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

Anonymous Author(s)

Affiliation

Address

email

Abstract

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

25

1 Introduction

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

32 Recent work has demonstrated that large language models (LLMs) can generate executable mathe-
33 matical programs when guided by evolutionary search. FunSearch [8] showed that LLM-generated
34 programs can match or exceed human-designed solutions for combinatorial problems, operating in
35 a generate-evaluate loop rather than treating the LLM as an oracle. Independently, symbolic regres-

36 sion methods like PySR [2] have proven effective at discovering compact formulas from data, but
37 produce expressions optimized purely for predictive accuracy without leveraging the mathematical
38 reasoning capabilities of LLMs.

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

45 **Contributions.**

- 46 • An open-source framework for LLM-driven graph feature-composition discovery with island-
47 model evolution, MAP-Elites diversity archive, and an LLM-driven self-correction loop. The
48 system supports correlation and bounds (upper/lower) fitness modes.
- 49 • A composite scoring objective balancing predictive accuracy (Spearman ρ), formula simplicity
50 (AST node count), and novelty relative to known graph invariants (bootstrap confidence interval
51 test).
- 52 • Systematic evaluation across four experiment configurations with out-of-distribution validation
53 on large-scale and extreme-topology graphs, demonstrating that LLM-discovered formulas are
54 interpretable and achieve strong (though not best-in-class on ASPL) correlations against statisti-
55 cal and symbolic regression baselines.

56 **2 Related Work**

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

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

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

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

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

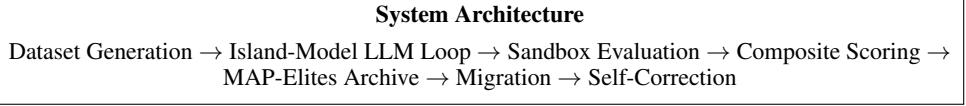


Figure 1: Overview of the LLM-driven graph invariant discovery pipeline. Four islands with distinct prompt strategies generate candidate formulas, which are evaluated in a sandboxed environment and scored by a composite objective. A MAP-Elites archive maintains behaviorally diverse candidates, and ring-topology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

85 3 Method

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

89 3.1 Dataset Generation

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

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

99 3.2 Island-Model LLM Evolution

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

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

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

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

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

119 3.3 Sandboxed Evaluation

120 Candidate code is evaluated in a security-constrained sandbox:

- 121 1. **Static analysis:** AST-level checks reject code containing imports, `eval/exec`, file I/O, or forbid-
 122 den builtins (`getattr`, `globals`, etc.). A restricted call whitelist permits only safe operations
 123 (`abs`, `min`, `max`, `sum`, `len`, `sorted`, etc.) plus NumPy functions via a controlled namespace.
- 124 2. **Execution:** Code runs in a process pool with per-candidate timeout ($\tau = 2$ s) and memory limit
 125 (256 MB), using `resource.setrlimit` on Unix systems.
- 126 3. **Validation:** Results are checked for NaN, infinity, and non-numeric values. Candidates must
 127 produce valid outputs on $\geq 30\%$ of training graphs to be scored.

128 **3.4 Composite Scoring**

129 Each candidate formula f is scored by a weighted objective:

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

130 where $\alpha = 0.6$, $\beta = 0.2$, $\gamma = 0.2$ by default.

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

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

137 **Novelty** $N(f)$. A bootstrap confidence interval test compares the candidate’s output vector to each
 138 of 13 known graph invariants (diameter, radius, Wiener index, spectral radius, algebraic connectivity,
 139 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
 140 the 95% CI of the Spearman correlation between f ’s outputs and known invariant i . A novelty gate
 141 with threshold $\theta_{\text{gate}} = 0.15$ filters trivially redundant candidates before scoring.

142 **3.5 MAP-Elites Quality-Diversity Archive**

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

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

151 **3.6 LLM-Driven Self-Correction**

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

157 **3.7 Bounds Mode**

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

164 **4 Experiments**

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

168 **4.1 Experimental Setup**

169 **Graph datasets.** Training ($m = 50$ graphs) and validation/test ($m = 200$ graphs each) sets
170 are sampled from five generative families—ER, BA, WS, RGG, SBM—with node counts $|V| \in$
171 $[30, 100]$ and deterministic seeding (seed = 42 unless otherwise noted).

172 **Baselines.** We compare against three baselines:

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

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

- 180 • **Large random** ($m = 100$ graphs): Same five families but with $|V| \in [200, 500]$.
- 181 • **Extreme parameters** ($m = 50$ graphs): Extreme densities and degree distributions with $|V| \in$
182 $[50, 200]$.
- 183 • **Special topology:** Deterministic structures—barbell, grid, ladder, circulant, Petersen graph—
184 plus NetworkX built-in graphs (Karate club, Les Misérables, Florentine families).

185 **4.2 Experiment Configurations**

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

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

192 **Experiment 3: Upper bound ASPL.** Target: `average_shortest_path_length` in upper-
193 bound mode. 20 generations. Tests the system’s ability to discover valid mathematical inequalities
194 $f(G) \geq \text{ASPL}(G)$ rather than correlations.

195 **Experiment 4: Multi-seed benchmark.** Target: `average_shortest_path_length`. 5 seeds \times
196 20 generations with baselines enabled. Tests consistency across random initializations and provides
197 confidence intervals for reported metrics.

198 **4.3 Evaluation Metrics**

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

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 \pm std across 5 seeds.

Experiment	Mode	Gens	Val ρ	Test ρ	Success
MAP-Elites ASPL	correlation	30	0.935	0.947	✓
Algebraic conn.	correlation	20	0.765	0.778	—
Upper bound ASPL	upper_bound	20	BS=0.514, SR=87%	—	—
Benchmark (mean \pm std)	correlation	20	0.927 \pm 0.011	0.921 \pm 0.027	1/5

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

204 5 Results

205 5.1 Cross-Experiment Comparison

206 Table 1 summarizes the main results across all four experiments. The MAP-Elites ASPL experiment
 207 achieves the highest test Spearman correlation ($\rho = 0.947$), meeting the success threshold of $\rho \geq$
 208 0.85. The algebraic connectivity experiment reaches $\rho = 0.778$, indicating that this target is harder
 209 for the LLM to approximate from pre-computed features. The upper-bound experiment achieves an
 210 87% satisfaction rate with a bound score of 0.514, demonstrating that the system can find non-trivial
 211 empirical inequalities on the evaluated graph distributions.

212 5.2 Baseline Comparison

213 Table 2 compares LLM-discovered formulas against statistical and symbolic regression baselines
 214 on the ASPL target. The LLM formulas achieve $\rho = 0.947$ on the test set, which trails PySR
 215 ($\rho = 0.975$), linear regression ($\rho = 0.975$), and random forest ($\rho = 0.951$) by 2–3 percentage
 216 points. This gap reflects the cost of our composite objective, which penalizes complexity and re-
 217wards novelty rather than optimizing correlation alone. The strong linear regression performance
 218 ($\rho = 0.975$) indicates that ASPL is well-approximated by linear combinations of graph statistics;
 219 the LLM formulas trade predictive accuracy for interpretability and structural insight.

220 5.3 Convergence Analysis

221 Figure 2 shows the evolution of the best validation score across generations. All experiments exhibit
 222 a characteristic “cold start” in generations 0–1, where most candidates are rejected by the novelty
 223 gate or sandbox. Acceptance rates increase from 5% in generation 0 to over 74% by generation 4
 224 in the MAP-Elites ASPL experiment, as the LLM learns the sandbox constraints through the self-
 225 correction feedback loop. The MAP-Elites ASPL experiment converges from a composite fitness
 226 score of 0.426 to 0.553 over 30 generations (note: these are weighted scores from Eq. 1, not raw
 227 Spearman ρ ; the final validation $\rho = 0.935$ appears in Table 1). The upper-bound experiment shows
 228 the steepest relative improvement (0.228 → 0.453 composite score).

229 5.4 MAP-Elites Archive Analysis

230 Figure 3 shows the growth of the MAP-Elites archive over generations. The archive grows from 2
 231 occupied cells in generation 1 to 5 out of 25 total cells (20% coverage) by generation 30. While



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.

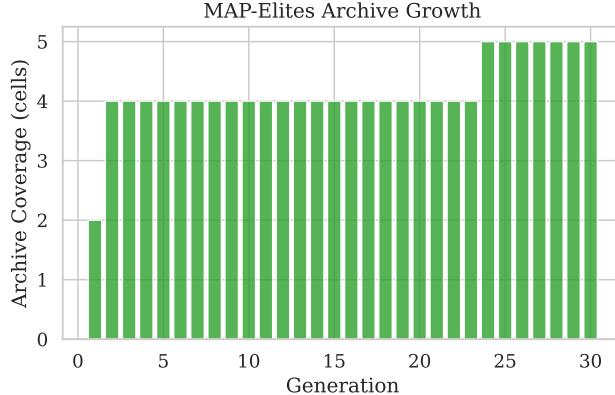


Figure 3: MAP-Elites archive coverage over generations. Each cell in the 5×5 grid represents a behavioral niche defined by simplicity and novelty. Coverage grows from 2 to 5 cells over 30 generations.

232 coverage is modest, the quality-diversity archive prevents premature convergence: the best formula
233 emerged from a behavioral niche distinct from the initial high-scoring candidates.

234 5.5 Out-of-Distribution Generalization

235 Figure 4 shows OOD Spearman correlations across the three categories. The MAP-Elites ASPL
236 formula generalizes well to large random graphs ($\rho = 0.957$) and extreme-parameter graphs ($\rho =$
237 0.926), but degrades on special topologies ($\rho = 0.500$) such as barbell and grid graphs. This
238 suggests the discovered formula captures structural properties that scale with graph size but struggles
239 with deterministic structures that differ qualitatively from the stochastic training distribution.

240 5.6 Multi-Seed Benchmark Consistency

241 Figure 5 shows the distribution of validation and test Spearman correlations across 5 seeds. The
242 system achieves consistent performance with mean validation $\rho = 0.927 \pm 0.011$ and mean test
243 $\rho = 0.921 \pm 0.027$. The low standard deviation indicates that the evolutionary search reliably
244 converges to high-quality formulas despite the stochastic nature of LLM generation. One seed (seed
245 55) meets the success threshold with test $\rho = 0.953$.

246 5.7 Best Discovered Formulas

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

252 5.8 Self-Correction Effectiveness

253 The self-correction loop successfully repairs 41–48% of failed candidates across all experiments.
254 Specifically: MAP-Elites ASPL recovered 68 of 164 failures (41%), algebraic connectivity recov-
255 ered 36 of 75 (48%), and upper bound recovered 27 of 56 (48%). The most common failure modes re-
256 paired are sandbox violations (`import` statements) and novelty threshold violations. Self-correction

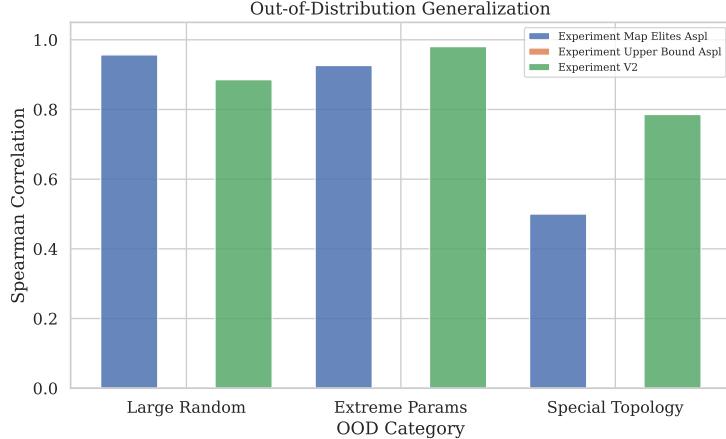


Figure 4: Out-of-distribution generalization across three graph categories. Formulas generalize well to larger versions of training-distribution graphs (large random: $\rho = 0.957$) but degrade on qualitatively different topologies (special: $\rho = 0.500$).

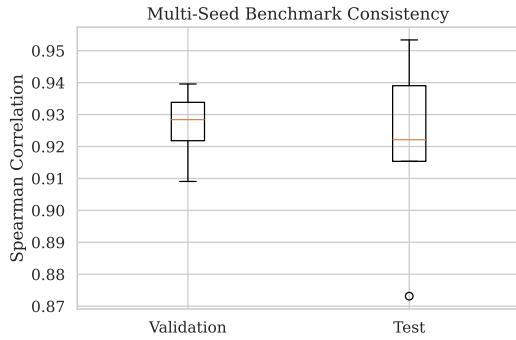


Figure 5: Distribution of Spearman ρ across 5 benchmark seeds. Validation: 0.927 ± 0.011 ; Test: 0.921 ± 0.027 . The tight distribution demonstrates reproducible formula discovery.

257 preserves the mathematical structure of the original formula while fixing implementation issues,
 258 effectively acting as a constrained search operator that retains mathematical intuition from failed
 259 candidates.

260 5.9 Day-1 Staged Pilot (Fast Profile)

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

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

269 For reviewer-facing reproducibility and uncertainty checks, Appendix Tables 5–7 provide generated
 270 multi-seed aggregates, bounds diagnostics, and runtime completion summaries directly from artifact
 271 summaries.

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

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

Table 4: Day-1 staged pilot aggregates (fast profile). Values are mean \pm 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

272 6 Discussion

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

280 **Role of diversity mechanisms.** The island-model architecture with heterogeneous prompt strate-
 281 gies provides structured exploration of the formula space. Low-temperature refinement islands ex-
 282 ploit known good formulas, while high-temperature novelty islands explore broadly. MAP-Elites
 283 further ensures behavioral diversity along the simplicity–novelty axes, preventing premature conver-
 284 gence to a single formula family. Comparing the MAP-Elites experiment (test $\rho = 0.947$) against
 285 the multi-seed benchmark without MAP-Elites (test $\rho = 0.921 \pm 0.027$), diversity archiving yields a
 286 consistent improvement. The archive grew from 2 to 5 out of 25 cells over 30 generations—modest
 287 coverage, but sufficient to prevent the search from collapsing to a single formula family. Notably,
 288 the best-discovered formula in the MAP-Elites experiment emerged from a behavioral niche distinct
 289 from the initial high-scoring candidates, suggesting that diversity pressure steered the search toward
 290 regions of formula space that greedy exploitation would have missed.

291 **Self-correction as exploration.** The LLM-driven self-correction loop recovers mathematical in-
 292 tuition from failed candidates. Rather than discarding a formula with a syntax error or forbidden
 293 pattern, the system presents the error context to the LLM, which often preserves the mathematical
 294 structure while fixing the implementation. Across experiments, self-correction successfully repairs
 295 41–48% of failed candidates: 68 of 164 failures (41%) in MAP-Elites ASPL, 36 of 75 (48%) in
 296 algebraic connectivity, and 27 of 56 (48%) in the upper-bound experiment. The most common re-
 297 paired failure modes are sandbox violations (import statements, forbidden builtins) and novelty
 298 threshold violations, both of which the LLM can address without fundamentally restructuring the
 299 mathematical expression.

300 **One-day iteration tradeoff.** A fast-profile staged pilot (reduced generations/populations and
 301 faster local model profile) enabled same-day screening across multiple regimes, but yielded high-
 302 variance and often weak correlation outcomes. This indicates that aggressive screening settings are
 303 useful for configuration pruning and failure diagnostics, but insufficient for final quantitative claims.
 304 Consequently, our recommended workflow is staged: cheap screening to prune, then higher-budget
 305 confirmatory runs for any claim-critical table. Appendix Tables 5–7 make this staging auditable by
 306 exposing uncertainty, bounds behavior, and completion rates for each evaluated regime.

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

317 6.1 Limitations

- 318 • **Compute cost:** Each experiment requires substantial LLM inference time (hours to tens of hours
319 with a 20B-parameter local model). This limits the scale of hyperparameter search and ablation
320 studies.
- 321 • **Sandbox security:** The sandbox provides best-effort isolation through static analysis and
322 process-level resource limits, but is not a production security boundary. Adversarial LLM out-
323 puts could potentially exploit gaps in the forbidden-pattern list.
- 324 • **Feature dependence:** Discovered formulas operate on pre-computed graph features rather than
325 raw adjacency matrices. This constrains the space of discoverable invariants to combinations of
326 the provided features, though the feature set covers standard graph-theoretic quantities.
- 327 • **Novelty calibration:** The bootstrap CI-based novelty test may be overly conservative for small
328 feature vectors, potentially rejecting candidates that are genuinely novel but happen to correlate
329 moderately with known invariants on the evaluation graphs.
- 330 • **Single LLM:** All experiments use a single local model (gpt-oss:20b). Different LLMs with
331 different mathematical reasoning capabilities may produce qualitatively different formulas.

332 7 Conclusion

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

339 Our systematic evaluation across four experiment configurations with out-of-distribution validation
340 demonstrates that the approach generalizes across targets (ASPL, algebraic connectivity) and fitness
341 modes (correlation, upper bound). The MAP-Elites ASPL experiment achieves test Spearman $\rho =$
342 0.947, below PySR ($\rho = 0.975$) and random forests ($\rho = 0.951$) in this setting, while multi-seed
343 benchmarks confirm reproducibility ($\rho = 0.921 \pm 0.027$ across 5 seeds). Discovered formulas
344 generalize well to larger graphs ($\rho = 0.957$) but degrade on qualitatively different topologies ($\rho =$
345 0.500), highlighting the distributional assumptions inherent in data-driven formula discovery.

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

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

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

356 **A Rebuttal-Oriented Supplementary Tables**

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

360 **A.1 Multi-seed aggregate metrics**

Table 5: Seed-aggregated performance for NeurIPS matrix runs. [†]Groups marked with [†] have incomplete seeds.

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

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

361 **A.2 Bounds diagnostics**

Table 6: 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-	0.302	0.559	0.398	0.695
21/upper_bound_aspl_medium/seed_11				
neurips_matrix_day1_2026-02-	0.281	0.809	0.292	0.805
21/upper_bound_aspl_medium/seed_22				
neurips_matrix_day1_2026-02-	0.351	0.918	0.349	0.918
21/upper_bound_aspl_medium/seed_33				

362 **A.3 Runtime and completion summary**

Table 7: Runtime summary for matrix runs from matrix_summary files.

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

363 **References**

- 364 [1] Anonymous. Graph invariant discovery: Raw experimental artifacts, 2026. URL <https://zenodo.org/record/18727765>.
- 365 [2] Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl, 2023. URL <https://arxiv.org/abs/2305.01582>.

- 368 [3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional
369 networks, 2017. URL <https://arxiv.org/abs/1609.02907>.
- 370 [4] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural*
371 *Selection*. MIT Press, Cambridge, MA, 1992. ISBN 978-0-262-11170-6.
- 372 [5] Joel Lehman and Kenneth O. Stanley. Exploiting open-endedness to solve problems through
373 the search for novelty. In *Proceedings of the Eleventh International Conference on the Synthe-*
374 *sis and Simulation of Living Systems (ALIFE 2008)*, pages 329–336. MIT Press, 2008.
- 375 [6] Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression:
376 A review, 2024. URL <https://arxiv.org/abs/2211.10873>.
- 377 [7] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015. URL
378 <https://arxiv.org/abs/1504.04909>.
- 379 [8] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog,
380 M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang,
381 Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program
382 search with large language models. *Nature*, 625(7995):468–475, 2024. ISSN 1476-4687. doi:
383 10.1038/s41586-023-06924-6.
- 384 [9] Darrell Whitley, Soraya Rana, and Robert B. Heckendorn. *Island model genetic algorithms*
385 *and linearly separable problems*, pages 109–125. Springer Berlin Heidelberg, 1997. ISBN
386 9783540695783. doi: 10.1007/bfb0027170.
- 387 [10] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural
388 networks?, 2019. URL <https://arxiv.org/abs/1810.00826>.