\text{out}}$ and USD prices per million tokens $c_{\text{in}}, c_{\text{out}}$,

$$T_{\text{in}}^{(\text{M})} = \frac{T_{\text{in}}}{10^6}, \quad T_{\text{out}}^{(\text{M})} = \frac{T_{\text{out}}}{10^6}, \quad C_{\text{USD}} = c_{\text{in}} T_{\text{in}}^{(\text{M})} + c_{\text{out}} T_{\text{out}}^{(\text{M})}. \tag{29}$$

Cost is computed per run before aggregation. It is a list-price estimate, excluding caching discounts and infrastructure costs.

L.4 Candidate-level metrics

Validity and uniqueness.
Let N_{gen}, N_{fail}, and N_{rep} be the logged generated, invalid, and duplicate counts.

$$V = 1 - \frac{N_{\text{fail}}}{N_{\text{gen}}}, \qquad U = 1 - \frac{N_{\text{rep}}}{N_{\text{gen}}}. \tag{30}$$

Both lie in $[0, 1]$: larger means fewer invalid proposals or repetitions, respectively. Molecular duplicates use canonical SMILES. Network uniqueness retains the runner's logged duplicate definition; it is not recomputed by the unordered-set deduplication used for selected-population metrics. Importantly, U is a non-repetition fraction over *all generated candidates*, not the fraction of valid candidates that are distinct. Invalid proposals contribute to N_{gen} but not N_{rep}, so U should be read alongside V. Kernel validity is $N_{\text{pass}}/N_{\text{eval}}$, where passing requires compilation and harness correctness.

Molecular Tanimoto diversity.
Let $B(x)$ be the active bits of a 2,048-bit Morgan fingerprint (radius 2, chirality disabled). For the n_t distinct molecules in $\mathcal{A}_{50}(t)$,

$$\begin{aligned} \text{Tan}(x, x') &= \frac{|B(x) \cap B(x')|}{|B(x) \cup B(x')|}, \\ \text{Div}_{\text{mol}}(t) &= \frac{2}{n_t(n_t - 1)} \sum_{i<k} [1 - \text{Tan}(x_i, x_k)]. \end{aligned} \tag{31}$$

Self-pairs are excluded. The statistic is undefined for fewer than two molecules and lies in $[0, 1]$ otherwise. Higher means more fingerprint variation among the selected 50 molecules, not necessarily more scaffold-level principles.

Code diversity.
Tokenize source into identifiers, digit sequences, and individual non-whitespace symbols. Let $B_5(x)$ be its set of consecutive five-token sequences. Within a window $\mathcal{W}$ of $n_{\mathcal{W}} \geq 2$ candidates,

$$\text{Div}_{\text{code}}(\mathcal{W}) = \frac{2}{n_{\mathcal{W}}(n_{\mathcal{W}} - 1)} \sum_{i<k} \left[1 - \frac{|B_5(x_i) \cap B_5(x_k)|}{|B_5(x_i) \cup B_5(x_k)|}\right]. \tag{32}$$

A shorter source contributes its whole token sequence; two empty sets have distance zero. Range $[0, 1]$: larger means more surface variation, not necessarily a different executed route.

M Oracle-principle hints

The hints of Table 23 are written as principle-level search directions. They name a region or role pattern but no candidate, scaffold, node set or ranked list.

Table 23: Oracle-principle hints used in the interventions.

Task	Hint
Molecular (L2)	Search around compact heteroaryl-amide chemotypes rather than only refining generic benzamide or para-methoxy aryl-amide families. Favor low-synthetic-complexity molecules with a small sulfur-containing or heteroaromatic ring, a simple aryl/benzyl partner, and light substituent tuning to keep QED high while driving the unwanted target probabilities near zero.
Dolphin (L2)	A direction is a heterogeneous degree profile: include one low-degree or leaf-like sentinel and combine it with low-to-mid and moderate/high-role nodes whose errors complement each other as a set.
Residence hall (L2)	Look for five sentinels that jointly cover different network roles across degree and neighborhood structure. Try heterogeneous complementarity, try a mixed low/mid/moderately-high profile and test replacements that preserve that role coverage.
Kernel, all problems (L1)	Optimization trajectories on this problem typically settle early on one overall design approach whose latency stays far above what the hardware allows, and further refinement within that approach does not close the gap. Whenever the implementations shown share the same overall algorithmic decomposition, do not keep tuning it; propose a candidate whose decomposition of the computation is fundamentally different.
GQA / MLA paged decode (L2)	Instead of one program per (sequence, head) that walks the whole KV sequence serially, split each sequence's KV range into several chunks processed by different programs in parallel, each producing partial attention statistics (running max, normalizer, accumulated output), and merge the chunks afterwards with log-sum-exp rescaling. Pack the heads that share the same KV as the row dimension of a matrix product so that scores and outputs run on tensor cores and each KV block is read once.
MLA paged prefill (L2)	Treat the computation as a two-dimensional tiling—a block of query tokens (with all heads, which share the same compressed KV) against a block of KV tokens—so that score computation and output accumulation are matrix products on tensor cores, with an online softmax carried across KV blocks and the causal mask applied only where a block crosses the diagonal. Long sequences can additionally be split along the KV dimension across programs and merged with log-sum-exp rescaling. Avoid host-side loops over sequences.
FP8 MoE (L2)	Do not dequantize the FP8 weights to fp32 or bf16 before the GEMMs. Feed the FP8 activations and weights directly to tensor-core matrix multiplies and apply the 128-block scales to the accumulator as the K dimension is traversed. Group the tokens by expert once and run the per-expert GEMMs as one grouped launch rather than a Python loop over experts, fusing the SwiGLU between the two GEMMs and the top-k weighted reduction after the second.
MLA paged decode (L3, example)	The L2 direction, followed by an outline: (1) choose a chunk length and a grid over (sequence, chunk); programs past the sequence length exit early. (2) Chunk kernel: load q_nope as a [16, 512] tile and q_pe as [16, 64]; iterate the chunk's KV positions in blocks, gather compressed-KV rows [block, 512] and rope rows [block, 64] through kv_indices; scores = dot(q_nope, $\text{ckv}^\top$) + dot(q_pe, $\text{kpe}^\top$) scaled by sm_scale; keep a per-head running max and normalizer; accumulate dot(P, ckv) into a [16, 512] tile; write partial max, normalizer and output to scratch indexed by (sequence, chunk). (3) Merge kernel per sequence and head: global max over chunks, rescale each chunk, sum normalizers and outputs, divide, emit the log-sum-exp. (4) Pick the chunk length so that sequences $\times$ chunks is a small multiple of the SM count.