From AI Assistant to AI Scientist: Autonomous Discovery of LLM-RL Algorithms with LLM Agents

Anonymous ACL submission

Abstract

Discovering improved policy optimization algorithms for language models remains a costly manual process requiring repeated mechanism-level modification and validation. Unlike simple combinatorial code search, this problem requires searching over algorithmic mechanisms tightly coupled with training dynamics while reusing empirical evidence across iterations. We propose POISE, a closed-loop framework for automated discovery of policy optimization algorithms for language models. POISE maintains a structured, genealogically linked archive linking proposals, executable implementations, standardized evaluations, and natural-language reflections to support evidence-driven iteration. In mathematical reasoning experiments starting from GRPO, POISE evaluates 64 candidate algorithms and discovers improved mechanisms, including analytic-variance scaling and validity masking. The best variant improves weighted Overall from 47.8 to 52.5 (+4.6) and increases AIME25 `pass@32` from 26.7% to 43.3%, demonstrating the feasibility of automated policy optimization discovery while supporting interpretable design principles.

1 Introduction

Reinforcement Learning (RL) has become an important post-training paradigm for Large Language Models (LLMs) (Ouyang et al., 2022; Shao et al., 2024). However, discovering improved policy optimization algorithms for language models remains a costly manual process. Designing an effective policy optimization algorithm involves more than hyperparameter tuning: researchers repeatedly modify and validate core mechanisms, including loss design, advantage estimation, and regularization. Each new hypothesis requires time-consuming, resource-intensive training and evaluation, burdening researchers and demanding substantial computational resources. LLMs have shown advanced reasoning and planning capabilities (Yang et al., 2025; Liu et al., 2025a; Erdogan et al.), and recent studies have begun developing LLM-driven systems for automated research (Lu et al., 2024; Schmidgall et al., 2025; Schmidgall and Moor, 2025). This raises a question: can LLMs discover new policy optimization algorithms for language models?

Although automated research has made remarkable progress in tasks such as machine-learning research automation (Liu et al., 2025c), program optimization (Fawzi and Paredes, 2023), and model architecture discovery (Liu et al., 2025b), discovering new policy optimization algorithms for language models still presents unique challenges. Unlike tasks framed as combinatorial searches over discrete program components, discovering policy optimization algorithms for language models requires searching over mechanisms tightly coupled with training dynamics. For example, small design choices, such as introducing a clipping objective as in PPO (Schulman et al., 2017), adding a KL penalty, or modifying advantage normalization, can substantially alter optimization behavior and training stability. Moreover, discovering a new algorithm requires more than proposing an implementation: it also requires identifying which mechanisms contribute to performance gains, which fail under empirical evaluation, and how accumulated evidence should inform the next testable hypothesis. This calls for a search process that systematically converts empirical feedback into reusable evidence and carries that evidence across iterations.

To address this challenge, we propose POISE (**P**olicy **O**ptimization through **I**terative **S**earch and **E**vidence), a closed-loop framework for the automated discovery of policy optimization algorithms for language models. Within POISE, algorithm discovery is carried out as Epistemic Evolutionary Search over an algorithm space with a reflection-augmented evolutionary solver. The framework maintains a structured archive with genealogical links that associates each algorithm proposal with

its executable implementation, standardized evaluation outcomes, and natural-language reflections, allowing later proposals to build on accumulated evidence instead of isolated trial-and-error. In mathematical reasoning experiments starting from a GRPO baseline (Shao et al., 2024), POISE evaluates 64 algorithms and discovers improved policy optimization mechanisms, including analytic-variance scaling and validity masking. The best variant improves weighted Overall from 47.8 to 52.5 (+4.6) and increases AIME25 `pass@32` from 26.7% to 43.3%, while the resulting lineages yield interpretable design principles.

Our main contributions are as follows:

- **Closed-Loop Discovery Framework:** We propose POISE, a closed-loop framework for automated discovery of policy optimization algorithms for language models. It combines reflection augmentation with a structured, genealogically linked archive supporting evidence-driven iteration from proposal generation to implementation, verification, evaluation, and archive update.
- **Empirical Validation of Automated Algorithm Discovery:** Starting from GRPO, POISE evaluates 64 candidate algorithms and discovers mechanisms that yield substantial gains on mathematical reasoning benchmarks. These results demonstrate the feasibility of automated policy optimization discovery for language models.
- **Interpretable Design Principles:** Beyond discovering stronger algorithms, our analysis extracts design principles from lineage evolution, including signal decoupling, conditional normalization, and correctness-first efficiency shaping. These principles clarify how and when such mechanisms help in sparse-reward language model policy optimization.

2 Related Work

LLM-driven Scientific Discovery leverages the knowledge, reasoning, and execution capabilities of Large Language Models (LLMs) to automate parts of the research process. This paradigm has evolved from basic experimental automation (Boiko et al., 2023) to complex algorithm design (Fawzi and Paredes, 2023; Shi et al.). More recently, "AI Scientist" agents have emerged, aiming for end-to-end pipelines that encompass literature review, coding, and experimentation (Lu et al., 2024; Yamada et al., 2025; Schmidgall et al., 2025; Gottweis et al., 2025; Mitchener et al., 2025). Progress has also been made in quality control for idea generation (Si et al.; Li et al., 2024; Keya et al., 2025). Furthermore, LLMs have demonstrated effective exploration within well-defined search spaces across domains like program optimization (Novikov et al., 2025), theorem proving (Trinh et al., 2024; Chervonyi et al., 2025), architecture search (Liu et al., 2025b; Cheng et al., 2025), and automated machine learning engineering (Chan et al.; Jiang et al., 2025; Liu et al., 2025c). However, extending automated discovery to RL algorithm design for language model policy optimization is more challenging because the search space is compositional, feedback is delayed and stochastic, and verification requires expensive distributed training runs. To address this, we propose POISE, an iterative evolutionary system guiding LLMs to design, verify, and refine policy-optimization algorithms for language models.

Policy Optimization for Large Language Models aims to enhance reasoning capabilities by training LLMs with algorithms like PPO (Schulman et al., 2017) and its computationally efficient variant GRPO (Shao et al., 2024). Recent studies have enhanced these baselines by refining specific algorithmic components, such as decoupled clipping in DAPO (Yu et al., 2025), soft adaptive gating in SAPO (Gao et al., 2025), asymmetric weighting in ASPO (Wang et al., 2025), and geometric mean aggregation in GMPO (Zhao et al., 2025). While these studies demonstrate the value of targeted component-level refinements, exploring the combinatorial design space of such mechanisms still largely relies on manual iteration. Unlike prior work, which typically relies on manually refining individual algorithmic components, our method introduces an LLM-driven evolutionary system to autonomously discover and compose algorithmic improvements under standardized settings.

3 Method

3.1 Problem Modeling and Methodology

We formulate automated discovery of policy optimization algorithms for language models as search over a space of feasible reinforcement learning algorithms. Let $\mathcal{Z}$ denote the space of feasible algorithms supported by our framework, where each algorithm $z \in \mathcal{Z}$ specifies a complete training procedure through the composition of algorithmic components (*e.g.*, loss functions, advantage estimators) and their associated hyperparameters.

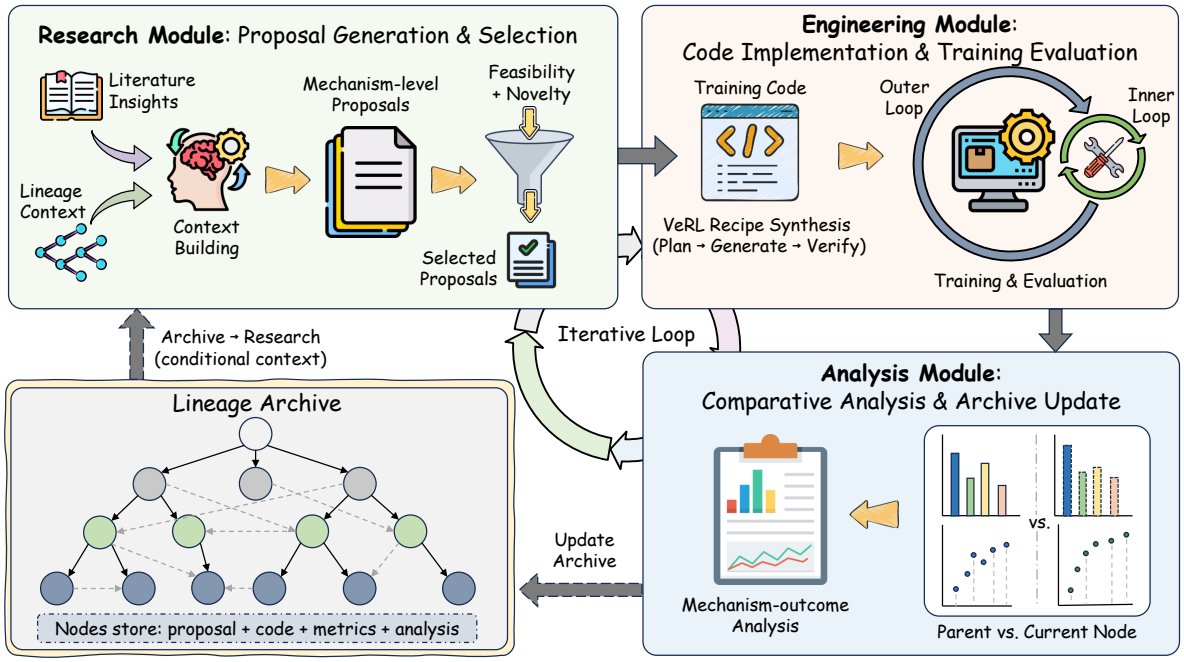

Figure 1: Overview of POISE. Phase I (Proposal Generation) generates candidate algorithms from historical evidence and prior knowledge; Phase II (Implementation, Verification, and Evaluation) implements proposals under shared interfaces, verifies fidelity to the intended algorithm, and evaluates them with a standardized protocol; Phase III (Reflective Analysis and Archive Update) interprets the results and updates the archive.

For any algorithm $z \in \mathcal{Z}$, a standardized evaluation pipeline $\mathcal{O}(z)$ executes the training-and-evaluation workflow and returns a training trajectory τ and a vector of evaluation metrics $\mathbf{y} \in \mathbb{R}^K$, *i.e.* $(\tau, \mathbf{y}) = \mathcal{O}(z)$. The trajectory τ captures training dynamics, such as learning curves, gradient norms, and stability signals, while $\mathbf{y}$ quantifies performance across K evaluation metrics.

Our objective is to discover an algorithm z^* that maximizes the expected scalar utility of the resulting metric vector:

$$z^* = \arg\max_{z \in \mathcal{Z}} \mathbb{E}\left[U\big(\mathbf{y}(z)\big)\right], \tag{1}$$

where the expectation is taken over the randomness of training and evaluation, and $U : \mathbb{R}^K \to \mathbb{R}$ maps the K evaluation metrics to a scalar utility.

Within POISE, algorithm discovery is implemented as **Epistemic Evolutionary Search over an algorithm space**, where epistemic underscores that search must reason under uncertainty about which design choices and lineages remain promising given accumulated archive evidence. As in Figure 1, POISE organizes this search into three phases: proposal generation (Phase I), implementation, verification, and evaluation (Phase II), and reflective analysis with archive update (Phase III). A reflection-augmented evolutionary solver maintains a **structured archive with genealogical links**, denoted $\mathcal{M}_t = \{e_1, \ldots, e_t\}$, across phases. Each entry $e_i = (z_i, \rho_i, \tau_i, \mathbf{y}_i, r_i)$ records the algorithm proposal, its executable implementation ρ_i, resulting empirical evidence as training trajectories τ_i and evaluation metrics $\mathbf{y}_i$, and natural-language reflection r_i. In Phase II, the evaluation pipeline $\mathcal{O}$ implements a proposal under shared interfaces, applies verification and consistency checks, and evaluates it under a standardized protocol. This evidence is then consumed by the reflective analysis and archive update operators in Phase III. Around this evaluation pipeline, the closed loop is coordinated by three operators:

1. **Proposal Operator ($\mathcal{P}$):** Given a selected parent entry $e_p \in \mathcal{M}_t$, it proposes improved candidates in the algorithm space conditioned on historical evidence, reflection, and external literature priors $\mathcal{L}$:

$$\{z_{t+1}^{(n)}\}_{n=1}^{N} \sim \mathcal{P}_{\text{LLM}}\left(\cdot \mid z_p, r_p, \mathcal{M}_t, \mathcal{L}\right).$$

2. **Reflective Analysis Operator ($\mathcal{A}$):** After evaluation via $(\tau_{t+1}, \mathbf{y}_{t+1}) = \mathcal{O}(z_{t+1})$, this operator converts training dynamics and metric outcomes into a mechanism-level diagnosis:

$$r_{t+1} = \mathcal{A}_{\text{LLM}}(z_{t+1}, \tau_{t+1}, \mathbf{y}_{t+1} \mid \mathcal{M}_t).$$

3. **Archive Update Operator ($\mathcal{U}$):** It updates the archive and lineage structure, and selects promising anchors for subsequent evolution:

$$\mathcal{M}_{t+1} = \mathcal{U}\big(\mathcal{M}_t, z_{t+1}, \rho_{t+1}, \tau_{t+1}, \mathbf{y}_{t+1}, r_{t+1}\big).$$

This formulation organizes algorithm search as an evidence-driven closed loop.

3.2 Phase I: Proposal Generation via Epistemic Evolutionary Search

The proposal operator $\mathcal{P}$ generates high-quality candidates conditioned on archive evidence, reflection, and literature priors $\mathcal{L}$. It prioritizes promising parents, constructs proposal context, and selects candidates for subsequent evaluation.

Lineage Prioritization. To identify evolutionary anchors, we construct an acquisition function $s(d)$ that balances strong existing lineages against epistemic uncertainty: Parent selection should therefore favor not only strong nodes, but also nodes likely to yield stronger descendants.

$$\begin{aligned} s(d) = {} & w_{\text{pareto}} \cdot \mathcal{U}_{\text{pareto}} + w_{\text{perf}} \cdot \mathcal{U}_{\text{perf}} \\ & + w_{\text{div}} \cdot \mathcal{U}_{\text{div}} + w_{\text{bayes}} \cdot \alpha_{\text{GP}}(d). \end{aligned}$$

Here, $\mathcal{U}_{\text{pareto}}, \mathcal{U}_{\text{perf}}$, and $\mathcal{U}_{\text{div}}$ denote Pareto-frontier strength, normalized performance, and diversity signals, respectively, while the weights $\mathbf{w}$ control their balance. Crucially, $\alpha_{\text{GP}}(d)$ provides an estimate of a node's descendant potential, namely whether it is likely to yield stronger descendants rather than merely scoring well currently. To avoid favoring immediate performance too myopically, we train this GP signal on a **Discounted Top-K Descendant Gain** target, where $U(\cdot)$ denotes the scalar node utility from the evaluation metrics, c_k ranges over the top-K descendants of d, and β sets the gain scale:

$$\begin{aligned} y_d = {} & \mathbb{E}_{k \in \text{top-K}}\big[\gamma^{\text{depth}(c_k, d)} \\ & \cdot \tanh\big((U(c_k) - U(d))/\beta\big)\big], \end{aligned}$$

We estimate α_{GP} with a GP-based UCB term over archived node features to preserve sensitivity to epistemic uncertainty.

Context Construction from the Archive and Literature. For a selected parent p, we construct a hybrid proposal context from archive evidence, lineage structure, and external priors. To provide complementary reference points, we assemble a tiered reference set: (i) Pareto-frontier references; (ii) complementary references chosen for structural contrast relative to the parent; and (iii) exploratory references sampled outside the direct lineage. We further incorporate literature priors from $\mathcal{L}$ when available. Together, these archive cases and literature priors shape proposal generation.

Population Evolution & Selection. From this context, the proposal operator identifies promising improvement directions and produces algorithm proposals. We adopt a **Population-Based Generation** strategy to sample N parallel candidates spanning diverse local directions. A screening stage filters and ranks candidates for logical consistency, implementation feasibility, novelty, and expected improvement relative to the parent. The highest-ranked candidate z_{t+1} is selected for implementation, verification, and evaluation.

3.3 Phase II: Implementation, Verification, and Evaluation

The evaluation pipeline $\mathcal{O}$ maps abstract proposals to reliable, comparable empirical evidence through implementation, verification, training, and standardized evaluation. This stage emphasizes faithful implementation, consistency checks, and standardized evaluation so observed differences can be attributed as much as possible to algorithmic changes rather than implementation mismatches.

Implementation and Verification under Shared Interfaces. Each proposal is implemented under a common training interface so algorithmic changes remain executable and directly comparable. Given a proposal z_{t+1}, the framework instantiates it as implementation ρ_{t+1} by modifying core optimization components such as advantage computation, objective design, or regularization. During implementation, the framework uses interface constraints and consistency checks to align the implementation with the intended algorithm. Before full evaluation, if verification or quick validation reveals clear mismatches with design intent or implementation faults, the candidate is revised within bounded correction loops so the resulting evidence remains reliable and comparable.

Standardized Evaluation Protocol. For fair comparison, candidates are evaluated under a unified protocol with aligned training budgets, backend settings, and metric definitions. The framework aggregates training trajectories τ_{t+1} and executes standardized tests to compute the evaluation met-

Algorithm	AIME24 pass@32	AIME25 pass@32	AMC pass@32	MATH acc@1	Minerva acc@1	Olympiad acc@1	Overall
GRPO	50.0	26.7	84.3	72.4	24.6	35.4	47.8
Core Signal Modeling							
VM-AV-GRPO	53.3 $_{+3.3}$	43.3 $_{+16.7}$	88.0 $_{+3.6}$	73.6 $_{+1.2}$	24.3 $_{-0.4}$	35.3 $_{-0.1}$	52.5 $_{+4.6}$
AV-GRPO	56.7 $_{+6.7}$	36.7 $_{+10.0}$	81.9 $_{-2.4}$	71.0 $_{-1.4}$	26.8 $_{+2.2}$	35.3 $_{-0.1}$	50.9 $_{+3.1}$
High-Performing Specializations							
MSA-GRPO	50.0 $_{+0.0}$	43.3 $_{+16.7}$	84.3 $_{+0.0}$	69.4 $_{-3.0}$	24.3 $_{-0.4}$	35.3 $_{-0.1}$	50.7 $_{+2.8}$
SVE-LNA-GRPO	50.0 $_{+0.0}$	30.0 $_{+3.3}$	89.2 $_{+4.8}$	73.0 $_{+0.6}$	23.2 $_{-1.5}$	32.7 $_{-2.7}$	48.7 $_{+0.9}$
Failure-side and Routing Variants							
FA-GRPO	53.3 $_{+3.3}$	36.7 $_{+10.0}$	83.1 $_{-1.2}$	72.2 $_{-0.2}$	22.4 $_{-2.2}$	35.0 $_{-0.4}$	49.9 $_{+2.1}$
PR-GRPO	53.3 $_{+3.3}$	36.7 $_{+10.0}$	80.7 $_{-3.6}$	71.2 $_{-1.2}$	23.9 $_{-0.7}$	33.2 $_{-2.2}$	49.4 $_{+1.5}$
CDA-GRPO	50.0 $_{+0.0}$	30.0 $_{+3.3}$	85.5 $_{+1.2}$	72.6 $_{+0.2}$	25.0 $_{+0.4}$	36.9 $_{+1.5}$	49.0 $_{+1.2}$

Table 1: Held-out reasoning evaluation results (percentage points). We report a compact set of strong and mechanism-representative variants in the main text; the full results table (all evaluated variants) is provided in the Appendix. Small colored numbers denote Δ vs. GRPO (green: improvement; red: regression). "Overall" is the fixed weighted average used by the evolutionary selector (weights in Section 4.1). "Olympiad" denotes OlympiadBench.

rics $\mathbf{y}_{t+1}$. Thus, archived nodes remain comparable and suitable for later attribution and selection.

3.4 Phase III: Reflective Analysis & Archive Update

Reflective Analysis. After evaluation, the Reflective Analysis Operator $\mathcal{A}$ converts quantitative signals into mechanism-level diagnoses. It analyzes the algorithmic change Δz, the training trajectories τ_{t+1}, and the evaluation metrics $\mathbf{y}_{t+1}$, and contrasts them with the parent to identify likely contributing effects. For example, it can relate KL spikes to numerical instability in advantage estimation, or connect gains to a particular signal-routing mechanism. Outcomes thus become reusable evidence rather than isolated score changes.

Epistemic Accumulation and Archive Update. The archive stores not only code and scores but also which design choices help, fail, or depend on context. As entries are integrated into $\mathcal{M}$, the framework updates its understanding of which design choices are effective, which are fragile, and when they remain valid. Reflections on strong nodes reinforce useful patterns; those on weak nodes delineate failure boundaries. Over time, the archive becomes a practical reference for proposals, and the regularities from it feed back into proposal generation as inductive bias.

4 Experiments

4.1 Experimental Setup

Training Setup. We implement candidate algorithms in a shared VERL training pipeline (Sheng et al., 2024), running on its distributed policy optimization backend. Algorithm evolution starts from the **GRPO** baseline (Shao et al., 2024), with **Gemini-3-pro-preview** (Google DeepMind) serving as the unified reasoning engine for all framework agents. Following Zhao et al. (2025), we use **Qwen2.5-Math-1.5B** (Yang et al., 2024) as the policy model for rapid iteration. For training data, we sample a 5000-example subset from MATH Levels 3–5 (Hendrycks et al., 2021). Hyperparameters remain fixed across iterations: learning rate 10^{-6}, global batch size 256, group size $G = 8$, and 8 training epochs. Maximum prompt and response lengths are set to 1024 and 3000, respectively. All runs use $8\times$ 80GB A100 GPUs.

Evaluation Protocol. We evaluate on a standardized reasoning suite spanning six datasets: AIME24 (MAA, 2024), AIME25 (MAA, 2025), AMC (MAA, 2023), MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). Inference uses vLLM (Kwon et al., 2023) with a maximum generation length of 4096. For AIME24, AIME25, and AMC, we report pass@32 under temperature 1.0. For the larger datasets MATH-500, Minerva, and OlympiadBench, we report standard acc@1 using deterministic generation. To align with our evolutionary objective, we compute a weighted **Overall** favoring harder benchmarks (0.2 for AIME-series datasets and 0.15 for the others).

4.2 Results

Main Results. In the main evolution run we evaluate 64 algorithms: GRPO and 63 evolved variants.

Figure 2: Training dynamics in the main experiment for GRPO and representative evolved variants. The plots compare entropy, reward, and response length over training.

Table 1 highlights representative algorithms, with full results in Section A.1. The strongest overall variant is VM-AV-GRPO, achieving an Overall of 52.5 and improving over GRPO by +4.6. Its largest gain appears on AIME25, where performance rises from 26.7 to 43.3, while AIME24, AMC, and MATH also improve over baseline. The search also uncovers benchmark-specific variants. Beyond VM-AV-GRPO, AV-GRPO attains the best AIME24 result (56.7), MSA-GRPO matches the best AIME25 score (43.3), and SVE-LNA-GRPO achieves the highest AMC score (89.2). These results show evolutionary search improves overall performance and discovers variants with stronger results on specific benchmarks.

Mechanism Families. Across the lineage, four recurrent mechanism families emerge. **Core Signal Modeling** (*e.g.*, AV-GRPO and VM-AV-GRPO) changes how sparse successes are scaled and compared. **Signal Routing and Conditional Normalization** (*e.g.*, PR-GRPO and CDA-GRPO) separates solved, failed, and mixed regimes and reduces interference among correctness, format, and efficiency signals. **Failure-side Control and Failure Reshaping** (*e.g.*, FA-GRPO and DFR-GRPO) provides directional learning signals even when all samples in a group are wrong. **Stabilization and Constraint Control** (*e.g.*, RF-GRPO and BN-GRPO) dampens update shocks and policy drift. This last family mainly serves as enabling or precursor control rather than as the strongest final design, while Sections A.4 and 5 explain how all four families interact across the lineage.

Figure 2 compares the training dynamics of GRPO against representative evolved variants. Relative to the baseline, stronger variants exhibit more stable entropy and reward trajectories, accompanied by differentiated behaviors in controlling response length. These process-level patterns suggest systematic differences in optimization behavior rather than isolated fluctuations in final scores.

4.3 Controllability Under Length-Compression Constraints

To test whether natural-language directives can effectively steer evolutionary search, we introduce a **length-compression constraint**: the system favors algorithms that substantially reduce reasoning output length while preserving overall accuracy.

Under this constraint, 6 of the 10 evolved variants reduce mean response length. Mean length is computed as the unweighted mean word count over the six evaluation datasets; 4 of these variants (DACE-GRPO, CAG-GRPO, DRE-GRPO, PAS-GRPO) simultaneously achieve shorter responses and higher Overall than GRPO. The strongest compression results arise from correctness-first optimization combined with conditional efficiency shaping. For instance, DACE-GRPO gates the length-efficiency term on correct samples by group pass rate, so compression concentrates on high-pass-rate groups while the accuracy signal is strengthened on harder ones. CAG-GRPO instead reformulates the efficiency term as a penalty-only cost gated by success rate. Conversely, variants with poorly calibrated efficiency constraints exhibit weaker accuracy-length trade-offs, indicating that compression pressure is more effective when introduced conditionally rather than indiscriminately.

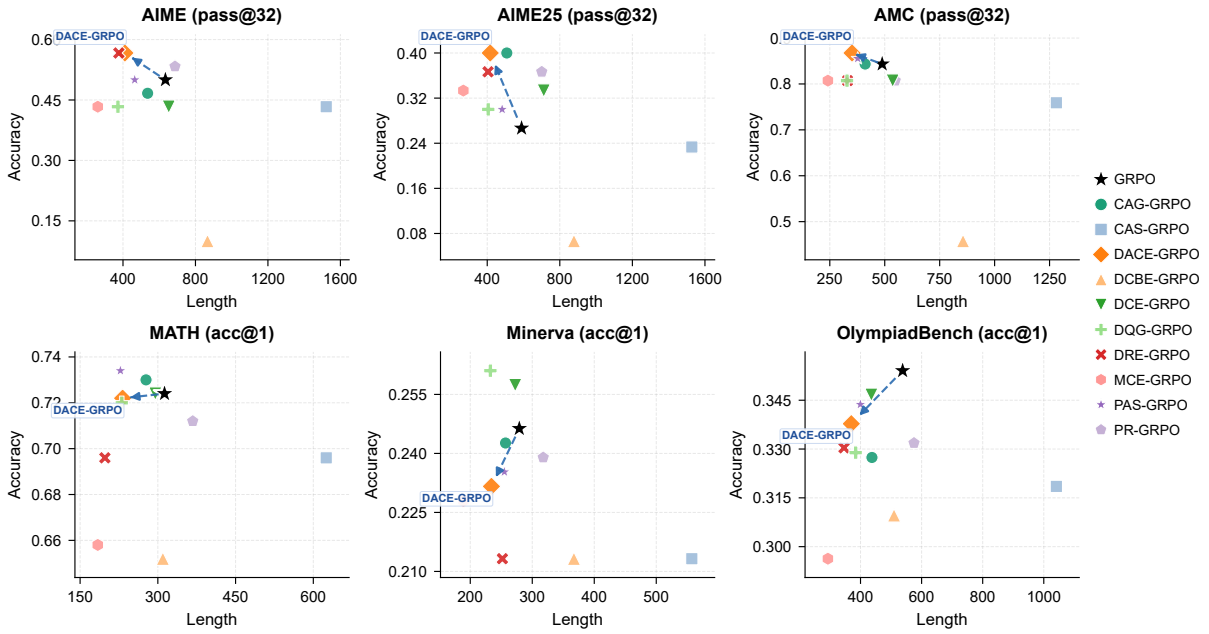

Figure 3: Accuracy-length trade-offs under the length-compression constraint.

Section A.5 and Table 4 summarize the corresponding dynamics and the full quantitative results for all 11 nodes in the branch.

Figure 3 and Table 4 show that DACE-GRPO provides the best balance, improving Overall from 47.8 to 51.7 (+3.9) while reducing mean output length from 473.6 to 335.7 words (−29.1%); CAG-GRPO achieves stable compression (length ratio 0.854) with positive Overall gain. DRE-GRPO and PAS-GRPO show moderate gains but follow the same direction of difficulty- and correctness-conditioned efficiency shaping. At the dataset level, MATH and AMC retain substantial compression headroom without sacrificing accuracy: for example, DACE-GRPO reaches 86.7 on AMC pass@32 while reducing mean length by 28.1%. OlympiadBench is more compression-sensitive, where shorter variants generally incur mild accuracy drops, indicating a stronger dependence on longer reasoning chains. Overall, these results indicate that natural-language directives can meaningfully steer search in a constrained accuracy-length trade-off space, and that correctness-first, conditionally shaped efficiency signals provide a viable route to effective compression.

5 Analysis

We begin by summarizing the full lineage by tree depth and then examine representative mechanism chains. Figure 4 shows that depth-wise averages are not monotonic, since deeper generations still include exploratory failures. Nevertheless, the frontier moves beyond the root: the cumulative best Overall increases from 47.8 for GRPO to 49.9 at depth 1, 50.9 at depth 3, and 52.5 at depth 4. Later depths still contain competitive descendants such as MSA-GRPO and CDA-GRPO. This suggests that the search remains productive even after the best observed node has already appeared. A complete view by tree depth is provided in Figure 6.

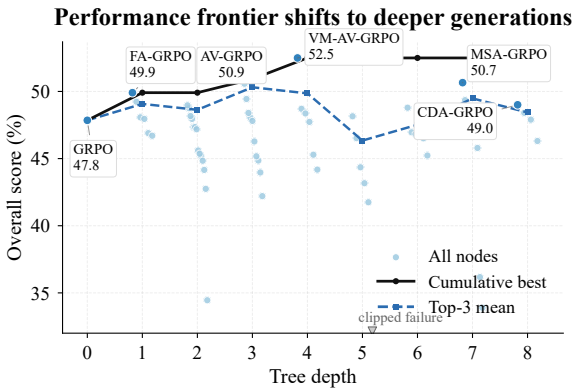

Figure 4: Performance frontier vs. tree depth in the base-branch lineage; dots are algorithms. Solid: cumulative best Overall; dashed: mean Overall of top-3 at each depth. Despite exploratory failures at depth, the frontier improves beyond the root.

The lineage provides retrospective evidence on parent retention. Across 21 comparable parent-selection rounds (each round selects 3 nodes in parallel to expand the next generation), the par-

ent that yields the strongest descendant is not the selected parent with the highest Overall score in 11 rounds. One representative round is as follows (numbers in parentheses denote Overall scores): ZA-GRPO (47.3), BN-GRPO (44.2), and Anchor-GRPO (45.2) are retained together. In that round, the strongest later descendant arises from BN-GRPO via VM-AV-GRPO (52.5), above the ZA-GRPO branch peak of 48.4. This pattern is consistent with retaining parents for future improvement rather than filtering by current Overall score. We provide a fuller round-level account and representative parent-descendant gains in Section A.4.1.

5.1 Mechanistic Analysis: Core Signal Modeling

A natural question is whether the framework recovers interpretable structure rather than merely tuning hyperparameters. The lineage from GRPO to VM-AV-GRPO illustrates this in the **Core Signal Modeling** family, supported by training-dynamics evidence (Figure 2), lineage trace (Section A.3), and trajectory- and algorithm-level details (Sections A.2 and A.4.2). Baseline GRPO underperforms on sparse-reward benchmarks (*e.g.*, AIME25): failure-dominated group statistics can dilute the learning signal from rare successes, while high-variance updates can hasten entropy collapse.

To address this bottleneck, the lineage introduces **Analytic-variance scaling** (AV-GRPO). Instead of relying on empirical batch variance, it anchors advantage normalization to the analytic variance of the reward distribution so that rarer successes can yield stronger positive updates. A subsequent refinement then exposes another failure mode: when "format-correct but reasoning-wrong" samples enter the statistics, they can receive positive advantages. In response, the lineage introduces **Validity Masking** (VM-AV-GRPO), which restores a strict gradient hierarchy (invalid $\ll$ valid-wrong $<$ valid-correct) by computing statistics exclusively on the valid subset. Together, analytic-variance scaling and validity masking are associated with a substantial AIME25 improvement from GRPO to VM-AV-GRPO, while Sections A.2 and A.4.2 trace the intermediate turning points and the corresponding algorithm details behind this refinement.

5.2 Learning from Negative Evidence

Beyond discovering novel algorithms, the framework can also use negative evidence to understand why an algorithm failed. This is evident in the lineage targeting all-fail stagnation (GRPO $\rightarrow$ FA-GRPO $\rightarrow$ DFR-GRPO), a representative instance of **Failure-side Control and Failure Reshaping**; Sections A.2 and A.4.2 provide the extended trajectory and algorithm-level formulations.

FA-GRPO first injects gradients into all-fail groups through fixed scalar penalties. However, post-run analysis of the training dynamics suggests that mixing such fixed penalties with normalized advantages causes a scale mismatch that destabilizes formatting. Furthermore, applying uniform negative advantages acts as "directionless repulsion," triggering policy drift. DFR-GRPO then refines this design by separating the signals. It decouples format and reasoning into independent streams and replaces uniform repulsion with a **zero-centered signal** derived from token entropy, so that more informative failures are compared against less informative ones rather than being pushed equally. This yields a more directional gradient even when all candidates are incorrect, alleviating directionless drift, albeit with trade-offs on the hardest benchmarks. Together with the **Signal Routing and Conditional Normalization** cases detailed in Sections A.4.2 and A.4.3, this cross-generational refinement highlights three recurring principles: rare successes should be amplified through calibrated signal modeling, flawed hypotheses should be removed through negative evidence, and secondary objectives should be introduced through signal decoupling under controlled conditions.

6 Conclusion

In this work, we show that the discovery of policy optimization algorithms for language models can be organized as a structured empirical loop of proposal, implementation, verification, evaluation, and reflection rather than treated solely as manual trial and error. POISE discovers stronger policy optimization variants from a GRPO baseline, identifies mechanisms such as analytic-variance scaling and validity masking, and further shows through the length-compression setting that natural-language directives can steer the search toward a specified region of the accuracy-length trade-off space. Taken together, the resulting lineages reveal recurring principles such as signal decoupling, conditional normalization, and correctness-first efficiency shaping, suggesting that part of algorithm design can be studied as an evidence-driven iterative process that complements direct manual design.

Limitations

Our study demonstrates the promise of automated mechanism discovery, but limitations remain. First, the full evolutionary loop is still computationally intensive, since each candidate requires end-to-end training and standardized evaluation; under a fixed budget, this constrains both the breadth of explored candidates and the number of controlled follow-up trials. Second, empirical evidence is concentrated on mathematical reasoning benchmarks and a limited range of model settings, so the transferability of the discovered mechanisms to broader regimes (*e.g.*, open-domain dialogue, code generation, and tool-use tasks) remains to be established. Third, although lineage analysis, training dynamics, and reflective diagnoses support mechanism-level interpretations, these interpretations should still be treated as evidence-grounded hypotheses rather than controlled causal identification of which component is responsible for each gain. Future work should focus on improving search efficiency, broadening cross-task and cross-scale validation, and developing stronger causal evaluation protocols.

Ethics Statement

We confirm that all authors are aware of the ACL Code of Ethics and commit to conducting this research in accordance with responsible research practices.

Data Use and Privacy. Our experiments use publicly available mathematical reasoning benchmarks (*e.g.*, AIME, AMC, MATH, Minerva, and OlympiadBench) and do not involve private or personally identifiable information. The training subset and evaluation data are used under the respective dataset terms and licenses. We do not collect user-level behavioral data or deploy systems that interact with human subjects.

Automation Risks and Responsible Use. This work studies automated algorithm discovery for LLM-RL. Although our setting is focused on mathematical reasoning, automated search systems can, in principle, generate unstable or misaligned optimization strategies if applied without safeguards. To reduce this risk, we use bounded verification loops, standardized evaluation, and human oversight in the research pipeline, and we report limitations regarding generalization and causal interpretation. We do not claim that the discovered mechanisms are universally safe or optimal outside the evaluated scope.

Use of AI Assistants. LLMs are used as research assistants within the proposed framework (for proposal generation, implementation support, and reflective analysis) and for manuscript polishing. Final scientific claims, experimental decisions, and interpretation of results are made and verified by the authors.

References

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. Nature, 624(7992):570–578.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, and 1 others. Mle-bench: Evaluating machine learning agents on machine learning engineering. In The Thirteenth International Conference on Learning Representations.

Junyan Cheng, Peter Clark, and Kyle Richardson. 2025. Language modeling by language models. arXiv preprint arXiv:2506.20249.

Yuri Chervonyi, Trieu H Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang H Nguyen, Marcelo Menegali, Junehyuk Jung, Junsu Kim, Vikas Verma, Quoc V Le, and 1 others. 2025. Gold-medalist performance in solving olympiad geometry with alphageometry2. Journal of Machine Learning Research, 26(241):1–39.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. In Forty-second International Conference on Machine Learning.

Alhussein Fawzi and Bernardino Romera Paredes. 2023. Funsearch: Making new discoveries in mathematical sciences using large language models. DeepMind Google.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. 2025. Soft adaptive policy optimization. arXiv preprint arXiv:2511.20347.

Google DeepMind. https://deepmind.google/models/gemini/pro/.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, and 1 others. 2025. Towards an ai co-scientist. arXiv preprint arXiv:2502.18864.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3828–3850.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. NeurIPS.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. 2025. Aide: Ai-driven exploration in the space of code. arXiv preprint arXiv:2502.13138.

Farhana Keya, Gollam Rabby, Prasenjit Mitra, Sahar Vahdati, Sören Auer, and Yaser Jaradeh. 2025. Sci-idea: Context-aware scientific ideation using token and sentence embeddings. arXiv preprint arXiv:2503.19257.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. Advances in neural information processing systems, 35:3843–3857.

Ruochen Li, Liqiang Jing, and Xinya Du. 2024. Learning to generate research idea with dynamic control. In 2nd AI4Research Workshop: Towards a Knowledge-grounded Scientific Research Lifecycle.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. arXiv preprint arXiv:2512.02556.

Yixiu Liu, Yang Nan, Weixian Xu, Xiangkun Hu, Lyumanshan Ye, Zhen Qin, and Pengfei Liu. 2025b. Alphago moment for model architecture discovery. arXiv preprint arXiv:2507.18074.

Zexi Liu, Yuzhu Cai, Xinyu Zhu, Yujie Zheng, Runkun Chen, Ying Wen, Yanfeng Wang, Siheng Chen, and 1 others. 2025c. Ml-master: Towards ai-for-ai via integration of exploration and reasoning. arXiv preprint arXiv:2506.16499.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob N Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. CoRR.

MAA. 2023. American mathematics contest 12 (amc 12).

MAA. 2024. American invitational mathematics examination (aime).

MAA. 2025. American invitational mathematics examination (aime).

Ludovico Mitchener, Angela Yiu, Benjamin Chang, Mathieu Bourdenx, Tyler Nadolski, Arvis Sulovari, Eric C Landsness, Daniel L Barabasi, Siddharth Narayanan, Nicky Evans, and 1 others. 2025. Kosmos: An ai scientist for autonomous discovery. arXiv preprint arXiv:2511.02824.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, and 1 others. 2025. Alphaevolve: A coding agent for scientific and algorithmic discovery. arXiv preprint arXiv:2506.13131.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744.

Samuel Schmidgall and Michael Moor. 2025. Agentrxiv: Towards collaborative autonomous research. arXiv preprint arXiv:2503.18102.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. arXiv preprint arXiv:2501.04227.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv:2409.19256.

Yu-Zhe Shi, Mingchen Liu, Fanxu Meng, Qiao Xu, Zhangqian Bi, Kun He, Lecheng Ruan, and Qining Wang. Hierarchically encapsulated representation for protocol design in self-driving labs. In The Thirteenth International Conference on Learning Representations.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. In The Thirteenth International Conference on Learning Representations.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. Nature, 625(7995):476–482.

Jiakang Wang, Runze Liu, Lei Lin, Wenping Hu, Xiu Li, Fuzheng Zhang, Guorui Zhou, and Kun Gai. 2025. Aspo: Asymmetric importance sampling policy optimization. arXiv preprint arXiv:2510.06062.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. arXiv preprint arXiv:2504.08066.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. arXiv preprint arXiv:2505.09388.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. CoRR.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, and 1 others. 2025. Geometric-mean policy optimization. arXiv preprint arXiv:2507.20673.

A Appendix

A.1 Full main-experiment results and lineage

We provide the full held-out evaluation results for all evaluated variants in Table 2. To better illustrate the evolutionary paths, we also visualize the complete lineage tree in Figure 5.

A.2 Detailed Algorithm Descriptions (Representative Subset)

Due to space constraints, we do not enumerate the full evolutionary catalog here. Instead, we summarize representative improvement and failure cases from the latest experiments. For each algorithm, we report **motivation**, **method (with key equations)**, and **results**. AIME24, AIME25, and AMC use pass@32; MATH/Minerva/OlympiadBench use acc@1; Overall is consistent with the main text.

A.2.1 GRPO Baseline

GRPO

Motivation. Under sparse correctness rewards, we need a stable relative learning signal without introducing a value network.
Method. For each prompt, GRPO samples G responses and computes group-relative advantages:

$$\mu_g = \frac{1}{G}\sum_{j=1}^{G} r_j,$$

$$\sigma_g = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu_g)^2 + \epsilon},$$

$$A_i = \frac{r_i - \mu_g}{\sigma_g}.$$

The policy is then updated with a PPO-style objective plus KL regularization (divergence to a fixed reference policy).
Results. Overall 47.8, AIME24 50.0, AIME25 26.7, AMC 84.3, MATH 72.4, Minerva 24.6, OlympiadBench 35.4.

A.2.2 Main Experiment: Representative Improving Algorithms

This representative subset focuses on the main improving and regressive cases needed for interpretation, rather than exhaustively mirroring the compact main-table mechanism-family grouping. Specifically, AV-GRPO and VM-AV-GRPO fall under **Core Signal Modeling**, FA-GRPO and DFR-GRPO under **Failure-side Control and Failure Reshaping**, and BN-GRPO serves as a representative **Stabilization and Constraint Control** case. MSA-GRPO is included here as an additional high-performing variant rather than as one of the main-table exemplars for the mechanism-family grouping, while the **Signal Routing and Conditional Normalization** family is analyzed in more detail in Section A.4.

AV-GRPO

Motivation. Proposal analysis suggests that batch-level scaling in BN-GRPO improves stability but can dilute sparse hard-task success signals.
Method. AV-GRPO replaces empirical scaling with analytic-variance scaling and stricter format constraints:

$$\mu_g = \frac{1}{G}\sum_i R_i,$$

$$\mu_{2,g} = \frac{1}{G}\sum_i R_i^2,$$

$$\sigma_{\text{analytic}} = \sqrt{\mu_{2,g} - \mu_g^2},$$

$$A_i = \frac{R_i - \mu_g}{\max(\sigma_{\text{analytic}}, \sigma_{\min})}.$$

For binary rewards, this reduces to $\sigma = \sqrt{\mu(1-\mu)}$, providing a more adaptive scale in rare-success regimes. In the proposal, this scaling-centered redesign is paired with a strict format penalty and a lower entropy coefficient, so the empirical gain should be interpreted as the effect of a small set of coordinated changes rather than a single isolated substitution.
Results. Overall 50.9, AIME24 56.7, AIME25 36.7, with gains over GRPO.

VM-AV-GRPO

Motivation. The training proposal identifies a residual mismatch in AV-GRPO: mixing invalid and valid samples in shared statistics can produce false-positive incentives for valid-but-wrong outputs.
Method. VM-AV-GRPO applies validity masking with Bayesian-smoothed valid-subset statistics (computed on the format-valid subset only), together with an explicit floor penalty for invalid outputs:

$$\mu_{\text{valid}} = \frac{\sum_{i\in V} R_i + \alpha}{|V| + \alpha + \beta},$$

$$\sigma_{\text{valid}} = \sqrt{\mu_{\text{valid}}(1 - \mu_{\text{valid}})},$$

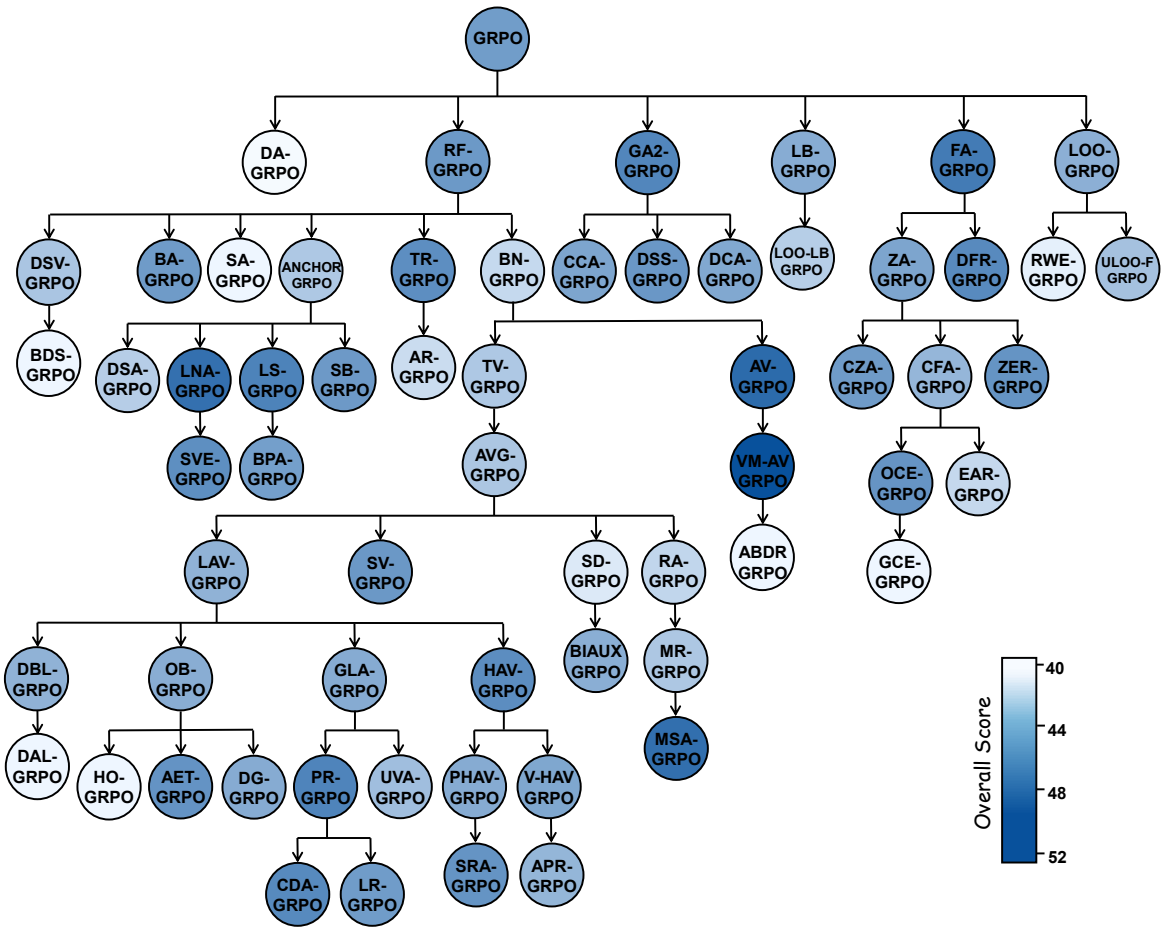

Figure 5: **Evolutionary lineage of GRPO variants in the main run.** Nodes denote algorithms, arrows denote parent-child inheritance, and node color encodes the weighted Overall (lighter is lower).

Algorithm	AIME24 pass@32	AIME25 pass@32	AMC pass@32	MATH acc@1	Minerva acc@1	Olympiad acc@1	Overall
GRPO	50.0	26.7	84.3	72.4	24.6	35.4	47.8
VM-AV-GRPO	53.3 +3.3	43.3 +16.7	88.0 +3.6	73.6 +1.2	24.3 -0.4	35.3 -0.1	52.5 +4.6
AV-GRPO	56.7 +6.7	36.7 +10.0	81.9 -2.4	71.0 -1.4	26.8 +2.2	35.3 -0.1	50.9 +3.1
MSA-GRPO	50.0 +0.0	43.3 +16.7	84.3 +0.0	69.4 -3.0	24.3 -0.4	35.3 -0.1	50.7 +2.8
LNA-GRPO	56.7 +6.7	33.3 +6.7	84.3 +0.0	73.6 +1.2	24.3 -0.4	35.0 -0.4	50.6 +2.7
FA-GRPO	53.3 +3.3	36.7 +10.0	83.1 -1.2	72.2 -0.2	22.4 -2.2	35.0 -0.4	49.9 +2.1
LS-GRPO	46.7 -3.3	36.7 +10.0	86.7 +2.4	71.4 -1.0	24.6 +0.0	35.7 +0.3	49.4 +1.6
PR-GRPO	53.3 +3.3	36.7 +10.0	80.7 -3.6	71.2 -1.2	23.9 -0.7	33.2 -2.2	49.4 +1.5
GA2-GRPO	53.3 +3.3	33.3 +6.7	83.1 -1.2	70.6 -1.8	24.3 -0.4	34.7 -0.7	49.2 +1.4
CDA-GRPO	50.0 +0.0	30.0 +3.3	85.5 +1.2	72.6 +0.2	25.0 +0.4	36.9 +1.5	49.0 +1.2
DFR-GRPO	46.7 -3.3	33.3 +6.7	85.5 +1.2	73.4 +1.0	25.7 +1.1	35.1 -0.3	49.0 +1.1
HAV-GRPO	43.3 -6.7	36.7 +10.0	88.0 +3.6	72.6 +0.2	22.8 -1.8	35.3 -0.1	48.8 +0.9
TR-GRPO	50.0 +0.0	33.3 +6.7	83.1 -1.2	72.4 +0.0	25.4 +0.7	33.0 -2.4	48.8 +0.9
SVE-LNA-GRPO	50.0 +0.0	30.0 +3.3	89.2 +4.8	73.0 +0.6	23.2 -1.5	32.7 -2.7	48.7 +0.9
AET-GRPO	46.7 -3.3	36.7 +10.0	84.3 +0.0	71.2 -1.2	23.5 -1.1	32.7 -2.7	48.4 +0.6
SRA-GRPO	46.7 -3.3	33.3 +6.7	86.7 +2.4	72.0 -0.4	23.5 -1.1	33.8 -1.6	48.4 +0.6
ZER-GRPO	50.0 +0.0	33.3 +6.7	80.7 -3.6	72.2 -0.2	24.3 -0.4	34.4 -1.0	48.4 +0.6
OCE-GRPO	46.7 -3.3	33.3 +6.7	84.3 +0.0	71.0 -1.4	24.3 -0.4	36.1 +0.7	48.4 +0.5
DSS-GRPO	53.3 +3.3	26.7 +0.0	83.1 -1.2	72.4 +0.0	23.5 -1.1	35.4 +0.0	48.2 +0.3
SV-GRPO	46.7 -3.3	30.0 +3.3	86.7 +2.4	72.6 +0.2	26.1 +1.5	33.3 -2.1	48.2 +0.3
RF-GRPO	50.0 +0.0	33.3 +6.7	80.7 -3.6	71.2 -1.2	23.5 -1.1	33.9 -1.5	48.1 +0.2
SB-GRPO	50.0 +0.0	30.0 +3.3	88.0 +3.6	71.6 -0.8	21.7 -2.9	32.4 -3.0	48.1 +0.2
DA-GRPO	53.3 +3.3	30.0 +3.3	79.5 -4.8	71.4 -1.0	23.9 -0.7	33.8 -1.6	48.0 +0.1
BA-GRPO	46.7 -3.3	33.3 +6.7	83.1 -1.2	71.4 -1.0	24.3 -0.4	34.2 -1.2	48.0 +0.1
LR-GRPO	46.7 -3.3	36.7 +10.0	84.3 +0.0	70.2 -2.2	21.7 -2.9	32.0 -3.4	47.9 +0.1
CZA-GRPO	43.3 -6.7	36.7 +10.0	83.1 -1.2	74.6 +2.2	20.2 -4.4	34.1 -1.3	47.8 +0.0
BPA-GRPO	50.0 +0.0	33.3 +6.7	83.1 -1.2	71.0 -1.4	18.8 -5.9	34.2 -1.2	47.7 -0.1
ZA-GRPO	50.0 +0.0	26.7 +0.0	81.9 -2.4	73.0 +0.6	24.3 -0.4	34.2 -1.2	47.3 -0.5
CCA-GRPO	50.0 +0.0	26.7 +0.0	81.9 -2.4	71.8 -0.6	25.0 +0.4	34.7 -0.7	47.3 -0.5
V-HAV-GRPO	43.3 -6.7	33.3 +6.7	85.5 +1.2	68.2 -4.2	23.5 -1.1	35.4 +0.0	47.2 -0.6
DCA-GRPO	43.3 -6.7	36.7 +10.0	81.9 -2.4	69.4 -3.0	23.5 -1.1	33.0 -2.4	47.2 -0.7
DG-GRPO	43.3 -6.7	30.0 +3.3	86.7 +2.4	71.8 -0.6	23.2 -1.5	33.6 -1.8	47.0 -0.9
PHAV-GRPO	46.7 -3.3	26.7 +0.0	84.3 +0.0	72.4 +0.0	24.6 +0.0	33.9 -1.5	47.0 -0.9
GLA-GRPO	40.0 -10.0	33.3 +6.7	86.7 +2.4	71.8 -0.6	22.1 -2.6	34.7 -0.7	47.0 -0.9
LB-GRPO	53.3 +3.3	23.3 -3.3	83.1 -1.2	69.0 -3.4	23.2 -1.5	35.1 -0.3	46.9 -1.0
OB-GRPO	53.3 +3.3	26.7 +0.0	78.3 -6.0	69.4 -3.0	22.8 -1.8	35.0 -0.4	46.8 -1.0
BIAUX-GRPO	50.0 +0.0	30.0 +3.3	83.1 -1.2	69.2 -3.2	20.6 -4.0	32.3 -3.1	46.8 -1.1
LOO-GRPO	53.3 +3.3	16.7 -10.0	85.5 +1.2	72.8 +0.4	24.3 -0.4	35.4 +0.0	46.7 -1.1
LAV-GRPO	50.0 +0.0	33.3 +6.7	79.5 -4.8	67.6 -4.8	21.0 -3.7	31.0 -4.4	46.5 -1.3
DBL-GRPO	43.3 -6.7	33.3 +6.7	80.7 -3.6	71.0 -1.4	22.8 -1.8	33.3 -2.1	46.5 -1.3
APR-GRPO	43.3 -6.7	26.7 +0.0	83.1 -1.2	71.2 -1.2	26.1 +1.5	35.0 -0.4	46.3 -1.5
CFA-GRPO	33.3 -16.7	33.3 +6.7	85.5 +1.2	75.0 +2.6	24.3 -0.4	34.8 -0.6	46.3 -1.6
UVA-GRPO	46.7 -3.3	20.0 -6.7	86.7 +2.4	73.0 +0.6	23.2 -1.5	33.5 -1.9	45.8 -2.1
ULOO-F-GRPO	36.7 -13.3	40.0 +13.3	77.1 -7.2	70.4 -2.0	20.2 -4.4	34.1 -1.3	45.6 -2.2
DSV-GRPO	43.3 -6.7	30.0 +3.3	80.7 -3.6	69.2 -3.2	23.2 -1.5	31.6 -3.9	45.4 -2.5
AVG-GRPO	43.3 -6.7	30.0 +3.3	83.1 -1.2	68.8 -3.6	21.0 -3.7	31.3 -4.1	45.3 -2.6
MR-GRPO	46.7 -3.3	30.0 +3.3	79.5 -4.8	67.8 -4.6	21.3 -3.3	30.7 -4.7	45.2 -2.6
TV-GRPO	40.0 -10.0	33.3 +6.7	78.3 -6.0	71.0 -1.4	21.3 -3.3	32.7 -2.7	45.2 -2.7
ANCHOR-GRPO	43.3 -6.7	26.7 +0.0	79.5 -4.8	72.4 +0.0	23.2 -1.5	32.7 -2.7	45.2 -2.7
DSA-GRPO	46.7 -3.3	23.3 -3.3	75.9 -8.4	73.6 +1.2	23.2 -1.5	33.0 -2.4	44.9 -3.0
LOO-LB-GRPO	43.3 -6.7	26.7 +0.0	79.5 -4.8	70.4 -2.0	22.8 -1.8	32.9 -2.5	44.8 -3.0
RA-GRPO	40.0 -10.0	26.7 +0.0	84.3 +0.0	69.8 -2.6	21.0 -3.7	31.7 -3.7	44.4 -3.5
EAR-GRPO	36.7 -13.3	26.7 +0.0	80.7 -3.6	72.2 -0.2	22.8 -1.8	34.4 -1.0	44.2 -3.7
BN-GRPO	40.0 -10.0	26.7 +0.0	77.1 -7.2	72.4 +0.0	22.1 -2.6	33.9 -1.5	44.2 -3.7
AR-GRPO	40.0 -10.0	23.3 -3.3	80.7 -3.6	71.6 -0.8	23.2 -1.5	33.2 -2.2	44.0 -3.9
SD-GRPO	33.3 -16.7	26.7 +0.0	81.9 -2.4	71.2 -1.2	22.8 -1.8	31.9 -3.6	43.2 -4.7
RWE-GRPO	36.7 -13.3	20.0 -6.7	77.1 -7.2	72.8 +0.4	24.3 -0.4	35.3 -0.1	42.7 -5.1
BDS-GRPO	36.7 -13.3	23.3 -3.3	75.9 -8.4	71.4 -1.0	21.3 -3.3	32.7 -2.7	42.2 -5.6
ABDR-GRPO	33.3 -16.7	20.0 -6.7	78.3 -6.0	72.4 +0.0	22.4 -2.2	34.1 -1.3	41.7 -6.1
DAL-GRPO	16.7 -33.3	26.7 +0.0	55.4 -28.9	70.8 -1.6	22.1 -2.6	35.1 -0.3	36.2 -11.7
GAA-GRPO	20.0 -30.0	10.0 -16.7	63.9 -20.5	71.2 -1.2	27.6 +2.9	35.0 -0.4	35.6 -12.2
SA-GRPO	20.0 -30.0	13.3 -13.3	54.2 -30.1	73.2 +0.8	23.9 -0.7	33.9 -1.5	34.5 -13.4
HO-GRPO	26.7 -23.3	6.7 -20.0	63.9 -20.5	67.2 -5.2	17.3 -7.4	33.2 -2.2	33.9 -14.0
GCE-GRPO	0.0 -50.0	0.0 -26.7	1.2 -83.1	53.0 -19.4	15.1 -9.6	24.4 -11.0	14.1 -33.8

Table 2: Full held-out reasoning evaluation results for all evaluated variants (percentage points). GRPO (first row) is the baseline. Small colored numbers denote Δ vs. GRPO (green: improvement; red: regression). "Overall" is the fixed weighted average used by the evolutionary selector (weights in Section 4.1). "Olympiad" denotes OlympiadBench.

$$A_i = \begin{cases} \dfrac{R_i - \mu_{\text{valid}}}{\max(\sigma_{\text{valid}}, \sigma_{\text{floor}})}, & i \in V \\ A_{\text{floor}}, & i \notin V. \end{cases}$$

It further applies validity-gated length normalization (only on valid samples) to negative valid advantages:

$$L_{\text{rel}} = \text{clip}\left(\frac{L_i}{\bar{L}_{\text{valid}}}, 0.5, 2.0\right),$$

$$A_i^{\text{final}} = \begin{cases} A_i / L_{\text{rel}}, & i \in V \wedge A_i < 0 \\ A_i, & \text{otherwise.} \end{cases}$$

Results. Overall 52.5, AIME25 43.3, AMC 88.0; higher than AV-GRPO on AMC (81.9).

MSA-GRPO

Motivation. The proposal separates high-information mixed groups from low-information uniform groups, arguing they should not drive updates at equal magnitude.

Method. MSA-GRPO introduces regime-aware anchor selection and scaling (Mixed: both correct and incorrect samples; Uniform: all same outcome):

$$\sigma_{\text{anchor}} = \begin{cases} \sigma_{\text{outcome}}, & \sigma_{\text{outcome}} > \epsilon \\ \sigma_{\text{total}}, & \text{otherwise} \end{cases}$$

$$A_{\text{scaled}} = A_{\text{raw}} \cdot \begin{cases} 1.0, & \text{Mixed} \\ \alpha_{\text{uniform}}, & \text{Uniform} \end{cases}$$

with multiplicative length scaling:

$$\text{ratio}_i = \frac{L_i}{\bar{L}_{\text{group}}},$$

$$A_{\text{final}} = \begin{cases} A_{\text{scaled}} \cdot \text{ratio}_i^{\lambda}, & A_{\text{scaled}} > 0 \\ A_{\text{scaled}} \cdot \text{ratio}_i^{-\lambda}, & A_{\text{scaled}} < 0. \end{cases}$$

Results. Overall 50.7, AIME25 43.3.

FA-GRPO

Motivation. Proposal evidence highlights all-fail stagnation under standard GRPO, motivating explicit signal shaping in all-fail groups.

Method. FA-GRPO uses a two-regime advantage:

$$A_{\text{final}} = \begin{cases} \text{clip}\left(\dfrac{R_i - \mu_g}{\sigma_g + \epsilon}, -3, 3\right), & R_{\max} > 0 \\ -2.0, & \text{invalid format} \\ -0.5, & \text{valid but wrong.} \end{cases}$$

This design separates structural errors from reasoning failures with different penalty magnitudes.

Results. Overall 49.9, AIME25 36.7.

DFR-GRPO

Motivation. Post-run diagnosis of FA-GRPO suggests that fixed penalties can destabilize formatting when mixed directly with normalized reasoning advantages, and that uniform negative penalties in all-fail groups create directionless repulsion rather than directional correction.

Method. DFR-GRPO decouples format control and reasoning quality into independent advantage streams:

$$A_i^{\text{total}} = A_i^{\text{reason}} + \lambda_{\text{fmt}} A_i^{\text{format}},$$

where A_i^{format} tracks format compliance separately from reasoning quality. In all-fail groups, the reasoning stream no longer uses a uniform constant penalty; instead, it assigns a zero-centered signal derived from token entropy, so that more informative failures are compared against less informative ones rather than being pushed equally. This turns failure-side updates from undirected repulsion into directional comparison among failures.

Results. Overall 49.0, AIME25 33.3, AMC 85.5. Relative to FA-GRPO, this revision primarily regularizes the failure-side signal path, while incurring trade-offs on some hard benchmarks.

A.2.3 Main Experiment: Stabilization and Regression Cases

BN-GRPO

Motivation. BN-GRPO aims to suppress noise amplification via group-centering and batch-scaling, while moving KL/entropy terms (entropy encourages exploration) into the loss.

Method.

$$A_{\text{raw},ij} = R_{ij} - \mu_i,$$

$$\sigma_{\text{batch}} = \sqrt{\frac{1}{B} \sum_{i,j} A_{\text{raw},ij}^2},$$

$$A_{\text{final},ij} = \text{clip}\left(\frac{A_{\text{raw},ij}}{\sigma_{\text{batch}} + \epsilon}, -3, 3\right),$$

$$L(\theta) = \mathbb{E}\left[L^{\text{CLIP}}(\theta) - \beta_{\text{KL}} D_{\text{KL}} + \beta_{\text{ent}} H(\pi)\right].$$

Results. Overall 44.2, AIME25 26.7. In this stage, the stabilization objective does not translate into hard-task gains.

SA-GRPO

Motivation. SA-GRPO uses fixed scaling and a dual-anchor baseline to avoid extreme advantages

in low-variance groups.
Method.

$$\mu_{\text{anchor}} = \alpha\mu_{\text{group}} + (1-\alpha)\mu_{\text{batch}},$$
$$A_{ij} = \frac{R_{ij} - \mu_{\text{anchor}}}{\sigma_{\text{fixed}}},$$

with decoupled regularization in loss:

$$L_{\text{total}} = L_{\text{CLIP}} - \beta_{\text{KL}} D_{\text{KL}} + \gamma_{\text{Ent}} H(\pi).$$

Results. Overall 34.5, AIME25 13.3, representing a regression case.

A.2.4 Response-Length Compression Experiment: Success and Failure

DACE-GRPO (Success)

Motivation. Compress response length while preserving hard-task correctness, avoiding "compress-first, regress-later" behavior.
Method. Difficulty-aware reweighting is applied via group pass rate (the fraction of correct samples within a prompt group):

$$\text{PassRate}_i = \frac{1}{G}\sum_{j=1}^{G} \mathbb{I}(r_{ij} > 0.5),$$

$$A_{\text{acc}}^{\text{new}} = A_{\text{acc}}\big(1 + \alpha(1 - \text{PassRate}_i)\big),$$
$$A_{\text{eff}}^{\text{new}} = A_{\text{eff}} \cdot \text{PassRate}_i,$$

$$A_{\text{total}} = A_{\text{acc}}^{\text{new}} + \beta A_{\text{eff}}^{\text{new}} \cdot \mathbb{I}(\text{Correct}_i).$$

A_eff is computed from the z-scored negative response lengths within the correct subset, so the compression signal only acts among correct samples and is then further gated by group pass rate. An annealed entropy target $H_{\text{target}}(t)$ further supports exploration early and consolidation later.
Results. Overall 51.7 (+3.9), mean length 473.6 → 335.7 (ratio 0.709); AIME25 40.0, AMC 86.7.

CAG-GRPO (Success)

Motivation. Prevent length collapse driven by disproportionate gradients from overly short solutions, by reformulating efficiency as a cost term rather than a positive bonus.
Method.

$$A_{\text{eff}} = \min\left(\frac{\mu_L - L_i}{\mu_L}, 0\right),$$
$$\beta = \beta_{\text{base}} \cdot SR^2,$$
$$A_{\text{total}} = A_{\text{acc}} + \beta A_{\text{eff}} \cdot \mathbb{I}(\text{Correct}_i).$$

Here μ_L is computed over the correct subset within a prompt group. The non-positive efficiency adjustment removes explicit bonuses for being shorter than the mean, reducing the risk of over-compression.
Results. Overall 49.5 (+1.6), mean length ratio 0.854; AIME25 40.0, AMC 84.3.

DCBE-GRPO (Failure)

Motivation. DCBE-GRPO introduces a relative-length-improvement efficiency term and difficulty gating, aiming for stable compression without accuracy collapse.
Method.

$$RLI_i = \frac{\mu_L - L_i}{\mu_L + \epsilon},$$
$$\alpha_{\text{diff}} = \left(\frac{N_{\text{correct}}}{N_{\text{total}}}\right)^2,$$

$$A_{\text{total}} = A_{\text{acc}} + \lambda \cdot \alpha_{\text{diff}} \cdot RLI_i \cdot \mathbb{I}(\text{Correct}_i).$$

where μ_L denotes the mean length computed over the correct subset within a prompt group. In the implemented recipe, this efficiency path is paired with a dynamic entropy controller. **Results.** Overall 27.8 (−20.0), mean length ratio 1.333 (length increases); AIME25 6.7, AMC 45.8. This therefore serves as a negative case for the combined design, where the RLI-style efficiency term alone does not explain the observed failure.

A.2.5 Summary

Across these representative cases, improvements are associated with finer signal hierarchy design (*e.g.*, validity separation and difficulty-aware reweighting), rather than stronger penalties alone. Failure cases indicate that stabilization or efficiency objectives can still degrade hard reasoning when not coordinated with task difficulty and signal structure.

A.3 Case Study of Evolutionary Discovery Workflow

To illustrate how the framework accumulates and reuses evidence, we present a detailed trace of a single evolutionary lineage, visualized in Figure 5 and grounded in the full held-out results in Table 2. This case study shows how stronger variants emerge under a shared empirical protocol, and how failures, revisions, and follow-up hypotheses are linked across iterations.

The chosen lineage follows the path: GRPO → RF-GRPO → BN-GRPO → AV-GRPO → VM-AV-GRPO. This path is selected because it captures two informative phases of the search process. In the mechanism-family grouping used in Section 4.2, it begins with **Stabilization and Constraint Control** (RF-GRPO, BN-GRPO) and culminates in **Core Signal Modeling** (AV-GRPO, VM-AV-GRPO):

1. **A Performance Jump:** Starting from insufficient use of failure-group signals and unstable sparse-success amplification, the lineage passes through stabilization attempts and a regression before reaching analytic-variance scaling, which is associated with a clear performance gain.

2. **A Follow-up Correction:** After that gain, the system continues to diagnose a residual false-positive incentive on valid-but-wrong samples and corrects it with validity masking, validity-gated length normalization, and adaptive entropy control.

Root: GRPO Baseline

Status: Baseline (Overall: 47.8, AIME25: 26.7%)

The lineage begins with standard GRPO. While competitive on broader benchmark coverage, the framework identifies a clear bottleneck on sparse-success benchmarks such as AIME25.

- **Diagnosed Deficiency:** Under sparse success, the training signal faces two structural challenges: (i) when only a few samples succeed in a group, the within-group variance can be tiny, inflating advantage scale and inducing update shocks and distribution collapse; and (ii) in all-wrong groups, rewards provide little discriminative relative structure, yielding weak directional learning signals. This makes it difficult to balance exploration with consolidating rare successful trajectories on hard prompts.

Attempt 1: Reference Fallback and Damped Normalization (RF-GRPO)

Status: Targeted gain (Overall: 48.1, AIME25: 26.7% → 33.3%)

To mitigate sparse-success shocks and failure-mode lock-in, the framework proposes RF-GRPO.

- **Mechanism:** It introduces damped advantage normalization to cap outlier updates, performance-gated KL fallback to the reference policy for all-fail groups, and hierarchical format rewards to provide denser structural feedback.
- **Outcome:** AIME25 improves over the baseline, but the overall gain remains limited and failure-group noise can still interfere with learning on the hardest problems.
- **Diagnosis:** In this setting, the stabilizing controls improve training stability, but a remaining issue persists: if failure-group noise is normalized without distinction, signal contamination can still hinder hard-task learning.

Stabilization Attempt: Batch-Level Scaling (BN-GRPO)

Status: Stable but regressive (Overall: 44.2, AIME25: 26.7%)

To reduce failure-noise contamination, the framework proposes BN-GRPO, decoupling group-centering and batch-level scaling, while moving KL/entropy regularization explicitly into the optimization objective.

- **Outcome:** While motivated by improved stability, evaluation shows a regression in aggregate performance and no improvement over the baseline on AIME25.
- **Takeaway:** Stabilization alone is not equivalent to hard-task progress. If the advantage scale is not explicitly aligned with success rarity, suppressing drift can also attenuate the positive signal needed for rare-success consolidation.

The Success Leap: Analytic-Variance Scaling (AV-GRPO)

Status: Breakthrough (Overall: 50.9, AIME25: 26.7% → **36.7%**)

Based on the diagnosis above, the framework proposes AV-GRPO, replacing empirical-scale heuristics with analytic-variance scaling for advantage normalization.

- **Mechanism 1: Analytic-Variance Advantage Scaling.** Advantages are normalized by a theoretically grounded variance term, so that rarer successes naturally produce stronger positive updates.
- **Mechanism 2: Stricter format constraints and conservative entropy regularization.** This reduces invalid exploration noise while preserving effective sparse-signal learning.
- **Evidence:** This iteration yields a clear jump on both aggregate and hard-task metrics, including an AIME24 score of 56.7, consistent with the view that stronger rare-signal amplification helps difficult reasoning.

The Research Loop: Validity-Masked Correction (VM-AV-GRPO)

Status: Refinement gain (Overall: 52.5, AIME25: 36.7% → 43.3%)

After the AV-GRPO breakthrough, the framework continues diagnosing a residual mismatch: when invalid samples are included in the same statistics, valid-but-wrong responses can be over-rewarded.

- **Mechanism:** VM-AV-GRPO applies validity masking with Bayesian-smoothed valid-subset statistics, assigns a floor penalty to invalid outputs, and combines validity-gated length normalization with adaptive entropy control.
- **Outcome & Diagnosis:** The update improves Overall and AIME25 while also yielding a concurrent gain on AMC (AV-GRPO: $81.9 \rightarrow$ VM-AV-GRPO: 88.0). This pattern is consistent with the view that separating structural validity from reasoning quality leads to a cleaner sparse-reward signal hierarchy.
- **Interpretation:** This step links an empirical gain to a residual mismatch and a subsequent mechanism revision, rather than treating the first improvement as the end of the search.

A.4 Detailed Evolutionary Trajectories and Evidence Accumulation

This subsection expands the concise analysis in Section 5 by making the intermediate turning points, failed hypotheses, and cross-lineage transfer patterns more explicit.

A.4.1 Retrospective Evidence on Parent Retention

Across the 21 comparable parent-selection rounds in the base branch, three parents are retained in each round, yielding 63 retained-parent decisions in total. These retained sets cover about 1.8 top-level branches on average, and 14 rounds span at least two branches.

More importantly, a parent's score at selection and its later contribution are not equivalent. Among the 62 retained-parent decisions whose descendants can be observed later in the lineage, 43 yield a descendant with higher Overall than the selected parent itself, with a median gain of $+2.6$ Overall; 27 improve by more than 3 points, and 12 improve by more than 5 points. In particular, 27 of the 33 selections whose parent is below the GRPO baseline eventually lead to descendants that surpass that baseline.

Table 3: Representative selected parents whose strongest descendants substantially outperform them. Scores are Overall. Several of these cases arise in rounds where a lower-scoring selected parent later outperforms the branch of an initially stronger selected parent.

Parent	Overall	Best descendant	Gain
BN-GRPO	44.2	VM-AV-GRPO (52.5)	+8.3
Anchor-GRPO	45.2	LNA-GRPO (50.6)	+5.4
RA-GRPO	44.4	MSA-GRPO (50.7)	+6.3
GLA-GRPO	47.0	PR-GRPO (49.4)	+2.4
CFA-GRPO	46.3	OCE-GRPO (48.4)	+2.1

The evidence becomes sharper when we compare parents selected in the same round. In more than half of the comparable rounds ($11/21$), the parent that eventually produces the strongest descendant is not the selected parent with the highest Overall score at selection time; across all comparable within-round parent pairs, $26/50$ reverse order between Overall score at selection and eventual best-descendant score. One representative round is as follows (numbers in parentheses denote Overall scores): ZA-GRPO (47.3), BN-GRPO (44.2), and Anchor-GRPO (45.2) are retained together; the strongest later descendant is VM-AV-GRPO (52.5) from BN-GRPO, whereas the ZA-GRPO branch peaks at 48.4. Another representative round is as follows (numbers in parentheses denote Overall scores): LAV-GRPO (46.5), GA2-GRPO (49.2), and RA-GRPO (44.4) are retained together; here the selected parent with the lowest Overall score, RA-GRPO, later reaches MSA-GRPO (50.7), exceeding the GA2-GRPO branch peak of 48.2. Representative cases are listed in Table 3. These observations support the intuition behind retaining parents for future improvement rather than filtering solely by current Overall score; they constitute retrospective rather than controlled causal evidence.

A.4.2 Evolution Dynamics: Exploration and Refinement in Mechanism Space

The evolutionary trajectory reveals a stepwise refinement process: each generation typically introduces a relatively localized mechanism revision, or a compact bundle of coordinated changes, on top of its parent and is then filtered by the evaluation protocol. To explain these turning points, we refer to three core training signals: (i) **advantage**, which governs the relative weight of rare successes versus failures in updates; (ii) **KL divergence**, which constrains deviation from a reference model and controls update size; and (iii) **entropy**, which helps

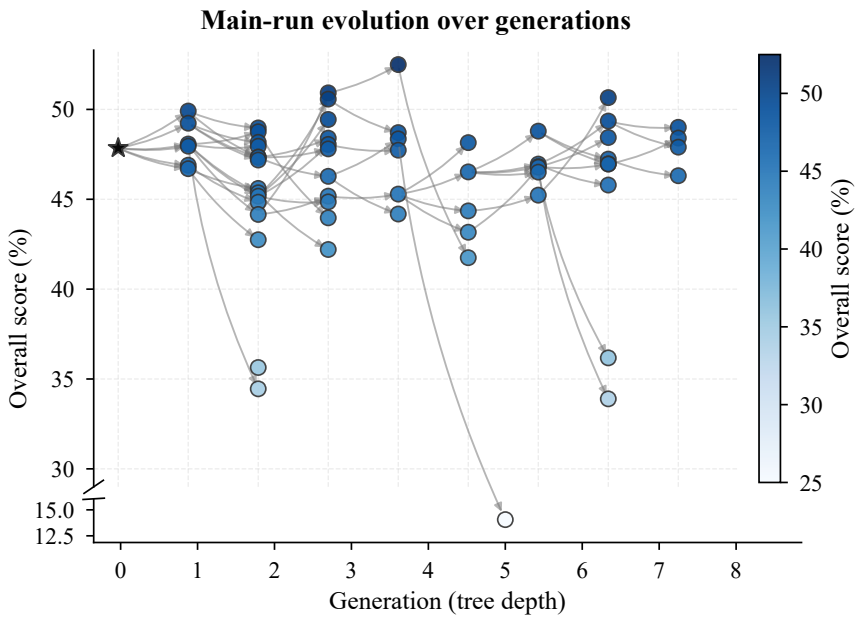

Figure 6: Complete view of the full base-branch lineage by tree depth. Nodes are plotted by tree depth and Overall score, and arrows indicate parent-child relations. Relative to the summary frontier view in Figure 4, this figure makes the underlying structure explicit: deeper generations still contain exploratory failures, and several strong descendants emerge only after intermediate regressions or branch-specific refinements.

diagnose premature collapse or excessive drift.

The traces suggest that performance jumps often coincide with restructuring and decoupling learning-signal pathways rather than simply increasing a single signal magnitude, together with recalibrating penalties on failures and stability constraints. We trace three representative lineages that align with three of the four mechanism families in the main text; the remaining family, **Stabilization and Constraint Control**, appears as an important precursor within the first lineage:

(i) Core Signal Modeling (with Stabilization and Constraint Control Precursors). GRPO (Overall 47.8, AIME25 26.7) → RF-GRPO (Overall 48.1, AIME25 33.3) → BN-GRPO (Overall 44.2, AIME25 26.7) → AV-GRPO (Overall 50.9, AIME25 36.7) → VM-AV-GRPO (Overall 52.5, AIME25 43.3). **Main transition:** This lineage refines how sparse successes are safely amplified. The GRPO baseline suffers from entropy collapse caused by gradient shocks from rare successes, and provides no usable relative signal in all-fail groups. RF-GRPO introduces a stabilizing control: damped normalization caps outlier advantages, dynamic reference fallback prevents deterministic failure modes, and hierarchical rewards provide dense format feedback. BN-GRPO further diagnoses a side effect of per-group whitening: failure groups dominated by format noise can be amplified to the same magnitude as success groups, diluting the rare-success signal needed for hard tasks. To break this bottleneck, AV-GRPO shifts to explicit rare-event modeling by anchoring normalization to analytic variance so that rarer successes can yield stronger positive updates. VM-AV-GRPO then addresses a false-positive failure mode: when invalid samples enter the statistics, valid-but-wrong outputs are often assigned positive advantages. It restores the strict gradient hierarchy (invalid $\ll$ valid-wrong $<$ valid-correct) via validity masking with valid-subset statistics, combining this with validity-gated length normalization and an adaptive entropy controller to further improve AIME25 and Overall, though navigating trade-offs on certain hard metrics.

(ii) Failure-side Control and Failure Reshaping. GRPO (Overall 47.8, AIME25 26.7) → FA-GRPO (Overall 49.9, AIME25 36.7) → DFR-GRPO (Overall 49.0, AIME25 33.3). **Main transition:** This lineage targets all-fail stagnation and related failure modes. FA-GRPO attempts to break the zero-gradient issue by branching on group success: it assigns hierarchical fixed penalties in all-fail groups (mild for logic errors, stronger for format errors). However, subsequent evidence reveals two pitfalls of fixed penalties: mixing scalars with

normalized advantages causes scale mismatch that destabilizes formatting, and applying uniform negative advantages acts as "directionless repulsion," often triggering drift and verbose failures. DFR-GRPO addresses this by separating the signals: it decouples format and reasoning into two independent advantage streams, and replaces uniform repulsion with a zero-centered signal derived from token entropy. This signal compares more informative failures against less informative ones, so the policy receives a directional gradient even when all candidates are incorrect, providing more robust learning signals in all-fail groups, albeit with trade-offs on the hardest benchmarks.

(iii) Signal Routing and Conditional Normalization. GRPO (Overall 47.8, AIME25 26.7) $\rightarrow$ GLA-GRPO (Overall 47.0, AIME25 33.3) $\rightarrow$ PR-GRPO (Overall 49.4, AIME25 36.7) $\rightarrow$ CDA-GRPO (Overall 49.0, AIME25 30.0). **Main transition:** This lineage illustrates the pitfalls of scalarizing multiple signals too early. While GLA-GRPO improves stability, it leaves all-fail groups with little usable learning signal. PR-GRPO makes routing explicit by partitioning groups into solved/failed/mixed regimes: it silences mastered groups, penalizes all-fail groups, and normalizes only mixed-group residuals to improve signal fidelity. CDA-GRPO addresses a complementary bottleneck: aggregating outcome, format, and efficiency (length) into a single scalar fundamentally causes interference, and hard-silencing solved groups creates an efficiency "deadzone." To fix this, CDA-GRPO splits advantages into independent streams and conditionally normalizes efficiency only within the correct subset ("first be correct, then be short"). It replaces brittle regime cliffs with a continuous optimistic baseline, which helps suppress short-and-wrong tendencies caused by signal conflict. This supports a core heuristic: secondary objectives should be introduced through signal decoupling and conditional normalization so that they complement, rather than dilute, the primary correctness signal.

A.4.3 Evidence Accumulation and Cross-Lineage Transfer

Based on standardized evaluation across all 64 nodes, we view each iteration as a mechanism-level experiment: each proposal states an anticipated bottleneck and a corrective hypothesis, while evaluation results and key training signals (advantage/KL/entropy) provide feedback for later revisions. Importantly, this feedback does not remain isolated within a single path. In later iterations, negative outcomes help mark out practical design limits, while components that repeatedly correlate with gains are reused in new combinations. We summarize three primary modes of cross-lineage knowledge transfer:

Negative-evidence-driven path pruning. Negative examples are often fed back into later proposals as algorithm-level "failure diagnoses" (*e.g.*, the FA$\rightarrow$DFR progression in Section A.4.2). For instance, FA-GRPO attempts to inject gradients into all-fail groups via fixed scalar penalties, but DFR-GRPO's diagnosis reveals that mixing fixed penalties with normalized advantages causes scale mismatch, and applying identical negative signals to all failures lacks directionality, inducing policy drift. Consequently, later designs tended to decouple format control from reasoning quality into comparable independent streams, and to introduce zero-centered token-entropy signals in all-fail groups to eliminate directionless repulsion.

Horizontal diffusion of stability mechanisms. Stability components frequently diffuse across paths. For example, advantage clipping and explicit KL regularization are reused across multiple representative paths (*e.g.*, BN-GRPO, DFR-GRPO, and later variants) to suppress excessively large updates when amplifying rare successes. Later variants incorporate these regularizers into the loss function and pair them with clipping to control drift, shifting stability control from a single global hyperparameter to more fine-grained local constraints. This reuse reduces the overhead of exploring new algorithmic combinations, allowing more of the search budget to be spent on signal-structure changes rather than repeated recovery from unstable training.

Recurring empirical regularities. As evolution deepens, recurring heuristics across branches gradually consolidate into stable empirical rules: first, in success-scarce regimes, penalties on failures should generally avoid coupling with length to prevent a bias toward short but uninformative answers; second, secondary objectives are better introduced via "signal decoupling and conditional normalization" so that the primary signal does not overwhelm finer-grained objectives. VM-AV-GRPO's fix for false-positive advantages on "valid-

but-wrong" samples, and CDA-GRPO's "first be correct, then be short" conditional logic, both follow the same pattern: establish comparable, routable learning signals first, then introduce secondary preferences under controlled conditions.

In summary, lineage evolution can be viewed as a process of evidence accumulation. Negative examples mark exploration boundaries, component reuse lowers exploratory overhead, and recurring regularities give later proposals a stronger starting point. This helps improve search efficiency and the quality of algorithm discovery under a fixed computational budget.

A.5 Training Dynamics under Length-Compression Constraints

Table 4 provides a quantitative summary of all 11 nodes in the length-compression branch (1 GRPO baseline + 10 evolved variants). To keep the table readable, we report four representative dataset columns explicitly, while Overall and mean length are still computed over the full six-dataset evaluation suite. Mean length is computed as the unweighted mean word count over the six evaluation datasets.

Figure 7 illustrates the training dynamics of representative successful variants evolved under the length-compression constraint. Together with the failure cases summarized in the representative algorithm descriptions (Section A.2), these traces suggest that effective compression is associated with controlled length reduction while maintaining comparatively stable entropy and reward trajectories, rather than applying an indiscriminately stronger length penalty.

Algorithm	AIME24 pass@32	AIME25 pass@32	AMC pass@32	MATH acc@1	Overall	Mean Length	Length Ratio
GRPO	50.0	26.7	84.3	72.4	47.8	473.6	1.000
DACE-GRPO	56.7	40.0	86.7	72.2	51.7	335.7	0.709
CAG-GRPO	46.7	40.0	84.3	73.0	49.5	404.4	0.854
DRE-GRPO	56.7	36.7	80.7	69.6	49.4	318.0	0.671
PR-GRPO	53.3	36.7	80.7	71.2	49.4	532.5	1.124
PAS-GRPO	50.0	30.0	85.5	73.4	48.5	367.7	0.776
DCE-GRPO	43.3	33.3	80.7	72.4	47.4	484.0	1.022
DQG-GRPO	43.3	30.0	80.7	72.0	46.4	325.5	0.687
MCE-GRPO	43.3	33.3	80.7	65.8	45.2	239.5	0.506
CAS-GRPO	43.3	23.3	75.9	69.6	43.1	1092.3	2.306
DCBE-GRPO	10.0	6.7	45.8	65.2	27.8	631.2	1.333

Table 4: Quantitative summary of all 11 nodes in the length-compression branch. Mean length denotes the unweighted mean word count across the six evaluation datasets. Length ratio is computed relative to the GRPO baseline.

Figure 7: Training dynamics under the length-compression constraint for representative successful variants. The plots compare entropy, reward, and response length over training.