Permutation-based Reasoning and Intelligence Search Method (PRISM)

Predicting When Search Helps with a Pre-Flight Protocol,
and an Exhaustive Map of Language-Model Instruction Order

Blessings Mambwe
ML Collective

August 8, 2026

arXiv:2608.08344v1 [cs.LG] 8 Aug 2026

Abstract

PRISM addresses permutation-valued decisions in systems whose components are fixed but whose order is not: compiler passes, preprocessing steps, neural modules, or reasoning instructions. Before selecting an optimizer, it asks whether structured search is likely to beat uniform sampling. PRISM's *pre-flight* uses a small fixed measurement budget to estimate one-move fitness autocorrelation and fitness–distance correlation, forecasts the search regime, selects a method, and records the forecast beside the outcome. On exactly enumerated landscapes, the protocol predicts its own failures: on a 5,040-ordering neural landscape, evolutionary search finds an optimum in 19 of 40 runs, versus 30 of 40 for uniform sampling, as near-zero local correlation forecasts. In the flagship application, all 720 orderings of six fixed reasoning instructions span 6.3% to 96.9% accuracy on a GSM8K subset. The position effects transfer across model families and survive automated rewriting of the instruction text. Cross-domain evidence includes six scientific-pipeline systems, 18 registered tabular architecture forecasts, and two preliminary reinforcement-learning ordering surfaces. Code and artifact index: https://github.com/bleymambwe/PRISM.

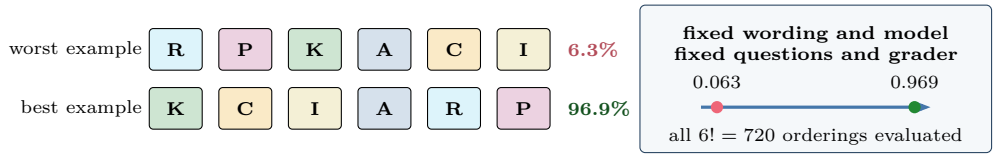


Figure 1: Order alone moves measured accuracy by 90.6 percentage points. The strips show one worst and one best ordering of RESTATE (R), IDENTIFY (I), PLAN (P), COMPUTE (C), CHECK (K), and ANSWER (A). Every ordering is evaluated on the same 32 GSM8K questions with fixed wording, deterministic decoding, and the same grader.

1 Introduction

Many systems have already selected the right components but have not justified their order. A compiler has passes, a scientific pipeline has preprocessing operations, a neural model has modules, and a prompt has instructions. Once the component multiset is fixed, a candidate is a permutation $\pi \in \mathbb{S}_n$, and the optimization problem is

$$\pi^* \in \arg\max_{\pi \in \mathbb{S}_n} F(\pi), \tag{1}$$

where F is the task score. Composition is generally non-commutative: moving a nonlinear operation, a normalizer, or an instruction changes what later components receive. The magnitude of this effect in the exhaustive instruction study is previewed in Figure 1.

Fixing the multiset is a genuine restriction on a broader configuration space. If n ordered slots draw from k component types, unrestricted assignment yields k^n sequences. With fixed counts $c_1, \ldots, c_k$, $\sum_i c_i = n$, the reachable arrangements number

$$|S(\mathbf{c})| = \frac{n!}{c_1! \cdots c_k!} \leq k^n, \tag{2}$$

because they form one subset of the length-n sequences over the same alphabet. When all components are distinct, as in the experiments here, $|S| = n!$. This restriction buys exact ground truth at small n: six instructions have 720 orderings, compared with $6^6 = 46{,}656$ unrestricted six-slot configurations.

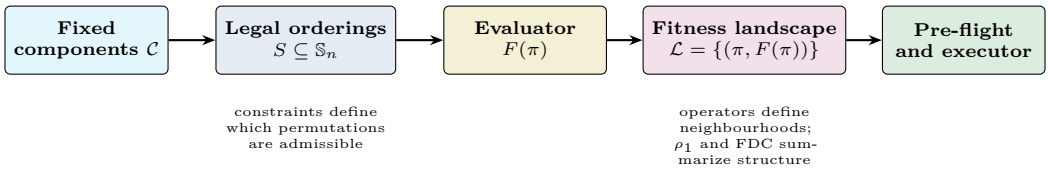


Figure 2: Objects required for a PRISM study. The landscape is not merely the search-space size: it is the scored structure induced by the evaluator and the chosen neighbourhood operators. The empirical landscape regimes are visualized in Figure 4.

Restriction is not simplification. Cardinality does not determine search difficulty. The exact parity landscape over $7! = 5{,}040$ orderings defeats elitist evolutionary search, which finds an optimum in 19 of 40 runs against 30 of 40 for uniform sampling. A matched-operator synthetic landscape over $16! \approx 2.09 \times 10^{13}$ orderings is solved in roughly 405 generations. A five-thousand-point space can be harder than a twenty-trillion-point one. The difference is structure: whether a search move preserves fitness information and whether better candidates tend to lie closer to an optimum. Neither property is visible in n, so both must be measured.

In this work, we introduce a pre-flight protocol that spends a small, fixed budget before the main search. It measures local fitness preservation and global guidance, checks whether the evaluator can distinguish candidates, forecasts whether structured search should help, and selects an *executor*—the method that spends the remaining budget. Possible executors include exact enumeration, uniform sampling, an evolutionary searcher, a surrogate, or a transfer-guided list. The forecast, rather than any single executor, is the contribution.

Our contributions are four results:

C1. A two-statistic pre-flight separates operator-aligned, near-random, and deceptive regimes and turns null results into testable forecasts (Sections 3 and 5).

C2. Exact enumeration exposes a decisive boundary: structured search can be worse than uniform sampling when its neighbourhood contains almost no fitness signal (Section 6).

C3. A complete 720-order map of fixed reasoning instructions reveals a 6.3%–96.9% accuracy range, transferable position effects, and ordering variation that survives automated wording optimization (Sections 7 and 8).

C4. Cross-domain studies cover six scientific-computing pipelines, registered tabular architecture forecasts, and preliminary option-ordering and curriculum-ordering surfaces in reinforcement learning (Sections 9 and 12).

The argument can be read as five claims: (1) order is a real optimization coordinate; (2) space size is not a difficulty measure; (3) search operators define the relevant locality; (4) exact ground truth makes failures visible; and (5) forecasts must be retained beside outcomes, including misses. The rest of the paper follows that order, then closes with cross-domain calibration and specific limitations. The final synthesis in Table 2 connects each headline finding to its practical recommendation.

Takeaway

Ordering is a real optimization surface, but the size of the space says nothing about whether searching it will pay. Difficulty must be measured from landscape structure before budget is spent.

2 Terminology and Prerequisites

PRISM assumes a constrained permutation problem rather than unrestricted system design. A *component* is a fixed item whose internal content does not change during the study. A *candidate* is one legal ordering π of those components, and the *evaluator* assigns its task score $F(\pi)$. The resulting *fitness landscape* is the set of legal orderings together with their scores. A move operator—swap, insertion, inversion, or scramble—defines a neighbourhood on that landscape. An *executor* is the method selected to spend the post-diagnostic evaluation budget, such as enumeration, sampling, a surrogate, or the evolutionary searcher. These objects and their dependencies are summarized in Figure 2.

Five prerequisites should hold before optimization begins: (1) the component set and legal-order constraints are explicit; (2) candidate scores are reproducible or accompanied by a declared noise model; (3) the evaluator has sufficient resolution to distinguish orderings; (4) repeated evaluations can be identified so distinct-evaluation budgets are comparable; and (5) at least one uninformed and one domain-specific baseline are available. If these conditions fail, the correct action is to repair the experimental design rather than interpret an optimizer ranking.

Protocol 1: PRISM pre-flight protocol

Input: frozen evaluator F, component multiset, probe budget B_p, main budget B, candidate executors.
Output: recorded forecast, selected executor, verified outcome.

1. Sample B_p distinct orderings; measure variance, tie density, and evaluator resolution.
2. Estimate $\rho_1(o)$ for each meaningful move o; estimate exact or proxy FDC.
3. Forecast the regime and select enumeration, sampling, evolution, a surrogate, or transfer.
4. Execute exactly B distinct evaluations; repeated proposals do not increase the count.
5. Record forecast, outcome, uncertainty, and any mismatch in the forecast ledger.

Table 1: Conservative decision rule. Thresholds are empirical operating bands, not universal constants.

Pre-flight observation	Action
Negligible variance or dense top ties	Improve the evaluator or sample modestly and stop.
One operator has clearly largest ρ_1	Use its neighbourhood if search is otherwise justified.
Materially negative FDC	Structured search may exploit global guidance.
Near-zero or modestly positive FDC	Use uniform sampling as the method of record.
Strongly positive FDC	Avoid local evolutionary search.

Takeaway

A permutation experiment is scientifically meaningful only when the components, legal space, evaluator, landscape neighbourhood, and comparison budget are defined before search.

3 The Pre-Flight Protocol

A pre-flight uses a small probe set $P \subset \mathbb{S}_n$ before the main optimization budget. Repeated orderings are served from cache and do not count again; we call this the *distinct-evaluation budget*. The probe first checks score variance and the density of top ties. If the evaluator cannot distinguish candidates, optimizer comparison is premature.

For a one-move operator o, local fitness preservation is estimated by

$$\rho_1(o) = \operatorname{corr}(F(\pi), F(o(\pi)))\,, \tag{3}$$

over sampled parent–child pairs. A positive value means that the neighbourhood preserves fitness information; a value near zero means that one move resembles a fresh draw. Global guidance is measured by fitness–distance correlation

$$\mathrm{FDC} = \operatorname{corr}(F(\pi), d(\pi, \Omega))\,, \tag{4}$$

where Ω is the exact optimum set when known and otherwise the best frozen probe sample. Because fitness is maximized, negative FDC indicates that better candidates tend to lie nearer the target; near-zero values imply weak global guidance; strongly positive values warn that local improvement points away from the optimum. The protocol below routes these measurements to the executor that spends the remaining budget, while quantitative decision bands appear in Table 1 and Figure 4.

Takeaway

Two cheap measurements separate locally informative, near-random, and deceptive regimes. The protocol can therefore select uniform sampling as a successful outcome rather than treating it as a failed optimizer.

Algorithm 1: Elitist permutation executor

Input: fitness F, move o, population size m, distinct-evaluation budget B
Output: best evaluated ordering

Initialize m random permutations and evaluate unseen candidates
while *fewer than B distinct orderings have been evaluated* **do**
    select a parent by tournament; propose $\pi' = o(\pi)$
    evaluate π' only if its cache key is new
    replace the worst non-elite member; retain the best-so-far ordering
return best cached ordering

4 Related Work

Landscape analysis and permutation search. One-step autocorrelation and fitness–distance correlation are classical measures of local preservation and global guidance [1, 2]. Permutation landscape analysis further connects distance metrics and move operators to position, precedence, and adjacency structure [3]; convergence under irreducible mutation and elitist retention is standard [4]. PRISM does not claim these ingredients as new. Its contribution is to operationalize them as a low-budget, recorded selector among enumeration, sampling, evolutionary search, and a structure-matched surrogate, then test the forecast on exact and live application landscapes.

Prompt order, position, and calibration. Lu et al. show that reordering few-shot demonstrations can produce large performance differences and that effective orders need not transfer across models [5]. Zhao et al. reduce prompt-induced instability through contextual calibration [6]. These studies motivate robustness interventions; PRISM instead asks whether an ordering surface has enough measurable structure to justify search. Lost-in-the-middle effects concern the placement of retrieved evidence in long contexts [7], while Sclar et al. quantify sensitivity to formatting choices [8]. The present experiments hold component wording and rendering fixed and vary only the order of instruction modules.

Option-order studies expose related positional bias in multiple-choice evaluation. Reordering answer options can substantially change measured performance [9], and selection bias toward option identifiers motivates explicit debiasing methods [10]. Those works permute answer choices; PRISM permutes the internal order of a fixed reasoning scaffold and characterizes the full six-component landscape. Recent causal attention and presentation-order accounts provide candidate mechanisms for why later information may be used differently [11, 12], but they do not determine whether a given ordering landscape is searchable.

Prompt optimization. Chain-of-thought and self-consistency alter elicited computation [13, 14]; APE, OPRO, and DSPy optimize prompt text or programs [15–17]. PRISM freezes component content and isolates order as a separate coordinate. The OPRO experiment therefore tests complementarity rather than claiming that ordering replaces content optimization.

5 Landscapes, Operators, and Locality

Search operators define what "nearby" means. Absolute-position landscapes reward elements in specific slots and align with swap moves. Precedence landscapes reward pairwise order and align with insertion. Adjacency landscapes reward neighbouring pairs and align with inversion. Scramble is a broad disruption rather than a precise neighbourhood. These types are diagnostic ideals, not a claim that real applications are pure. The three operator-landscape pairings are illustrated in Figure 3.

The difference is operationally large. At $n = 16$, the matched operator solves the absolute-position objective in 405 generations, the precedence objective in 396, and the adjacency objective in 1,199, while mismatched operators often reach the 10,000-generation cap without an optimum. We call such a run *censored*: it exhausts its cap without success, and it remains in the denominator rather than being discarded. When the type is unknown, a uniform four-operator portfolio removes censoring on these typed landscapes at an observed 1.1–4.1× cost.

The pre-flight automates the choice. With 100 move pairs per operator, the pair-sampled ρ_1 selector identifies the matched move with mean accuracy 0.925 across typed synthetic and enumerated language-model landscapes. Representation can matter even more than population tuning: on a 5,040-order precedence landscape, a ridge surrogate using pairwise precedence features reaches the unique optimum in 14.8 evaluations, bootstrap 95% interval [14,15], versus 54.2 [50,59] for matched elitist search.

Takeaway

Search operators define locality. A mismatched move can prevent success rather than merely slow it, while a representation aligned with the measured structure can outperform population search.

6 Ground Truth and the Falsification

Search evaluation is circular when the optimizer's best observation is also treated as the answer key. Repeated stochastic queries make the maximum optimistically biased, and a method that explores more candidates can

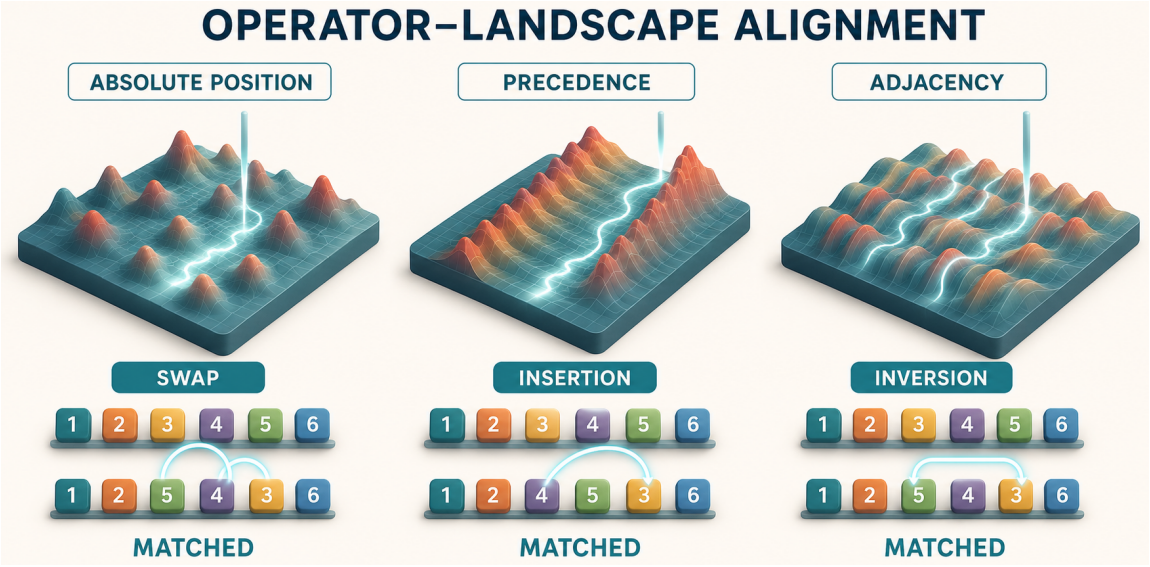


Figure 3: Operator-to-landscape alignment. The conceptual illustration pairs absolute position with swap, precedence with insertion, and adjacency with inversion. It is a schematic, not a measured surface; the observed censoring and locality statistics are reported in the text and Figure 4.

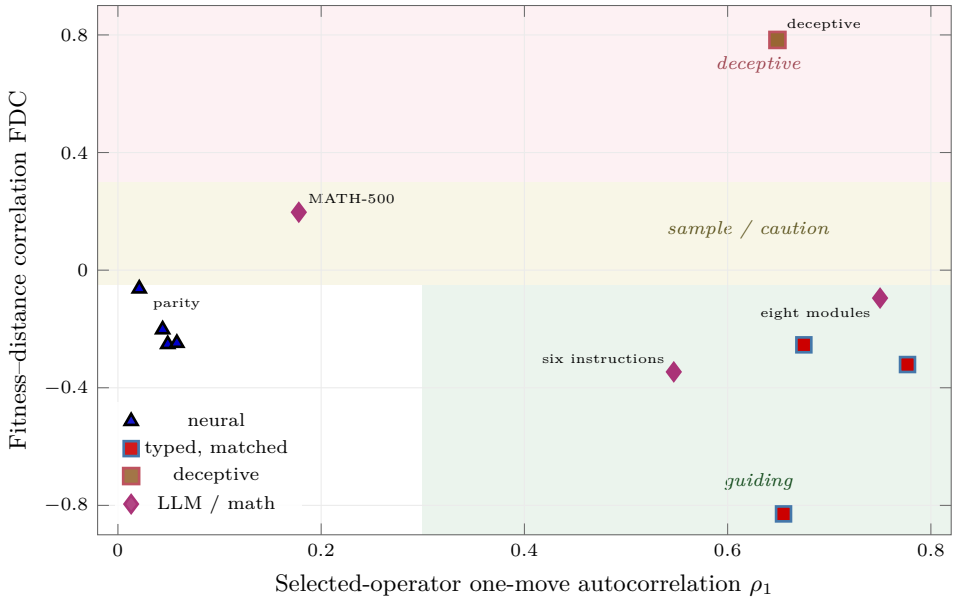


Figure 4: The two pre-flight axes separate guiding, near-random, and deceptive regimes. The four neural, three typed synthetic, and one deceptive point use exact FDC and the largest ρ_1 among swap, insertion, and inversion; scramble is excluded because identity moves can inflate its correlation. Three later language-model and mathematics probes use the same axes with frozen-sample approximations where exact enumeration is unavailable. The shaded regions summarize the conservative rule in Table 1; optimum density and evaluator resolution remain additional gates.

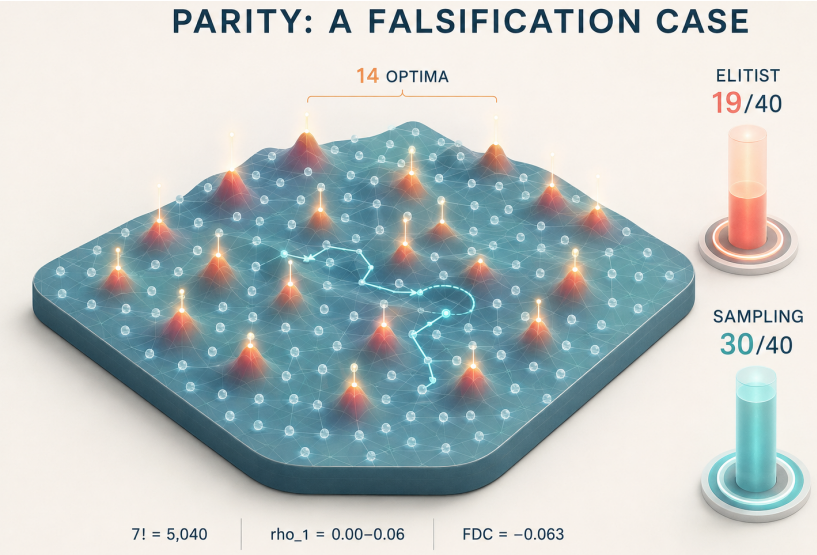


Figure 5: Parity is the central falsification case. The schematic marks 14 optima among 5,040 orderings and audited hit counts of 19/40 for elitist search versus 30/40 for sampling. Wilson score 95% intervals are [0.33,0.63] and [0.60,0.86].

redefine the apparent target in its own favour. We therefore enumerate every ordering whenever feasible, freeze sampled pools otherwise, and compare methods at equal distinct-evaluation budgets. Exact landscapes report the optimum set, its density, hit probability, and regret rather than distance to a method-dependent best observation. The complete parity counterexample and its audited hit counts are summarized in Figure 5.

The decisive test is a deterministic neural parity landscape with seven fixed modules. It contains 14 optima among $7! = 5{,}040$ orderings. The study was intended to show that scale-aware tuning would preserve an evolutionary advantage. It did not. Default elitist search found an optimum in 4 of 15 exploratory runs. Larger populations and stronger mutation removed population collapse but required 384 mean evaluations, versus 318 for uniform sampling.

The failure is structural rather than a tuning anecdote. Swap, insertion, and inversion have ρ_1 between 0.00 and 0.06, and exact FDC $= -0.063$. One local move therefore carries almost no information about the next candidate, and global distance provides little additional guidance. Aging and surrogate variants raise coverage, but no tested structured method establishes a decisive advantage over the uninformed baseline. The pre-flight predicts this boundary before the search comparison is run.

This negative result changed the research claim. The supported statement is not "evolution beats random"; it is that the benefit of structured permutation search is conditional on measurable landscape structure. A correct forecast that sampling should be competitive saves budget and is a success for the protocol.

Takeaway

Structured search is not universally better than random sampling. On a landscape with almost no local signal, the honest baseline wins, and the pre-flight identifies that condition in advance.

7 The Instruction-Ordering Landscape

The flagship landscape freezes six one-sentence modules—RESTATE, IDENTIFY, PLAN, COMPUTE, CHECK, and ANSWER—and changes only their order. Each permutation is rendered as numbered steps before the same 32-question GSM8K subset [18]. Gemini 2.5 Flash-Lite is queried with deterministic decoding; answers are scored by a frozen numeric extractor; every ordering–question response is cached. All $6! = 720$ orderings are evaluated.

Accuracy ranges from 0.063 to 0.969, with mean 0.717 and standard deviation 0.230. The effect is not merely a few exceptional prompts: marginal position averages in Figure 6 reveal a coherent mechanism.

Every one of the three worst orderings places ANSWER before COMPUTE; the best examples reverse that precedence. This is consistent with the idea that presentation order should respect the order in which information is used [12], while not identifying a unique causal mechanism. The pre-flight selects insertion: $\rho_1 = 0.547$, compared with 0.411 for swap and 0.297 for inversion. Exact FDC $= -0.346$ confirms guiding structure.

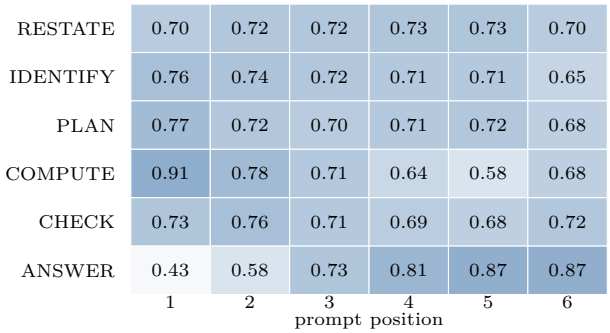
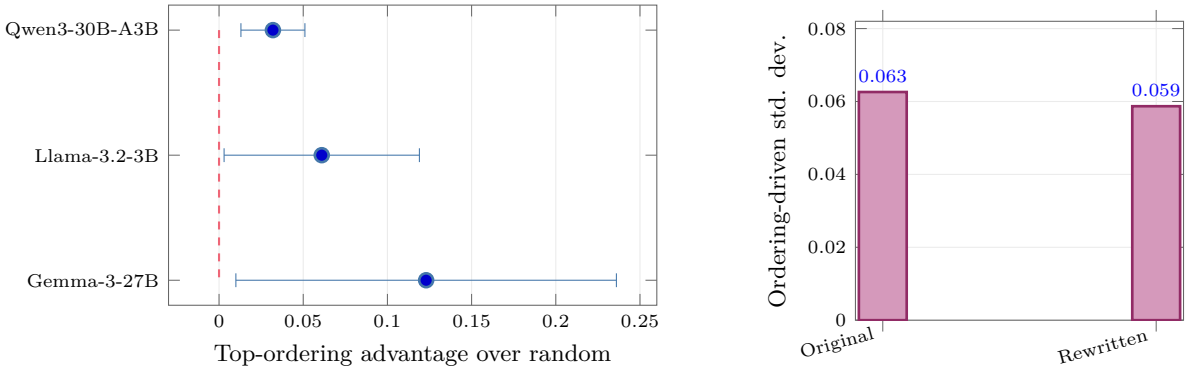


Figure 6: ANSWER is favoured late and COMPUTE early. Each cell is mean accuracy over all 720 orderings in which the named module occupies the given position. ANSWER rises from 0.435 in the first position to 0.870 in the last; COMPUTE falls from 0.908 first to 0.585 fifth.



(a) Cross-family transfer with 95% intervals.

(b) Variation survives OPRO rewriting.

Figure 7: Ordering structure transfers and is complementary to wording optimization. Left: source top orderings beat random for Qwen [+0.013,+0.051], Llama [+0.003,+0.118], and Gemma [+0.008,+0.234]. Right: after 24 OPRO rewriting rounds, ordering-driven variation retains 94% of its original standard deviation.

Guidance does not guarantee a large optimizer advantage. Sixty orderings, 8.3% of the space, attain the maximum measured accuracy. Across 40 replay runs, evolutionary search reaches an optimum in 9.9 mean distinct evaluations, bootstrap percentile 95% interval [7,13], versus 10.8 [8,14] for uniform sampling. The intervals overlap. An earlier 15-run estimate suggested a large speedup; the higher-powered audit withdrew that claim.

Two additional modules expand the space to $8! = 40{,}320$. A probe of 222 unbiased orderings on 20 questions finds accuracy from 0.10 to 1.00 and selects insertion at $\rho_1 = 0.75$, but approximate FDC $= -0.095$ forecasts that sampling will be competitive. Evolutionary search and uniform sampling both reach measured accuracy 1.0 at budgets 30, 60, and 120 in all four replay runs. At this resolution, one question changes accuracy by 0.05 and perfect ties are dense; the protocol recommends a modest sample or a better evaluator rather than a more elaborate searcher.

Takeaway

A fixed set of six reasoning instructions spans 90.6 accuracy points under order alone, with interpretable position effects. Dense optima nevertheless make uniform sampling competitive.

8 Transfer and Baselines

The exact six-module landscape supports transfer at three resolutions. First, a score constructed from its module–position table correlates with fitness on the 222 unbiased eight-module orderings at Spearman $\rho = 0.665$, bootstrap 95% interval [0.582,0.733]. Second, source-selected top orderings are tested on fixed pools for Gemma-3-27B, Qwen3-30B-A3B, and Llama-3.2-3B. Fine rank transfer varies, but the practical top-ordering advantage is positive on all three targets, with bootstrap intervals excluding zero. Third, on a filtered MATH-500 hard-reasoning pool [19], the easy-task position score correlates with target fitness at $r = 0.432$, while the pre-flight's random-competitive forecast is confirmed by overlapping search intervals. Figure 7 reports both the cross-family effects and the persistence of ordering variation after wording optimization.

The wording baseline directly tests the objection that order sensitivity is only a symptom of poor instruction text. OPRO raises held-out accuracy by +0.032, paired bootstrap 95% interval [+0.013,+0.052], yet the

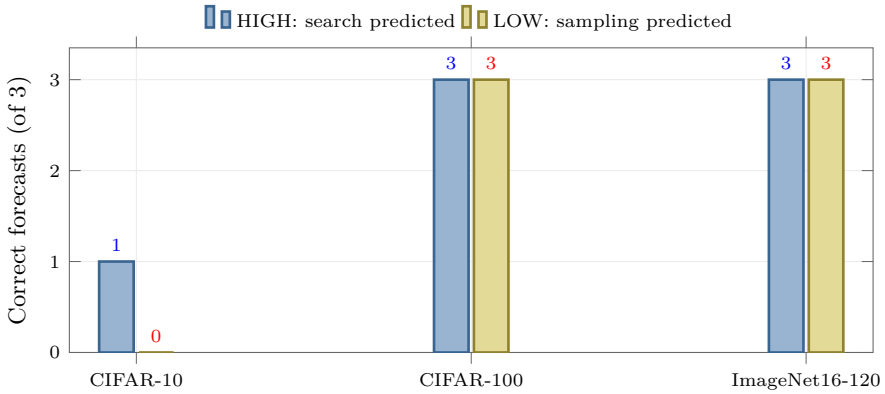


Figure 8: Registered NAS forecast calibration by dataset and stratum. Each bar summarizes three fixed-operation slices. Thirteen of 18 forecasts are correct. CIFAR-100 and ImageNet16-120 are correct in all cells; the CIFAR-10 LOW misses are conservative under-calls because search wins.

ordering standard deviation changes only from 0.0626 to 0.0587. Content and order are therefore separate optimization coordinates.

We also compare OPRO with the evolutionary executor on the same 25-evaluation ordering budget, using five shared runs. Final best-found accuracy is 0.500 [0.467,0.527] for evolution and 0.507 [0.480,0.540] for OPRO; their difference is −0.007 [−0.053,+0.040]. The comparison is a tie at the tested budget, consistent with the near-zero-guidance forecast. The point is not that the methods are identical, but that the pre-flight correctly avoids promising a directional win.

Takeaway

Ordering is a separate optimization axis from wording. Rewriting improves the average prompt but does not flatten the ordering surface, while optimizer differences vanish in the forecast near-random regime.

9 Cross-Domain Calibration

The protocol is also tested outside prompting. A scientific machine learning suite based on sparse identification of nonlinear dynamics [20] permutes six preprocessing operations for sparse identification of nonlinear dynamics. All 720 orderings are enumerated for each of six systems. Ordering changes fitness in every system; insertion has the largest ρ_1 in all six; and the evolutionary executor has more successful runs than uniform sampling in all six, with pronounced advantages in four. The main miss is diagnostic: Cayley distance is aligned with swaps and underestimates guidance on several insertion-shaped pipeline landscapes.

A tabular neural architecture search study using a compact NAS-Bench-201 backend [21] registers 18 HIGH/LOW forecasts on fixed-operation multiset slices before search. Thirteen are correct. CIFAR-100 and ImageNet16-120 are each 6/6; all five misses occur on CIFAR-10-valid, and the three LOW misses are conservative because search wins despite the cautious forecast. These studies test calibration, not a claim that PRISM is a new pipeline learner or architecture-search algorithm. The registered architecture forecasts are broken down by dataset and stratum in Figure 8.

A *forecast ledger* pairs every recorded call with its outcome, including corrections, ties, and null results. The complete numerical record appears in Table 5; retaining misses is what exposes the current distance-metric limitation rather than converting it into a post hoc success.

Across the full program, recorded external-model inference expenditure remains below US$20, and the synthetic, neural, scientific-pipeline, and tabular architecture studies use CPU execution. Exact tables, frozen pools, caches, and registered forecasts are retained so a wrong call remains auditable rather than being rewritten as a successful post hoc explanation.

10 Limitations

Exact neural ground truth stops at seven modules and exact language-model ground truth at six; the eight-module study observes 222 of 40,320 orderings. The language evidence covers two mathematical reasoning benchmarks and frozen model snapshots, not general generation, retrieval, coding, or future model versions. Fixed question pools and graders also bound resolution: with 20 binary questions, one answer moves fitness by 0.05 and produces dense ties.

Table 2: Final summary of the paper's key findings, limitations, and recommended actions.

Focus	Key finding	Takeaway or action
Searchability	Space size does not predict difficulty; measured locality and guidance do	Run the PRISM pre-flight before committing the main budget
Executor selection	No executor dominates across guiding, near-random, and deceptive regimes	Select enumeration, sampling, evolution, a surrogate, or transfer conditionally
Exact falsification	On parity, elitist search hits 19/40 optima versus 30/40 for sampling	Treat a correct sampling forecast as a successful result
Instruction order	Six fixed modules span 6.3%–96.9% accuracy and show coherent position effects	Treat ordering as a distinct optimization coordinate
Transfer and baselines	Position structure transfers imperfectly and survives wording optimization	Revalidate rankings on each target and optimize wording and order separately
Cross-domain calibration	Scientific pipelines and NAS slices reveal both correct calls and distance-metric misses	Retain a forecast ledger and revise diagnostics from failures
Scope	Evidence is concentrated at small n, fixed evaluators, and bounded task families	Replicate at larger scale with aligned metrics, uncertainty, and safety constraints

Exact FDC requires known optima. Sampled studies use the best frozen probe observation, which can miss the relevant basin, while swap-aligned Cayley distance conservatively under-calls some insertion-shaped pipelines. Precedence-aligned distance remains future work. The OPRO comparison has five runs and supports a tie rather than fine equivalence.

Takeaway

The evidence supports calibrated decisions on measured landscapes, not universal optimizer superiority or guaranteed transfer and calibration at larger n.

11 Conclusion

Permutation-space difficulty follows structure rather than cardinality: a 5,040-order parity landscape can be harder for elitist search than a matched synthetic space with roughly twenty trillion orderings. PRISM therefore measures locality, global guidance, score variance, and tie density before allocating budget. Its exhaustive instruction map spans 6.3%–96.9% accuracy, yet dense optima and weak guidance still make uniform sampling the correct executor.

Recommendations. Define legal orderings, enumerate small spaces, verify evaluator resolution, align distance with the move, and compare executors under one distinct-evaluation budget. Register the forecast first and retain ties, censored runs, null results, and misses.

Future work and further research. Priorities are precedence- and adjacency-aligned distances, uncertainty-aware probes, adaptive probe allocation, scalable surrogates, and independent replication across tasks and model families. Language-agent pipelines, compiler and database plans, scientific workflows, and laboratory automation offer low-cost replay; clinical, security, financial, and autonomous applications require safety constraints and prospective human oversight. The 24-domain roadmap in Table 3 separates opportunities from validated evidence.

Takeaway

PRISM is a falsifiable protocol for deciding when to search, when to sample, and when to improve the evaluator first; its value lies in calibrated choice rather than universal optimizer superiority.

12 Applications and Experimental Roadmap

The protocol applies to fixed component sets whose order can change without changing their content. Table 3 maps 24 problem domains where ordering may affect accuracy, cost, latency, safety, robustness, or resource use. Each study must define the legal space, verify score resolution, register the forecast, and compare with sampling and the strongest domain baseline; the entries are testable opportunities rather than claims of validated gains.

Table 3: Twenty-four problem domains in which PRISM can test ordering effects. Each entry defines a legal permutation unit and an impact objective; it identifies a testable opportunity rather than a validated win. New studies still require evaluator-resolution, safety, pre-flight, and equal-budget baseline checks.

Domain	Permuted unit and impact objective
Language-model prompting	Reasoning or tool instructions; accuracy, robustness, token use
Agent workflows	Retrieval, planning, action, checking; success, calls, latency
Compiler optimization	Legal passes; runtime, code size, energy
Neural architecture	Fixed blocks or edge operations; accuracy and efficiency
Scientific discovery	Preprocessing and identification stages; recovery and stability
Data engineering	Cleaning, joining, transformation stages; quality and cost
Clinical diagnostics	Test, triage, and review steps; safety, time, accuracy
Genomics	Quality control, alignment, and variant calling; sensitivity and runtime
Drug discovery	Screening, synthesis, and assay stages; yield and experimental cost
Manufacturing	Process operations; yield, defects, energy
Robotics	Perception, planning, and control modules; success and safety
Reinforcement learning	Options or curriculum tasks; sample efficiency and return
Cybersecurity response	Detection, containment, recovery steps; risk and response time
Network service chains	Routing, filtering, inspection functions; throughput and latency
Cloud deployment	Build, test, deploy, migrate stages; downtime and cost
Database execution	Joins, filters, and aggregations; latency and resource use
Image processing	Denoising, normalization, segmentation; quality and accuracy
Signal processing	Filters and transforms; signal quality and latency
Remote sensing	Correction, fusion, classification stages; mapping accuracy
Financial risk	Cleaning, modelling, stress testing, reporting; stability and compliance
Supply chains	Sourcing, stocking, routing steps; cost and service level
Maintenance	Inspection, diagnosis, repair actions; reliability and downtime
Education	Lessons, examples, and exercises; mastery and retention
Laboratory automation	Preparation, mixing, measurement, quality control; reproducibility and time

Takeaway

High-impact use requires a legal ordering space, a resolvable evaluator, a registered forecast, and equal-budget domain baselines.

Acknowledgments. The author gratefully acknowledges God for guidance and strength, and thanks Jake Beck for reviewing the manuscript and providing constructive feedback. This work was conducted within the ML Collective research community. The author retains responsibility for the analyses and conclusions.

References

[1] Edward Weinberger. "Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference". In: *Biological Cybernetics* 63 (1990), pp. 325–336. DOI: 10.1007/BF00202749.

[2] Terry Jones and Stephanie Forrest. "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms". In: *Proceedings of the Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1995, pp. 184–192.

[3] Vincent A. Cicirello. "On Fitness Landscape Analysis of Permutation Problems: From Distance Metrics to Mutation Operator Selection". In: *Applied Sciences* 12.19 (2022), p. 9976. DOI: 10.3390/app12199976. arXiv: 2208.11188 [cs.NE].

[4] Günter Rudolph. "Convergence Analysis of Canonical Genetic Algorithms". In: *IEEE Transactions on Neural Networks* 5.1 (1994), pp. 96–101. DOI: 10.1109/72.265964.

[5] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2022, pp. 8086–8098. DOI: 10.18653/v1/2022.acl-long.556.

[6] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. "Calibrate Before Use: Improving Few-Shot Performance of Language Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 12697–12706. URL: https://proceedings.mlr.press/v139/zhao21c.html.

[7] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. "Lost in the Middle: How Language Models Use Long Contexts". In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 157–173. DOI: 10.1162/tacl_a_00638.

[8] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I Learned to Start Worrying about Prompt Formatting". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=RIu5lyNXjT.

[9] Pouya Pezeshkpour and Estevam Hruschka. "Large Language Models Sensitivity to The Order of Options in Multiple-Choice Questions". In: *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics, 2024, pp. 2006–2017. DOI: 10.18653/v1/2024.findings-naacl.130.

[10] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. "Large Language Models Are Not Robust Multiple Choice Selectors". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=shr9PXz7T0.

[11] Hyunjong Ok and Jaeho Lee. *Lost in the Prompt Order: Revealing the Limitations of Causal Attention in Language Models*. 2026. arXiv: 2601.14152 [cs.CL].

[12] Yue Zhou, Henry Peng Zou, Barbara Di Eugenio, and Yang Zhang. "The Curse of Verbalization: How Presentation Order Constrains LLM Reasoning". In: *Findings of the Association for Computational Linguistics: EACL 2026*. Association for Computational Linguistics, 2026, pp. 4175–4185. DOI: 10.18653/v1/2026.findings-eacl.218.

[13] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2022. arXiv: 2201.11903 [cs.CL].

[14] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. 2022. arXiv: 2203.11171 [cs.CL].

[15] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. *Large Language Models Are Human-Level Prompt Engineers*. 2022. arXiv: 2211.01910 [cs.LG].

[16] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. *Large Language Models as Optimizers*. 2023. arXiv: 2309.03409 [cs.LG].

[17] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. *DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines*. 2023. arXiv: 2310.03714 [cs.CL].

[18] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Rei Nakano, Christopher Hesse, and John Schulman. *Training Verifiers to Solve Math Word Problems*. 2021. arXiv: 2110.14168 [cs.LG].

[19] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. *Measuring Mathematical Problem Solving With the MATH Dataset*. 2021. arXiv: 2103.03874 [cs.LG].

[20] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. "Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems". In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937. DOI: 10.1073/pnas.1517384113.

[21] Xuanyi Dong and Yi Yang. *NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search*. 2020. arXiv: 2001.00326 [cs.CV].

Figure 9: Evolutionary executor loop. A population is evaluated, its best member is retained, tournament selection chooses parents, and the selected mutation operator produces the next generation. The convergence statements below apply to this executor, not to enumeration or uniform sampling.

Table 4: Headline comparisons and uncertainty statements retained in the short paper. Bootstrap entries are percentile intervals over the archived replay units; hit-rate entries use Wilson score intervals for binomial proportions.

Comparison	Method A	Method B	Interpretation
Parity optimum hits, 40 runs	elitist 19/40 [0.33,0.63]	random 30/40 [0.60,0.86]	no evolutionary advantage
Six-instruction evaluations to optimum	evolution 9.9 [7,13]	random 10.8 [8,14]	intervals overlap
MATH-500 final best-found accuracy	guided 0.407 [0.367,0.447]	random 0.397 [0.367,0.433]	random-competitive call holds
OPRO ordering search	evolution 0.500 [0.467,0.527]	OPRO 0.507 [0.480,0.540]	difference spans zero
Wording level change	+0.032 [0.013,0.052]	SD ratio 0.94	wording helps; ordering remains

A Executor Guarantees and Scope

The protocol's empirical claims do not depend on a novel convergence theorem. The evolutionary executor inherits the standard reachability argument for elitist evolutionary algorithms with an irreducible mutation kernel [4]. The executor cycle to which the following guarantees apply is shown in Figure 9.

Proposition A.1 (Reachability)**.** *Let $\mathbb{S}_n$ be finite. If mutation has positive probability of proposing every transposition and selection can choose every current population member with positive probability, then every permutation is reachable from every population state in finitely many steps with positive probability.*

Proof. Every permutation can be transformed into every other by a finite sequence of transpositions. The stated mutation and selection conditions assign positive probability to each step of such a sequence. Their finite product is positive. □

Theorem A.2 (Almost-sure optimum discovery)**.** *Under the conditions of Theorem A.1, an elitist executor run without a finite budget discovers a global optimum with probability one and never loses the best observed fitness.*

Proof. From any non-optimal state there is a finite positive-probability path to an optimum. Finiteness gives a positive lower bound on reaching an optimum over sufficiently long blocks. The probability of avoiding all such blocks tends to zero. Elitism preserves the best fitness after discovery. □

The guarantee is qualitative. A direct worst-case bound scales with the number of permutations and is therefore factorial; it does not explain the empirical $n^3 \log n$-shaped fits observed on matched synthetic landscapes through $n = 16$. Those fits are conditional evidence about particular structures and moves, not a universal runtime theorem.

For aging replacement, the incumbent optimum can leave the active population. Nevertheless, the best-ever record still converges almost surely under the same reachability conditions. This is why the paper distinguishes active-population convergence from record convergence and makes all finite-budget claims empirical.

Table 5: Numerical forecast ledger. Correct, mixed, tie, and null outcomes remain in the record.

Study	Recorded call	Outcome	Verdict
Six instructions	insertion; guiding structure	insertion signal; search/sampling intervals overlap	scoped
Eight modules	sampling competitive	both methods reach 1.0 at every budget	correct
Cross-family	source top orderings transfer	positive effect on all three targets; exact ranks vary	mixed
MATH-500	sampling competitive	guided, evolutionary, and random intervals overlap	correct
Wording and search	order survives rewriting; search comparison uncertain	SD ratio 0.94; final difference spans zero	correct
Pipelines and NAS	cautious pre-search calls	pipeline distance under-calls; NAS 13/18	partial

B Additional Statistical Record

B.1 Forecast ledger

B.2 Selector reliability

At equal pre-flight budgets, pair-sampled ρ_1 is more reliable than random-walk autocorrelation or negative mean absolute fitness change. Averaged across typed synthetic and the enumerated instruction landscape over 200 resamples per cell, selector accuracies are:

Estimator	$B = 50$	$B = 100$	$B = 500$
Pair-sampled ρ_1	**0.831**	**0.925**	**0.998**
Random-walk ρ_1	0.734	0.848	0.992
Negative mean absolute change	0.827	0.915	0.991

B.3 Pipeline and architecture-search details

The six pipeline systems comprise damped and cubic oscillators, Van der Pol, Lotka–Volterra, Lorenz-63, and Rössler dynamics. Fitness ranges under order alone are 0.40–1.00, 0.70–1.00, 0.31–0.88, 0.00–1.00, 0.89–0.99, and 0.37–0.81, respectively. Evolutionary-versus-random hit counts over 40 runs are 21/16, 34/30, 40/20, 40/37, 40/23, and 40/14.

The architecture-search forecast matrix contains three HIGH and three LOW slices per dataset. CIFAR-100 and ImageNet16-120 are correct in all six cells. On CIFAR-10-valid, one HIGH call is correct, two HIGH calls miss, and all three LOW calls are conservative misses because search wins. Thus 13 of 18 registered forecasts are correct, and every LOW miss errs toward underusing rather than overselling search.

B.4 Completed experiment coverage

Table 6 maps the complete research program into experiment families. The short main text emphasizes the results that determine the current claim; the archive retains per-run data, negative redesigns, corrections, and implementation details.

C Reproducibility, Resource Accounting, and Data Schema

All externally hosted model evaluations use frozen question pools, deterministic decoding where supported, canonical ordering identifiers, and append-only response records. Repeated orderings are resolved from the archived record and do not count as new distinct evaluations. Recorded external-model inference expenditure across the program remains below US$20; the synthetic, neural, scientific-pipeline, reinforcement-learning, and tabular architecture studies use CPU execution.

The data schema preserves one row per unique ordering with a stable study identifier, canonical ordering key, component list, fitness statistic, sample provenance, evaluator version, exact-space status, and resource record. Derived neighbourhood tables retain parent and child orderings, the move operator, both scores, and the sampling seed; distance tables declare the metric and whether the target is an exact optimum or a frozen proxy.

The canonical project repository, including the paper source, experiment summaries, data manifests, and regeneration instructions, is available at `https://github.com/bleymambwe/PRISM`.

Table 6: Coverage of completed experiment families through August 3, 2026.

Experiment family	Included studies	Retained conclusion
Reproduction and benchmark repair	Historical neural toys; failed deep-stack redesign; residual, seeded v4 benchmarks	Headlines reproduce with corrections; deterministic exact ground truth replaces noisy best-ever fitness.
Operator and landscape science	Typed synthetic landscapes; deceptive objective; uniform and adaptive portfolios	Operator matching is necessary on plateau-rich landscapes; portfolio removes configuration failure at bounded observed cost; deception remains a limit.
Exact neural search tests	Enumerated XOR and parity landscapes from 120 to 5,040 orderings	Exact hit rate and regret expose both success and the pivotal random-sampling falsification.
Diagnosis and method selection	Locality study; aging and restart replacement; precedence surrogate; 40-seed audit; ρ_1 ablation	Measurements select operators, replacement, and representation; pair-sampled ρ_1 is the most reliable tested low-budget selector.
Language-model ordering	Exhaustive and sampled instruction landscapes; size and family transfer; guided evaluation; MATH-500; OPRO rewriting and search	Ordering has a large structured effect, transfers imperfectly, survives wording optimization, and does not guarantee a search-speed win.
Cross-domain calibration	Six-system scientific-pipeline suite; 18 tabular NAS forecasts; two lightweight RL ordering pilots	Scientific and NAS studies calibrate the protocol beyond prompting; RL shows preliminary executor transfer but has not yet passed a registered pre-flight.

Table 7: Resource accounting for externally hosted language-model studies. Expenditure is reported for reproducibility and is not used as a performance metric.

Study surface	Evaluation coverage	Observed cost	Reproducibility record
Six instruction modules	$720 \times 32 \approx 23{,}000$ responses	US$3–5	exhaustive ordering-by-question table
Eight instruction modules	23,805 model evaluations	US$5.56	fixed sampled pool with per-response records
Cross-family transfer	11,955 model evaluations	US$0.66	shared prompts, orderings, and question set across three families
MATH-500	31,998 model evaluations	US$4.43	pilot-selected model, frozen pool, and registered forecast
Wording and search baselines	12,724 target evaluations	US$2.68	held-out scoring and shared evaluation budgets