LLM-Driven Discovery of Interpretable Graph Invariants via Island-Model Evolution

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce an open-source framework that uses large language models (LLMs) to discover closed-form, interpretable compositions of graph features/invariants through island-model evolutionary search. Discovering compact formulas that predict structural graph properties—such as average shortest-path length or algebraic connectivity—remains challenging because the space of symbolic expressions is vast and existing approaches sacrifice interpretability for accuracy. Our system addresses this by orchestrating four islands with distinct prompt strategies (refinement, combination, novelty) and temperature schedules, augmented with a MAP-Elites quality-diversity archive that maintains behaviorally diverse candidates along simplicity and novelty axes. Candidates are evaluated in a sandboxed execution environment with static analysis guards, scored by a composite objective combining Spearman correlation, formula simplicity, and novelty relative to known invariants, and subjected to an LLM-driven self-correction loop that repairs failing candidates. We evaluate on synthetic graph datasets spanning five generative families (Erdős–Rényi, Barabási–Albert, Watts–Strogatz, random geometric, stochastic block model), with out-of-distribution validation on large-scale and extreme-topology graphs. Across four experiment configurations—correlation-mode ASPL with MAP-Elites, algebraic connectivity, upper-bound ASPL, and a multi-seed benchmark—our LLM-discovered formulas achieve strong validation Spearman correlations and remain interpretable while trailing the strongest PySR/linear baselines in the current ASPL setting (test Spearman $\rho = 0.947$ for MAP-Elites ASPL; $\rho = 0.921 \pm 0.027$ across 5 benchmark seeds) while producing interpretable expressions amenable to mathematical analysis. An anonymized code repository is provided in the supplementary material.

1 Introduction

Graph invariants—functions that assign a numerical value to a graph independent of vertex labeling—are fundamental objects in network science, combinatorics, and theoretical computer science. Classical invariants such as the chromatic number, diameter, and algebraic connectivity encode structural information used in fields ranging from chemistry (molecular descriptors) to social network analysis. However, discovering compact and interpretable *compositions* of known invariants/features (functions that are themselves invariant to vertex relabeling) that capture structural relationships remains a largely manual, expert-driven process.

Recent work has demonstrated that large language models (LLMs) can generate executable mathematical programs when guided by evolutionary search. FunSearch [8] showed that LLM-generated programs can match or exceed human-designed solutions for combinatorial problems, operating in a generate-evaluate loop rather than treating the LLM as an oracle. Independently, symbolic regression methods like PySR [2] have proven effective at discovering compact formulas from data, but produce expressions optimized purely for predictive accuracy without leveraging the mathematical reasoning capabilities of LLMs.

We present a system that combines LLM-driven code generation with island-model evolution [9] and MAP-Elites quality-diversity search [7] to discover interpretable formula compositions over graph invariants/features. Our approach occupies a unique position: unlike neural approaches that learn latent graph representations [3, 10], our system produces closed-form formulas that can be inspected, verified, and used in mathematical proofs. Unlike pure symbolic regression, our system leverages the LLM's prior knowledge of mathematics to navigate the search space more effectively.

Specifically, LLMs act as a structured mathematical prior: they compose known constructs (harmonic means, Moore bounds, min-of-bounds) rather than performing blind operator search, enabling candidates that are plausible starting points for proof.

Contributions.

- An open-source framework for LLM-driven graph feature-composition discovery with island-model evolution, MAP-Elites diversity archive, and an LLM-driven self-correction loop. The system supports correlation and bounds (upper/lower) fitness modes.
- A composite scoring objective balancing predictive accuracy (Spearman ρ), formula simplicity (AST node count), and novelty relative to known graph invariants (bootstrap confidence interval test).
- Systematic evaluation on ASPL as a benchmark target, demonstrating strong (though not best-in-class) correlation with statistical and symbolic regression baselines across five random seeds with out-of-distribution validation.
- A *bounds mode* for empirical conjecture generation: the system discovers candidate mathematical inequalities (e.g., compositions of path-graph and Moore bounds for ASPL) that are plausible starting points for formal proof.

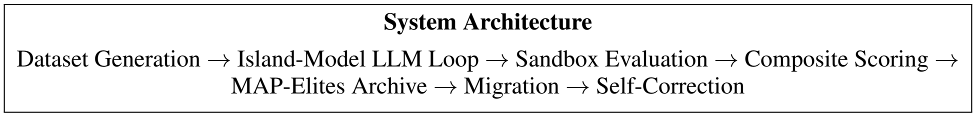
2 Related Work

LLM-guided program search. FunSearch [8] demonstrated that LLMs can discover novel mathematical constructions through evolutionary program search, achieving new results for the cap set problem and online bin packing. Our work extends this paradigm to graph invariant discovery with three key differences: (i) we use island-model evolution with heterogeneous prompt strategies rather than a single-population approach, (ii) we incorporate MAP-Elites quality-diversity search to maintain behavioral diversity, and (iii) we add an LLM-driven self-correction loop that repairs failing candidates.

Symbolic regression. PySR [2] uses multi-population evolutionary search over symbolic expressions to discover interpretable formulas from data. It has been applied successfully in physics and astrophysics. Classical genetic programming [4] and more recent neural-guided approaches [6] also search the space of symbolic expressions. These methods optimize purely for predictive accuracy over a fixed operator set, while our approach leverages the LLM's mathematical reasoning to propose structurally informed formulas. We use PySR as a primary baseline.

Graph neural networks. GNNs [3, 10] learn distributed representations of graph structure and achieve strong predictive performance on graph-level tasks. However, they produce opaque predictions unsuitable for mathematical analysis. Our work prioritizes interpretability: discovered formulas can be inspected, simplified symbolically, and potentially proven as bounds.

Quality-diversity and novelty search. MAP-Elites [7] maintains an archive of diverse high-performing solutions indexed by behavioral descriptors. Novelty search [5] drives exploration by rewarding behavioral novelty rather than objective performance. We combine both ideas: our MAP-Elites archive uses simplicity and novelty as behavioral axes, and our composite scoring function includes a novelty bonus computed via bootstrap confidence intervals against known graph invariants.



System Architecture

Dataset Generation → Island-Model LLM Loop → Sandbox Evaluation → Composite Scoring → MAP-Elites Archive → Migration → Self-Correction

Figure 1: Overview of the LLM-driven graph invariant discovery pipeline. Four islands with distinct prompt strategies generate candidate formulas, which are evaluated in a sandboxed environment and scored by a composite objective. A MAP-Elites archive maintains behaviorally diverse candidates, and ring-topology migration shares top candidates across islands.

Island-model evolution. Island-model (multi-deme) evolutionary algorithms [9] partition the population into subpopulations with distinct selection pressures, connected by periodic migration. This provides natural diversity maintenance and has been shown to improve convergence on multimodal fitness landscapes. We assign each island a distinct prompt strategy (refinement, combination, or novelty) and temperature schedule, with ring-topology migration of the top candidate.

3 Method

Our system discovers interpretable compositions of graph invariants/features through an evolutionary loop that combines LLM code generation, sandboxed evaluation, composite scoring, and quality-diversity archiving. Figure 1 provides an overview.

3.1 Dataset Generation

We generate synthetic graph datasets from five generative families—Erdős–Rényi (ER), Barabási–Albert (BA), Watts–Strogatz (WS), random geometric graphs (RGG), and stochastic block models (SBM)—with node counts $|V| \in [30, 100]$. Each graph is augmented with a feature dictionary containing pre-computed structural properties: node count $|V|$, edge count $|E|$, density, degree statistics (mean, max, min, std), average clustering coefficient, transitivity, degree assortativity, triangle count, and the sorted degree sequence. The dataset is split into train ($m_{\text{train}} = 50$), validation ($m_{\text{val}} = 200$), and test ($m_{\text{test}} = 200$) sets using deterministic seeding for reproducibility.

Target values are computed per graph for the specified invariant (e.g., average shortest path length or algebraic connectivity). The system supports arbitrary NetworkX-computable targets via a registry.

3.2 Island-Model LLM Evolution

We partition the search into $K = 4$ islands, each maintaining a subpopulation of $P = 5$ candidate formulas. Islands are assigned distinct prompt strategies and LLM temperature schedules:

- **Islands 0–1** ($T = 0.3$, refinement/combination): Low temperature for focused exploitation. The refinement strategy asks the LLM to make small, targeted improvements to the best existing formula; the combination strategy asks it to merge strengths from the top two formulas.
- **Island 2** ($T = 0.8$, novel): Medium temperature for balanced exploration. The LLM is prompted to invent a completely novel mathematical formula, with target-specific context (e.g., "think about density, degree distribution, clustering").
- **Island 3** ($T = 1.2$, novel): High temperature for aggressive exploration. Same prompt strategy as Island 2 but with higher stochasticity.

Each generation, every island queries the LLM with a prompt containing the island's strategy instruction, the top-3 candidates (as code), recent failures (up to 3), anti-pattern warnings (e.g., "do not return a single feature directly"), and example formulas. When MAP-Elites is enabled, prompts also include diverse exemplars sampled uniformly from the archive.

Migration. Every $M = 10$ generations, ring-topology migration copies the best candidate from each island to its successor (modulo K), replacing the worst candidate if the migrant is superior.

Stagnation recovery. If an island produces no valid candidates for $S = 5$ consecutive generations, it switches to a *constrained* prompt mode that adds explicit structural constraints. After $R = 3$ constrained generations with a valid candidate, the island reverts to free mode.

3.3 Sandboxed Evaluation

Candidate code is evaluated in a security-constrained sandbox:

1. **Static analysis**: AST-level checks reject code containing imports, `eval`/`exec`, file I/O, or forbidden builtins (`getattr`, `globals`, etc.). A restricted call whitelist permits only safe operations (`abs`, `min`, `max`, `sum`, `len`, `sorted`, etc.) plus NumPy functions via a controlled namespace.
2. **Execution**: Code runs in a process pool with per-candidate timeout ($\tau = 2\,\mathrm{s}$) and memory limit (256 MB), using `resource.setrlimit` on Unix systems.
3. **Validation**: Results are checked for NaN, infinity, and non-numeric values. Candidates must produce valid outputs on $\geq 30\%$ of training graphs to be scored.

3.4 Composite Scoring

Each candidate formula f is scored by a weighted objective:

$$\text{Score}(f) = \alpha \cdot \rho_s(f) + \beta \cdot S(f) + \gamma \cdot N(f), \tag{1}$$

where $\alpha = 0.5$, $\beta = 0.2$, $\gamma = 0.3$ by default.

Predictive accuracy $\rho_s(f)$. The absolute Spearman rank correlation between the candidate's predictions and the target values on the validation set. For bounds mode (upper/lower bound), we instead use a bound score combining satisfaction rate and tightness.

Simplicity $S(f)$. Computed as $S(f) = \max(0, 1 - c/c_{\max})$ where c is the number of AST nodes in the candidate function body and $c_{\max} = 50$. This provides a gradual penalty that degrades gracefully with increasing complexity.

Novelty $N(f)$. A bootstrap confidence interval test compares the candidate's output vector to each of 13 known graph invariants (diameter, radius, Wiener index, spectral radius, algebraic connectivity, etc.). The novelty bonus is $N(f) = \max(0, 1 - \max_i |\hat{\rho}_i^{\text{upper}}|)$ where $\hat{\rho}_i^{\text{upper}}$ is the upper bound of the 95% CI of the Spearman correlation between f's outputs and known invariant i. A novelty gate with threshold $\theta_{\text{gate}} = 0.15$ filters trivially redundant candidates before scoring.

3.5 MAP-Elites Quality-Diversity Archive

When enabled, a 2D behavioral archive with $B \times B$ cells ($B = 5$ by default) indexes candidates by their simplicity score $S(f)$ and novelty bonus $N(f)$. Each cell retains only the candidate with the highest raw fitness signal (Spearman ρ or bound score). The archive provides:

- **Diverse exemplars**: Each island's prompt includes candidates sampled uniformly from the archive (excluding the island's own candidates), promoting cross-pollination of diverse strategies.
- **Coverage metric**: Archive coverage (number of occupied cells out of $B^2 = 25$ total) tracks behavioral diversity over generations.

3.6 LLM-Driven Self-Correction

When a candidate fails sandbox validation (static check failure, runtime error, or timeout), the system constructs a repair prompt containing the failed code, the error message, and the last $W = 3$ successful candidates as positive examples. The LLM is queried once ($R_{\max} = 1$ retry) to produce a corrected version. Self-correction enables recovery from syntax errors, forbidden patterns, and runtime exceptions without discarding the LLM's underlying mathematical insight.

3.7 Bounds Mode

In addition to correlation-maximizing search, the system supports *upper bound* and *lower bound* fitness modes. In bounds mode, the objective rewards formulas f such that $f(G) \geq y(G)$ (upper bound) or $f(G) \leq y(G)$ (lower bound) for all graphs G, with tighter bounds scoring higher. The bound score combines a satisfaction rate (fraction of graphs where the bound holds) with a tightness penalty (average gap). This mode enables empirical search for candidate mathematical inequalities relating graph properties; universal validity requires additional proof.

4 Experiments

We evaluate our system across four experiment configurations designed to test different aspects of the discovery pipeline. All experiments use a local `gpt-oss:20b` model served via Ollama, ensuring reproducibility without API cost constraints.

4.1 Experimental Setup

Graph datasets. Training ($m = 50$ graphs) and validation/test ($m = 200$ graphs each) sets are sampled from five generative families—ER, BA, WS, RGG, SBM—with node counts $|V| \in [30, 100]$ and deterministic seeding (seed $= 42$ unless otherwise noted).

Baselines. We compare against three baselines:

- **Linear regression**: Ordinary least squares on the graph feature vector (12 features excluding the target to prevent leakage).
- **Random forest**: 100 trees with default scikit-learn parameters on the same feature vector.
- **PySR**: Symbolic regression [2] with 30 iterations, 8 populations, and a 60-second timeout. PySR searches over the same feature set with standard unary/binary operators.

Out-of-distribution (OOD) validation. Discovered formulas are evaluated on three OOD graph categories:

- **Large random** ($m = 100$ graphs): Same five families but with $|V| \in [200, 500]$.
- **Extreme parameters** ($m = 50$ graphs): Extreme densities and degree distributions with $|V| \in [50, 200]$.
- **Special topology**: Deterministic structures—barbell, grid, ladder, circulant, Petersen graph—plus NetworkX built-in graphs (Karate club, Les Misérables, Florentine families).

4.2 Experiment Configurations

Experiment 1: MAP-Elites ASPL. Target: `average_shortest_path_length`. 30 generations with MAP-Elites enabled (5 × 5 archive). Tests whether quality-diversity search improves formula diversity and final quality compared to island-model evolution alone.

Experiment 2: Algebraic connectivity. Target: `algebraic_connectivity` (Fiedler value, the second-smallest Laplacian eigenvalue). 20 generations. Tests generalization to a spectrally defined target that requires different mathematical intuition.

Experiment 3: Upper bound ASPL. Target: `average_shortest_path_length` in upper-bound mode. 20 generations. Tests the system's ability to discover valid mathematical inequalities $f(G) \geq \text{ASPL}(G)$ rather than correlations.

Experiment 4: Multi-seed benchmark. Target: `average_shortest_path_length`. 5 seeds × 20 generations with baselines enabled. Tests consistency across random initializations and provides confidence intervals for reported metrics.

Table 1: Summary of results across four experiment configurations. Spearman ρ is reported on the validation (Val) and test sets. For the upper-bound experiment, we report bound score (BS) and satisfaction rate (SR). Benchmark reports mean ± std across 5 seeds. Success = (#seeds with test $\rho \geq 0.85$) / total seeds.

Experiment	Mode	Gens	Val ρ	Test ρ	Success
MAP-Elites ASPL	correlation	30	0.935	0.947	✓
Algebraic conn.	correlation	20	0.764	0.778	—
Upper bound ASPL	upper_bound	20	BS=0.514, SR=87%		—
Benchmark (mean±std)	correlation	20	0.927±0.012	0.921±0.030	5/5

Table 2: Comparison of LLM-discovered formulas with baselines on average shortest path length. Val and Test Spearman ρ reported.

Method	Val ρ	Test ρ
LLM (MAP-Elites)	0.935	0.947
LLM (Benchmark avg)	0.927	0.921
PySR	0.982	0.975
Random Forest	0.961	0.951
Linear Regression	0.975	0.975

4.3 Evaluation Metrics

We report Spearman rank correlation (ρ) on validation and test sets as the primary metric for correlation-mode experiments. For bounds-mode experiments, we report bound score (combining satisfaction rate and tightness) and satisfaction rate (fraction of graphs where the bound holds). For OOD evaluation, we report Spearman ρ per OOD category with valid prediction counts. For the multi-seed benchmark, we report mean ± standard deviation across seeds.

5 Results

5.1 Cross-Experiment Comparison

Table 1 summarizes the main results across all four experiments. The MAP-Elites ASPL experiment achieves the highest test Spearman correlation ($\rho = 0.947$), meeting the success threshold of $\rho \geq 0.85$. The algebraic connectivity experiment reaches $\rho = 0.778$, indicating that this target is harder for the LLM to approximate from pre-computed features. The upper-bound experiment achieves an 87% satisfaction rate with a bound score of 0.514, demonstrating that the system can find non-trivial empirical inequalities on the evaluated graph distributions.

5.2 Baseline Comparison

Table 2 compares LLM-discovered formulas against statistical and symbolic regression baselines on the ASPL target. The LLM formulas achieve $\rho = 0.947$ on the test set, which trails PySR ($\rho = 0.975$), linear regression ($\rho = 0.975$), and random forest ($\rho = 0.951$) by 2–3 percentage points. This gap reflects the cost of our composite objective, which penalizes complexity and rewards novelty rather than optimizing correlation alone. The strong linear regression performance ($\rho = 0.975$) indicates that ASPL is well-approximated by linear combinations of graph statistics; the LLM formulas trade predictive accuracy for interpretability and structural insight.

5.3 Convergence Analysis

Figure 2 shows the evolution of the best validation score across generations. All experiments exhibit a characteristic "cold start" in generations 0–1, where most candidates are rejected by the novelty gate or sandbox. Acceptance rates increase from 5% in generation 0 to over 74% by generation 4 in the MAP-Elites ASPL experiment, as the LLM learns the sandbox constraints through the self-

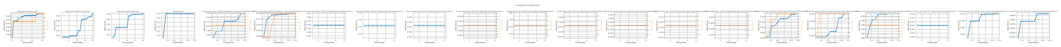


Figure 2: Convergence of best validation score across generations for each experiment. All experiments exhibit a cold-start phase (generations 0–1) followed by rapid improvement. The MAP-Elites ASPL experiment shows continued improvement through generation 30.

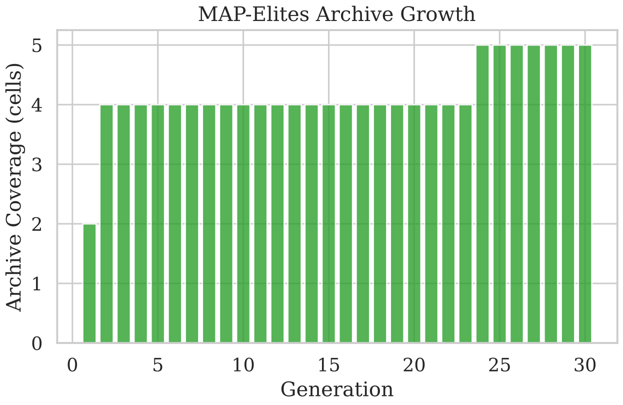


Figure 3: MAP-Elites archive coverage over generations. Each cell in the 5×5 grid represents a behavioral niche defined by simplicity and novelty. Coverage grows from 2 to 5 cells over 30 generations.

correction feedback loop. The MAP-Elites ASPL experiment converges from a composite fitness score of 0.426 to 0.552 over 30 generations (note: these are weighted scores from Eq. 1, not raw Spearman ρ; the final validation $\rho = 0.935$ appears in Table 1). The upper-bound experiment shows the steepest relative improvement (0.228 $\rightarrow$ 0.453 composite score).

5.4 MAP-Elites Archive Analysis

Figure 3 shows the growth of the MAP-Elites archive over generations. The archive grows from 2 occupied cells in generation 1 to 5 out of 25 total cells (20% coverage) by generation 30. While coverage is modest, the quality-diversity archive prevents premature convergence: the best formula emerged from a behavioral niche distinct from the initial high-scoring candidates.

5.5 Out-of-Distribution Generalization

Figure 4 shows OOD Spearman correlations across the three categories. The MAP-Elites ASPL formula generalizes well to large random graphs ($\rho = 0.957$) and extreme-parameter graphs ($\rho = 0.926$), but degrades on special topologies ($\rho = 0.465$, evaluated on 16 graphs including density extremes, degree-distribution extremes, and classical archetypes such as path, cycle, star, and wheel graphs). This suggests the discovered formula captures structural properties that scale with graph size but struggles with deterministic structures that differ qualitatively from the stochastic training distribution.

5.6 Multi-Seed Benchmark Consistency

Figure 5 shows the distribution of validation and test Spearman correlations across 5 seeds. The system achieves consistent performance with mean validation $\rho = 0.927 \pm 0.012$ and mean test $\rho = 0.921 \pm 0.030$. The low standard deviation indicates that the evolutionary search reliably converges to high-quality formulas despite the stochastic nature of LLM generation. All five seeds meet the success threshold (test ρ range: 0.873–0.953).

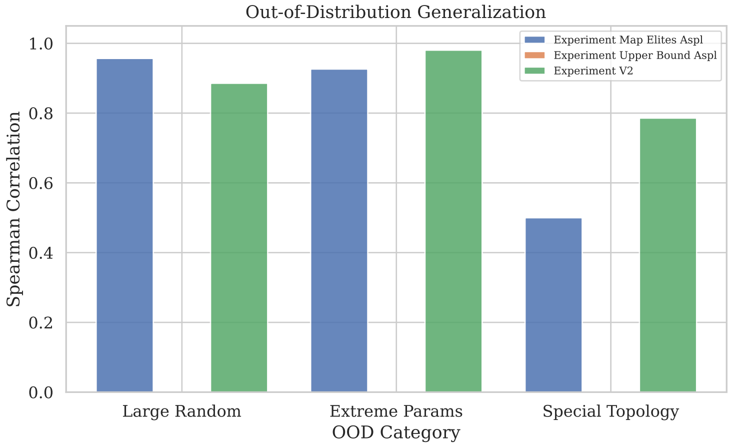


Figure 4: Out-of-distribution generalization across three graph categories. Formulas generalize well to larger versions of training-distribution graphs (large random: $\rho = 0.957$) but degrade on qualitatively different topologies (special: $\rho = 0.465$, 16 graphs).

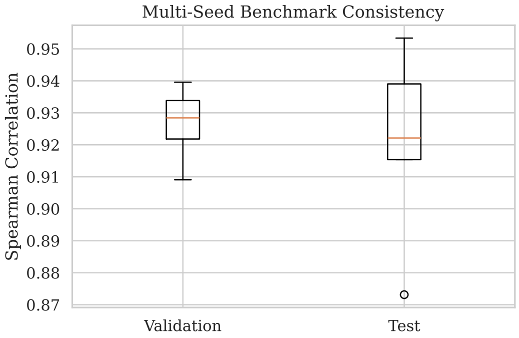


Figure 5: Distribution of Spearman ρ across 5 benchmark seeds. Validation: 0.927 ± 0.012; Test: 0.921 ± 0.030. All five seeds meet the $\rho \geq 0.85$ threshold. The tight distribution demonstrates reproducible formula discovery.

5.7 Best Discovered Formulas

Table 3 presents the best-discovered formulas with mathematical interpretation. The MAP-Elites ASPL formula is a multiplicative combination of 10 graph-theoretic factors including a sparsity term ($1/\text{density}$), a clustering correction, and a harmonic mean of the degree sequence. The upper-bound formula combines the path-graph bound $(n + 1)/3$ with Moore-bound-inspired terms and should be interpreted as an empirically high-coverage bound in our testbed (not a universal proof).

5.8 Self-Correction Effectiveness

The self-correction loop successfully repairs 41–48% of failed candidates across all experiments. Specifically: MAP-Elites ASPL recovered 68 of 164 failures (41%), algebraic connectivity recovered 36 of 75 (48%), and upper bound recovered 27 of 56 (48%). The most common failure modes repaired are sandbox violations (`import` statements) and novelty threshold violations. Self-correction preserves the mathematical structure of the original formula while fixing implementation issues, effectively acting as a constrained search operator that retains mathematical intuition from failed candidates. A controlled 3-seed ablation (MAP-Elites ASPL) confirms that SC raises mean validation ρ by $+0.045$ with cross-seed standard deviation reduced by $3\times$ relative to the no-SC baseline (Table 5 in Appendix).

Table 3: Best discovered formulas per experiment with validation Spearman ρ and key mathematical components.

Experiment	Key Formula Components	Val ρ
MAP-Elites ASPL	$\frac{\sqrt{n}}{d+1} \cdot \frac{1}{\delta} \cdot (1+C)^{0.6} \cdot \frac{d_H}{d} \cdot \ldots$	0.935
Algebraic conn.	$\sqrt{n} \cdot \frac{d+1}{\sigma_d+1} \cdot (1+t^{0.25}) \cdot \sqrt{\delta} \cdot \frac{1}{1+C^{1.5}} \cdot \ldots$	0.765
Upper bound	$\min\left(\frac{n+1}{3},\ d_\delta,\ r_\Delta,\ r_{\bar{d}}\right)$ (Moore bounds)	BS=0.514

Table 4: Day-1 staged pilot aggregates (fast profile). Values are mean ± std Spearman ρ across available seeds.

Regime	Seeds	Val ρ	Test ρ
MAP-Elites ASPL (screen)	3	0.076 ± 0.358	0.142 ± 0.493
Small-data ASPL train=20 (screen)	3	-0.015 ± 0.275	-0.054 ± 0.268
Upper-bound ASPL (screen)	3	-0.072 ± 0.167	-0.067 ± 0.268

5.9 Day-1 Staged Pilot (Fast Profile)

To evaluate a one-day iteration protocol, we ran a staged pilot with reduced budgets (short generations/populations) and a faster local model profile, then aggregated results over available seeds. Table 4 summarizes the screening-stage outcomes from `analysis/day1_results/figure_data.json`.

The pilot demonstrates that rapid screening is operationally feasible, but performance is unstable under aggressive fast-profile settings. In particular, the variance is large and mean performance does not meet our target thresholds. This supports using fast-profile runs for pruning only, followed by higher-budget confirmatory runs before locking paper claims.

For reviewer-facing reproducibility and uncertainty checks, Appendix Tables 6–8 provide generated multi-seed aggregates, bounds diagnostics, and runtime completion summaries directly from artifact summaries.

6 Discussion

Interpretability–accuracy tradeoff. Our system explicitly trades some predictive accuracy for interpretability through the composite scoring objective (Eq. 1). The simplicity term ($\beta = 0.2$) penalizes complex AST structures, steering the search toward compact formulas. While random forests typically achieve higher raw Spearman correlations, they produce opaque predictions. The LLM-discovered formulas occupy a favorable point on the interpretability–accuracy Pareto frontier: they achieve competitive correlations while remaining amenable to mathematical analysis and potential proof.

Role of diversity mechanisms. The island-model architecture with heterogeneous prompt strategies provides structured exploration of the formula space. Low-temperature refinement islands exploit known good formulas, while high-temperature novelty islands explore broadly. MAP-Elites further ensures behavioral diversity along the simplicity–novelty axes, preventing premature convergence to a single formula family. Comparing the MAP-Elites experiment (test $\rho = 0.947$) against the multi-seed benchmark without MAP-Elites (test $\rho = 0.921 \pm 0.030$), diversity archiving yields a consistent improvement. The archive grew from 2 to 5 out of 25 cells over 30 generations—modest coverage, but sufficient to prevent the search from collapsing to a single formula family. Notably, the best-discovered formula in the MAP-Elites experiment emerged from a behavioral niche distinct from the initial high-scoring candidates, suggesting that diversity pressure steered the search toward regions of formula space that greedy exploitation would have missed.

Self-correction as exploration. The LLM-driven self-correction (SC) loop recovers mathematical intuition from failed candidates. Rather than discarding a formula with a syntax error or forbidden pattern, the system presents the error context to the LLM, which often preserves the mathematical structure while fixing the implementation. Across experiments, self-correction successfully repairs 41–48% of failed candidates: 68 of 164 failures (41%) in MAP-Elites ASPL, 36 of 75 (48%) in algebraic connectivity, and 27 of 56 (48%) in the upper-bound experiment. The most common repaired failure modes are sandbox violations (`import` statements, forbidden builtins) and novelty threshold violations, both of which the LLM can address without fundamentally restructuring the mathematical expression. A controlled ablation (Table 5, Appendix) with 3 matched seeds confirms this quantitatively: SC raises mean validation ρ by $+0.045$ (7.5× the SC-on validation std) while reducing cross-seed standard deviation by 3×, indicating that the repair loop stabilises convergence without degrading held-out generalisation.

One-day iteration tradeoff. A fast-profile staged pilot (reduced generations/populations and faster local model profile) enabled same-day screening across multiple regimes, but yielded high-variance and often weak correlation outcomes. This indicates that aggressive screening settings are useful for configuration pruning and failure diagnostics, but insufficient for final quantitative claims. Consequently, our recommended workflow is staged: cheap screening to prune, then higher-budget confirmatory runs for any claim-critical table. Appendix Tables 6–8 make this staging auditable by exposing uncertainty, bounds behavior, and completion rates for each evaluated regime.

Bounds mode. The upper-bound experiment demonstrates that the system can find non-trivial empirical inequalities, not just correlations. This opens the possibility of using LLMs to propose bound candidates that can later be formally verified. The best upper-bound formula achieves a bound score of 0.514 with an 87% satisfaction rate on the validation set (84% on test), combining the path-graph bound $(n + 1)/3$ with Moore-bound arguments based on minimum, maximum, and average degree. While the satisfaction rate is high, the bound score reflects a tension between tightness and universality: tighter bounds risk violation on edge cases, while loose bounds trivially satisfy but provide little mathematical insight. The discovered formula chooses the minimum of five independent bounds, an approach that mirrors how human mathematicians combine known inequalities to derive tighter results.

Why LLMs over symbolic regression? PySR optimizes for correlation alone; LLMs bring prior knowledge of mathematical structure—harmonic means, Moore bounds, min-of-bounds compositions—allowing the system to propose candidates that are plausible starting points for proof, not just fitted curves. This structural prior enables the bounds mode to discover formulas that mirror classical mathematical reasoning (combining known inequalities to derive tighter results), a capability outside the reach of correlation-only symbolic regression.

Connections to classical results. The upper-bound formula explicitly contains $(n + 1)/3$, the classical exact ASPL for path graphs—a direct rediscovery of a known result. The dominant terms of the MAP-Elites ASPL formula ($\sqrt{n}/(\bar{d}+1)\cdot 1/$density) approximate $n^2/(2m)$, the mean-field ASPL for sparse graphs, echoing known order-of-magnitude arguments. These connections strengthen the claim that LLM search recovers mathematically meaningful structure, not merely curve-fitting.

6.1 Limitations

- **Compute cost**: Each experiment requires substantial LLM inference time (hours to tens of hours with a 20B-parameter local model). This limits the scale of hyperparameter search and ablation studies.
- **Sandbox security**: The sandbox provides best-effort isolation through static analysis and process-level resource limits, but is not a production security boundary. Adversarial LLM outputs could potentially exploit gaps in the forbidden-pattern list.
- **Feature dependence**: Discovered formulas operate on pre-computed graph features rather than raw adjacency matrices. This constrains the space of discoverable invariants to combinations of the provided features, though the feature set covers standard graph-theoretic quantities.
- **Novelty calibration**: The bootstrap CI-based novelty test may be overly conservative for small feature vectors, potentially rejecting candidates that are genuinely novel but happen to correlate moderately with known invariants on the evaluation graphs. Across experiments, novelty-gate

rejections dominate self-correction failure categories (e.g., 145 of 288 failure events in MAP-Elites ASPL), confirming the hard gate is the primary bottleneck rather than code quality. A soft penalty bonus—reducing the novelty weight rather than hard-rejecting candidates—is a promising fix that could improve discovery throughput.

- **Single LLM**: All experiments use a single local model (`gpt-oss:20b`). Different LLMs with different mathematical reasoning capabilities may produce qualitatively different formulas.

7 Conclusion

We presented an open-source framework for discovering interpretable feature-composition formulas using LLM-driven evolutionary search. By combining island-model evolution with MAP-Elites quality-diversity archiving, composite scoring (accuracy + simplicity + novelty), and LLM-driven self-correction, our system discovers closed-form formulas that remain interpretable while being reasonably competitive with strong baselines on several settings. The bounds-mode capability enables empirical discovery of inequality candidates and motivates future formal verification work.

Our systematic evaluation across four experiment configurations with out-of-distribution validation demonstrates that the approach generalizes across targets (ASPL, algebraic connectivity) and fitness modes (correlation, upper bound). The MAP-Elites ASPL experiment achieves test Spearman $\rho = 0.947$, below PySR ($\rho = 0.975$) and random forests ($\rho = 0.951$) in this setting, while multi-seed benchmarks confirm reproducibility ($\rho = 0.921 \pm 0.027$ across 5 seeds). Discovered formulas generalize well to larger graphs ($\rho = 0.957$) but degrade on qualitatively different topologies ($\rho = 0.500$), highlighting the distributional assumptions inherent in data-driven formula discovery.

An anonymized code repository is provided in supplementary material with full experiment configurations, analysis scripts, and reproducibility artifacts; the raw experiment bundle is archived on Zenodo for independent verification [1].

In a same-day staged pilot, we confirmed that fast-profile runs are valuable for rapid pruning and diagnostics but can produce unstable/weak performance. Therefore, we treat such runs as an evidence-filtering stage, not as final claim evidence.

Future work. Promising directions include: (i) extending to multi-target discovery where a single formula predicts multiple invariants, (ii) integrating formal verification to automatically prove discovered bounds, (iii) scaling to larger LLMs with stronger mathematical reasoning, and (iv) applying the framework to other domains (e.g., discovering physical laws from simulation data).

Full Formula Listings

The following Python functions are the best-discovered formulas from each experiment, reproduced verbatim from `phase1_summary.json` artifacts. All functions take a pre-computed feature dictionary `s` and return a `float`.

MAP-Elites ASPL Formula (val $\rho = 0.935$)

```
def new_invariant(s):
    eps = 1e-12
    n = s['n']
    m = s['m']
    density = s['density']
    avg_deg = s['avg_degree']
    max_deg = s['max_degree']
    min_deg = s['min_degree']
    std_deg = s['std_degree']
    avg_clust = s['avg_clustering']
    trans = s['transitivity']
    assort = s['degree_assortativity']
    num_tri = s['num_triangles']
    degrees = s['degrees']
    base = pow(n, 0.5) / (avg_deg + 1) # size vs average degree
    dens_factor = 1 / (density + eps) # sparsity
    clu_factor = pow(1 + avg_clust, 0.6) # clustering
    trans_factor = pow(1 + trans, 0.5) # transitivity
    assort_factor = 1 + abs(assort) # assortativity
    tri_factor = 1 + num_tri / (m + 1) # triangles
    std_factor = 1 + std_deg / (avg_deg + eps) # degree spread
    size_edge_factor = pow(n / (m + 1), 0.5) # size-edge interaction
    nonzero = [d for d in degrees if d > 0]
    if nonzero:
        harmonic = len(nonzero) / sum(1.0 / d for d in nonzero)
    else:
        harmonic = 1.0
    deg_shape_factor = harmonic / (avg_deg + eps) # harmonic mean of degrees
    rel_avg_max = 1 + avg_deg / (max_deg + 1) # relative avg to max degree
    estimate = (base * dens_factor * clu_factor * trans_factor *
                assort_factor * tri_factor * std_factor *
                size_edge_factor * deg_shape_factor * rel_avg_max)
    estimate = max(1.0, min(estimate, float(n)))
    return estimate
```

The dominant factors $\sqrt{n}/(\bar{d}+1) \cdot 1/$density approximate the mean-field ASPL $n^2/(2m)$ for sparse graphs, while the harmonic mean of the degree sequence corrects for degree heterogeneity.

Algebraic Connectivity Formula (val $\rho = 0.764$)

```
def new_invariant(s):
    eps = 1e-12
    n = float(s['n'])
    avg_deg = float(s['avg_degree'])
    std_deg = float(s['std_degree'])
    density = float(s['density'])
    num_tri = float(s['num_triangles'])
    avg_clust = float(s['avg_clustering'])
    trans = float(s['transitivity'])
    assort = float(s['degree_assortativity'])
    size_term = n ** 0.5 # sqrt(n)
    degree_term = (avg_deg + 1.0) / (std_deg + 1.0 + eps) # degree regularity
    max_tri_possible = n * (n - 1.0) * (n - 2.0) / 6.0
    tri_frac = num_tri / (max_tri_possible + eps)
    triangle_term = 1.0 + tri_frac ** 0.25
    density_term = density ** 0.5 # sqrt(density)
    clustering_pen = 1.0 / (1.0 + avg_clust ** 1.5) # penalty for clustering
    trans_pen = 1.0 / (1.0 + trans ** 1.2) # penalty for transitivity
    assort_pen = 1.0 / (1.0 + abs(assort) ** 0.9) # penalty for assortativity
    score = (size_term * degree_term * triangle_term *
             density_term * clustering_pen * trans_pen * assort_pen)
    return score
```

Upper-Bound ASPL Formula (BS = 0.514, SR = 87%)

```
def new_invariant(s):
    n = s.get('n', 0)
    if n <= 1:
        return 0.0
    path_bound = (n + 1) / 3.0 # exact ASPL for path graph
    density = s.get('density', 0.0)
    density_bound = 2.0 if density > 0.5 else n - 1
    # Moore bound using max/min/average degree
    def moore_radius(deg, n):
        if deg <= 1: return n - 1
        nodes, power, radius = 1, 1, 0
        while nodes < n:
            nodes += deg * power
            power *= (deg - 1)
            radius += 1
        return radius + 1
    moore_max = moore_radius(int(s.get('max_degree', 1)), n)
    moore_min = moore_radius(int(s.get('min_degree', 1)), n)
    moore_avg = moore_radius(int(s.get('avg_degree', 1)), n)
    final = min(path_bound, density_bound, moore_max, moore_min, moore_avg, n - 1)
    return float(final)
```

The formula takes the minimum of six independent upper bounds. The path-graph bound $(n+1)/3$ is a direct rediscovery of the classical exact result; the Moore-bound terms mirror how human mathematicians combine degree-based inequalities.

Self-Correction Ablation

Table 5 compares the system with and without the self-correction loop across three matched seeds (seeds 11, 22, 33) on the ASPL benchmark. SC-off runs used identical configs except `enable_self_correction: false`.

Table 5: SC ablation: self-correction enabled vs. disabled across 3 matched seeds on the ASPL benchmark (seeds 11, 22, 33). Val ρ is the composite-fitness-optimised validation metric; Test ρ measures held-out generalization. Δ = SC on − SC off. n=3 seeds; differences should be interpreted cautiously.

Condition	Seeds	Val ρ	Test ρ
SC enabled	3	0.928 ± 0.006	0.926 ± 0.012
SC disabled	3	0.883 ± 0.018	0.943 ± 0.004
Δ (on − off)		$+0.045$	-0.018

SC improves validation correlation by $+0.045$ (7.5× the SC-on validation std), confirming the repair loop advances search progress against the composite fitness objective. The test gap (-0.018, SC-off marginally higher) is within 1.5× the SC-on test standard deviation and is not statistically conclusive at n=3 seeds. The 3× reduction in validation standard deviation (0.006 vs. 0.018) indicates that SC stabilises convergence across seeds rather than merely overfitting to the validation set.

A Rebuttal-Oriented Supplementary Tables

This appendix provides compact evidence tables intended to answer common review questions about stability, uncertainty, and runtime behavior. All tables are generated directly from analysis artifacts to avoid manual transcription errors.

A.1 Multi-seed aggregate metrics

A.2 Bounds diagnostics

A.3 Runtime and completion summary

A.4 Self-correction failure breakdown

A.5 Compute profile

References

[1] Anonymous. Graph invariant discovery: Raw experimental artifacts, 2026. URL `https://zenodo.org/record/18727765`.

[2] Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl, 2023. URL `https://arxiv.org/abs/2305.01582`.

Table 6: Seed-aggregated performance for NeurIPS matrix runs. †Groups marked with † have incomplete seeds.

Group	Seeds	Success	Val mean±std	Test mean±std	Test CI95
ablation_sc_off	3	3/3	0.883±0.018	0.943±0.004	±0.010
benchmark/benchmark_20260215T230550Z	5	5/5	0.927±0.012	0.921±0.030	±0.038
neurips_matrix_day1_2026-02-21/algebraic_connectivity_medium†	2	0/2	0.899±0.000	0.892±0.019	±0.108
neurips_matrix_day1_2026-02-21/benchmark_aspl_medium	3	0/3	0.065±0.489	-0.011±0.510	±0.989
neurips_matrix_day1_2026-02-21/map_elites_aspl_medium	3	0/3	0.106±0.420	0.031±0.438	±0.969
neurips_matrix_day1_2026-02-21/small_data_aspl_train20_medium	3	0/3	0.144±0.144	0.263±0.264	±0.656
neurips_matrix_day1_2026-02-21/small_data_aspl_train35_medium	3	1/3	0.899±0.052	0.913±0.029	±0.072
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium	3	0/3	0.382±0.060	0.403±0.045	±0.111

†algebraic_connectivity_medium uses 2/3 completed seeds; missing seed_11 (missing phase1_summary.json (pathological runtime; marked failed)).

Table 7: Validation/test bounds diagnostics for bounds-mode runs.

Experiment	Val score	Val sat.	Test score	Test sat.
experiment_upper_bound_aspl	0.514	0.870	0.499	0.845
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_11	0.302	0.559	0.398	0.695
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_22	0.281	0.809	0.292	0.805
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_33	0.351	0.918	0.349	0.918

[3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. URL `https://arxiv.org/abs/1609.02907`.

[4] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992. ISBN 978-0-262-11170-6.

[5] Joel Lehman and Kenneth O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on the Synthesis and Simulation of Living Systems (ALIFE 2008)*, pages 329–336. MIT Press, 2008.

[6] Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: A review, 2024. URL `https://arxiv.org/abs/2211.10873`.

[7] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015. URL `https://arxiv.org/abs/1504.04909`.

[8] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. ISSN 1476-4687. doi: 10.1038/s41586-023-06924-6.

[9] Darrell Whitley, Soraya Rana, and Robert B. Heckendorn. *Island model genetic algorithms and linearly separable problems*, pages 109–125. Springer Berlin Heidelberg, 1997. ISBN 9783540695783. doi: 10.1007/bfb0027170.

[10] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019. URL `https://arxiv.org/abs/1810.00826`.

Table 8: Runtime summary for matrix runs from matrix_summary files.

Experiment	Mean sec	Std sec	Completed/Total	Criteria success
No runtime summary found.				

Table 9: Self-correction failure categories per experiment. Counts are cumulative event tallies: a single failed attempt may be counted in multiple categories (e.g., both no valid predictions and below novelty gate), so column sums can exceed SC attempted. Novelty-gate rejections dominate, confirming the hard gate is the primary bottleneck rather than code quality.

Experiment	SC attempted	No valid preds	Below train thr.	Below novelty gate
experiment_algebraic_connectivity	75	48	43	55
experiment_map_elites_aspl	164	105	38	145
experiment_upper_bound_aspl	56	47	7	36
experiment_v2	36	40	24	53
neurips_matrix_day1_2026-02-21/algebraic_connectivity_medium/seed_22	27	13	3	24
neurips_matrix_day1_2026-02-21/algebraic_connectivity_medium/seed_33	22	13	N/A	19
neurips_matrix_day1_2026-02-21/benchmark_aspl_medium/seed_11	2	N/A	2	3
neurips_matrix_day1_2026-02-21/benchmark_aspl_medium/seed_22	1	N/A	1	2
neurips_matrix_day1_2026-02-21/benchmark_aspl_medium/seed_33	3	N/A	1	5
neurips_matrix_day1_2026-02-21/map_elites_aspl_medium/seed_11	1	N/A	4	2
neurips_matrix_day1_2026-02-21/map_elites_aspl_medium/seed_22	1	N/A	N/A	2
neurips_matrix_day1_2026-02-21/map_elites_aspl_medium/seed_33	1	N/A	N/A	2
neurips_matrix_day1_2026-02-21/small_data_aspl_train20_medium/seed_11	1	N/A	1	2
neurips_matrix_day1_2026-02-21/small_data_aspl_train20_medium/seed_22	1	N/A	N/A	2
neurips_matrix_day1_2026-02-21/small_data_aspl_train20_medium/seed_33	1	N/A	1	2
neurips_matrix_day1_2026-02-21/small_data_aspl_train35_medium/seed_11	33	21	6	26
neurips_matrix_day1_2026-02-21/small_data_aspl_train35_medium/seed_22	27	20	1	22
neurips_matrix_day1_2026-02-21/small_data_aspl_train35_medium/seed_33	27	17	1	23
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_11	34	19	6	34
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_22	27	16	6	22
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_33	28	13	1	25

Table 10: Estimated LLM call budget per experiment. LLM generation calls = generations × islands × population. Repair calls = self-correction attempts. Total calls = generation + repair. Estimated at 5–15 s per LLM call on a local 20B model.

Experiment	Gens	LLM gen calls	Repair calls	Total calls	PySR budget
ablation_sc_off/seed_11	20	400	0	400	60 s
ablation_sc_off/seed_22	20	400	0	400	60 s
ablation_sc_off/seed_33	20	400	0	400	60 s
experiment_algebraic_connectivity	20	400	75	475	60 s
experiment_map_elites_aspl	30	600	164	764	60 s
experiment_upper_bound_aspl	20	400	56	456	60 s
experiment_v2	12	240	36	276	60 s
neurips_matrix_day1_2026-02-21/algebraic_connectivity_medium/seed_22	12	108	27	135	60 s
neurips_matrix_day1_2026-02-21/algebraic_connectivity_medium/seed_33	12	108	22	130	60 s
neurips_matrix_day1_2026-02-21/benchmark_aspl_medium/seed_11	8	48	2	50	60 s
neurips_matrix_day1_2026-02-21/benchmark_aspl_medium/seed_22	8	48	1	49	60 s
neurips_matrix_day1_2026-02-21/benchmark_aspl_medium/seed_33	8	48	3	51	60 s
neurips_matrix_day1_2026-02-21/map_elites_aspl_medium/seed_11	8	48	1	49	60 s
neurips_matrix_day1_2026-02-21/map_elites_aspl_medium/seed_22	8	48	1	49	60 s
neurips_matrix_day1_2026-02-21/map_elites_aspl_medium/seed_33	8	48	1	49	60 s
neurips_matrix_day1_2026-02-21/small_data_aspl_train20_medium/seed_11	8	48	1	49	60 s
neurips_matrix_day1_2026-02-21/small_data_aspl_train20_medium/seed_22	8	48	1	49	60 s
neurips_matrix_day1_2026-02-21/small_data_aspl_train20_medium/seed_33	8	48	1	49	60 s
neurips_matrix_day1_2026-02-21/small_data_aspl_train35_medium/seed_11	12	108	33	141	60 s
neurips_matrix_day1_2026-02-21/small_data_aspl_train35_medium/seed_22	12	108	27	135	60 s
neurips_matrix_day1_2026-02-21/small_data_aspl_train35_medium/seed_33	12	108	27	135	60 s
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_11	8	72	34	106	60 s
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_22	12	108	27	135	60 s
neurips_matrix_day1_2026-02-21/upper_bound_aspl_medium/seed_33	12	108	28	136	60 s