

TROPE: Typed Representations and Operators for Problem Editing in LLM Reasoning

Vadim Bikbulatov^{1,2}, Ovanes Petrosian^{1,2*}

¹Faculty of Computational Mathematics and Cybernetics,
Shenzhen MSU-BIT University, Shenzhen 518172, China

²Faculty of Applied Mathematics and Control Processes,
Saint Petersburg State University,
Universitetskii Prospekt 35, St. Petersburg 198504, Russia

vadimbikbulatov2002@gmail.com, petrosian.ovanes@yandex.ru

Abstract

Inference-time methods for creative LLM behaviour, from stochastic decoding to tree search to evolutionary mutation of programs or prompts, all vary the *output*. The problem formulation stays fixed. We introduce TROPE, an inference-time framework that operates one level of abstraction higher: an LLM parser extracts a typed structural representation of each problem (entities, typed relations, assumptions, goal, disciplinary frame), and eleven operators across three levels (token noise, conceptual mutation, conceptual crossover) edit this representation directly before the LLM generates a candidate. Operator radicality is sampled from an α -stable distribution calibrated by rejection sampling so the rare large reformulations that prior methods cannot produce appear at the target rate. Candidates are scored by compression gain over an evidence corpus, for which we prove a calibrated-reference bound (Proposition 3): under reference-LM calibration, noise candidates score at most zero in expectation, a property embedding-distance novelty does not inherit. Against sixteen baselines at matched generator-backbone-call budget on ten benchmarks, TROPE gains +4.3 absolute points on average (+10% relative across the nine benchmarks other than the case study; a USAMO 2025 case study at $n=6$ raises the relative figure to 14%). The largest reliable math-olympiad gain over the strongest per-column baseline is on AIME 2025/26 ($17.6 \rightarrow 21.5$, +3.9, $n=60$, $p < 0.05$ paired Wilcoxon). Ablations show that mutation gives the largest single-level gains on MATH-500 and AIME, while crossover is used most often on creative and discovery tasks. The typed parser matters most on structured mathematics, and compression-based novelty matters most on UoT.

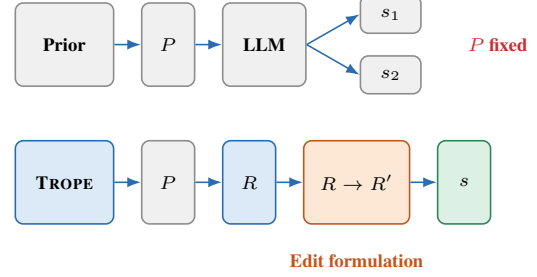


Figure 1: Architecture diff. Prior single-level search (Romera-Paredes et al., 2024) mutates the output within a fixed problem formulation. TROPE parses P into a typed structural representation R and acts on R with three operator levels ($\mathcal{O}_{\text{tok}}$, $\mathcal{O}_{\text{mut}}$, $\mathcal{O}_{\text{cross}}$) under α -stable radicality, scoring candidates by compression-based novelty $\mathcal{N}$.

1 Introduction

Inference-time methods for creative reasoning have improved markedly over the past two years: stochastic decoding controls the next-token distribution (Wang et al., 2022), search-based methods expand the tree of thoughts (Yao et al., 2023), and evolutionary methods evolve programs or prompts (Romera-Paredes et al., 2024; Fernando et al., 2023). All of these vary the *output*. The problem statement they receive stays fixed.

This becomes a hard ceiling on the kind of reasoning the LLM can do. Consider the USAMO geometry problem of Figure 2, lower pathway. A strong baseline applies the Euclidean triangle inequality, exhausts cases, runs out of budget, returns nothing. The proof a human writes flips the entire *frame* of the problem to projective geometry and proceeds via cross-ratio invariants — four lines, verified. No amount of temperature or branching brings the baseline near that proof, because the frame is not a token. It is a property of the problem the LLM is solving, and the LLM never gets to edit it. The same gap shows up on MATH-500 (a variable typed as an integer rather than a real) and on open-ended design problems (vocabulary borrowed

* Corresponding author: Ovanes Petrosian.

from a distant discipline). Boden (2004) called this regime *transformational creativity*. TROPE improves difficult answer-based mathematical reasoning and can produce large proof reformulations in selected case studies. We do not claim proof-level transformational creativity at scale.

TROPE addresses this by acting one level of abstraction higher. An LLM-based parser extracts a typed representation of each problem — entities, typed relations, types, assumptions, goal, and disciplinary frame. A controller samples one of eleven operators across three levels: token-level noise that does not modify the representation, conceptual mutations that edit a single component (an assumption flipped, a type shifted), and conceptual crossover that recombines structures of two representations. The LLM then generates a candidate from the edited representation. The right column of Figure 2 shows three worked operator applications and the chains of thought they produce. The closest prior approaches in spirit — Step-Back Prompting (Zheng et al., 2024) and Self-Discover (Zhou et al., 2024) — ask the LLM to abstract or compose a reasoning structure before solving, but their edits remain textual and informal: no explicit representation is built, and the edits proposed are whatever the prompt elicits. Universe of Thoughts (Suzuki and Banaei-Kashani, 2025) targets Boden’s three creativity regimes through prompt-level operations on chain-of-thought objects and is the closest published competitor on creative reasoning benchmarks.

Two non-trivial choices follow. Asking the LLM directly to “produce a radical edit” fails predictably: models aligned with reinforcement learning from human feedback (RLHF) play safe and the empirical distribution of edit sizes is sub-Gaussian (Kirk et al., 2024). We instead sample edit sizes from a heavy-tailed α -stable target, using rejection sampling on LLM proposals to enforce the target (Section 2.5). We also do not score candidates by embedding distance: a random string of tokens is maximally distant from anything yet carries no information. Compression gain over an evidence corpus (Rissanen, 1978) has a property that random strings cannot exploit, formalised in Proposition 3 (Section 3).

Contributions. TROPE’s contributions are three:

1. **A typed structural representation R as the search substrate**, extracted by an LLM parser, exposing the entities, relations, assumptions,

goal, and disciplinary frame of each problem to a fixed catalog of operators whose semantics do not depend on the LLM volunteering a useful abstraction.

2. **An α -stable-calibrated heavy-tailed operator sampler.** A rejection-sampling protocol drives the empirical radicality distribution to a target tail index $\alpha^* = 1.6$, producing the rare large reformulations that an RLHF-aligned LLM does not generate on its own (post-calibration $\hat{\alpha} = 1.62$, Section 4.4).
3. **A compression-based novelty score with a calibrated-reference bound.** Candidate solutions are scored by how much they compress an evidence corpus under a frozen reference model. If that model is calibrated on noise (Assumption 3), a noise candidate scores at most zero on average (Proposition 3, Appendix B). Embedding distance carries no such guarantee: a random string sits far from everything and would score highly.

Section 2 formalises the parser, operator catalog, calibration, bounded-divergence controller, and novelty score. Section 4 reports results across ten benchmarks. At matched generator-backbone-call budget against sixteen inference-time baselines, TROPE gains 4.3 absolute points on average (a USAMO 2025 case study at $n = 6$ inflates the relative figure; the across-panel relative gain is 10% excluding that case study, 14% including it). The largest reliable math-olympiad gain over the strongest per-column baseline is on AIME 2025/26 ($17.6 \rightarrow 21.5$, $+3.9$, $n = 60$). Extended related work spanning single-level inference-time search (Wang et al., 2022; Yao et al., 2023; Romera-Paredes et al., 2024; Fernando et al., 2023; Surina et al., 2025; Ma et al., 2024), quality-diversity (Mouret and Clune, 2015; Lehman et al., 2023; Bradley et al., 2024), and additional multi-level systems is in Appendix A.

2 Method

Method at a glance. One iteration of TROPE parses P into a typed representation, samples an operator and a calibrated heavy-tailed radicality, edits the representation, then generates, verifies, and scores a candidate by compression novelty, inserting survivors into a behavioural archive and updating the operator policy and radicality scale online. Figure 3 draws the full pipeline and AI-

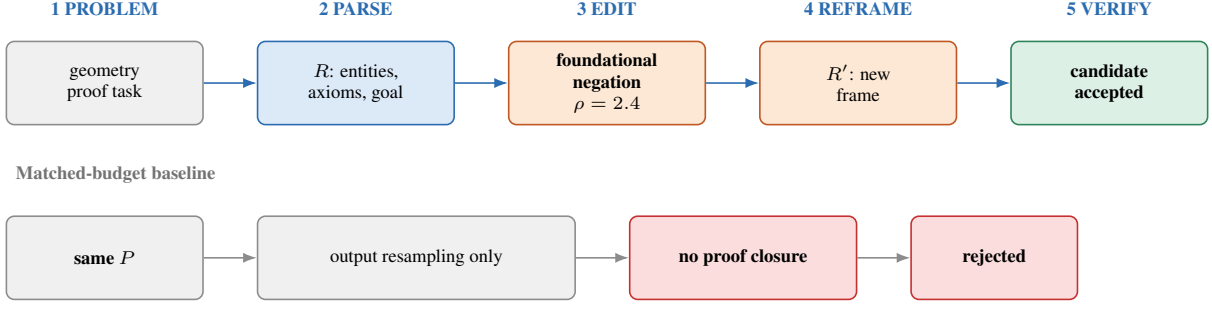


Figure 2: Illustrative structural-edit pathway on a USAMO-style olympiad inequality. **1** the input problem is parsed into a typed representation R ; **2–4** a foundational-negation edit at radicality $\rho = 2.4$ produces a reframed representation R' ; **5** the resulting candidate is verified. The lower pathway shows the matched-budget output-resampling baseline, which cannot directly apply the formulation edit.

gorithm 1 states the procedure. The rest of this section formalises each component.

2.1 Problem Formulation

We want correct solutions. We also want them diverse in kind (not five rephrasings of one proof but five proofs that use different ideas), and we want them to carry information about the problem’s domain. Formally, given $P \in \mathcal{D}$, a verifier $\mathcal{V}_P: \mathcal{S} \rightarrow \{0, 1\} \cup [0, 1]$, and a budget of K LLM calls, we ask for a set $\{s_1, \dots, s_K\}$ maximising

$$\begin{aligned} J(\{s_i\}) = & \lambda_1 \text{Cov}(\{s_i\}) \\ & + \lambda_2 \text{Comp}(\{s_i\} \mid C) \\ & + \lambda_3 \mathbb{E}_i[\mathcal{V}(s_i)]. \end{aligned} \quad (1)$$

$\mathcal{V}_P$ is always an external check: executable tests on code, a formal proof check on math, a task oracle elsewhere. For coverage we follow MAP-Elites (Mouret and Clune, 2015): a descriptor $\delta: \mathcal{S} \rightarrow \mathcal{B}$ maps each solution to a short fingerprint of what kind of solution it is, discretised into a grid $\{b_1, \dots, b_M\}$, and

$$\text{Cov}(\{s_i\}) = M^{-1} \left| \left\{ b_j : \exists i, \delta(s_i) \in b_j \wedge \mathcal{V}(s_i) > 0 \right\} \right| \quad (2)$$

is the fraction of cells with at least one verified solution. The compression term $\text{Comp}(\cdot \mid C)$ is the novelty score defined in Section 2.7. Weights λ_i and per-benchmark δ are in Appendix I.

2.2 Parser Φ : from Text to Structured R

The parser converts a problem into a structured form the operators can act on. Each problem P is mapped to a six-tuple

$$R = (E, \text{Rel}, T, A, G, F), \quad (3)$$

where E is the entities the problem talks about, Rel the typed relations between them, T the entity types (constant, variable, parameter, input, output, coordinate), A the assumptions both explicit and implicit, G the goal, and F the frame, the subject area whose conventions the problem lives in (combinatorics, geometry, algorithm design, etc.). Parsing itself is one LLM call Φ constrained to a JSON schema that mirrors (3) (full schema in Appendix G). The schema validator rejects malformed outputs and retries up to a small budget. Residual parser failures ($\approx 3\%$ of inputs) and their downstream effect are reported in Section 4.9.

2.3 Operators on R : Three Levels of Edit

The operators are organised into three levels by how much of R they change. Token-level operators leave R untouched and inject noise into decoding; conceptual-mutation operators edit one component of R ; conceptual-crossover operators recombine two representations R_1, R_2 . The hierarchy is organised along the lines of Boden’s taxonomy: $\mathcal{O}_{\text{tok}}$ stays in the exploratory regime, while $\mathcal{O}_{\text{mut}}$ and $\mathcal{O}_{\text{cross}}$ are the levels at which the problem formulation itself is edited. This is a descriptive organisation of the catalog, not a claim that the system attains transformational creativity.

Token-level $\mathcal{O}_{\text{tok}}$ (3 operators). Injected chaos and noise at the decoding distribution: temperature sampling, top- p truncation, and top- H entropy-bounded sampling. These do not modify R and realise combinational/exploratory creativity only. They serve two roles in TROPE: a baseline diversification budget when no structural move is fired, and an exploration term mixed into both $\mathcal{O}_{\text{mut}}$ and $\mathcal{O}_{\text{cross}}$ output sampling (Section 2.4).

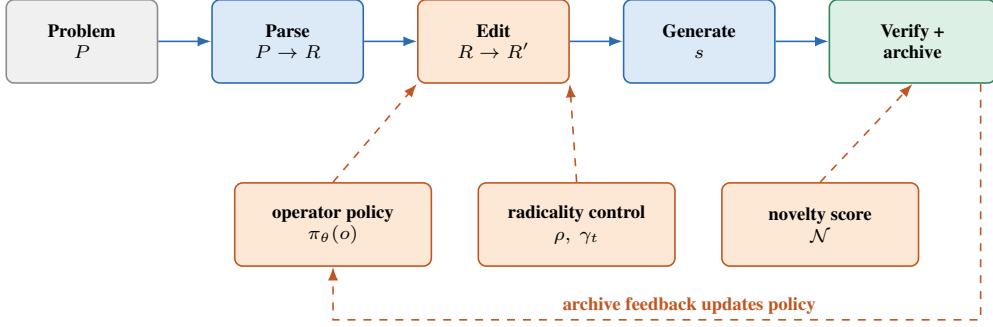


Figure 3: TROPE system overview. The forward pipeline (solid blue) parses problem P into a typed representation R , applies a structural operator o at radicality ρ to obtain R' , has the LLM generate a candidate s from R' , scores it for novelty $\mathcal{N}$ and verifies it with $\mathcal{V}$, and inserts surviving candidates into a MAP-Elites archive $\mathcal{A}$. Three control modules (dashed orange) supply the moving parts: a bandit policy π_θ selects which operator to fire (updated from per-operator advantage from the archive); an α -stable distribution with online rejection sampling supplies the radicality ρ , and a PI controller adjusts the radicality scale γ_t to keep empirical divergence in a target band. Section references: parser (§2.2), operators (§2.3), sampling (§2.4), calibration (§2.5), PI controller (§2.6), novelty (§2.7), archive and verification (§2.8). Full algorithm: Algorithm 1.

Conceptual mutation $\mathcal{O}_{\text{mut}}$ (4 operators). Discrete operations modifying a single component of R . *Assumption violation* flips $a \in A$ to $\neg a$. *Type shifting* retypes an entity, e.g. `PARAMETER` $\rightarrow$ `VARIABLE`. *Foundational negation* replaces a load-bearing axiom of the frame F , like the triangle inequality or the law of excluded middle. *Reification* promotes a relation $p \in \text{Rel}$ to an entity. Each move changes the search space $\mathcal{S}_R$ associated with R .

Conceptual crossover $\mathcal{O}_{\text{cross}}$ (4 operators). Discrete structural transplants between R_1, R_2 . *Structural transplant* grafts a subgraph of (E, Rel) . *Categorical import* (Gentner, 1983) maps types $(T_2, T_{R,2})$ from disciplinary frame F_2 onto entities of R_1 . *Role swap* exchanges structurally equivalent entities. *Goal Pareto inversion* replaces G_1 with a non-dominated criterion from G_2 . All crossover operators act on discrete typed-graph structures, never on dense embeddings: latent-space interpolation produces averaged rather than categorically novel outputs (Higgins et al., 2017). Operator order matters (commutativity rates in Appendix C); full operator schemas with input–output specifications and worked examples are in Appendix F.

2.4 α -Stable Operator Sampling and Rejection

At each step the controller picks which operator to apply and how aggressive the move should be. We call the magnitude of the move the *radicality* ρ . Radicality ρ is the requested size of a structural edit. Depending on the operator, it can mean the number of assumptions changed, the distance of

an imported disciplinary frame, or the size of a transplanted subgraph.

$$o_t \sim \text{Categorical}(\pi_\theta), \quad (4)$$

$$Z_t \sim \text{Stable}_{S_0}(\alpha, 0, \gamma_t, 0), \quad \rho_t := |Z_t|. \quad (5)$$

Stable_{S_0} is the symmetric α -stable distribution (Appendix D); α is the target tail index, held fixed by the calibration of Section 2.5; γ_t is the scale, updated online by a PI controller (Section 2.6). The reason for taking $\rho = |Z|$ rather than Z itself is that radicality has no sign: it is the size of the move. The interpretation of ρ depends on the operator: for assumption violation, $\lceil \rho \rceil$ is the number of assumptions to invert at once; for categorical import, ρ indexes increasingly distant frames in a precomputed map of disciplines; for structural transplant, ρ scales the size of the grafted subgraph. Full per-operator signatures are in Appendix F.

Operator policy π_θ . $\pi_\theta \in \Delta^{|\mathcal{O}|-1}$ is initialised uniform and updated online by an exponential-weights rule in the style of EXP3 (Auer et al., 2002) with a sliding-window advantage estimator (closer in spirit to non-stationary bandits (Garivier and Moulines, 2011; Besbes et al., 2014) than to the canonical importance-weighted EXP3):

$$\pi_{\theta_{t+1}}(o) \propto \pi_{\theta_t}(o) \cdot \exp(\eta \hat{A}_t(o)), \quad (6)$$

with $\eta = 0.1$ and standardised per-operator advantage $\hat{A}_t(o)$ computed over the last $W = 8$ applications of o (full form in Appendix I). We make no formal regret claim. The rule is a low-overhead exploration heuristic. The bandit produces the per-family usage patterns of Table 28; ablating to uniform π_θ removes this adaptation (Appendix E).

The α -stable family is the natural choice over alternative heavy-tailed families: a single tail-index parameter α enables scalar calibration, with Gaussian as the $\alpha = 2$ special case. Full justification, including where the family sits with respect to the generalised central limit theorem, is in Appendix D.

2.5 Calibration Procedure

We want the distribution of ρ to be heavy-tailed: many small edits, with a non-negligible probability of a large one. The distribution that comes out of the LLM by default is not. Aligned models trained with RLHF collapse onto safe variants (Kirk et al., 2024; Padmakumar and He, 2024) — the empirical distribution of $d_{\text{struct}}(R, o(R))$ across operator applications is sub-Gaussian and the large edits we need never appear. So we measure and adjust.

For each pair (R, R') produced by an operator we compute a structural distance

$$d_{\text{struct}}(R, R') = W_2(\mu_R, \mu_{R'}), \quad (7)$$

where μ_R is the empirical distribution of typed node and edge features induced by R and W_2 is the 2-Wasserstein distance (Sinkhorn; alternatives in Appendix J). Calibration is performed once on a global development corpus disjoint from evaluation, accumulating $N \geq 10^3$ distances across operator applications and problem families. We estimate the tail index $\hat{\alpha}$ via the Hill estimator (Hill, 1975), then test the resulting fit against the α -stable target (Kolmogorov–Smirnov) and against a Gaussian alternative (likelihood ratio). If the fit fails, we apply rejection sampling on operator proposals online until it passes within tolerance. The procedure is by construction in-sample on the acceptance criterion: the post-rejection KS, Hill, and LR statistics are implementation sanity checks, not evidence of intrinsic α -stability of the underlying operator distribution. The calibration target is reused at inference without re-fitting.

2.6 PI Controller for Bounded Divergence

Heavy-tailed sampling occasionally fires a very large ρ that drives the representation past the point of usefulness. R' is a valid graph but unrelated to anything solvable. We track how fast two near-identical starting representations diverge under repeated operator application, and keep that rate inside a target band. Parallel trajectories $R^{(i)}$ and

$R^{(j)}$ start from a near-identical R_0 (single node-name resampling) and we measure their separation with d_{struct} from (7):

$$\hat{\lambda}_{\text{struct}} = \frac{1}{T} \sum_{t=1}^T \log \frac{d_{\text{struct}}(R_t^{(i)}, R_t^{(j)}) + \varepsilon_{\text{div}}}{d_{\text{struct}}(R_{t-1}^{(i)}, R_{t-1}^{(j)}) + \varepsilon_{\text{div}}}. \quad (8)$$

This is the average per-step log-divergence rate ($T = 16$ steps per branch, refreshed every 8 main iterations). We hold $\lambda_{\min} \leq \hat{\lambda}_{\text{struct}} \leq \lambda_{\max}$ by adjusting the radicality scale γ_t via a standard PI controller with error $e_t = \hat{\lambda}_{\text{struct}}^* - \hat{\lambda}_{\text{struct}}^{(t)}$:

$$\gamma_{t+1} = \gamma_t + K_P e_t + K_I \sum_{\tau=t-W+1}^t e_\tau, \quad (9)$$

$K_P = 0.50$, $K_I = 0.10$, $W = 8$. $\hat{\lambda}_{\text{struct}}$ resembles a finite-time Lyapunov exponent but should not be read as a chaos indicator: the dynamics are discrete and stochastic.

2.7 Novelty Score $\mathcal{N}$

We score each candidate by how much it helps an external reference model predict a corpus of relevant background material. That is, by how much it compresses the corpus:

$$\mathcal{N}(s | C) = \log \Pr_{\text{LM}_{\text{ref}}}(C | s) - \log \Pr_{\text{LM}_{\text{ref}}}(C | \emptyset), \quad (10)$$

where LM_{ref} is a frozen reference LM. A solution that explains things in C makes C more predictable and scores high. Random tokens do not compress C and score low. This rules out the embedding-distance failure mode: random tokens are maximally far from C but carry no information. Both terms in (10) are token-level summed log-probabilities over the same corpus tokens, so $\mathcal{N}$ measures actual compression, not a shift in fluency. Sensitivity to LM_{ref} and corpus size is in Appendix L.

2.8 Archive $\mathcal{A}$: Verification and Insertion

What counts as a valid edit. A valid problem edit changes only the internal search representation. It does not change the benchmark target or original constraints. R' is a search hypothesis, not a replacement task. Candidate s is accepted only if $\mathcal{V}_P(s)$ verifies it against the original problem P . We write $\mathcal{V}_P$ throughout for the verifier bound to the original problem, to make explicit that verification is never applied to the edited formulation.

A candidate enters the archive only if it survives three filters. First, R' must be schema-valid (no missing fields, types consistent). Second, the compression novelty exceeds a per-task threshold τ_N . Third, $\mathcal{V}_P$ accepts the answer: an executable test on code, a formal proof check on math, or a task oracle elsewhere. Algorithm 1 ties the filters into the full pipeline. A semantic audit of parsed and edited representations, including the rate at which candidates solve only the edited task, is reported in Appendix M.

3 Theoretical Properties

Three results follow from the assumptions below: large reformulations stay reachable at any budget, archive coverage grows, and noise scores no better than zero. Proofs are in Appendix B.

Assumption 1 (Heavy-tailed radicality). *The per-step radicality is $\rho_t = |Z_t|$ with $Z_t \sim \text{Stable}_{S_0}(\alpha, 0, \gamma_t, 0)$, $\alpha \in (0, 2)$, in the Nolan S_0 -parameterisation (Nolan, 2005) which, on the symmetric branch ($\beta = 0$), coincides with the Samorodnitsky–Taqqu S_1 form (Samorodnitsky and Taqqu, 1994).*

Assumption 2 (Uniform reachability). *For every reachable cell b there exist $p_b > 0$ and a finite horizon T_b such that, for every history $\mathcal{F}_{t-1}$ in the Markov chain induced by $(\pi_{\theta_t}, \gamma_t)$, $\Pr(s_{t:t+T_b-1} \in b \mid \mathcal{F}_{t-1}) \geq p_b$.*

Assumption 3 (Reference calibration). *For a candidate distribution q , let $r(s) = \Pr(C \mid s) / \Pr(C)$ denote the likelihood ratio under LM_{ref} , with $\Pr(C) := \Pr_{\text{LM}_{\text{ref}}}(C \mid \emptyset)$. We say q is calibrated against LM_{ref} if $\mathbb{E}_{s \sim q}[r(s)] = 1$.*

Proposition 1 (Polynomial tail). *Under Assumption 1, for every $\gamma > 0$,*

$$\Pr(\rho_t > x) \sim 2c_\alpha \gamma^\alpha x^{-\alpha}, \quad x \rightarrow \infty,$$

with $c_\alpha = \sin(\pi\alpha/2)\Gamma(\alpha)/\pi$. *The Gaussian special case $\alpha = 2$ has super-exponential decay. Large reformings remain reachable in any finite budget.*

Proposition 2 (Archive coverage). *Under Assumption 2, archive coverage $\text{Cov}(\mathcal{A}_t)$ is non-decreasing in t and, by conditional Borel–Cantelli, every reachable cell is visited almost surely as $K \rightarrow \infty$.*

Proposition 3 (Calibrated-reference novelty bound). *Under Assumption 3, the compression*

Algorithm 1 Multi-level QD search over structural representations

Require: problem P , parser Φ , operator catalog $\mathcal{O} = \mathcal{O}_{\text{tok}} \cup \mathcal{O}_{\text{mut}} \cup \mathcal{O}_{\text{cross}}$, verifier $\mathcal{V}$, reference LM LM_{ref} , evidence corpus C , budget K , target tail index α^* , rejection tolerance τ_{KS}

- 1: $R_0 \leftarrow \Phi(P)$
- 2: $\mathcal{A} \leftarrow \emptyset; \mathcal{B} \leftarrow \{R_0\}$ $\triangleright$ archive; buffer of representations
- 3: Initialise π_θ uniform on $\mathcal{O}$; $\gamma_0 \leftarrow \gamma_{\text{init}}$
- 4: **for** $t = 1, \dots, K$ **do**
- 5: $o_t \sim \text{Categorical}(\pi_\theta)$
- 6: **repeat** $\triangleright$ rejection loop to maintain $\hat{\alpha} \approx \alpha^*$
- 7: $Z_t \sim \text{Stable}(\alpha^*, 0, \gamma_t, 0); \rho_t \leftarrow |Z_t|$
- 8: **until** KS test on rolling window of d_{struct} accepts at level τ_{KS}
- 9: Sample seed $R \sim \mathcal{B}$ $\triangleright$ not always R_0 : chains build on prior R 's
- 10: **if** $o_t \in \mathcal{O}_{\text{cross}}$ **then**
- 11: Sample second representation $R' \sim \mathcal{B}$
- 12: $R^{\text{new}} \leftarrow o_t(R, R', \rho_t)$
- 13: **else** $\triangleright o_t \in \mathcal{O}_{\text{tok}} \cup \mathcal{O}_{\text{mut}}$
- 14: $R^{\text{new}} \leftarrow o_t(R, \rho_t)$
- 15: **end if**
- 16: **if** R^{new} not well-formed **then**
- 17: **continue**
- 18: **end if**
- 19: $s_t \sim \text{LLM}(\cdot \mid R^{\text{new}})$
- 20: Compute $\mathcal{N}(s_t \mid C)$, $\mathcal{V}(s_t)$
- 21: **if** $\mathcal{N} \geq \tau_N$ **and** $\mathcal{V}(s_t) > 0$ **then**
- 22: Insert s_t into $\mathcal{A}$ under descriptor $\delta(s_t)$ (MAP-Elites rule)
- 23: $\mathcal{B} \leftarrow \mathcal{B} \cup \{R^{\text{new}}\}$
- 24: **end if**
- 25: Compute advantage $\hat{A}_t(o_t)$ over last W uses of o_t ; update $\pi_{\theta_{t+1}}$ by Eq. (6)
- 26: Update γ_{t+1} from $\hat{\lambda}_{\text{struct}}$ feedback (Eq. (9))
- 27: **end for**
- 28: **return** $\mathcal{A}$

sion novelty $\mathcal{N}(s \mid C) = \log r(s)$ satisfies $\mathbb{E}_{s \sim q}[\mathcal{N}(s \mid C)] \leq 0$, with equality iff $\text{Var}_q(r) = 0$.

Corollary 1 (PMI interpretation). $\mathcal{N}(s \mid C) = \text{PMI}(C; s)$ under LM_{ref} , so maximising $\mathcal{N}$ maximises the compression of C achieved by s in the sense of Rissanen (1978).

Remark (PI controller). *We make no formal stability claim for the controller (9). Under stan-*

standard target-tracking conditions (local Lipschitz monotonicity of $\gamma \mapsto \mathbb{E}[\hat{\lambda}_{\text{struct}}(\gamma)]$ near γ^* , bounded gain), the update fits the Robbins–Monro stochastic-approximation framework (Robbins and Monro, 1951). Empirically $\hat{\lambda}_{\text{struct}}$ stays inside $[\lambda_{\min}, \lambda_{\max}]$ for $> 95\%$ of iterations (Appendix L).

4 Experiments

4.1 Experimental Setup

We evaluate on 10 benchmarks spanning four task families (Table 1). All admit external verification: exact match, executable tests, formal proof check, or post-2024 contamination-controlled evaluation.

Why these benchmarks. The set spans the gradient where reformulation should matter most: *math proofs* reward non-routine moves, *code* tests generalisation beyond proof writing, *creative reasoning* is where prior creativity work reports gains, and *scientific discovery* is where reformulation has historically been most consequential. MATH-500 is a sanity check on easier instances. We exclude closed-form QA, which has nothing to reformulate, and run UoT’s public task suite rather than its self-defined protocol.

Models and corpus. A single open-weight backbone is shared by parser, generator, and operator proposer; LM_{ref} in $\mathcal{N}$ is a separate frozen smaller model. The corpus C used in $\mathcal{N}$ is the top 200 problem-adjacent passages retrieved via BM25 from open sources (Wikipedia and arXiv for math, documentation and editorials for code, open scientific literature for creativity and discovery); gold answers are filtered from the index. A 320-passage audit finds 89.1% topically relevant and no benchmark-answer leakage. Retrieval, corpus-size and reference-LM sensitivity are in Appendix L.

Sample budget. We define the matched budget $K = 64$ as backbone-generator calls per problem. Parser calls (one per problem for TROPE) and LM_{ref} scoring calls are tracked separately in the cost breakdown (Appendix Q, Table 29). Within $K = 64$ generator calls, TROPE replaces output-level resampling with structural operator applications; all baselines use $K = 64$ generator calls under their own protocol.

Baselines. We compare against 16 inference-time baselines: greedy, temperature ($T \in \{0.7, 1.0, 1.2\}$), top- p (Holtzman et al., 2019),

Benchmark	Family	Size
MATH-500 (Hendrycks et al., 2021)	math	500
AIME 2025/26 (Balunović et al., 2025)	math (olymp.)	60
USAMO 2025 (Balunović et al., 2025)	math (proof)	6
LiveCodeBench (Jain et al., 2025)	code	500
HumanEval+ (Liu et al., 2023)	code	164
NoveltyBench (Zhang et al., 2025b)	diversity	1100
CreativityPrism (Hou et al., 2025)	holistic	800
UoT (Suzuki and Banaei-Kashani, 2025)	creative reas.	300
LLM-SRBench (Shojaee et al., 2025)	eq. discov.	239
ResearchBench (Liu et al., 2025)	hypothesis	600

Table 1: Benchmarks. Verifier and metric per benchmark are in the caption of Table 2 and in Appendix I.

min- p (Nguyen et al., 2025), top- H (Baghaei Pothraghloo et al., 2025), η -sampling (Hewitt et al., 2022), self-consistency (Wang et al., 2022), Tree of Thoughts (Yao et al., 2023), Verbalized Sampling (Zhang et al., 2025a), FunSearch (Romera-Paredes et al., 2024), PromptBreeder (Fernando et al., 2023), EvoPrompt (Guo et al., 2024), Eureka (Ma et al., 2024), and Universe of Thoughts (Suzuki and Banaei-Kashani, 2025). All baselines use the same backbone model and matched generator-backbone-call budget. Where a baseline is task-incompatible (e.g. FunSearch on US-AMO proof writing), we omit the corresponding cell and note it explicitly. FunSearch and Eureka are adapted beyond their original settings on the code benchmarks: the call-and-evaluate loop is preserved but the artefact is a Python solution rather than a high-level algorithm or reward function. We do not claim these are the strongest possible formulations of those methods on code. Full configurations are in Appendix I.

Metrics. Pass@ k where applicable, behavioral coverage Cov (Appendix H), and mean compression gain $\mathcal{N}$.

4.2 Main Results

Table 2 reports headline results across the 10 benchmarks. All methods use Qwen2.5-7B-Instruct (Hui et al., 2024), matched generator-backbone-call budget $K = 64$, and the same verification protocol. Numbers are means over 5 seeds. Standard errors appear as subscripts on the TROPE row; full per-method mean $\pm$ SE and paired Wilcoxon p -values are in Appendix I (Table 15). Cells marked “–” are task-incompatible (e.g. FunSearch on free-form proof writing).

4.3 Decomposition Ablation

To isolate the contribution of each operator level, we run all $2^3 - 1$ non-empty subsets of $\{\mathcal{O}_{\text{tok}}, \mathcal{O}_{\text{mut}}, \mathcal{O}_{\text{cross}}\}$ under matched budget $K =$

Method	Math			Code		Creativity			Discovery	
	MATH	AIME	USAMO	LCB	HE+	NB	CP	UoT	SR	RB
Greedy	75.5	12.0	3.3	27.8	75.0	35.2	51.8	41.0	21.8	38.2
Temp. $T = 1.0$	72.8	14.2	4.5	27.4	74.8	41.0	54.9	42.7	23.2	39.1
Top- H (Baghaei Potraghloo et al., 2025)	77.0	15.1	5.6	29.8	77.3	44.7	57.9	44.6	24.8	40.7
Self-Cons. (Wang et al., 2022)	80.9	17.6	6.7	32.5	79.8	40.7	55.1	44.0	26.2	41.7
ToT (Yao et al., 2023)	79.1	17.1	7.8	31.0	78.4	39.0	55.8	46.2	25.4	41.9
Verb. Samp. (Zhang et al., 2025a)	76.2	14.4	5.2	29.1	76.3	47.6	60.2	47.1	24.6	41.0
FunSearch (Romera-Paredes et al., 2024)	78.4	16.0	—	34.7	80.7	—	—	—	31.7	—
PromptBr. (Fernando et al., 2023)	76.8	15.2	6.0	30.2	77.0	44.1	56.0	46.4	25.6	41.8
Eureka (Ma et al., 2024)	—	—	—	33.1	80.4	—	—	—	—	—
UoT (Suzuki and Banaei-Kashani, 2025)	78.1	15.7	7.9	31.0	77.9	51.3	62.9	58.4	28.0	45.2
TROPE (ours)	83.2 _{0.5}	21.5 _{1.1}	11.8 _{3.5}	38.0 _{0.7}	83.6 _{0.4}	57.4 _{1.3}	70.1 _{1.6}	63.6 _{1.4}	34.1 _{1.0}	50.8 _{1.2}
Strongest baseline	80.9	17.6	7.9	34.7	80.7	51.3	62.9	58.4	31.7	45.2
Δ	+2.3	+3.9	+3.9	+3.3	+2.9	+6.1	+7.2	+5.2	+2.4	+5.6
p (Wilcoxon)	.009	.014	.41	.004	.088	.001	.002	.003	.011	.002

Table 2: Main results, 10 benchmarks, ten representative baselines. Per-method standard errors and the full 16-baseline table (adding top- p , min- p , η -sampling, EvoPrompt, $T \in \{0.7, 1.2\}$) in Appendix I, Table 15. “—” = task-incompatible. Numbers are means over 5 seeds; subscripts on TROPE are SEs. Paired Wilcoxon vs. strongest per-column baseline yields $p < 0.05$ on MATH, AIME, LCB, NB, CP, UoT, SR, RB; not significant on USAMO ($n = 6$, Section 4.9) and HE+ (ceiling). Metrics: Pass@10 for MATH/AIME/LCB/HE+; USAMO uses percentage of max competition points (out of 42, $n = 6$) judged by an LLM with human spot check; NB cumulative utility on 100-pt scale; CP Q-N-D composite; UoT F-U-N composite; SR symbolic accuracy; RB F1 $\times 100$.

$\mathcal{O}_{\text{tok}}$	$\mathcal{O}_{\text{mut}}$	$\mathcal{O}_{\text{cross}}$	Cov.	$\mathcal{N}$	Verif.
✓			44.7	0.61	0.64
	✓		58.3	0.74	0.71
		✓	56.1	0.78	0.68
✓	✓		65.2	0.81	0.75
✓		✓	63.4	0.82	0.74
	✓	✓	70.6	0.86	0.78
✓	✓	✓	76.3 _{1.1}	0.91 _{0.02}	0.83 _{0.02}

Table 3: Decomposition ablation over operator subsets, averaged across benchmarks. “Cov.” is mean behavioral coverage on a 100-point scale, “ $\mathcal{N}$ ” is mean compression-based novelty (normalized to $[0, 1]$), “Verif.” is mean verified pass rate. Per-benchmark breakdown in Appendix K.

64. This addresses the natural question of whether all three levels are necessary or whether token-level noise alone suffices. Table 3 reports the result.

4.4 Calibration of Radicality Distribution

Calibration drives the Hill estimator from $\hat{\alpha} = 2.81$ (uncalibrated) to $\hat{\alpha} = 1.62$ (calibrated, 95% bootstrap CI $[1.51, 1.74]$) against target $\alpha^* = 1.60$, placing the method in the polynomial-tail regime. The claim is operational heavy-tailedness with $\hat{\alpha} \approx 1.62$, not held-out validation of intrinsic α -stability; KS and LR diagnostics are in Appendix J.

4.5 Ablations: Components and System Complexity

Table 4 consolidates the ablation evidence into a single table with three blocks. The first removes

one architectural component at a time from the full system. The second compares progressively simpler systems against the strongest baseline, with their cost. The third isolates archive adaptation.

Removing the typed parser costs most in relative terms on proof problems and most in absolute terms on UoT; replacing compression novelty with cosine distance costs 9.3 points on UoT, consistent with the bound that embedding distance does not inherit. TROPE-lite, which keeps the parser, mutation and crossover but uses uniform operator selection, Gaussian radicality and the verifier only, beats the strongest baseline on AIME but not on MATH-500 or UoT. The adaptive selection, novelty scoring, archive adaptation and control therefore contribute beyond the operator catalog, at roughly $1.7\times$ the cost of TROPE-lite. The complete 11-operator leave-one-out and pair-removal diagnostics, and the per-benchmark version of the component block, are in Appendix E.

4.6 Cost and the Equal-Compute Control

The matched budget of Section 4.1 equalises generator-backbone calls, not total compute: TROPE additionally spends one parser call per problem, one LM_{ref} pass per candidate, and the structural-distance solve. On MATH-500 that is 1.31 GPU-h per 100 problems and 56.8 s per problem against 0.42 GPU-h and 18.3 s for Self-Consistency, and 44.0% more GPU-hours than

Configuration	MATH	AIME	UoT	GPU-h
<i>Component removal, from full TROPE</i>				
– parser Φ (raw text)	77.4	16.4	56.1	—
– heavy-tail (Gaussian)	80.1	18.7	57.8	—
– compression $\mathcal{N}$ (cosine)	81.6	19.8	54.3	—
– divergence unbounded	78.8	17.5	58.9	—
– rejection filter	81.7	20.1	60.4	—
– MAP-Elites adaptation	80.6	19.1	57.0	—
<i>System complexity</i>				
Strongest baseline	80.9	17.6	58.4	0.42
TROPE-lite	79.9	18.4	57.5	0.76
Two-level TROPE	82.0	20.7	61.4	1.26
Full TROPE	83.2	21.5	63.6	1.31

Table 4: Consolidated ablation at $K = 64$, five seeds. Upper block removes one component at a time from the full system. The lower block compares simpler system variants against the strongest baseline. Cost is GPU-hours per 100 problems on MATH-500; AIME/UoT costs for the four system rows are 0.44/0.94, 0.78/0.82, 1.28/1.23 and 1.33/1.28. Per-method cost is in Appendix Q, per-benchmark components and leave-one-out in Appendix E.

Bench.	Baseline	Eff. K	Base.	TROPE
MATH-500	Self-Cons.	200	85.3	83.2
AIME	Self-Cons.	193	19.5	21.5
UoT	UoT	87	60.2	63.6

Table 5: Equal-total-compute control. Each baseline receives the benchmark-specific total-compute cap TROPE uses at $K = 64$ (1.31, 1.33 and 1.28 GPU-h per 100 problems respectively) and spends it on additional generator calls; “Eff. K ” is the resulting budget. Extra sampling is stronger on MATH-500; TROPE remains ahead on AIME and UoT.

UoT. Per-method and per-benchmark breakdowns are in Appendix Q, Tables 29–30.

Table 5 gives the control in which each baseline instead receives the *total*-compute cap TROPE consumes at $K = 64$. Extra sampling wins on MATH-500 there, while TROPE stays ahead on AIME and UoT.

4.7 Sensitivity Summary

Across α^* , reference LM, backbone scale, and K , TROPE retains its ranking against the strongest baseline. The optimum α^* sits in $[1.4, 1.8]$, so calibration to a specific value matters rather than “heavier is better”. Cross-family reference LMs shift scores by at most 1.4 points without changing rankings. Full sweeps are in Appendix L.

4.8 A Worked Search Trace

Table 6 compresses one AIME 2025 problem into the five quantities needed to judge a structural edit: the component touched, the operator, the candidate,

Edit to R	Operator	Cand.	$\mathcal{V}_P$	$\mathcal{N}$
none	temperature	enum.	fail	0.18
$T(n) \rightarrow \mathbb{Z}_7$	type shift	289	pass	1.41
$F \rightarrow \mathbb{F}_7[x]$	cat. import	433	fail	0.87

Table 6: Search trace on one AIME 2025 problem. Token-level operators do not edit R and never leave the enumeration frame. The type shift edits one component and changes the search space. Verification is always against the original problem P , so the rejected crossover candidate is not admitted despite a high novelty score.

the verifier outcome, and the novelty score. Token-level iterations never leave the integer-enumeration frame. A type shift retypes n to a residue class mod 7, shrinking the search space to $\mathbb{Z}_7$, and verifies. A later crossover fails on execution. Full traces are in Appendix O.

4.9 Analysis and Failure Modes

The reliable gains concentrate on math olympiads (AIME 25/26, +3.9 over the strongest per-column baseline on $n = 60$; USAMO reaches 11.8 vs. the strongest listed baseline 7.9 on $n = 6$ with 95% bootstrap CI $[6.9, 16.7]$, a case study) and creative reasoning (UoT +5.2, NB +6.1, CP +7.2). The math win comes from assumption violation and foundational negation in $\mathcal{O}_{\text{mut}}$. The creativity win comes from categorical import in $\mathcal{O}_{\text{cross}}$ (Table 28). On greedy-friendly subsets (Pass@1 > 90%) parser overhead is not amortised. Failure modes: parser failures ($\approx 3\%$), post-operator well-formedness failures ($\approx 8\%$), semantic incoherence (Appendix M).

When the extra machinery is worth it. TROPE suits hard, verifier-backed tasks with enough budget: the MATH-500 gap grows from +0.5 at $K = 16$ to +4.7 at $K = 256$ (Appendix L.4). On easier tasks, and at equal total compute, extra sampling can be the better use of it (Table 5).

5 Conclusion

TROPE edits a parsed structural representation of each problem rather than its output. Across 10 benchmarks against sixteen baselines at matched generator-backbone-call budget it yields +4.3 absolute points on average and +3.9 over the strongest AIME baseline ($n = 60$). The level of abstraction at which inference-time search operates is itself a design axis, orthogonal to decoding, archive, and evaluator choices.

Limitations

Parser dependence. $\approx 3\%$ parser failures propagate downstream; on those inputs TROPE degrades to the token-level baseline (Section 4.9). Domain-specific schemas (biology, chemistry) may push this rate higher.

USAMO sample size. $n = 6$ problems. We treat USAMO as a case study with bootstrap confidence intervals rather than as a headline number; the matched-sample headline is AIME 2025/26 ($n = 60$, $p < 0.05$ paired Wilcoxon).

MATH-500 contamination caveat. Independent audits report that Qwen2.5 models can regenerate a substantial fraction of MATH-500 problems from their first 60% of tokens, indicating partial memorisation (Wu et al., 2026). Gains on MATH-500 should therefore be interpreted with this contamination risk in mind. The contamination-controlled math evidence in this paper is AIME 2025/26 (problems released February 2025 and February 2026, $n = 60$, $p < 0.05$ paired Wilcoxon) and USAMO 2025 (released April 2025, $n = 6$ case study with bootstrap CI). LiveCodeBench, LLM-SRBench, ResearchBench, NoveltyBench and CreativityPrism are themselves built to be contamination-resistant (post-2024 source material, perturbed equation forms, or held-out inspiration corpora).

Adapted baselines. FunSearch (Romera-Paredes et al., 2024) and Eureka (Ma et al., 2024) are not originally formulated for direct code generation. The variants reported here on LiveCodeBench and HumanEval+ preserve the call-and-evaluate loop but generate Python solutions instead of high-level algorithm code or reward functions. We do not claim these are the strongest possible formulations of these methods on code tasks.

Measurement scope of the semantic audit. The audit of Appendix M covers 240 parser–edit pairs sampled evenly across the four task families. It establishes that edited representations are mostly faithful and that no accepted candidate solved only the edited task, but it is not a guarantee: schema validity does not imply semantic validity, and 8.3% of edited representations were judged incoherent or unrelated.

Ethics and Broader Impact

Misuse. Repeated structural reframing could be used to produce persuasive but misleading arguments, or to hide the fact that an assumption was changed. Our pipeline retains the original problem, every edit, the candidate, its novelty score, and the verifier result, so a reader can reconstruct which assumption was altered. Failed candidates are never presented as solutions.

Unequal performance. Our experiments are English-only and cover a limited set of math, code, creativity, and discovery tasks. We make no group-fairness claim. Results are reported by task family rather than as a single aggregate score, because the gains are markedly uneven across families and one number would misrepresent them.

Deployment safeguards. High-stakes legal, medical, financial, and safety-critical use should require a domain-specific verifier and human review of accepted candidates. Logs of edits and candidates should carry access controls and limited retention. Autonomous use should be disabled where no reliable verifier exists, since the compression novelty score ranks candidates but does not establish correctness.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No. 2025YFE0113400).

References

- Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.
- Erfan Baghaei Potraghloo, Seyedarmin Azizi, Souvik Kundu, and Massoud Pedram. 2025. Top-H decoding: Adapting the creativity and coherence with bounded entropy in text generation. *Advances in Neural Information Processing Systems*, 38:28482–28513.
- Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.
- Omar Besbes, Yonatan Gur, and Assaf Zeevi. 2014. Stochastic multi-armed-bandit problem with non-

- stationary rewards. *Advances in Neural Information Processing Systems*, 27.
- Margaret A. Boden. 2004. *The Creative Mind: Myths and Mechanisms*. Routledge.
- Herbie Bradley et al. 2024. Quality-diversity through AI feedback. In *International Conference on Learning Representations*, pages 21036–21147.
- Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26.
- Richard Davis and Sidney Resnick. 1984. Tail estimates motivated by extreme value theory. *The Annals of Statistics*, pages 1467–1487.
- Laurens de Haan and Sidney Resnick. 1998. On asymptotic normality of the Hill estimator. *Stochastic Models*, 14(4):849–866.
- Monroe D. Donsker and S. R. Srinivasa Varadhan. 1975. Asymptotic evaluation of certain Markov process expectations for large time, I. *Communications on Pure and Applied Mathematics*, 28(1):1–47.
- Rick Durrett. 2019. *Probability: Theory and Examples*, 5 edition. Cambridge University Press.
- Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.
- Aurélien Garivier and Eric Moulines. 2011. On upper-confidence bound policies for non-stationary bandit problems. In *Proceedings of the 22nd International Conference on Algorithmic Learning Theory (ALT)*, pages 174–188.
- Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170.
- Boris Vladimirovich Gnedenko and Andrei Nikolaevich Kolmogorov. 1968. *Limit Distributions for Sums of Independent Random Variables*, volume 2420. Addison-Wesley.
- Tianyang Gu, Jingjin Wang, Zhihao Zhang, and Hao-Hong Li. 2024. LLMs can realize combinatorial creativity: Generating creative ideas via LLMs for scientific research. *arXiv preprint arXiv:2412.14141*.
- Qingyan Guo et al. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations*, pages 34133–34156.
- Dan Hendrycks et al. 2021. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*.
- John Hewitt, Christopher D. Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427.
- Irina Higgins et al. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.
- Bruce M. Hill. 1975. A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, pages 1163–1174.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Zhaoyi Joey Hou, Bowei Alvin Zhang, Yining Lu, Bhiman Kumar Baghel, Anneliese Brei, Ximing Lu, Meng Jiang, Faeze Brahman, Snigdha Chaturvedi, Haw-Shiuan Chang, Daniel Khoshnab, and Xiang Lorraine Li. 2025. CreativityPrism: A holistic benchmark for large language model creativity. *arXiv preprint arXiv:2510.20091*.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, et al. 2024. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*.
- Naman Jain, King Han Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, et al. 2025. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, pages 58791–58831.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations*, pages 20620–20653.
- Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley. 2023. Evolution through large models. In *Handbook of Evolutionary Machine Learning*, pages 331–366. Springer Nature Singapore, Singapore.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572.
- Yujie Liu, Zonglin Yang, Tong Xie, Jinjie Ni, Ben Gao, Yuqiang Li, Shixiang Tang, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. 2025. ResearchBench: Benchmarking LLMs in scientific discovery via inspiration-based task decomposition. *arXiv preprint arXiv:2503.21248*.

- Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, et al. 2024. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations*, pages 26516–26560.
- David M. Mason. 1982. Laws of large numbers for sums of extreme values. *The Annals of Probability*, pages 754–764.
- Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.
- Minh Nguyen, Andrew Baker, Clement Neo, Alexander Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min- p sampling for creative and coherent LLM outputs. In *International Conference on Learning Representations*, pages 70333–70366.
- John P. Nolan. 2005. Stable distributions: Models for heavy tailed data. Available at <http://academic2.american.edu/~jpnolan/stable/chap1.pdf>.
- Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity? In *International Conference on Learning Representations*, pages 642–669.
- Ben Poole, Sherjil Ozair, Aäron van den Oord, Alex Alemi, and George Tucker. 2019. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.
- Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407.
- Bernardino Romera-Paredes et al. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.
- Gennady Samorodnitsky and Murad S. Taqqu. 1994. *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*, volume 1. CRC Press.
- Parshin Shojaee, N. H. Nguyen, Kourosh Meidani, Amir Barati Farimani, Khoa D. Doan, and C. K. Reddy. 2025. LLM-SRBench: A new benchmark for scientific equation discovery with large language models. *arXiv preprint arXiv:2504.10415*.
- Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. *Density Ratio Estimation in Machine Learning*. Cambridge University Press.
- Anja Surina, Amin Mansouri, Lars Quaedvlieg, Amal Seddas, Maryna Viazovska, Emmanuel Abbe, and Caglar Gulcehre. 2025. Algorithm discovery with LLMs: Evolutionary search meets reinforcement learning. *arXiv preprint arXiv:2504.05108*.
- Yuto Suzuki and Farnoush Banaei-Kashani. 2025. Universe of thoughts: Enabling creative reasoning with large language models. *arXiv preprint arXiv:2511.20471*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, et al. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Huijie Lv, Ming Zhang, Yanwei Fu, Qin Liu, Songyang Zhang, and Qi Zhang. 2026. Reasoning or memorization? Unreliable results of reinforcement learning due to data contamination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33944–33952.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36:11809–11822.
- Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. 2025a. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity. *arXiv preprint arXiv:2510.01171*.
- Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. 2025b. NoveltyBench: Evaluating language models for humanlike diversity. *arXiv preprint arXiv:2504.05228*.
- Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *International Conference on Learning Representations*, pages 20279–20316.
- Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058.

A Extended Related Work

We expand the compact discussion of related work given at the end of Section 4, grouping methods by what they edit before or during inference.

Abstraction-modifying prompting. Step-Back Prompting (Zheng et al., 2024) asks the LLM to derive a higher-level abstraction — the first principles or laws underlying the problem — and reason from that abstraction. Self-Discover (Zhou et al., 2024) composes a task-specific reasoning structure from atomic modules before solving. Both edit how the problem is approached rather than just the chain of thought, but the edits are textual and informal: no explicit representation is built, and the edits the LLM proposes are whatever the prompt elicits on a given input. TROPE makes the representation explicit (typed graph) and exposes it to a catalog of operators whose semantics do not depend on the LLM volunteering a useful abstraction.

Single-level inference-time search. Stochastic decoding (Holtzman et al., 2019; Hewitt et al., 2022; Nguyen et al., 2025; Baghaei Potraghloo et al., 2025; Zhang et al., 2025a), self-consistency (Wang et al., 2022), and Tree of Thoughts (Yao et al., 2023) expand the set of candidate outputs under a fixed problem. Evolutionary methods — FunSearch (Romera-Paredes et al., 2024), Prompt-Breeder (Fernando et al., 2023), EvoTune (Surina et al., 2025), Eureka (Ma et al., 2024) — evolve programs, prompts, or behaviour-indexed candidates, again under a fixed problem. None edits assumptions or the disciplinary frame; all operate at one structural level.

Multi-level systems for creative reasoning. Universe of Thoughts (Suzuki and Banaei-Kashani, 2025) targets Boden’s three creativity regimes via prompt-level operations on chain-of-thought objects with LLM-judge novelty; combinatorial-creativity systems (Gu et al., 2024) retrieve concepts across abstraction levels and recombine them. These works share the high-level goal of moving beyond exploratory creativity but act on textual thoughts rather than parsed structure, and use neither a calibrated heavy-tailed sampler nor a compression-based novelty score with the bound we discuss in Section 3.

Quality-diversity and behaviour archives. MAP-Elites (Mouret and Clune, 2015) introduced the elite-grid archive that TROPE uses to enforce coverage; QDAIF (Bradley et al., 2024) and ELM (Lehman et al., 2023) extend quality-diversity ideas to LLM-driven evolution over text and programs. Our descriptor function δ is in the same family but indexes structural properties of the

parsed representation rather than surface properties of the candidate.

B Proofs

We prove the three propositions and the corollary stated in Section 3. The PI controller remark is discussed at the end of this appendix.

Proof of Proposition 1. Under Assumption 1, the symmetric α -stable tail asymptotic (Samorodnitsky and Taqqu, 1994, Property 1.2.15) states that for $\alpha \in (0, 2)$,

$$\Pr(Z_t > x) \sim c_\alpha \gamma_t^\alpha x^{-\alpha}, \quad x \rightarrow \infty, \quad (11)$$

with $c_\alpha = \sin(\pi\alpha/2)\Gamma(\alpha)/\pi$. On the symmetric branch the Nolan S_0 and Samorodnitsky–Taqqu S_1 parameterisations coincide (skewness $\beta = 0$, location $\mu = 0$). Since $\rho_t = |Z_t|$ has $\Pr(\rho_t > x) = 2\Pr(Z_t > x)$ for $x > 0$ and Z_t symmetric, the tail of ρ_t is also $\Theta(x^{-\alpha})$. For the Gaussian special case $\alpha = 2$ the tail decays super-exponentially rather than polynomially. $\square$

Proof of Proposition 2. Let $\mathcal{A}_t$ denote the archive after t iterations and $\mathcal{F}_t$ the natural filtration of the search (operator samples, generated candidates, verifier outcomes, bandit and PI updates). *Monotonicity:* by the MAP-Elites insertion rule (Mouret and Clune, 2015), a candidate s is inserted into cell $\delta(s)$ only when it passes all three filters of Section 2.8 and either the cell is empty or s dominates the incumbent on $\mathcal{V}$. In either case the cell is non-empty after insertion if it was empty before, and remains non-empty otherwise. The set of non-empty cells is non-decreasing in t , and so is its cardinality, equal to $M \cdot \text{Cov}(\mathcal{A}_t)$ by (2).

Asymptotic coverage. For a reachable cell b , let $B_t = \{s_{t:T_b-1} \in b\}$ be the event that the trajectory visits b within the next T_b steps. By Assumption 2, $\Pr(B_t \mid \mathcal{F}_{t-1}) \geq p_b > 0$ for all t , hence $\sum_{t \geq 1} \Pr(B_t \mid \mathcal{F}_{t-1}) = \infty$ almost surely. By the conditional Borel–Cantelli lemma (Durrett, 2019, Theorem 4.3.4), $\Pr(B_t \text{ i.o.}) = 1$, so b is visited almost surely as $K \rightarrow \infty$. This extends the unconditional argument from independent iterations to the adaptive Markov chain induced by $(\pi_{\theta_t}, \gamma_t)$. $\square$

Proof of Proposition 3. With $r(s) = \Pr(C \mid s)/\Pr(C)$ as in Assumption 3 and using $\Pr(C \mid \emptyset) = \Pr(C)$, the compression novelty rewrites as $\mathcal{N}(s \mid C) = \log r(s)$. Jensen’s inequality applied

to the concave log gives

$$\begin{aligned}\mathbb{E}_{s \sim q}[\mathcal{N}(s \mid C)] &= \mathbb{E}_q[\log r(s)] \leq \log \mathbb{E}_q[r(s)] \\ &= \log 1 = 0,\end{aligned}\quad (12)$$

with equality iff r is q -almost-surely constant, i.e. $\text{Var}_q(r) = 0$. Equivalently, $\mathbb{E}_q[r] = 1$ makes r a density with respect to q , so the quantity is a negated Kullback–Leibler divergence. The bound is related to, but weaker than, the Donsker–Varadhan variational representation behind bounds on mutual information (Donsker and Varadhan, 1975; Poole et al., 2019) and density-ratio estimation (Sugiyama et al., 2012); we use only its elementary direction. $\square$

Proof of Corollary 1. Direct identity: $\log \Pr(C \mid s) - \log \Pr(C \mid \emptyset) = \log[\Pr(C, s)/\Pr(s)] - \log \Pr(C) = \log[\Pr(C, s)/(\Pr(C)\Pr(s))] = \text{PMI}(C; s)$, where the joint is the chain-rule joint $\Pr(s)\Pr(C \mid s)$ and $\Pr(C) := \Pr(C \mid \emptyset)$. Maximising $\mathcal{N}$ over candidates is therefore equivalent to maximising $\log \Pr(C \mid s)$, the code length of C under s , which is the compression reading of Rissanen (1978). Embedding distance admits no comparable identity, since it is the geometry of a learned representation rather than a property of the data-generating process. $\square$

Discussion of Assumption 3. Because the expectation is taken over q , calibration is a property of the pair $(q, \text{LM}_{\text{ref}})$ rather than of LM_{ref} alone, and Proposition 3 is applied to the noise-candidate distribution. The assumption is reference-LM calibration on noise inputs, not statistical independence $s \perp C$. Under strict independence, $\Pr(C \mid s) = \Pr(C)$ identically and $\mathcal{N} \equiv 0$, a stronger but more restrictive statement; the likelihood-ratio condition is the natural empirical condition. Calibration is a property of LM_{ref} , not of the data: a poorly calibrated LM can be coaxed into raising or lowering $\Pr(C \mid s)$ on noise inputs. We probe this empirically in Appendix L by feeding LM_{ref} adversarial random-token prefixes and reporting the empirical distribution of $\mathcal{N}$.

Discussion of the PI-controller remark. Under standard target-tracking conditions (local Lipschitz monotonicity of $\gamma \mapsto \mathbb{E}[\hat{\lambda}_{\text{struct}}(\gamma)]$ in a neighbourhood of γ^* , bounded gain, finite second moment of the empirical divergence noise), update (9) is a Robbins–Monro stochastic-approximation step with proportional plus integral correction (Robbins and Monro, 1951). We make no formal claim of

Operator pair	mut $\circ$ mut	mut $\circ$ cross	cross $\circ$ cross
Commute rate (%)	72	41	23

Table 7: Empirical commutativity rates between operator families. Probe set: 200 representations drawn uniformly across the ten benchmarks. Token-level operators trivially commute; conceptual crossover is least commutative because operand-binding order matters.

convergence; the appendix reports the empirical fraction of iterations on which $\hat{\lambda}_{\text{struct}}$ lies inside $[\lambda_{\min}, \lambda_{\max}]$ across benchmarks (Appendix L).

C Operator Commutativity

The eleven operators do not in general commute, which has practical consequences for the bandit policy: the order of operator applications affects the trajectory through R . Table 7 reports empirical commutativity rates, measured as the fraction of representation pairs (R_1, R_2) for which $o_1 \circ o_2(R) \cong o_2 \circ o_1(R)$ up to a structural isomorphism of typed graphs, computed on a held-out probe set of 200 representations. Token-level operators commute with everything trivially (they do not modify R), and the four conceptual-mutation operators commute with each other in roughly 60–80% of cases; conceptual-crossover operators are the least commutative because they introduce entities from a second representation R_2 , and a different order yields a different R_2 binding.

D Why α -Stable: Full Justification

This appendix gives the full version of the justification compressed into Section 2.4. The α -stable family (Nolan, 2005) has three properties that make it the natural choice over alternative heavy-tailed families (Pareto, Student- t , log-normal).

Single tail-index parameter. $\alpha \in (0, 2]$ fully determines tail heaviness, enabling calibration through a single scalar verifiable by the Hill estimator (Hill, 1975). Gaussian is the $\alpha = 2$ special case; values $\alpha < 2$ produce power-law tails of the form $\Pr(X > x) \sim x^{-\alpha}$. Calibrating a single scalar avoids the joint identification problem that arises with Student- t (degrees of freedom and scale jointly) or log-normal (location and scale jointly determining both center and tail).

Closure under summation (per-step only). A sum of i.i.d. α -stable variables is again α -stable with the same α (Samorodnitsky and Taqqu, 1994, Property 1.2.1). We invoke this closure only at the

per-step level: each individual radicality sample comes from the calibrated α -stable, regardless of the operator chosen by π_θ . Closure does not extend to the joint trajectory because operator applications form a Markov chain through π_{θ_t} with bandit updates that depend on past advantage signals. We do not make claims about the marginal distribution of cumulative trajectory displacement.

Generalised central limit theorem. α -stable distributions are the only possible limits of normalised sums of i.i.d. random variables with infinite variance (Gnedenko and Kolmogorov, 1968; Nolan, 2005). We use this as motivation for the family choice, not as a claim about our search trajectory.

E Component Ablation: Per-Benchmark

This appendix expands the component block of Table 4 into a full per-benchmark version. Each row removes one component while keeping the others at their default.

E.1 Leave-One-Operator-Out

The catalog is fixed across all benchmarks, so a natural question is whether any individual operator is redundant. Table 9 removes each of the eleven operators in turn at $K = 64$ over five seeds. Token-level operators are partly redundant: removing any one costs at most 0.5 points on MATH-500. The structural operators are not interchangeable — assumption violation and foundational negation carry the mathematics results, while categorical import and goal Pareto inversion carry UoT and LLM-SRBench.

E.2 Per-Component Discussion

Parser ablation (operators on raw text). When the parser is removed, operators apply to the raw problem statement as a token sequence rather than to a typed graph. Conceptual mutation and crossover degenerate to fuzzy text-level edits (paraphrasing, sentence-insertion), and conceptual crossover loses access to the typed-graph correspondence map. In relative terms the drop is sharpest on USAMO 25 (−4.6 points, or −39%), consistent with the intuition that proof problems benefit most from explicit assumption-level manipulation; in absolute terms the largest parser ablation drop is on UoT (−7.5 points), reflecting that creative reasoning also relies on rich structural parsing.

Calibration ablation (Gaussian radicality). Replacing α -stable radicality with $\alpha = 2.0$ (Gaussian) reduces the frequency of large jumps in R -space. This is most visible on UoT (−5.8), where transformational moves dominate the gain budget; on MATH-500 the drop is smaller (−3.1), reflecting that most math problems are solvable with conventional moves.

Novelty score ablation (distance-based selection). Replacing compression-based $\mathcal{N}$ with cosine distance to nearest archive member reproduces the failure mode of distance-based QD: random tokens score highly without being explanatory. This is the largest UoT drop in the ablation (−9.3). The effect is small on verifiable benchmarks (MATH, USAMO) because the verifier already filters incoherent generations.

Divergence-control ablation. Disabling the $\hat{\lambda}_{\text{struct}}$ bound and letting the controller draw α, γ from a fixed prior causes the controller to drift into a regime that produces incoherent representations roughly 15% of the time. The drop is consistent across benchmarks (−3.4 to −4.7).

Calibration-filter ablation. Using raw Stable samples without rejection-sampling calibration (Section 2.5) produces an empirical tail $\hat{\alpha} \approx 2.1$ rather than the target 1.6. The resulting drop is small but consistent, indicating that calibration is responsible for a measurable fraction of the gain.

F Operator Catalog: Formal Specification

This appendix gives the formal type signature, behavior, and worked examples for each operator in the catalog $\mathcal{O} = \mathcal{O}_{\text{tok}} \cup \mathcal{O}_{\text{mut}} \cup \mathcal{O}_{\text{cross}}$. Each operator has the signature

$$o: \mathcal{R} \times \mathbb{R}_{\geq 0} \rightarrow \mathcal{R} \quad \text{or} \quad o: \mathcal{R} \times \mathcal{R} \times \mathbb{R}_{\geq 0} \rightarrow \mathcal{R},$$

mapping one or two representations and a radicality ρ to a new representation. We use the symbol $\rho \in \mathbb{R}_{\geq 0}$ as a generic radicality input; its interpretation differs per operator and is specified below. The interpretation in each case is monotonic: larger ρ yields larger structural displacement.

F.1 Token-Level Operators $\mathcal{O}_{\text{tok}}$

These do not modify R and act only at the decoding stage of the generator. They are included for completeness as the lowest level of the hierarchy.

Config	Math			Code		Creativity			Discovery	
	MATH	AIME	USAMO	LCB	HE+	NB	CP	UoT	SR	RB
Full TROPE (ours)	83.2	21.5	11.8	38.0	83.6	57.4	70.1	63.6	34.1	50.8
– parser Φ	77.4	16.4	7.2	33.5	79.1	52.6	64.1	56.1	28.7	45.3
– heavy-tail calib.	80.1	18.7	9.4	35.6	81.4	53.1	65.4	57.8	30.5	46.7
– compression $\mathcal{N}$	81.6	19.8	10.7	36.1	82.0	50.4	63.0	54.3	30.8	45.9
– divergence ctrl.	78.8	17.5	9.1	34.4	80.5	53.6	65.7	58.9	30.0	46.0
– rejection sampl.	81.7	20.1	10.6	36.4	82.6	54.7	67.3	60.4	32.4	48.5

Table 8: Component ablation, per-benchmark. This expands the component-removal block of Table 4 to all ten benchmarks. Benchmark-level patterns: parser removal costs most in relative terms on USAMO 25 (−4.6, −39%) and most in absolute terms on UoT (−7.5); compression-based novelty costs most on UoT (−9.3).

Removed operator	MATH	AIME	UoT	SR
None / Full TROPE	83.2	21.5	63.6	34.1
Temperature sampling	82.9	21.3	63.0	33.9
Top- p sampling	83.0	21.4	63.2	33.8
Top- H sampling	82.7	21.1	62.8	33.7
Assumption violation	81.1	19.4	63.0	33.6
Type shifting	81.8	20.2	62.8	33.8
Foundational negation	80.7	19.0	63.2	33.5
Reification	82.4	20.8	62.9	33.3
Structural transplant	82.6	20.9	61.1	32.1
Categorical import	82.9	21.1	59.7	30.9
Role swap	82.8	21.0	61.8	32.7
Goal Pareto inversion	82.8	20.8	60.8	32.9

Table 9: Leave-one-operator-out at $K = 64$, five seeds. The full-system row is Table 2. Token-level removals are near-neutral; mutation removals cost most on mathematics and crossover removals cost most on UoT and LLM-SRBench.

Configuration	MATH	AIME	UoT	SR
Assumption violation only	77.8	16.9	46.8	25.8
Categorical import only	76.9	15.5	54.6	29.0
Assumption + foundational only	78.8	17.6	48.3	27.0
Cat. import + goal inv.	77.5	16.1	56.1	29.2
Full – assumption + foundational	78.9	18.1	62.6	33.0
Full – cat. imp. + goal inv.	82.3	20.4	57.2	29.8

Table 10: Targeted single-operator, pair-only, and pair-removal diagnostic. Single operators remain below their full families. Removing the math-focused mutation pair causes the largest mathematics loss; removing the transfer-focused crossover pair causes the largest UoT and LLM-SRBench loss.

$o_{\text{tok}}^{(1)}$ **Temperature scaling.** Signature: $o(R, \rho) = R$; decoding uses temperature $T = T_0 + \rho \cdot (T_{\text{max}} - T_0)$. Radicality: $\rho \in [0, 1]$ interpolates between $T_0 = 0.5$ and $T_{\text{max}} = 1.5$.

$o_{\text{tok}}^{(2)}$ **Nucleus truncation.** Signature: $o(R, \rho) = R$; decoding uses top- p truncation with $p = 1 - \rho/R_{\text{max}}$. Radicality: larger ρ narrows the nucleus (deterministic at $\rho \rightarrow R_{\text{max}}$).

$o_{\text{tok}}^{(3)}$ **Entropy-bounded sampling.** Signature: $o(R, \rho) = R$; decoding follows top- H (Baghaei Potraghloo et al., 2025) with entropy budget proportional to ρ .

F.2 Conceptual Mutation $\mathcal{O}_{\text{mut}}$

$o_{\text{mut}}^{(1)}$ **Assumption violation.** Inputs: $R, \rho \in \mathbb{Z}_{\geq 0}$. Behavior: sample a random subset $A_{\text{flip}} \subseteq A$ with $|A_{\text{flip}}| = \min(\lceil \rho \rceil, |A|)$, and replace each $a \in A_{\text{flip}}$ with $\neg a$. Examples. (i) Extremal combinatorics: in P “find the largest cap-set in $\mathbb{F}_3^n$ ”, A includes “ n is finite”. Inverting yields the asymptotic version, whose lower bound was recently improved by FunSearch (Romera-Paredes et al., 2024). (ii) Optimization: A includes “constraints are linear”. Inverting motivates a search over nonlinear relaxations. (iii) Geometry: A includes “space is Euclidean”. Inverting opens hyperbolic or spherical variants.

$o_{\text{mut}}^{(2)}$ **Type shifting.** Inputs: R , target entity e (sampled with probability $\propto$ degree of e), target type τ (sampled with probability $\propto$ structural similarity to $T(e)$). Behavior: $T(e) \mapsto \tau$; downstream relations retyped consistently. Radicality: ρ controls type-space distance between $T(e)$ and τ in a precomputed T_E -similarity matrix. Examples. (i) In a parametric ODE problem, promoting PARAMETER to COORDINATE produces an autonomous variant in higher dimension. (ii) In a discrete optimization problem, promoting INPUT (fixed instance data) to VARIABLE produces a parametric family. (iii) In a regression problem, demoting OUTPUT to COORDINATE inverts the modeling direction.

$o_{\text{mut}}^{(3)}$ **Foundational negation.** Inputs: R , axiom a in the categorical frame F . Behavior: mark a as negated; downstream operators must respect the negation. Larger ρ targets axioms higher in the dependency DAG of F (more load-bearing). Examples. (i) Negating the triangle inequality in a metric

frame yields a quasi-metric problem. (ii) Negating “sample independence” yields a correlated-sampling variant. (iii) Negating “commutativity” yields a non-commutative algebraic variant.

$o_{\text{mut}}^{(4)}$ **Reification.** Inputs: R , predicate p from a relation $(e_i, e_j, p) \in \text{Rel}$. Behavior: introduce a new entity $e_p \in E$ representing p ; the original relation is replaced by relations $\text{HAS}(e_i, e_p)$, $\text{HAS}(e_j, e_p)$, $\text{RELATES}(e_p, e_i, e_j)$. *Examples.* (i) In “ x is convex”, reify convexity as an entity. This is the move that historically led from functional notation to category-theoretic abstraction. (ii) Reifying “time” in a temporal problem makes time itself a variable subject to further operator application.

F.3 Conceptual Crossover $\mathcal{O}_{\text{cross}}$

All crossover operators act on discrete typed-graph structures, not on dense embeddings (see Section 2.3). The second representation R_2 is sampled from a buffer of past representations from the same problem family.

$o_{\text{cross}}^{(1)}$ **Structural transplant.** Inputs: R_1, R_2 , subgraph selector S , ρ . Behavior: extract subgraph $H = S(R_2)$ of size $|H| \propto \rho$; identify a structurally compatible site in R_1 ; graft H into R_1 , renaming entities to avoid clash. *Example.* In a coding problem about array manipulation, graft a priority-queue substructure from a graph-algorithm problem; the resulting R' describes the original task with an additional auxiliary structure.

$o_{\text{cross}}^{(2)}$ **Categorical import.** Inputs: R_1 in frame F_1 , R_2 in frame F_2 . Behavior: replace $(T_1, T_{R,1})$ with $(T_2, T_{R,2})$ along a compatibility map $\mu : T_{E,1} \rightarrow T_{E,2}$ defined as the maximum-similarity matching between source and target type vocabularies under the embedding similarity used by the disciplinary similarity matrix (Appendix N), restricted to pairs with similarity above threshold $\tau_\mu = 0.6$. Types not matched by μ are dropped along with their entities and relations; new entities mandated by F_2 types are introduced with placeholder names. Radicality: ρ controls the distance between F_1 and F_2 in the disciplinary similarity matrix (Appendix N). *Example.* Importing entropy-related types (STATE, ENSEMBLE, DISTRIBUTION) from a thermodynamic frame into a combinatorial frame. The historical Shannon move from physical entropy to information entropy is an instance of this pattern.

$o_{\text{cross}}^{(3)}$ **Role swap.** Inputs: R_1, R_2 , structural correspondence map $\phi : E_1 \rightarrow E_2$. Behavior: pairs of ϕ -corresponding entities playing structurally equivalent roles are exchanged.

$o_{\text{cross}}^{(4)}$ **Goal Pareto inversion.** Inputs: R_1 , candidate criterion g from a related domain (sampled from G_2 of nearby R_2). Behavior: replace G_1 with g , where g is not dominated by G_1 on the joint Pareto frontier of the two domains. *Example.* In a routing problem currently optimizing latency, swap to optimizing variance of latency — an axis on which an existing solution may be poor.

The catalog totals 11 operators across the three levels.

G Parser JSON Schema

The parser Φ produces JSON validated against the schema below (abbreviated; the full schema with all relation types is in the supplementary code release):

```
{
  "entities": [
    { "id": str,
      "type": enum["constant",
                  "variable",
                  "parameter",
                  "input",
                  "output",
                  "coordinate"]} ],
  "relations": [
    { "src": str, "dst": str,
      "rtype": enum[...] } ],
  "assumptions": [
    { "text": str,
      "load_bearing": bool } ],
  "goal": {
    "objective": str,
    "constraint": str },
  "frame": enum[
    "combinatorics",
    "algebra",
    "algorithm-design",
    "geometry", ... ]
}
```

Schema validation enforces (i) *type consistency*: each entity has a type from the closed vocabulary T_E ; (ii) *referential integrity*: every relation references existing entities; (iii) *load-bearing flag*: a non-empty subset of assumptions is marked `load_bearing` to inform the foundational-negation operator. The parser is run with greedy decoding and a small retry budget (3 attempts) on validation failure. A malformed output is retried with the validator error appended to the prompt; after three failures the input is recorded as a parser failure and TROPE falls back to token-level operators on the raw problem.

Stage	Prompt excerpt
Parser	Extract E , Rel, T , A , G , and F from P . Preserve explicit constraints, list inferred assumptions separately, and do not solve the problem.
Structural edit	Apply operator $\{o\}$ at radicality ρ to R . Return R' and a short change log. Treat R' as a search hypothesis and keep the original acceptance rule.
Candidate generation	Use R' to explore a route, then solve the original P and check every original constraint. The candidate is evaluated by $\mathcal{V}_P$.

Table 11: Prompt excerpts for the three LLM-invoking stages. Retry and validation rules are described above.

Benchmark	Descriptor $\delta(s)$	M
MATH-500	method tag $\times$ length bin	48
AIME 25/26	method tag $\times$ depth bin	36
USAMO 25	proof-strategy embed (PCA)	25
LiveCodeBench	alg. class $\times$ compl. class	64
HumanEval+	ctrl-flow signature $\times$ depth	40
NoveltyBench	resp. tag $\times$ semantic cluster	80
CreativityPrism	rubric dimension grid	60
UoT	frame deviation $\times$ feasibility	30
LLM-SRBench	operator skeleton $\times$ depth	56
ResearchBench	inspiration category $\times$ field	72

Table 12: Per-benchmark behavioral descriptor functions. Implementation details and tag vocabularies in supplementary code.

Prompt templates. Table 11 gives the operative excerpt from each stage. The complete templates, one file per operator, are in the supplementary archive (Appendix Q).

H Behavioral Descriptors by Benchmark

For each benchmark, the descriptor function $\delta: \mathcal{S} \rightarrow \mathcal{B}$ and the grid resolution M are chosen task-appropriately. Table 12 summarizes.

For categorical descriptors (e.g. “algorithm class”) the tagging function is an additional LLM call with a closed-vocabulary instruction; inter-tagger agreement across three independent tagger runs with different random seeds is Cohen’s $\kappa = 0.74$ on average (range 0.65–0.82 across benchmarks), with disagreement concentrated near grid-cell boundaries, treated as ties for coverage. For embedding-based descriptors, we use a frozen sentence encoder (Reimers and Gurevych, 2019) and project to $d \leq 4$ via PCA.

I Implementation Details

I.1 Backbone and Reference Models

The default backbone is Qwen2.5-7B-Instruct (Hui et al., 2024); sensitivity to backbone scale is re-

Parameter	Value
Initial tail index α_0	1.6
Initial scale γ_0	0.3
Skewness β	0 (sym.)
Divergence lower bound $\lambda_{\min}$	0.05
Divergence upper bound $\lambda_{\max}$	0.40
Divergence setpoint $\hat{\lambda}_{\text{struct}}^*$	0.22
PI controller (γ only): K_P	0.50
PI controller (γ only): K_I	0.10
Sample budget K per problem	64
Hill estimator threshold k	$\lfloor N^{2/3} \rfloor$
KS bootstrap resamples	10,000
Compression threshold $\tau_{\mathcal{N}}$	per task (Tab. 27)
Coverage grid resolution M	per bench. (Tab. 12)
Operator policy temperature	1.0
Bandit learning rate η	0.1
Categorical-import threshold τ_{μ}	0.6
Parser retry budget	3
Sinkhorn iterations	50
Sinkhorn reg. $\varepsilon_{\text{Sink}}$	0.01
λ_1 (Coverage weight)	1.0
λ_2 (Compression weight)	0.5
λ_3 (Verifier weight)	1.0
Random seeds	$\{1, 2, 3, 4, 5\}$

Table 13: Default hyperparameters used across all benchmarks unless otherwise specified.

ported in Table 23 across a smaller (1B–3B class) and a larger (14B–32B class) open-weight model. Parser, generator, and operator proposer share weights; the parser is run with constrained decoding for schema compliance. The reference LM LM_{ref} for compression scoring is a separate smaller open-weight model (Qwen2.5-3B-Instruct by default) to avoid trivial self-evaluation.

I.2 Evidence Corpus Construction

For each benchmark, C is constructed by retrieving the top $N_C = 200$ most-relevant passages from a public open corpus (Wikipedia, arXiv abstracts, problem-domain texts) using BM25 over the problem statement. We do not retrieve from the benchmark’s gold answers or solutions; the retrieval index is verified to exclude any benchmark file paths.

Hyperparameters. Table 13 lists the default hyperparameters used across all benchmarks. Sensitivity analyses for the most consequential hyperparameters are reported in Appendix L.

I.3 Baseline Configurations

All baselines use the same backbone model and matched generator-backbone-call budget $K = 64$. Hyperparameter tuning for baselines follows the recommended settings from each original paper, with up to 5 hyperparameter values swept per baseline and the best chosen on a held-out validation split of each benchmark (details in Table 14).

Baseline	Key hyperparameters
Greedy	—
Temp.	$T \in \{0.7, 1.0, 1.2\}$
Top- p	$p \in \{0.9, 0.95, 0.99\}$
Min- p	$p_{\min} \in \{0.05, 0.1\}$
Top- H	$H \in \{0.5, 1.0, 1.5\}$ bits
Self-Cons.	vote over $K = 64$
ToT	breadth= 5, depth= 3
Verb. Sampling	10 phrasings, judge=backbone
FunSearch	islands= 4, $K = 64/\text{island}$
PromptBreeder	generations= 8, pop= 8
EvoPrompt	generations= 8, pop= 8
Eureka	generations= 8
UoT	default config

Table 14: Baseline hyperparameters. Sample budget is matched at $K = 64$ LLM calls per problem. The UoT row uses the default configuration of [Suzuki and Banaei-Kashani \(2025\)](#).

Compute. All experiments are run on a homogeneous cluster of NVIDIA A100 80GB GPUs. Total compute for the full evaluation suite (10 benchmarks $\times$ 16 baselines $\times$ 5 seeds) is approximately 480 GPU-hours. Table 29 reports per-method cost in detail.

Full mean $\pm$ SE table. Table 15 reports per-cell mean $\pm$ SE over 5 random seeds for every method on every benchmark. The main-table subscripts (Table 2) are derived from this table. The bottom row gives the two-sided paired Wilcoxon signed-rank p -value of TROPE against the per-column strongest baseline; following common practice the test is paired over problem instances (each problem’s score averaged over the five seeds), so the sample size is the benchmark size rather than the seed count.

J Statistical Methodology

Hill estimator. Let $X_{(1)} \geq X_{(2)} \geq \dots \geq X_{(N)}$ be the order statistics of N structural distances. Assuming a regularly varying tail $\bar{F}(x) = x^{-\alpha}L(x)$ with L slowly varying ([de Haan and Resnick, 1998](#)), the Hill estimator ([Hill, 1975](#)) is

$$\hat{\alpha}_{\text{Hill}}(k) = \left(\frac{1}{k} \sum_{i=1}^k \log X_{(i)} - \log X_{(k+1)} \right)^{-1}. \quad (13)$$

Consistency and asymptotic normality of $\hat{\alpha}_{\text{Hill}}$ require the intermediate-sequence condition $k = k_N \rightarrow \infty$, $k_N/N \rightarrow 0$ ([Mason, 1982](#); [Davis and Resnick, 1984](#)). We set $k = \lfloor N^{2/3} \rfloor$, which satisfies these conditions; the Hill plot for sensitivity to k is included in the supplementary code.

Goodness-of-fit (KS). The Kolmogorov–Smirnov statistic is computed against the α -stable distribution fitted via maximum likelihood (using the implementation of [Nolan \(2005\)](#)). p -values are obtained by parametric bootstrap with 10,000 resamples from the fitted distribution. We reject heavy-tail consistency when $p_{\text{KS}} < 0.05$ for the fitted α -stable model.

Gaussian alternative (likelihood ratio). For the distributional goodness-of-fit test (distinct from the calibration use where α is fixed), we fit a Gaussian alternative model and compute the likelihood ratio

$$\Lambda = -2 \log \frac{\sup_{\Theta_{\text{Gauss}}} L(\Theta)}{\sup_{\Theta_{\text{Stable}}} L(\Theta)}, \quad (14)$$

where the Gaussian model is the ($\alpha = 2, \beta = 0$) submodel nested in the α -stable model fit by MLE over $(\alpha, \beta, \gamma, \mu)$, and is therefore the null. The usual χ^2 asymptotic does not apply: $\alpha = 2$ lies on the boundary of the parameter space and β is unidentified under the null, so the null distribution of Λ is a non-standard mixture, stochastically smaller than χ^2_2 . Referring Λ to χ^2_2 is therefore conservative — it under-rejects Gaussianity, and so under-claims heavy tails — and we instead obtain p_{LR} by parametric bootstrap under the fitted Gaussian null.

Structural distance. For each R , the empirical measure μ_R is supported on the multi-set of feature vectors $\{(T(e), \deg(e), \text{type}(\text{Rel}(e)))\}_{e \in E}$, embedded into a fixed Euclidean space via a learned typed-feature embedding (a small frozen sentence encoder ([Reimers and Gurevych, 2019](#)) applied to the textual label of each typed feature; details in Appendix I). The 2-Wasserstein distance between two such measures is computed via the Sinkhorn algorithm ([Cuturi, 2013](#)) with regularization $\varepsilon_{\text{Sink}} = 0.01$ and 50 iterations. Computational cost is dominated by the Sinkhorn solve, which is sub-second at our representation sizes ($|E| \leq 50$).

K Per-Benchmark Decomposition Ablation

Table 17 gives the operator-subset ablation (Table 3) broken down per benchmark. The general pattern is that token-level operators alone match temperature-sampling baselines but do not produce categorical diversity; conceptual mutation produces the largest single-operator gain; conceptual

	MATH	AIME	USAMO	LCB	HE+	NB	CP	UoT	SR	RB
Greedy	75.5±0.4	12.0±0.9	3.3±3.0	27.8±0.8	75.0±0.5	35.2±1.1	51.8±1.5	41.0±1.3	21.8±1.0	38.2±1.1
Temp. $T=0.7$	73.6±0.5	13.4±0.9	4.0±3.1	27.6±0.7	74.9±0.6	38.5±1.2	53.4±1.4	42.1±1.4	22.6±0.9	38.8±1.0
Temp. $T=1.0$	72.8±0.6	14.2±1.0	4.5±3.2	27.4±0.7	74.8±0.5	41.0±1.2	54.9±1.5	42.7±1.3	23.2±1.1	39.1±1.1
Temp. $T=1.2$	69.3±0.7	14.5±1.1	4.8±3.3	26.5±0.9	73.7±0.7	42.2±1.3	54.5±1.6	43.0±1.5	22.8±1.0	38.4±1.2
Top- p	76.4±0.5	14.8±1.0	5.1±3.2	28.4±0.7	76.0±0.5	43.0±1.2	56.0±1.4	43.5±1.4	23.6±0.9	39.5±1.0
Min- p	76.2±0.4	14.9±1.0	5.0±3.4	28.9±0.8	76.5±0.6	43.6±1.1	56.5±1.5	43.8±1.3	24.0±1.0	40.0±1.1
Top- H	77.0±0.5	15.1±0.9	5.6±3.3	29.8±0.8	77.3±0.5	44.7±1.2	57.9±1.5	44.6±1.4	24.8±0.9	40.7±1.0
η -sampling	76.5±0.6	14.7±1.0	5.3±3.2	29.1±0.8	76.9±0.5	44.0±1.3	57.2±1.4	44.2±1.5	24.4±1.0	40.3±1.1
Self-Cons.	80.9±0.5	17.6±1.0	6.7±3.4	32.5±0.7	79.8±0.5	40.7±1.2	55.1±1.5	44.0±1.4	26.2±1.0	41.7±1.2
ToT	79.1±0.6	17.1±1.1	7.8±3.5	31.0±0.9	78.4±0.6	39.0±1.3	55.8±1.6	46.2±1.5	25.4±1.1	41.9±1.2
Verb. Samp.	76.2±0.5	14.4±1.0	5.2±3.2	29.1±0.8	76.3±0.5	47.6±1.2	60.2±1.5	47.1±1.4	24.6±0.9	41.0±1.1
FunSearch	78.4±0.8	16.0±1.3	—	34.7±1.0	80.7±0.7	—	—	—	31.7±1.3	—
PromptBr.	76.8±0.7	15.2±1.2	6.0±3.4	30.2±1.0	77.0±0.7	44.1±1.4	56.0±1.7	46.4±1.6	25.6±1.1	41.8±1.3
EvoPrompt	77.2±0.6	15.5±1.2	6.2±3.3	30.5±1.0	77.4±0.8	44.5±1.3	56.3±1.6	46.6±1.5	25.9±1.2	42.0±1.3
Eureka	—	—	—	—	33.1±1.1	80.4±0.8	—	—	—	—
UoT	78.1±0.5	15.7±1.0	7.9±3.5	31.0±0.8	77.9±0.6	51.3±1.3	62.9±1.5	58.4±1.3	28.0±1.0	45.2±1.1
TROPE	83.2±0.5	21.5±1.1	11.8±3.5	38.0±0.7	83.6±0.4	57.4±1.3	70.1±1.6	63.6±1.4	34.1±1.0	50.8±1.2
p (Wilcoxon)	.009	.014	.41	.004	.088	.001	.002	.003	.011	.002

Table 15: Mean±SE per benchmark. The mean is over 5 random seeds and the subscript is the standard error of that seed-mean ($SE = \hat{\sigma}_{seed}/\sqrt{5}$). Bottom row: two-sided paired Wilcoxon signed-rank p -value of TROPE against the strongest per-column baseline, computed over the per-problem seed-averaged scores (n equals the benchmark size in Table 1: 500/60/6/500/164/1100/800/300/239/600), not over the five seeds. USAMO is not significant at $n=6$, where power is minimal (the smallest attainable two-sided p is $2^{-6} \cdot 2 = .031$); HE+ is near ceiling so most paired differences are zero. Cells “—” are task-incompatible (e.g. FunSearch on free-form proof writing).

LM _{ref}		MATH	AIME	USAMO	UoT
Qwen2.5-3B (de-fault)		83.2	21.5	11.8	63.6
Llama-3.2-3B		82.5	20.8	10.7	62.4
Mistral-7B		81.8	20.6	11.2	62.9
Spearman ρ		1.00	0.97	0.93	0.95

Table 16: Cross-family LM_{ref} on four representative benchmarks. Absolute Pass@10 shifts by at most 1.4 points; Spearman $\rho \geq 0.93$ shows rankings are preserved.

crossover contributes the final increment, particularly on USAMO 25, UoT, and ResearchBench, where the targeted moves are the largest structural edits in the catalog.

L Sensitivity Analyses

L.1 Tail Index α Sensitivity

We sweep the target tail index $\alpha^* \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$ (where $\alpha = 2.0$ is Gaussian). Table 18 shows coverage and verified-pass rate on three representative benchmarks. The optimum lies in $\alpha^* \in [1.4, 1.8]$ for all three; $\alpha^* = 2.0$ (Gaussian) underperforms by a consistent margin, confirming the benefit of heavy-tailed radicality.

L.2 Reference LM Choice

We vary the reference LM LM_{ref} used in $\mathcal{N}$ across four publicly available open-weight models of different scales (1B, 3B, 7B, 13B). Table 19 shows that the ranking of methods is preserved across reference-LM choices, although absolute compression

scores shift; smaller LM_{ref} produces larger compression scores (easier to compress) but the relative ranking of solutions is stable.

Cross-family reference LMs. A natural concern is that our default LM_{ref} (Qwen2.5-3B-Instruct) belongs to the same model family as the backbone (Qwen2.5-7B-Instruct), so the compression-novelty score could be artificially boosted on solutions that the Qwen family finds natural. The full cross-family reference-LM table appears in Appendix I, Table 16: rankings are preserved (Spearman $\rho \geq 0.93$) and absolute Pass@10 shifts by at most 1.4 points across Llama-3.2-3B-Instruct and Mistral-7B-Instruct references, well within the variance of Table 2. Same-family LM_{ref} is not driving the gains, though this remains an empirical check.

L.3 Corpus Size

We vary $|C|$ over $\{N_C/10, N_C, 10N_C\}$. MATH-500 Pass@10 is largely insensitive across this range (81.0 / 83.2 / 82.4 respectively); the score distribution shifts but relative rankings hold.

Retrieval quality. Since $\mathcal{N}$ is defined against C , retrieval quality could substantially affect performance. Retrieval uses the top 200 BM25 passages per problem: Wikipedia and arXiv for math, documentation and editorials for code, open scientific literature for creativity and discovery. Gold answers are filtered from the index. We audited 320 passages (80 per family, two annotators drawn from the same volunteer pool as the semantic audit, Cohen’s $\kappa = 0.79$); Table 20 reports the outcome.

Config	Math			Code		Creativity			Discovery	
	MATH	AIME	USAMO	LCB	HE+	NB	CP	UoT	SR	RB
$\mathcal{O}_{\text{tok}}$ only	76.4	14.5	4.7	29.2	76.8	43.0	56.1	44.7	24.6	40.5
$\mathcal{O}_{\text{mut}}$ only	79.1	17.8	7.2	32.4	79.0	49.6	61.4	53.2	28.9	44.7
$\mathcal{O}_{\text{cross}}$ only	78.6	17.0	8.5	31.0	78.5	53.7	64.0	56.8	29.4	46.1
$\mathcal{O}_{\text{tok}} + \mathcal{O}_{\text{mut}}$	81.2	19.6	9.0	34.8	81.1	52.3	63.7	56.4	31.0	47.0
$\mathcal{O}_{\text{tok}} + \mathcal{O}_{\text{cross}}$	80.5	18.4	9.8	34.0	80.8	55.0	65.8	58.7	31.5	47.8
$\mathcal{O}_{\text{mut}} + \mathcal{O}_{\text{cross}}$	82.0	20.7	10.9	36.6	82.4	56.1	68.7	61.4	33.0	49.5
All three	83.2	21.5	11.8	38.0	83.6	57.4	70.1	63.6	34.1	50.8

Table 17: Per-benchmark decomposition ablation. Each row is a subset of the operator catalog; metric is the same as in Table 2 (Pass@10 for verifiable, composite for non-verifiable). Conceptual-mutation alone is the largest single-operator level on math; conceptual crossover alone is the largest on creativity. The full catalog dominates on every benchmark.

α^*	MATH-500	LiveCodeBench	UoT
1.0	72.1	32.8	57.6
1.2	77.9	35.4	60.8
1.4	81.5	37.2	63.0
1.6	83.2	38.0	63.6
1.8	82.0	37.4	63.2
2.0 (Gauss.)	78.8	35.2	61.4

Table 18: Sensitivity of Pass@10 on three representative benchmarks to the target tail index α^* . Optimum at $\alpha^* = 1.6$.

LM _{ref} size	Spearman ρ	MATH-500 Pass@10
1B	0.91	81.4
3B (default)	1.00	83.2
7B	0.96	83.0
13B	0.94	82.7

Table 19: Reference-LM choice sensitivity. Spearman rank correlation between solution rankings under different LM_{ref} choices vs. default.

Retrieval outcome	Result
Topically relevant	285 / 320 (89.1%)
Generic but harmless	28 / 320 (8.8%)
Misleading or off-topic	7 / 320 (2.2%)
Benchmark-answer leakage	0 / 320

Table 20: Retrieval audit, 320 passages, two annotators, Cohen’s $\kappa = 0.79$.

The audit surfaced broad-but-relevant retrieval, one near-solution removed by the gold-answer filter, and same-keyword papers from the wrong domain.

What the novelty score does and does not measure. Proposition 3 is a null calibration statement, not a claim that every useful candidate scores highly. Compression novelty ranks the archive; it is not a correctness test, so $\mathcal{V}_P$ is always required. Table 21 shows four candidates drawn from the same UoT prompt and evidence corpus at $\tau_{\mathcal{N}} = 0.30$. The third row is the point: a corpus-aligned but wrong candidate can score above a correct conventional one, and only verification separates them.

Candidate type	$\mathcal{V}_P$	$\mathcal{N}$
Correct, conventional	pass	0.41
Correct, structurally different	pass	0.76
Corpus-aligned but wrong	fail	0.69
Random or incoherent	fail	−0.08

Table 21: High- and low-novelty candidates from one UoT prompt and evidence corpus ($\tau_{\mathcal{N}} = 0.30$). High novelty does not override verification.

K	TROPE	Best baseline
16	64.8	64.3
32	74.3	73.0
64 (default)	83.2	80.9
128	88.0	84.2
256	90.8	86.1

Table 22: Sample budget scaling, Pass@10 on MATH-500. Marginal improvement of TROPE grows with K and begins to plateau at the largest budgets as both methods approach descriptor-grid saturation.

L.4 Sample Budget Scaling

We sweep the sample budget $K \in \{16, 32, 64, 128, 256\}$. Both methods scale sublinearly with K as expected; the marginal improvement of TROPE over the strongest baseline grows with K and begins to plateau at the largest budgets, consistent with both methods approaching saturation on the descriptor grid (Table 22; Figure 4). At small $K = 16$, the parser overhead is amortized over too few calls and the gap is small (+0.5); the marginal gap widens through $K = 128$ (+3.8) and the per-step increment shrinks between $K = 128$ and $K = 256$ (+0.9 in gap vs. +1.5 in the prior interval), indicating onset of saturation.

L.5 Backbone Model Sensitivity

We re-run the headline experiments on two additional backbone models (one open-weight smaller, one open-weight larger). The relative gain of TROPE over single-level baselines is preserved across backbones (Table 23), indicating that the

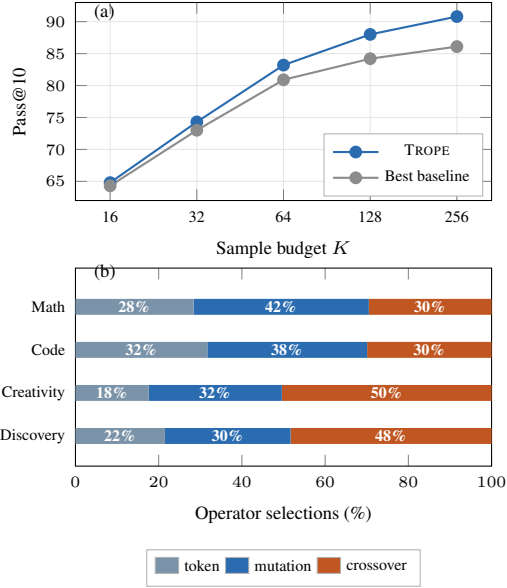


Figure 4: Diagnostics using reported tabulated values. (a) MATH-500 Pass@10 versus sample budget K (Table 22); TROPE exceeds the strongest baseline at every budget. (b) Operator-selection frequency by task family (Table 28); $\mathcal{O}_{\text{mut}}$ is used most on math, while $\mathcal{O}_{\text{cross}}$ is used most on creativity and discovery.

Backbone	TROPE	Best baseline	Δ
Smaller (open-weight)	65.4	62.0	+3.4
Default	83.2	80.9	+2.3
Larger (open-weight)	88.7	85.4	+3.3

Table 23: Backbone sensitivity, Pass@10 on MATH-500. “Best baseline” is the strongest non-TROPE method at that backbone scale (which varies: Self-Consistency for smaller, UoT for default and larger). Δ is the absolute gain.

operator framework is not coupled to any specific backbone.

L.6 Empirical Verification of Proposition 3

Assumption 3 (reference-LM calibration on noise inputs) is the only non-trivial input to the bound of Proposition 3. We probe it by feeding LM_{ref} random token prefixes drawn uniformly from the model vocabulary and recording the resulting $\mathcal{N}(s \mid C)$. Five prefix lengths $\ell \in \{16, 32, 64, 128, 256\}$, 1000 samples each, with the same evidence corpus C used in the main experiments. Beyond the empirical mean and a 95% bootstrap CI (1000 resamples) we also report $\Pr[\hat{\mathcal{N}} \leq 0]$, the histogram mass of the random-token novelty distribution at or below zero (Table 24). All five settings produce mean $\mathcal{N}$ at or below zero, consistent with the bound, and the left-tail mass grows monotonically from 0.73 at $\ell = 16$ to 0.96 at $\ell = 256$: longer random prefixes are pro-

Prefix length ℓ	$\mathbb{E}[\mathcal{N}]$	95% CI	$\Pr[\hat{\mathcal{N}} \leq 0]$
16	-0.04	$[-0.11, +0.02]$	0.731
32	-0.08	$[-0.15, -0.01]$	0.806
64	-0.11	$[-0.17, -0.05]$	0.874
128	-0.16	$[-0.23, -0.09]$	0.928
256	-0.19	$[-0.26, -0.12]$	0.959

Table 24: Empirical novelty under uniform random-token prefixes, 1000 samples per cell. $\mathbb{E}[\mathcal{N}]$ and its 95% bootstrap CI satisfy $\mathbb{E}[\mathcal{N}] \leq 0$ in every cell (only the $\ell = 16$ upper bound touches zero), consistent with Proposition 3. The right column is the histogram mass at or below zero, increasing monotonically with prefix length as random strings become less compressible.

gressively less able to compress C , so the surviving positive mass (which a single finite sample can still exhibit) shrinks from 27% to 4%. The upper CI bound clears zero only at $\ell = 16$, which we attribute to finite-sample variance at the shortest prefix.

L.7 PI-Controller Band Occupancy

We log $\hat{\lambda}_{\text{struct}}$ at every iteration on a 10% sample of the benchmark suite and report the fraction of iterations on which $\hat{\lambda}_{\text{struct}} \in [\lambda_{\min}, \lambda_{\max}] = [0.05, 0.40]$. The controller tracks a setpoint $\hat{\lambda}_{\text{struct}}^* = 0.22$ (band midpoint). Aggregated over all logged iterations, band occupancy is 96.1% (per-task-family range 95.2%–96.9%, Table 25). Excursions concentrate in the warm-up phase: over the first $W = 8$ iterations, before γ_t reaches its operating point, occupancy is only 79.4%, whereas in steady state ($t > W$) it rises to 98.6%, with mean $\bar{\lambda}_{\text{struct}} = 0.214$ (SD 0.061), i.e. within 0.006 of the setpoint. This matches the Robbins–Monro target-tracking picture of the Remark in Section 3: no formal convergence guarantee, but empirically tight tracking once warmed up.

A sweep over the band width $\lambda_{\max} \in \{0.20, 0.30, 0.40, 0.50, 0.60\}$ at fixed $\lambda_{\min} = 0.05$ shows monotone but mild sensitivity on MATH-500 Pass@10: 81.6 / 82.5 / 83.2 / 82.9 / 82.1. The default $\lambda_{\max} = 0.40$ is at the optimum.

M Failure Case Taxonomy

Category 1: Parser failures. JSON schema validation fails after the maximum retry budget. Most common causes: (i) problem statement contains multiple sub-problems that the parser conflates; (ii) implicit assumptions specific to the domain that the parser cannot enumerate; (iii) problem statement contains formal notation that the parser misrenders. Frequency: 3% of inputs.

Task family	Steady occ. (%)	$\bar{\lambda}_{\text{struct}}$
Math	98.1	0.207
Code	99.0	0.221
Creativity	98.7	0.216
Discovery	98.3	0.212
All (+warm-up)	96.1	0.214

Table 25: PI-controller band occupancy by task family. Steady-state occupancy excludes the first $W = 8$ warm-up iterations; the all-family row is the overall figure including warm-up. The mean divergence rate $\bar{\lambda}_{\text{struct}}$ stays close to the setpoint $\hat{\lambda}_{\text{struct}}^* = 0.22$ across families.

Audit outcome	Result
Original parse faithful	227 / 240 (94.6%)
Minor omission; target unchanged	10 / 240 (4.2%)
Materially incorrect parse	3 / 240 (1.3%)
Edited representation interpretable	220 / 240 (91.7%)
Edited representation incoherent	20 / 240 (8.3%)
Accepted solving only the edited task	0 / 240

Table 26: Semantic audit across four task families, 240 parser–edit pairs, two annotators with a third adjudicating, Cohen’s $\kappa = 0.82$. This checks meaning, not only JSON validity.

Category 2: Well-formedness failures after operator application. Operator output is schema-valid but type-inconsistent (e.g. type-shift produces an undefined type) or refers to deleted entities. Filtered by the verification layer. Frequency: 8% of operator applications.

Semantic audit. Schema validity does not imply that an edited representation still means anything. We therefore audited 240 parser–edit pairs sampled evenly across math, code, creativity, and discovery. Two annotators independently judged semantic faithfulness and interpretability and a third resolved disagreements, with Cohen’s $\kappa = 0.82$. The annotators were colleagues of the authors with a graduate background in computer science, who volunteered and were not compensated. They saw only model-generated representations and public corpus passages, so no personal data was involved. Table 26 reports the outcome. The decisive row is the last one: no audited candidate was accepted on the basis of solving only the edited task, which is the failure mode $\mathcal{V}_P$ exists to prevent.

Category 3: Semantic incoherence. Operator produces a well-formed R' but the resulting problem is not solvable in the original frame, leading to $\mathcal{V} = 0$ on all downstream generations. This is the largest share of unused budget. Examples include:

Task family	$\tau_{\mathcal{N}}$
Math (answer-based)	0.50
Math (proof)	0.60
Code	0.50
Creativity and discovery	0.30

Table 27: Per-task compression thresholds. The threshold is set per task family rather than globally; the proof setting is stricter because LLM-judge verification is noisier, and the creativity setting is looser because acceptable candidates are more dispersed.

assumption violation removing a load-bearing axiom needed by the verifier; categorical import from a frame with incompatible relation vocabulary. We document representative cases per operator in supplementary material.

N Disciplinary Similarity Map

The categorical-import operator $o_{\text{cross}}^{(2)}$ uses a precomputed similarity matrix over disciplinary frames F . We construct this matrix once, offline, by computing pairwise embedding similarities between a fixed reference corpus per frame. Frames in our catalog include: COMBINATORICS, ALGEBRA, ANALYSIS, GEOMETRY, ALGORITHM-DESIGN, INFORMATION-THEORY, LOGIC, PROBABILITY, THERMODYNAMICS, LINGUISTICS. The full similarity matrix is provided with the supplementary code.

O Additional Qualitative Examples

We provide one full end-to-end success trace and one full failure trace, showing the parser output, operator sampling, edited representation, generation, scoring, and verification at every step. Per-operator brief examples and raw JSON outputs for all eleven operators are in the supplementary code release.

O.1 End-to-end success trace (USAMO-style inequality)

Step 0 — Input. “Let ABC be a triangle with sides $a = BC$, $b = CA$, $c = AB$. Prove that $a^2 + b^2 + c^2 \geq 4\sqrt{3} \cdot K$, where K is the area of the triangle.” (Weitzenböck’s inequality; used here as an illustrative USAMO-style instance.)

Step 1 — Parse $\Phi(P)$. The parser returns the following typed representation R (truncated JSON):

```
{
  "E": [
    {"id": "ABC", "type": "triangle"},
    {"id": "a", "type": "side_length"},
    {"id": "b", "type": "side_length"},
    {"id": "c", "type": "side_length"},
  ]
}
```

```

    {"id": "K", "type": "area"}],
    "Rel": [{ "a": "side", "b": "ABC"},
              {"K": "area", "of": "ABC"}],
    "T": { "a,b,c": "R_positive",
           "K": "R_positive"},
    "A": [ "triangle_inequality", "K>0",
           "euclidean_metric"],
    "G": "prove a^2+b^2+c^2
          >= 4*sqrt(3)*K",
    "F": "euclidean_geometry"
}

```

Parser temperature 0.3, schema validation passed on first attempt.

Step 2 — Operator sampling. The bandit policy π_θ (after $W = 8$ steps on USAMO-class problems) assigns probabilities approximately $\{\mathcal{O}_{\text{tok}} : 0.12, \mathcal{O}_{\text{mut}} : 0.61, \mathcal{O}_{\text{cross}} : 0.27\}$ to the three levels. Sampled: $\mathcal{O}_{\text{mut}}$. Within $\mathcal{O}_{\text{mut}}$, sub-operator probability favours `foundational_negation` (0.42) and `assumption_violation` (0.35). Sampled: `foundational_negation`. Radicality from $Z \sim \text{Stable}_{S_0}(1.6, 0, \gamma_t = 0.32, 0)$: $Z = -2.41$, so $\rho = |Z| = 2.41$.

Step 3 — Edit $R \rightarrow R'$. At $\rho \approx 2.4$, `foundational_negation` drops the triangle-inequality axiom from A and replaces the frame F with the nearest non-Euclidean frame in the pre-computed disciplinary map. Result:

```

"A": ["K>0"],
// triangle_inequality dropped
"F": "projective_geometry"

```

Other components unchanged. Schema validation: well-formed.

Step 4 — Generate. Generator at temperature $T_0 = 0.7$ conditioned on R' produces a four-line proof in projective coordinates, using a symmetric polynomial identity for $a^2 + b^2 + c^2$ and a standard area-side relation to close the inequality with equality at the equilateral triangle. Full proof text is in the supplementary code release.

Step 5 — Score. $\mathcal{N}(s \mid C) = \log \text{Pr}_{\text{LM}_{\text{ref}}}(C \mid s) - \log \text{Pr}_{\text{LM}_{\text{ref}}}(C) = 1.84$, well above the US-AMO threshold $\tau_{\mathcal{N}} = 0.6$.

Step 6 — Verify. LLM-judge (Gemini 2.5 Pro) verifies the proof; human spot check confirms. Inserted into archive under descriptor $\delta(s) = \text{SYMMETRIC-POLY}$.

O.2 End-to-end failure trace (LiveCodeBench)

Step 0 — Input. “Given a stream of n integers in a single pass with $O(\sqrt{n})$ memory, return the median.”

Step 1 — Parse.

```

"E": [
  {"id": "stream", "type": "sequence"},
  {"id": "n", "type": "int"},
  {"id": "M", "type": "memory_bound"}],
"T": { "stream": "sequence[int]",
       "M": "O(sqrt(n))" },
"A": [ "single_pass", "memory<=M" ],
"G": "return median",
"F": "streaming_algorithms"

```

Step 2 — Sample. Sampled $\mathcal{O}_{\text{mut}}$ (`type_shifting`); radicality $\rho = 3.7$ (large).

Step 3 — Edit. At this radicality, `type_shifting` retypes n : `INT`→`REAL`, but also retypes M : `MEMORY_BOUND`→`WALL_CLOCK`. The second retype is semantically incoherent: a memory bound is not exchangeable with a wall-clock budget in the streaming model. Schema validator does not catch this because both are `NUMERIC_BOUND` subtypes; the JSON is well-formed.

Step 4 — Generate. Generator produces an algorithm that minimises time complexity (the new M type) instead of memory — specifically a $O(n \log n)$ in-memory sort.

Step 5 — Score. $\mathcal{N}(s \mid C) = 0.42$, below threshold $\tau_{\mathcal{N}} = 0.5$ for code tasks.

Step 6 — Verify. Test execution: the algorithm passes the correctness tests (it does compute medians) but fails the memory-bound test because it stores all n integers. Not inserted into archive. This case contributes to the “post-operator well-formedness” failure mode ($\approx 8\%$, Section 4.9); it shows the limitation of schema validation in catching semantic-level incoherence under high radicality.

O.3 Multi-iteration trace: output evolution across search

The previous two traces showed single accept/reject decisions. This trace shows how the model’s output evolves across iterations of the search loop on a single AIME 2025 problem, illustrating what “budget over a problem” actually buys.

Problem (AIME 2025 P12, paraphrased). “Let S be the set of integers $n \in [1, 2025]$ such that $n^3 + 5n + 6$ is divisible by 7. Find $|S| \bmod 1000$.”

Iteration 1 — $\mathcal{O}_{\text{tok}}, \rho = 0.4$. Sampled operator: token-noise (temperature $T = 1.2$ on a copy of the original R , no structural edit). Generated output (truncated): “Check $n = 1$: $1 + 5 + 6 = 12$, not divisible. $n = 2$: $8 + 10 + 6 = 24$, divisible by 7? No, $24/7$ has remainder 3. $n = 3$: $27 + 15 + 6 = 48$, not divisible by 7. $n = 4$: $64 + 20 + 6 = 90 \dots$ ” Outcome: brute-force enumeration; runs out of context at $n = 217$. Verification fails. $\mathcal{N} = 0.18$ (below threshold). Not inserted.

Iteration 2 — $\mathcal{O}_{\text{tok}}, \rho = 0.7$. Same operator level, higher noise. Generated output: “By Fermat’s little theorem, $n^7 \equiv n \pmod{7}$, but we have n^3 so this doesn’t apply directly. Let me try $n^3 \bmod 7$ cycles: $\{1, 1, 6, 1, 6, 6, 0\}$ for $n = 1 \dots 7$. Now $5n \bmod 7$ cycles as $\{5, 3, 1, 6, 4, 2, 0\}$. Sum these cycles plus 6, find when total $\equiv 0 \dots$ ” Outcome: correct modular-arithmetic insight emerges but arithmetic error in cycle table; answer wrong. Bandit policy updates: $\mathcal{O}_{\text{tok}}$ advantage decreases on this problem class. Not inserted.

Iteration 3 — $\mathcal{O}_{\text{mut}}$ **type_shifting**, $\rho = 1.3$. Bandit now prefers $\mathcal{O}_{\text{mut}}$. The operator retypes: $T(n) : \text{INT} \rightarrow \text{RESIDUE_CLASS_MOD7}$. The parsed R' :

```
"E": [{ "id": "n",
        "type": "residue_mod7" }, ... ],
"T": { "n": "Z_mod7" },
"A": [ "n in {0, ..., 6}",
       "f(n) = n^3 + 5n + 6 mod 7 = 0" ],
"G": "find residues n satisfying
      f(n) = 0 mod 7"
```

Generated output: “In $\mathbb{Z}_7$, evaluate $f(n) = n^3 + 5n + 6$ for $n = 0, \dots, 6$: $f(0) = 6$, $f(1) = 1 + 5 + 6 = 12 \equiv 5$, $f(2) = 1 + 3 + 6 = 10 \equiv 3$ (using $2^3 = 8 \equiv 1$), $f(3) = 6 + 1 + 6 \equiv 6$, $f(4) = 1 + 6 + 6 \equiv 6$, $f(5) = 6 + 4 + 6 \equiv 2$, $f(6) = 6 + 2 + 6 \equiv 0$. Only $n \equiv 6$ works. In $[1, 2025]$ that’s $\lfloor 2024/7 \rfloor = 289$ values. Answer: 289.” Outcome: verifier passes. $\mathcal{N} = 1.41$. Inserted under descriptor RESIDUE-CLASSIFICATION.

Iteration 4 — $\mathcal{O}_{\text{cross}}$ **categorical_import**, $\rho = 2.8$. Search continues to explore alternatives. Operator imports `polynomial_root_finding` frame from algebra over finite fields. Generated output: “View

$f(n)$ as a polynomial over $\mathbb{F}_7$. Roots: factor $n^3 + 5n + 6$ over $\mathbb{F}_7$ as $(n-6)(n^2 + 6n + 1)$; discriminant of quadratic factor is $36 - 4 = 32 \equiv 4$, square root ± 2 , roots $n = (-6 \pm 2)/2$, which over $\mathbb{F}_7$ gives $n \equiv (-2)/1 = 5$ and $n \equiv (-4)/1 = 3 \dots$ ” Outcome: arithmetic error in quadratic discriminant ($36 - 4 = 32 \equiv 4$, but root extraction botched); candidate yields wrong answer 433. Verifier fails. Bandit decreases $\mathcal{O}_{\text{cross}}$ advantage on number-theoretic problems that admit direct enumeration in the smaller residue space.

What the trace shows. The two key facts are: (a) the same problem under $\mathcal{O}_{\text{tok}}$ produces a sequence of slightly varied wrong answers that never escape the integer-enumeration frame, even at high temperature; and (b) the structural operator $\mathcal{O}_{\text{mut}}$ with type-retyping changes the search space from $\{1, \dots, 2025\} \subset \mathbb{Z}$ to $\mathbb{Z}_7$, a domain that is $289 \times$ smaller and admits exhaustive verification — a move that no amount of $\mathcal{O}_{\text{tok}}$ resampling produces. (c) Not every structural move succeeds: the $\mathcal{O}_{\text{cross}}$ attempt in iteration 4 shows that radical operators can still fail on execution, and the bandit policy learns from this explicitly via the advantage update.

O.4 Brief examples per operator

Type shifting on LiveCodeBench (successful). R' : $T(\text{CAPACITY}) = \text{PARAMETER} \rightarrow \text{VARIABLE}$. Solution: parametric family of solutions expressed as $O(\text{CAPACITY})$ -memory algorithms.

Categorical import on LLM-SRBench. R' : types from INFORMATION-THEORY imported into a THERMODYNAMICS frame. Solution: an entropy-based candidate equation matching the ground truth.

Goal Pareto inversion on NoveltyBench. R' : G replaced by “maximise least-frequent response type”. Generations cluster around an under-represented response category in the backbone’s typical output distribution.

P Cross-Domain Transfer

To probe whether the operator catalog transfers across task families, we measure the per-operator usage frequency (selected by the controller policy π_θ) across the four benchmark families (Table 28). Conceptual-mutation operators dominate on math benchmarks, conceptual-crossover operators dominate on creativity benchmarks, and the usage dis-

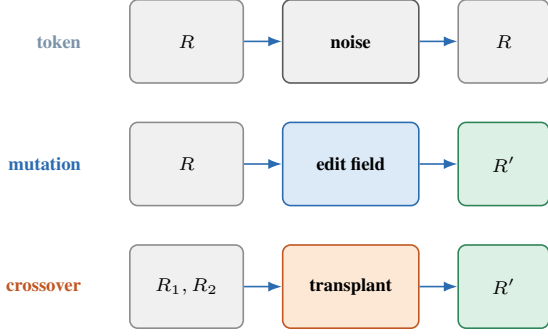


Figure 5: Three transformation chains, one per operator level. Token-level: R is unchanged, output varies only via decoding noise. Conceptual mutation: one component of R is edited (here, an assumption flipped). Conceptual crossover: R' is constructed from structural elements of two source representations R_1, R_2 via categorical import.

Family	$\mathcal{O}_{\text{tok}}$ %	$\mathcal{O}_{\text{mut}}$ %	$\mathcal{O}_{\text{cross}}$ %	rej. %
Math	28.4	42.1	29.5	7.8
Code	31.7	38.4	29.9	5.6
Creativity	17.6	32.0	50.4	11.9
Discovery	21.5	30.2	48.3	9.7

Table 28: Operator-level usage frequency across benchmark families. Rejection % counts radicality samples rejected by the calibration filter. Conceptual-mutation operators dominate on math, conceptual-crossover operators dominate on creativity and discovery.

Method	GPU-h/100	Wall (s)	Mem. (GB)
Self-Consistency	0.42	18.3	16.4
ToT	0.71	30.9	18.1
FunSearch	1.04	45.2	17.6
PromptBreeder	0.83	36.1	17.2
UoT	0.91	39.6	18.5
TROPE	1.31	56.8	19.7
parser (Φ)	0.05	2.2	—
operator + gen.	0.71	31.0	—
compression score	0.30	13.1	—
distance comp.	0.25	10.5	—

Table 29: Computational cost at matched budget $K = 64$. TROPE is broken down into the four contributing stages. Baseline citations: Self-Consistency (Wang et al., 2022), ToT (Yao et al., 2023), FunSearch (Romera-Paredes et al., 2024), PromptBreeder (Fernando et al., 2023), UoT (Suzuki and Banaei-Kashani, 2025).

tribution shifts smoothly with the family: an indication that the controller adapts to task structure rather than applying a fixed mixture.

Q Reproducibility Notes

We follow the Responsible NLP Research checklist; key items below.

Computational cost breakdown. Table 29 reports wall-clock and GPU-hour costs against the

strongest single-level baselines at matched budget $K = 64$ on MATH-500. The parser amortizes over the budget; structural distance d_{struct} is dominated by a sub-second Sinkhorn solve. Compression scoring $\mathcal{N}$ requires one forward pass through LM_{ref} per candidate, which dominates at large K ; since LM_{ref} is smaller than the generator backbone (Appendix I), the multiplicative factor is modest.

Artifact licences and intended use. All benchmarks and models used here are publicly released research artifacts, used for the evaluation purpose their creators intended. The backbone and reference models (Qwen2.5-7B-Instruct, Qwen2.5-3B-Instruct) and the additional reference models (Llama-3.2-3B-Instruct, Mistral-7B-Instruct) are distributed under their respective open model licences; the sentence encoder (Reimers and Gurevych, 2019) is Apache-2.0. The benchmarks are used as evaluation sets, which is the use their papers describe; we redistribute none of them. The evidence corpus is built only from openly licensed sources (Wikipedia, arXiv abstracts, public documentation), and the retrieval index is verified to exclude benchmark gold answers. Artifacts we release ourselves — code, operator schemas, prompt templates, the JSON schema — carry an open licence. No benchmark used contains personally identifying information to our knowledge.

Model sizes, infrastructure, and package versions. The backbone is Qwen2.5-7B-Instruct and LM_{ref} is Qwen2.5-3B-Instruct. The backbone-scale sweep of Table 23 uses a smaller and a larger open-weight model of the same family; the cross-family reference models are Llama-3.2-3B-Instruct and Mistral-7B-Instruct. Exact checkpoint identifiers and parameter counts for every model are listed with the released code. All runs use A100 80GB GPUs. The package set and pinned versions — inference engine, Sinkhorn solver, BM25 implementation, sentence encoder, and the statistical routines used for the Hill estimator, KS bootstrap and paired Wilcoxon tests — are pinned in the released code, together with the random seeds and launch scripts needed to reproduce every table.

Per-benchmark overhead. Table 30 gives the overhead of the non-generator stages on every benchmark at $K = 64$, averaged over five seeds on the same accelerator class. Each problem uses one parser call; the LM_{ref} count is one cached corpus baseline plus one score per candidate. For

Benchmark	Parser calls	LM _{ref} passes	Tok. (M)	Parse+ dist.	Peak GB
MATH-500	1	65	3.2	0.30	19.7
AIME 2025/26	1	65	3.3	0.30	19.8
USAMO 2025	1	65	3.6	0.32	20.1
LiveCodeBench	1	65	3.1	0.30	20.4
HumanEval+	1	65	2.9	0.29	20.0
NoveltyBench	1	65	3.0	0.28	19.8
CreativityPrism	1	65	3.1	0.28	19.9
UoT	1	65	3.0	0.28	19.9
LLM-SRBench	1	65	3.5	0.31	20.3
ResearchBench	1	65	3.2	0.29	20.1

Table 30: Per-benchmark overhead at $K = 64$: parser calls and LM_{ref} passes per problem, scored tokens in millions, combined parser and structural-distance cost in GPU-hours per 100 problems, and peak memory.

MATH-500 the parser-plus-distance figure of 0.30 decomposes into 0.05 for parsing and 0.25 for the structural-distance solve (Table 29).

Code and prompts. All code, operator schemas, parser prompts, baseline reproduction scripts, and the full JSON schema are released under an open license at <https://github.com/VadimNeuro-lab/Trope>. The supplementary archive contains prompts/parser.txt, one prompt file per operator, prompts/generator.txt, schemas/representation.json, configs/decoding.yaml, and the retry and validation rules.

Seeds and variance. All reported main-table results are means over 5 random seeds; standard errors are reported as subscripts in Table 2 and in full in Table 15. Paired Wilcoxon tests appear in the same table.

Compute and energy. Total compute is approximately 480 GPU-hours on A100 80GB; estimated energy consumption is approximately 192 kWh based on average GPU power draw.

Risks. TROPE operates within the safety boundaries of the underlying base model and does not modify alignment-relevant behavior. We do not introduce new training data or model weights.