SYMBOLICARENA: A UNIFIED INFRASTRUCTURE FOR BENCHMARK DISTILLATION AND DYNAMIC EVALUATION IN SYMBOLIC REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Symbolic regression (SR) seeks concise and interpretable mathematical expressions from data for scientific equation discovery. Existing SR benchmarks face a tradeoff between evaluation cost and benchmark validity. Repeated evaluation of large task pools is expensive, and compact benchmarks lack systematic evidence of preserved task diversity and algorithm discriminability. ***SymbolicArena*** provides a unified infrastructure for benchmark distillation and dynamic evaluation. The framework standardizes 664 heterogeneous tasks with executable ground truth expressions and distills the *Full Task Set* into *Core50*, a validated benchmark of 50 tasks. The distillation process preserves task coverage and algorithm discrimination under explicit balance constraints. *SymbolicArena* applies a unified execution protocol to heterogeneous SR algorithms and produces comparable outputs and search trajectories. Multi Axis Evaluation characterizes numerical quality, symbolic quality, and search behavior. *Core50* reduces evaluation workload by 92.5% and maintains agreement with *Full Task Set* evaluations. Experiments show that *SymbolicArena* achieves 72.6% to 86.7% lower approximation error than alternative selectors, further supporting its fidelity to the *Full Task Set*. Evaluation reveals a substantial gap between numerical fitting and symbolic recovery across current SR methods, suggesting that reliable equation recovery remains an open challenge.

1 INTRODUCTION

Symbolic regression (SR) aims to recover compact and interpretable mathematical expressions from data, providing an approach for scientific equation discovery and symbolic modeling (Langley, 1981; Schmidt & Lipson, 2009). Recent progress in SR has expanded from evolutionary search (Cranmer, 2023; Koza, 1992) to neural generation (Landajuela et al., 2022; Valipour et al., 2021) and large language model assisted discovery (Grayeli et al., 2024; Shojaee et al., 2025a). The growing number of SR algorithms increases the demand for reliable and consistent evaluation.

Existing SR benchmarks provide resources for algorithm comparison. Large benchmark collections support broad task coverage at substantial cost. Compact collections reduce evaluation cost, yet evidence that compact collections preserve conclusions from larger task populations remains limited. A reliable compact benchmark requires validation beyond task diversity. Selected tasks maintain algorithm discrimination and preserve conclusions from a larger task population. A systematic strategy for constructing compact and reliable SR benchmarks remains unavailable.

The evaluation challenge becomes stronger as SR algorithms adopt diverse search mechanisms and output representations. Existing benchmarks mainly emphasize task collection and final performance measurement. Reliable comparison also requires consistent execution and evaluation criteria that characterize different aspects of algorithm behavior.

SymbolicArena provides a unified framework for benchmark distillation and evaluation in symbolic regression. Compact benchmark construction is formulated as a distillation problem. The framework combines task information and algorithm responses from 664 standardized SR tasks to construct *Core50*, a validated benchmark of 50 tasks. The distillation process preserves task coverage and algorithm discrimination at a reduced evaluation budget. Beyond benchmark construction,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

SymbolicArena provides a unified evaluation protocol for diverse SR algorithms. The framework standardizes execution and records search trajectories. Multi Axis Evaluation organizes metrics into numerical quality, symbolic quality, and search behavior.

Experiments on 15 representative SR algorithms validate the fidelity and diagnostic value of *SymbolicArena*. *Core50* reduces approximation error to the *Full Task Set* by 72.6% to 86.7% relative to alternative selectors. Across nine algorithms at both scales, Spearman rank correlation reaches 0.9833 for ID and 0.9667 for OOD. The OOD ordering of three validation algorithms, FePySR, SymbolFit, and JAXSR, is preserved. The main contributions are as follows.

Benchmark Distillation. We formulate compact SR benchmark construction as a distillation problem and derive *Core50*, a validated 50 task subset from 664 executable tasks.

Unified Evaluation. *SymbolicArena* standardizes heterogeneous SR algorithms in a unified execution protocol, enabling reproducible comparison across representative methods.

Multi Axis Evaluation. We evaluate numerical quality, symbolic quality, and search behavior, revealing tradeoffs beyond final numerical error.

2 Related Work

Symbolic Regression Benchmarks. SR evaluation ranges from classical equation sets to broader benchmark suites. Nguyen (Uy et al., 2011), Keijzer (Keijzer, 2003), Korns (Korns, 2011), and Vladislavleva (Vladislavleva et al., 2009) provide controlled settings for equation recovery and extrapolation. SRBench (La Cava et al., 2021) expands evaluation across synthetic and real world regression problems, and SRBench 2025 (Imai Aldeia et al., 2025) considers predictive accuracy and expression complexity. SRSD (Matsubara et al., 2024) focuses on scientific equation rediscovery, and LLM SRBench (Shojaee et al., 2025b) evaluates LLM based equation discovery. Table 1 compares representative benchmarks and evaluation capabilities. Existing SR benchmarks provide broad task coverage and specialized evaluation protocols. Validation of compact benchmark fidelity remains limited, particularly for preserving task coverage and algorithm discrimination.

Table 1: **Comparison with representative symbolic regression benchmarks.**

Benchmark	Tasks	Focus	HP	CP	VC	ST
SRBench (La Cava et al., 2021)	252	Algorithm comparison	✓	✓	×	×
SRBench 2025 (Imai Aldeia et al., 2025)	24	Accuracy and complexity	✓	✓	×	×
SRSD (Matsubara et al., 2024)	240	Scientific equation rediscovery	×	×	×	×
LLM-SRBench (Shojaee et al., 2025b)	240	LLM based equation discovery	×	×	×	×
Nguyen (Uy et al., 2011)	12	Equation recovery and extrapolation	×	×	×	×
Keijzer (2003)	15					
Korns (2011)	15					
Vladislavleva (2009)	8					
SymbolicArena	**664→50**	**Validated benchmark distillation**	✓	✓	✓	✓

HP: heterogeneous task pool. **CP**: cross paradigm evaluation. **VC**: validated compact benchmark. **ST**: search trajectory recording.

Efficient Benchmarking. Efficient evaluation has received attention in machine learning. tinyBenchmarks (Polo et al., 2024), Anchor Points (Vivek et al., 2024), and SubLIME (Saranathan et al., 2025) investigate compact subset selection for approximating benchmark performance or preserving model rankings. Prior research concentrates on NLP and general machine learning. Compact SR evaluation remains less explored due to heterogeneous tasks and diverse algorithm responses. A broader review of related work is provided in Appendix A.

3 SymbolicArena

SymbolicArena contains two stages for compact and reproducible SR evaluation, as shown in Figure 1. **Benchmark distillation** constructs a compact task set that preserves task coverage and algo-

rithm responses from the *Full Task Set*. **Unified evaluation** applies a common execution protocol to the *Core50* and measures numerical quality, symbolic quality, and search behavior.

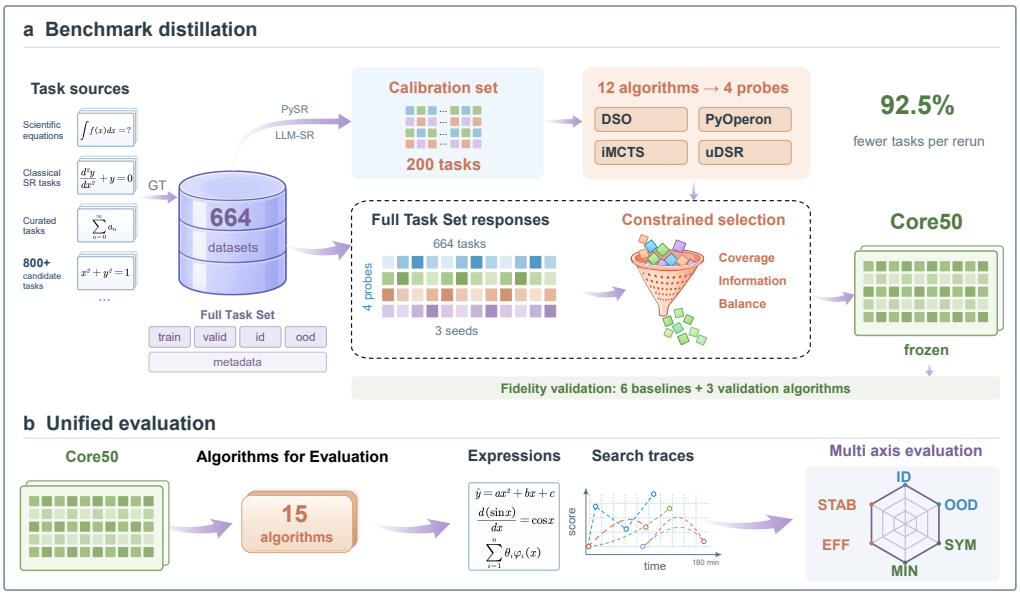


Figure 1: **Overview of *SymbolicArena*. a.** Benchmark distillation retains 664 tasks from 800 candidates, forming the *Full Task Set*. The *Calibration Set* tasks support selection of the *four probes*. Task properties and probe responses across *Full Task Set* guide selection of 50 tasks in *Core50*. **b.** *SymbolicArena* runs 15 SR algorithms on *Core50* using a common protocol and records outputs and search trajectories. Multi Axis Evaluation reports numerical quality (ID and OOD), symbolic quality (SYM and MIN), and search behavior (EFF and STAB).

3.1 Benchmark Distillation

We formulate compact SR evaluation as a subset selection problem. The goal is to identify a task subset whose aggregate scores and algorithm separation remain consistent with the *Full Task Set*. The construction proceeds from task standardization to calibration selection, probe selection, and final benchmark distillation. The details and the task manifest appear in Appendix C and Table 7.

***Full Task Set* Construction.** To support consistent evaluation across heterogeneous SR tasks, *SymbolicArena* maps each candidate task to a unified schema. The schema specifies feature ordering, target information, and fixed data splits. Ground truth expressions and structural metadata are retained for symbolic analysis and duplicate control.

A validation pipeline checks expression parsing, variable consistency, and target recomputation. The candidate pool exceeds 800 tasks. The 664 valid tasks form *Full Task Set*, denoted by $\mathcal{D}_{664}$:

$$\mathcal{D}_{664} = \left\{ \left(f_i^*, D_i^{\text{train}}, D_i^{\text{ID}}, D_i^{\text{OOD}}, m_i \right) \right\}_{i=1}^{664},$$

where f_i^* denotes the ground truth expression, D_i denotes the fixed data splits, and m_i represents task metadata. Detailed validation procedures are provided in Appendix B.

Calibration Task Selection. The *Calibration Set* is constructed to identify informative tasks from the *Full Task Set*. PySR and LLM-SR serve as initial probes because the search mechanisms produce complementary response patterns. Each probe runs once on every task in *Full Task Set* to obtain an initial performance signal. Calibration selection only identifies informative tasks.

Tasks are ranked according to the normalized performance difference between PySR and LLM-SR on ID and OOD evaluation. The *Calibration Set* contains tasks with strong probe disagreement and tasks solved by a single probe. Additional tasks with moderate disagreement are included to maintain broader task coverage. The construction audit is provided in Appendix C.1.

Probe Selection. Probe selection maximizes mean calibration usability subject to coverage of at least three primary paradigms. Calibration usability combines finite result rate and resistance to numerical explosion. The 12 algorithms on the 200 calibration tasks remain eligible. Among four algorithm subsets covering at least three primary paradigms, the unique optimum is

$$\mathcal{A}_4 = \{\mathrm{DSO}, \mathrm{PyOperon}, \mathrm{iMCTS}, \mathrm{uDSR}\}.$$

The four probes cover Neural Policy Search, Evolutionary Search, and Structured Search. Definitions and selection results appear in Appendix C.2.

***Core50* Construction.** The four probes are evaluated on 664 tasks in the *Full Task Set* using three random seeds. Error summaries, output validity, and seed variation characterize each task. Structural descriptors encode task properties. Response descriptors capture performance differences among the probes. Task informativeness combines discrimination and construction stability as $\mathrm{Info}_i = \mathrm{Disc}_i \cdot \mathrm{Stab}_i$. *Core50* is selected as a feasible subset of 50 tasks by considering task coverage, mean task informativeness, and distributional balance. The selection maximizes the objective function

$$J(S) = 0.45\,\mathrm{Coverage}(S) + 0.35\,\mathrm{MeanInfo}(S) + 0.20\,\mathrm{Balance}(S), \qquad S \in \mathcal{F},$$

where $\mathrm{MeanInfo}(S) = |S|^{-1} \sum_{i \in S} \mathrm{Info}_i$, $|S| = 50$. The coverage is defined as follows

$$\mathrm{Coverage}(S) = 0.6\,\mathrm{StructuralCoverage}(S) + 0.4\,\mathrm{ResponseCoverage}(S).$$

$\mathrm{Balance}(S)$ measures how well the subset represents the distributions of task difficulty and failure modes. The feasible set $\mathcal{F}$ controls source composition and duplicate membership. Additional constraints regulate difficulty and failure patterns. Details appear in Appendix C.3.

We optimize the objective using greedy initialization followed by feasible single task swaps, and freeze the best observed subset as *Core50* before fidelity validation and formal evaluation.

3.2 Unified Evaluation Protocol

SymbolicArena evaluates 15 SR algorithms on *Core50* under a standardized execution protocol. Each algorithm uses a common wrapper. Inputs comprise the training split, feature names, and target information. ID and OOD test splits and ground truth expressions are reserved for evaluation.

Algorithm specific environments remain isolated. Data loading, metric computation, and expression processing follow a shared implementation. The execution layer records standardized outputs and trajectories of expressions selected by each algorithm's native rule. The records support reproducible comparison and search analysis. Implementation details are provided in Appendix D.

3.3 Multi Axis Evaluation

Multi Axis Evaluation assesses numerical quality with ***ID*** and ***OOD***. ***SYM*** and ***MIN*** assess symbolic quality. ***EFF*** and ***STAB*** describe search behavior. Details are provided in Appendix E.

Opus 5 postprocessing simplifies final expressions and provides symbolic judgments for SYM and the structural component of STAB. MIN is computed from the frozen simplified expressions. Processed expressions and judgments are frozen before metric aggregation. The postprocessing procedure is detailed in Appendix E.3.

The metrics are defined on a $[0, 1]$ scale. All metrics except STAB use empirical averages across tasks and random seeds. STAB is aggregated across runs within each task and averaged across tasks. In the following formulas, $\mathbb{E}$ denotes the corresponding empirical average.

In-distribution quality ***ID*** measures numerical predictive quality on the ID test set

$$\mathrm{Score}^{\mathrm{ID}} = \mathbb{E}\left[\, q^{\mathrm{ID}} \,\right], \qquad q^{\mathrm{ID}} = \phi(\mathrm{NMSE}^{\mathrm{ID}}),$$

where $\mathrm{NMSE}^{\mathrm{ID}}$ is the normalized mean squared error on the ID test set. For a valid NMSE value x, the clipped log error is

$$r(x) = \mathrm{clip}(\log_{10}(\max(x, \epsilon)), \ell_{\min}, \ell_{\max})\,.$$

Numerical quality is obtained by linear rescaling

$$\phi(x) = \frac{\ell_{\max} - r(x)}{\ell_{\max} - \ell_{\min}}, \qquad \phi(x) \in [0, 1].$$

The bounds are $\ell_{\min} = -12$ and $\ell_{\max} = 2$, with $\epsilon = 10^{-12}$. Lower error yields higher quality. Invalid or unevaluable outputs receive quality zero.

Out-of-distribution quality ***OOD*** measures numerical predictive quality on the OOD test set with the same map

$$\text{Score}^{\text{OOD}} = \mathbb{E}\left[\, q^{\text{OOD}} \,\right], \qquad q^{\text{OOD}} = \phi(\text{NMSE}^{\text{OOD}}),$$

where NMSE^{OOD} is the normalized mean squared error on the OOD test set.

Symbolic fidelity ***SYM*** prioritizes symbolic equivalence judgments

$$\text{Score}^{\text{SYM}} = \mathbb{E}\left[\, m^{\text{SYM}} \,\right], \qquad m^{\text{SYM}} = \begin{cases} 1, & Eq = 1, \\ 0.5(S_{\text{tree}} F_v F_o)^{1/3}, & Eq = 0, \end{cases}$$

where the indicator Eq records the evaluator's `equivalent` judgment for the processed prediction and reference. The quantities S_{tree}, F_v, and F_o measure tree similarity, variable recovery, and operator recovery, respectively, as defined in Appendix E.4.

Minimality ***MIN*** measures expression minimality relative to the reference

$$\text{Score}^{\text{MIN}} = \mathbb{E}\left[\, m^{\text{MIN}} \,\right], \qquad m^{\text{MIN}} = \min\left(1, \frac{C^{ref}}{C^{pred}}\right),$$

where C^{ref} and C^{pred} denote the expression tree sizes of the simplified reference and prediction.

Efficiency ***EFF*** measures how rapidly search approaches the best numerical quality reached within a run. Let T denote the number of checkpoints on the fixed time grid

$$\text{Score}^{\text{EFF}} = \mathbb{E}[m^{\text{EFF}}], \qquad m^{\text{EFF}} = \frac{1}{T}\sum_{t=1}^{T} \frac{q(t)}{q^*},$$

where the numerical quality is $q(t) = (q^{\text{ID}}(t) + q^{\text{OOD}}(t))/2$, and the normalization reference is $q^* = \max_{1 \le t \le T} q(t)$. A run with $q^* = 0$ receives $m^{\text{EFF}} = 0$. The experiment records one checkpoint per minute, and T is expressed in minutes. At each checkpoint, $q(t)$ evaluates the expression selected by the algorithm's native rule. Test quality does not guide candidate selection.

Stability ***STAB*** combines numerical consistency, output validity, and structural consistency

$$\text{Score}^{\text{STAB}} = \mathbb{E}[m^{\text{STAB}}], \qquad m^{\text{STAB}} = (NVC)^{1/3},$$

where N, V, and C denote numerical consistency, valid run fraction, and structural consistency. Detailed definitions are provided in Appendix E.5.

4 Experiments

Experiments examine benchmark fidelity, generalization beyond construction probes, and diagnostic value.

4.1 Experimental Setup

Construction and fidelity evaluation. *Core50* construction follows Section 3.1. The four probes are evaluated on all 664 tasks using three random seeds. Fidelity validation uses a separate one hour evaluation of nine algorithms on *Full Task Set*. FePySR (Yu et al., 2026), SymbolFit (Tsoi et al., 2025), and JAXSR (Kitchin, 2024) are excluded from benchmark construction and serve as validation algorithms. Scores on *Core50* use the same evaluation records restricted to the frozen 50 tasks. PySR and LLM-SR use seed 1314. The remaining seven algorithms use seeds 520, 521, and 522. Probe selection sensitivity appears in Appendix F.

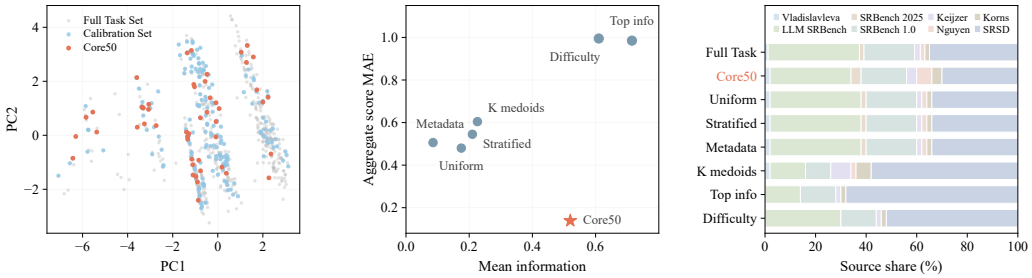
Formal evaluation. The comparison evaluates 15 algorithms on frozen *Core50* using seeds 520, 521, and 522. Appendix H presents the algorithms and configurations. Runs have a three hour budget. Clean evaluation provides the six axis results. Supplementary experiments perturb training labels at 1% and 5%. Construction, fidelity validation, and formal evaluation use separate records. Metric definitions appear in Section 3.3. The execution protocol is described in Appendix D.

4.2 Fidelity to the Full Task Set

***Core50* best preserves aggregate benchmark behavior.** We evaluate how closely each 50 task subset reproduces aggregate probe scores on *Full Task Set*. Six alternative selectors are defined in Appendix C.4. All selectors use the same construction records. Let $A_p(S)$ denote the mean task score of probe p on task set S. Aggregate fidelity is measured by

$$\mathrm{MAE}(S, \mathcal{D}_{664}) = \frac{1}{|\mathcal{A}_4|} \sum_{p \in \mathcal{A}_4} \left| A_p(S) - A_p(\mathcal{D}_{664}) \right| .$$

Lower MAE indicates closer agreement with *Full Task Set*.



a. Coverage b. Tradeoff c. Composition

Figure 2: **Aggregate fidelity and task representation. a.** PCA projection of structural and probe response descriptors for *Full Task Set*, *Calibration Set*, and *Core50*. **b.** Mean task information and aggregate score MAE for *Core50* and six alternative selectors. **c.** Source proportions for *Full Task Set* and the evaluated 50 task subsets.

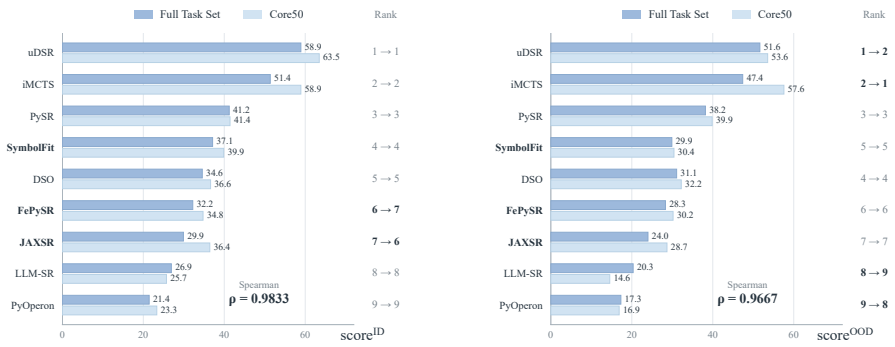
Core50 achieves the lowest aggregate score MAE of 0.1388. In Figure 2a, *Core50* occupies the major regions of the joint structural and response space, indicating broad representation of the task population. In Figure 2b, top information and difficulty balanced selection achieve higher mean task information yet produce much larger MAE. In Figure 2c, several baselines closely match the source proportions of *Full Task Set* yet retain higher MAE. The results indicate that benchmark fidelity benefits from both representative coverage and informative algorithm responses. Optimizing either factor alone is insufficient among the evaluated selectors.

Fidelity on algorithms excluded from construction. We evaluate nine algorithms on both *Full Task Set* and *Core50* using identical evaluation records. Figure 3a and Figure 3b report ID and OOD score and rank agreement across the two scales. FePySR, SymbolFit, and JAXSR are excluded from benchmark construction and serve as validation algorithms. Across all nine algorithms, Spearman correlations reach $\rho = 0.9833$ for ID and $\rho = 0.9667$ for OOD. The three validation algorithms preserve their OOD ordering, and one ID pair changes order. The strong agreement provides evidence that *Core50* fidelity is not confined to the construction probes.

4.3 Diagnostic Analysis of SR Methods

The validated *Core50* enables diagnostic comparison of 15 representative SR algorithms. Table 2 reports point estimates on clean *Core50*. Task bootstrap intervals are provided in Appendix G.2, and axis correlations appear in Appendix G.1.

No evaluated method dominates all dimensions. Table 2 shows that the highest point estimates occur in different methods across numerical, symbolic, and search metrics. iMCTS leads ID and



a. ID b. OOD

Figure 3: **Numerical score and rank agreement.** ID and OOD scores for nine algorithms on *Full Task Set* and *Core50*. FePySR, SymbolFit, and JAXSR are excluded from benchmark construction. Scores use identical evaluation records and a one hour budget.

Table 2: Six-axis profiles of 15 symbolic regression methods on *Core50* (clean). Methods are grouped by their primary search paradigm. Higher values denote better performance on each axis.

		Numerical		Symbolic		Search	
Paradigm	**Algorithm**	**ID ↑**	**OOD ↑**	**SYM ↑**	**MIN ↑**	**EFF ↑**	**STAB ↑**
Structured Search	iMCTS	**77.91**	**73.35**	42.15	80.37	91.66	23.31
	QLattice	34.79	25.37	29.82	55.12	87.81	31.12
	JAXSR	37.99	30.21	32.00	75.03	**99.99**	**97.43**
Evolutionary Search	PySR	**70.34**	**66.28**	**44.60**	**82.62**	91.94	31.03
	PyOperon	38.89	30.64	25.06	44.88	97.28	15.34
	gplearn	31.37	24.96	28.25	53.55	94.67	5.86
	SymbolFit	59.04	48.46	29.96	38.92	87.30	5.38
Neural Policy Search	uDSR	64.84	55.94	41.25	51.41	85.83	20.20
	DSO	38.56	34.14	43.19	79.90	89.60	31.92
Hybrid Search	FePySR	59.84	55.45	**48.13**	**89.55**	93.56	46.53
	RAG-SR	48.58	35.45	20.66	11.25	**99.74**	1.37
Transformer Methods	E2ESR	57.09	42.40	29.96	31.50	90.95	55.70
	TPSR	27.60	20.84	29.60	42.46	87.33	**84.20**
LLM Assisted Search	DrSR	41.86	32.82	39.51	63.94	83.93	10.92
	LLM-SR	42.59	32.78	35.89	59.06	80.55	17.72

Display scale 0–100. Best (1st) Second-best (2nd)

OOD, FePySR leads SYM and MIN, and JAXSR leads EFF and STAB. PySR ranks second on ID, OOD, SYM, and MIN. Figure 4 further reveals distinct profile shapes across methods, indicating strong capability specialization.

Strong numerical accuracy does not guarantee symbolic recovery. Table 2 shows that strong numerical performance does not consistently correspond to strong symbolic fidelity. iMCTS leads both numerical axes yet attains a SYM score of 42.15. A clean `Nguyen-9` case in Appendix G.4 provides a direct example. Near zero numerical error can coexist with failure of strict symbolic equivalence. Final numerical error alone is insufficient to characterize equation recovery.

Search behavior is distinct from attained quality. Table 2 also reveals a clear separation between search behavior and attained solution quality. JAXSR reaches near maximal EFF and STAB yet attains moderate ID and OOD scores. iMCTS shows a contrasting profile, leading both numerical axes with a STAB score of 23.31. A high EFF score can reflect early attainment of a modest

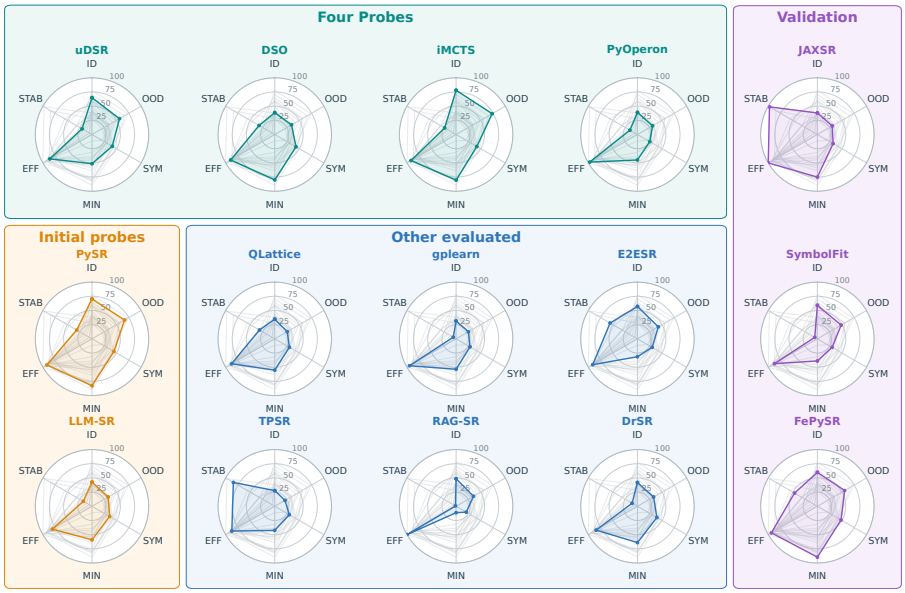


Figure 4: **Per algorithm six axis profiles on *Core50*.** Each panel displays the six axis scores of one algorithm. Normalized scores are multiplied by 100 for a common 0 to 100 display scale. Gray profiles show the remaining algorithms for reference. The panels reveal differences beyond a single numerical ranking.

peak because EFF is normalized to each run's observed maximum. High STAB indicates repeatability and does not directly imply high attained quality. Search progress and repeatability provide complementary information to final solution quality.

5 Limitations

Benchmark scope and fidelity. *Core50* is distilled from the current *Full Task Set* using responses from the *four probes*. Fidelity is validated only for the current task distribution and the algorithmic paradigms represented in the evaluation. New task families or substantially different search paradigms may require renewed fidelity validation of *Core50*. *Core50* also inherits the coverage boundary of the *Full Task Set*. Scientific domains and operator families absent from the current 664 tasks remain outside the validated scope.

Algorithm configuration. The evaluation uses official default or recommended configurations when available and applies only adaptations required by the shared protocol. Algorithm specific hyperparameter optimization is not included. Absolute scores and relative rankings may change with dedicated tuning. The reported results characterize each method in a standardized benchmark configuration.

6 Conclusion

SymbolicArena provides a unified framework for efficient and reproducible symbolic regression evaluation. The framework standardizes 664 executable SR tasks and distills *Core50*, reducing the evaluation workload by 92.5%. Relative to six alternative 50 task selectors, *Core50* reduces aggregate score MAE by 72.6% to 86.7% and preserves strong ranking agreement with *Full Task Set*. Validation on algorithms excluded from benchmark construction further supports the fidelity of the compact benchmark.

Evaluation of 15 representative SR methods reveals clear differences across numerical quality, symbolic quality, and search behavior. No evaluated method dominates all dimensions. Strong numerical accuracy does not consistently correspond to faithful symbolic recovery, and strong search progress

or repeatability does not imply high attained solution quality. The results show that final numerical error alone provides an incomplete view of SR performance. *SymbolicArena* offers a reproducible benchmark and evaluation protocol for studying such differences at substantially lower cost.

Reproducibility Statement. *SymbolicArena* fixes task manifests, data splits, random seeds, execution budgets, algorithm configurations, and evaluation protocols. Appendices B to E document benchmark construction, execution procedures, and metric definitions. Appendix I documents model assisted symbolic evaluation, prompt templates, and information access controls. The supplementary material provides the manifests, scripts, configurations, and frozen symbolic judgments required to reproduce the reported results.

AI USE STATEMENT

Generative AI tools assisted manuscript drafting and revision, LaTeX editing, figure preparation, literature retrieval, code inspection, and analysis of existing experimental records. AI tools also supported research discussion, experimental design review, consistency checks, and interpretation of experimental results. All methodological decisions, experimental designs, and reported results were reviewed and verified by the authors. Opus 5 assisted final expression simplification, equivalence assessment, and structural judgments for seed pairs in the evaluation pipeline. The resulting processed expressions and judgments were frozen before metric aggregation and retained for auditing.

REFERENCES

Anthropic. System card: Claude opus 5. https://www.anthropic.com/claude-opus-5-system-card, July 2026.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International Conference on Machine Learning*, pp. 936–945, 2021.

Kevin René Broløs, Meera Vieira Machado, Chris Cave, Jaan Kasak, Valdemar Stentoft-Hansen, Victor Galindo Batanero, Tom Jelen, and Casper Wilstrup. An approach to symbolic regression using Feyn, 2021. URL https://arxiv.org/abs/2104.05417.

Bogdan Burlacu, Gabriel Kronberger, and Michael Kommenda. Operon C++: An efficient genetic programming framework for symbolic regression. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 1562–1570. Association for Computing Machinery, 2020. doi: 10.1145/3377929.3398099.

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

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Arya Grayeli, Atharva Sehgal, Omar Costilla-Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44678–44709, 2024. doi: 10.52202/079017-1419.

Zhengyao Huang, Daniel Zhengyu Huang, Tiannan Xiao, Dina Ma, Zhenyu Ming, Hao Shi, and Yuanhui Wen. Improving Monte Carlo tree search for symbolic regression, 2025. URL https://arxiv.org/abs/2509.15929.

Guilherme Seidyo Imai Aldeia, Hengzhe Zhang, Geoffrey Bomarito, Miles Cranmer, Alcides Fonseca, Bogdan Burlacu, William G. La Cava, and Fabrício Olivetti de França. Call for action: Towards the next generation of symbolic regression benchmark. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 2529–2538. Association for Computing Machinery, 2025. doi: 10.1145/3712255.3734309.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10269–10281, 2022. doi: 10.52202/068431-0746.

Pierre-Alexandre Kamienny, Guillaume Lample, Sylvain Lamprier, and Marco Virgolin. Deep generative symbolic regression with Monte-Carlo tree search. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15655–15668. PMLR, 2023. URL https://proceedings.mlr.press/v202/kamienny23a.html.

Maarten Keijzer. Improving symbolic regression with interval arithmetic and linear scaling. In *Genetic Programming: Proceedings of EuroGP 2003*, volume 2610 of *Lecture Notes in Computer Science*, pp. 70–82. Springer, 2003. doi: 10.1007/3-540-36599-0_7.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124, 2021. URL https://aclanthology.org/2021.naacl-main.324/.

John Kitchin. JAXSR: JAX-based symbolic regression. Software repository, 2024. URL https://github.com/jkitchin/jaxsr.

Michael F. Korns. Accuracy in symbolic regression. In *Genetic Programming Theory and Practice IX*, Genetic and Evolutionary Computation, pp. 129–151. Springer, 2011. doi: 10.1007/978-1-4614-1770-5_8.

John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/c0c7c76d30bd3dcaefc96f40275bdc0a-Abstract-round1.html.

Mikel Landajuela, Chak Lee, Jiachen Yang, Ruben Glatt, Claudio P. Santiago, Ignacio Aravena, Terrell N. Mundhenk, Garrett Mulcahy, and Brenden K. Petersen. A unified framework for deep symbolic regression. In *Advances in Neural Information Processing Systems*, volume 35, pp. 33985–33998, 2022. doi: 10.52202/068431-2463.

Pat Langley. Data-driven discovery of physical laws. *Cognitive Science*, 5(1):31–54, 1981.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10351–10367, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.

Yoshitomo Matsubara, Naoya Chiba, Ryo Igarashi, and Yoshitaka Ushiku. Rethinking symbolic regression datasets and benchmarks for scientific discovery. *Journal of Data-centric Machine Learning Research*, 1(3):1–38, 2024. URL https://openreview.net/forum?id=qrUdrXsiXX.

Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Alicia Parrish, Hannah Rose Kirk, Jessica Quaye, Charvi Rastogi, Douwe Kiela, David Jurado, David Kanter, Rafael Mosquera, Will Cukierski, Juan Ciro, Lora Aroyo, Bilge Acun, Lingjiao Chen, Mehul Raje, Max Bartolo, Evan Sabri Eyuboglu, Amirata Ghorbani, Emmett Goodman, Addison Howard, Oana Inel, Tariq Kane, Christine R. Kirkpatrick, D. Sculley, Tzu-Sheng Kuo, Jonas W. Mueller, Tristan Thrush, Joaquin Vanschoren, Margaret Warren, Adina Williams, Serena Yeung, Newsha Ardalani, Praveen Paritosh, Ce Zhang, James Zou, Carole-Jean Wu, Cody Coleman, Andrew Ng, Peter Mattson, and Vijay Janapa Reddi. DataPerf: Benchmarks for data-centric AI development. In *Advances in Neural Information Processing Systems*, volume 36, pp. 5320–5347, 2023. doi: 10.52202/075280-0235.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017. doi: 10.7717/peerj-cs.103.

T. Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P. Santiago, Daniel M. Faissol, and Brenden K. Petersen. Symbolic regression via neural-guided genetic programming population seeding. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24912–24923, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d073bb8d0c47f317dd39de9c9f004e9d-Abstract.html.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2519–2536, 2024. doi: 10.18653/v1/2024.naacl-long.139.

Brenden K. Petersen, Mikel Landajuela, T. Nathan Mundhenk, Claudio P. Santiago, Sookyung Kim, and Joanne T. Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=m5Qsh0kBQG.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 34303–34326, 2024. URL https://proceedings.mlr.press/v235/polo24a.html.

Gayathri Saranathan, Cong Xu, Mahammad Parwez Alam, Tarun Kumar, Martin Foltin, Soon Yee Wong, and Suparna Bhattacharya. SubLIME: Subset selection via rank correlation prediction for data-efficient LLM evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30572–30593, 2025. doi: 10.18653/v1/2025.acl-long.1477.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan K. Reddy. Transformer-based planning for symbolic regression. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45907–45919, 2023. doi: 10.52202/075280-1990.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=m2nmp8P5in.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SRBench: A new benchmark for scientific equation discovery with large language models, 2025b. URL https://arxiv.org/abs/2504.10415.

Trevor Stephens. gplearn: Genetic programming in Python, with a scikit-learn inspired API. Software repository, 2016. URL https://github.com/trevorstephens/gplearn.

Ho Fung Tsoi, Dylan Rankin, Cecile Caillol, Miles Cranmer, Sridhara Dasu, Javier Duarte, Philip Harris, Elliot Lipeles, and Vladimir Loncar. SymbolFit: Automatic parametric modeling with symbolic regression. *Computing and Software for Big Science*, 9:12, 2025. doi: 10.1007/s41781-025-00140-9.

Nguyen Quang Uy, Nguyen Xuan Hoai, Michael O'Neill, R. I. McKay, and Edgar Galván-López. Semantically-based crossover in genetic programming: Application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, 2011. doi: 10.1007/s10710-010-9121-2.

Mojtaba Valipour, Bowen You, Maysum Panju, and Ali Ghodsi. SymbolicGPT: A generative transformer model for symbolic regression, 2021. URL https://arxiv.org/abs/2106.14131.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, 2024. doi: 10.18653/v1/2024.eacl-long.95.

Ekaterina J. Vladislavleva, Guido F. Smits, and Dick den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via Pareto genetic programming. *IEEE Transactions on Evolutionary Computation*, 13(2):333–349, 2009. doi: 10.1109/TEVC.2008.926486.

Runxiang Wang, Boxiao Wang, Kai Li, Yifan Zhang, and Jian Cheng. DrSR: LLM-based scientific equation discovery with dual reasoning from data and experience, 2025. URL https://arxiv.org/abs/2506.04282.

Zhiming Yu, Wangtao Lu, and Xin Lai. FePySR: A neural feature extraction framework for efficient and scalable symbolic regression, 2026. URL https://arxiv.org/abs/2605.12704.

Hengzhe Zhang, Chao Chen, Bing Xue, Wolfgang Banzhaf, and Mengjie Zhang. RAG-SR: Retrieval-augmented generation for neural symbolic regression. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NdHka08uWn.

APPENDIX CONTENTS

A Related Work **15**

B *Full Task Set* Construction and Standardization **16**

B.1 Task Sources and Admission 16

B.2 Unified Task Representation 16

B.3 *Full Task Set* Composition 17

C *Core50* Distillation and Validation **18**

C.1 *Calibration Set* Mining 18

C.2 Selection of Four Probes 18

C.3 *Core50* Construction 20

C.4 Selection Baselines and Fidelity Validation 21

C.5 *Core50* Manifest 23

D Unified Execution Protocol **25**

D.1 Wrapper and Execution Architecture 25

D.2 Unified Run Procedure 26

D.3 Recorded Results and Search Trajectories 26

D.4 Execution Isolation and Reproducibility Boundary 26

E Multi Axis Evaluation Details **27**

E.1 Numerical Quality Mapping 27

E.2 Numerical Quality 27

E.3 Opus Assisted Symbolic Postprocessing 27

E.4 Symbolic Quality 28

E.5 Search Behavior 29

E.6 Bootstrap Confidence Intervals 30

E.7 Supplementary Training Noise Robustness 30

F Sensitivity Analysis **31**

F.1 Probe4 Selection Stability and Replacement Structure 31

F.2 *Core50* Construction 32

G Additional Experimental Results **34**

G.1 Correlation Among Evaluation Axes 34

G.2 Uncertainty in Six Axis Scores 34

G.3 Robustness to Training Noise 34

G.4 Numerical Approximation and Symbolic Recovery 36

H Evaluated Algorithms and Experimental Settings **38**

I LLM Usage and Information Boundaries **40**

I.1 Llama in LLM-SR and DrSR 40
I.2 Opus 5 in Symbolic Evaluation 41
I.3 Information Isolation and Leakage Control 42

A Related Work

Symbolic Regression Algorithms. Genetic programming and evolutionary optimization provide established approaches to symbolic regression (Burlacu et al., 2020; Cranmer, 2023). Learning methods guide expression search using several complementary strategies. Deep Symbolic Regression uses reinforcement learning to optimize expression generation (Petersen et al., 2021). Deep Generative Symbolic Regression (Kamienny et al., 2023), iMCTS (Huang et al., 2025), and TPSR (Shojaee et al., 2023) explore tree search and planning. Neural models also guide symbolic search (Landajuela et al., 2022; Mundhenk et al., 2021). NeSymReS and End-to-End Symbolic Regression generate expressions directly from data (Biggio et al., 2021; Kamienny et al., 2022). LLM-SR (Shojaee et al., 2025a) and LaSR (Grayeli et al., 2024) use large language models for equation discovery. RAG-SR (Zhang et al., 2025) and DrSR (Wang et al., 2025) explore further LLM based discovery strategies. The diversity of search strategies and inductive biases motivates a unified evaluation across paradigms.

Symbolic Regression Benchmarks. Existing SR benchmarks support broad algorithm comparison and controlled scientific equation discovery. The Keijzer benchmark (2003) evaluates numerical fitting, linear scaling, and extrapolation. The Vladislavleva benchmark (2009) emphasizes model complexity, numerical accuracy, and extrapolation performance. The Korns benchmark (2011) assesses the exact recovery of complex multivariate expressions. Nguyen evaluates elementary analytical function recovery and genetic programming search capabilities (Uy et al., 2011). SRBench evaluates contemporary SR algorithms across synthetic and real-world regression problems (La Cava et al., 2021). Its recent extension broadens algorithm coverage and compares predictive accuracy, expression complexity, and computational cost (Imai Aldeia et al., 2025). SRSD reconstructs scientific equation-discovery tasks with physically meaningful sampling ranges and expression-level similarity measures (Matsubara et al., 2024). LLM-SRBench evaluates LLM-based equation discovery on tasks designed to reduce direct formula memorization (Shojaee et al., 2025b). The benchmarks provide valuable datasets and evaluation protocols, yet construction, execution, and diagnostic coverage remain heterogeneous. A unified framework that preserves benchmark fidelity and supports cross-paradigm evaluation is still missing.

Benchmark Distillation. Efficient benchmarking seeks to reduce evaluation cost and preserve conclusions from the full benchmark. Perlitz et al. (2024) study how smaller evaluation sets can retain reliable model comparisons. tinyBenchmarks selects compact subsets to approximate full benchmark performance (Polo et al., 2024). Anchor Points identifies representative examples for data-efficient evaluation (Vivek et al., 2024). SubLIME considers subset selection based on model-ranking preservation (Saranathan et al., 2025). Compact evaluation sets can support reliable model comparison. Symbolic regression introduces additional requirements because task structures and algorithm behaviors are heterogeneous.

Dynamic and Multi Axis Evaluation. Recent benchmarking studies assess model behavior beyond a single aggregate score. Dynabench introduced evaluation that evolves alongside model development (Kiela et al., 2021). Dynaboard extended dynamic benchmarking toward multidimensional evaluation and standardized comparison across systems (Ma et al., 2021). DataPerf emphasized reproducible and data-centric benchmarking protocols (Mazumder et al., 2023). SR benchmarks also provide numerical, symbolic, and computational diagnostics (La Cava et al., 2021; Matsubara et al., 2024; Shojaee et al., 2025b). Existing evaluation focuses largely on final outcomes using protocols specific to each benchmark. *SymbolicArena* complements existing protocols with standardized execution and Multi Axis Evaluation. The framework incorporates search trajectories and failure analysis.

B *Full Task Set* CONSTRUCTION AND STANDARDIZATION

Full Task Set is a population of 664 symbolic regression tasks collected from multiple benchmark sources. Construction covers task admission, representation standardization, and validation. Duplicate control and composition auditing complete the process. Ground truth formulas and static metadata are used during *Full Task Set* construction. Algorithm response features are excluded.

B.1 TASK SOURCES AND ADMISSION

The initial pool exceeds 800 symbolic regression tasks collected from modern benchmarks and classical equation sets. Modern sources include SRSD (Matsubara et al., 2024) and LLM-SRBench (Shojaee et al., 2025b). SRBench and its 2025 extension provide further tasks (Imai Aldeia et al., 2025; La Cava et al., 2021). Classical sources comprise the Keijzer (2003), Korns (2011), and Vladislavleva (2009) benchmarks. The Nguyen equation set provides further tasks (Uy et al., 2011).

Admission to *Full Task Set* requires all four conditions below.

- The ground truth formula is unambiguous, parseable as a symbolic expression tree, and executable as a numerical program.
- Variables, constants, and operators are resolvable and consistent with the dataset columns.
- Train, ID test, and OOD test splits are available or reproducibly generated from the executable ground truth program.
- Metadata for provenance, structural analysis, and split reconstruction are available. Duplicate control metadata are also available.

Tasks with ambiguous or unresolvable definitions are excluded. Admission yields the 664 tasks in *Full Task Set*.

B.2 UNIFIED TASK REPRESENTATION

Each admitted task is converted to a common executable schema. The main field groups are summarized in Table 3.

Table 3: **Unified schema of *Full Task Set*.**

Field group	Stored information
Identity and provenance	`dataset_id`, source family, and subgroup. Original basename and source benchmark
Variables and target	Ordered feature names and target name. Number of features and dummy variable indicator
Ground truth expression	Raw source formula and executable expression. Normalized symbolic forms and expression tree
Data splits	Fixed train, ID test, and OOD test samples. Split sizes and parameters required for deterministic regeneration
Structural metadata	Formula complexity and operator types. Variable count bin and OOD type
Duplicate metadata	Semantic duplicate group identifier and provenance links used for duplicate auditing

The raw source formula is preserved for provenance. A normalized symbolic representation is generated separately for structural comparison across sources. Expression normalization provides a unified representation by standardizing equivalent expressions and enforcing deterministic serialization.

Structural analysis uses the resulting expression strings, tree representations, and normalized operator forms. Additional representations are retained for duplicate detection and recovery analysis. The representations support symbolic equivalence analysis, tree comparison, and structural consistency analysis. Recovery analysis covers variables and operators.

The training split serves as the input for symbolic expression search. The ID test split measures generalization within the observed training distribution, and the OOD test split measures extrapolation beyond it. Compatible official splits are retained. Missing splits are generated deterministically from the executable ground truth program. Task metadata record the parameters required to regenerate the splits.

All materialized splits remain fixed and are shared across algorithms. Official numerical metrics are computed on the frozen ID and OOD test splits.

Validation and duplicate control. All standardized tasks are validated by replaying executable ground truth expressions, checking fixed data splits, and reproducing the normalized symbolic representations. Tasks with unresolved inconsistencies are excluded. Semantically equivalent or closely related tasks are assigned a shared `semantic_duplicate_group`, which prevents multiple members of one group from entering *Core50*. The resulting *Full Task Set* forms an executable and reproducible population for subsequent benchmark construction.

B.3 *Full Task Set* COMPOSITION

The 664 tasks cover eight benchmark sources, multiple formula types, and three complexity levels, as summarized in Table 4. All statistics are derived solely from task formulas and static metadata and define the population available for subsequent *Core50* distillation.

Table 4: **Composition of *Full Task Set* (664 tasks).**

Dimension	Categories and task counts
Task sources	LLM-SRBench (240), SRSD (232), SRBench 1.0 (133), Keijzer (15), SRBench 2025 first-principles subset (12), Nguyen (12), Korns (12), Vladislavleva (8)
Formula types	Rational (282), Trigonometric (137), Polynomial (124), Exponential / logarithmic (86), Mixed (34), Unknown (1)
Formula complexity	Simple (284), Moderate (188), Complex (192)

Counts appear in parentheses. Each task belongs to one category within each dimension according to its recorded metadata.

Operator count complexity. Formula complexity is characterized using an operator count extracted from the Python abstract syntax tree of each task's `formula.py`. For task i, the count is defined as

$$C_i = N_i^{\mathrm{BinOp}} + N_i^{\mathrm{UnaryOp}} + N_i^{\mathrm{Compare}},$$

where N_i^{BinOp}, N_i^{UnaryOp}, and N_i^{Compare} denote the numbers of binary operation, unary operation, and comparison nodes in the complete source file. Function calls are excluded from the operator count, and auxiliary functions in `formula.py` contribute to C_i when they contain any counted node type. For example, the stored implementation of Keijzer-1 has $C_i = 7$. The target expression $0.3x_1 \sin(2\pi x_1)$ contributes four binary operation nodes, and auxiliary logic in the same source file contributes three comparison nodes, placing the task in the Moderate group. The 664 tasks are partitioned according to the empirical one third and two thirds quantiles of C_i, yielding boundaries of 5 and 8

$$\text{Simple} : C_i \leq 5, \qquad \text{Moderate} : 6 \leq C_i \leq 8, \qquad \text{Complex} : C_i \geq 9.$$

Tasks with equal operator counts remain in the same group, producing 284 Simple, 188 Moderate, and 192 Complex tasks. The unequal group sizes arise from ties near the quantile boundaries. The resulting categories describe relative operator count complexity in the stored source implementation. Function calls such as `sin` and `exp` receive no additional complexity weight, and auxiliary implementation logic can affect the count. The categories serve as static descriptive metadata for *Full Task Set* composition, separate from empirical task difficulty.

C *Core50* DISTILLATION AND VALIDATION

Core50 construction proceeds from *Calibration Set* mining to *four probes* selection and final task selection. Construction records are separated from fidelity validation and formal evaluation.

C.1 *Calibration Set* MINING

Calibration Set contains 200 tasks selected to expose informative differences among SR algorithms. PySR and LLM-SR serve as initial probes, with one evaluation per probe on every task in *Full Task Set*. Calibration selection is separate from the later selection of *four probes*.

For task i and algorithm a, let $\ell_{i,a}^{\mathrm{id}}$ and $\ell_{i,a}^{\mathrm{ood}}$ denote the clipped $\log \mathrm{NMSE}$ on the ID and OOD test splits. The performance gaps are

$$g_i^{id} = \ell_{i,\mathrm{LLMSR}}^{id} - \ell_{i,\mathrm{PySR}}^{id}, \qquad g_i^{ood} = \ell_{i,\mathrm{LLMSR}}^{ood} - \ell_{i,\mathrm{PySR}}^{ood}.$$

A positive gap indicates an advantage for PySR, and a negative gap indicates an advantage for LLM-SR. The disagreement score combines the ID and OOD differences

$$G_i = \frac{|g_i^{id}| + |g_i^{ood}|}{2}.$$

Calibration Set contains three diagnostic groups. The high disagreement group contributes 140 tasks from the largest response gaps. One sided evaluability contributes 20 tasks for which only one initial probe obtains a successful solution. The moderate gap group contributes 40 tasks from intermediate disagreement regions and broadens coverage beyond extreme cases.

Across source families, *Calibration Set* contains 70 tasks from SRSD, 70 from LLM-SRBench, and 33 from SRBench1.0. Korns contributes 9 tasks. Keijzer, Nguyen, and SRBench2025 contribute 6 tasks each.

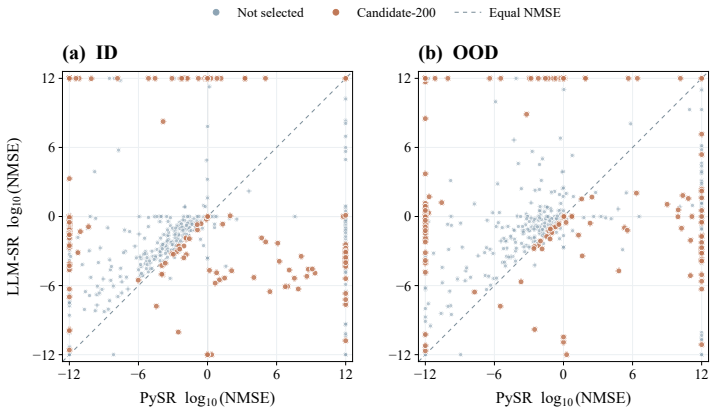


Figure 5: **Dual probe mining diagnostic.** Each point represents one task in the PySR and LLM-SR response space. Off diagonal points indicate large performance differences. Selected *Calibration Set* tasks cover distinct ID and OOD response regions.

Figure 5 shows that *Calibration Set* retains highly separable tasks and less extreme response regions.

C.2 SELECTION OF FOUR PROBES

Probe selection uses one recorded run for each of 12 algorithms on the 200 calibration tasks. The objective favors reliable responses subject to coverage of at least three primary paradigms. The criterion prioritizes reliable response availability and coverage across search paradigms. Task discriminability is measured later from the selected probes on *Full Task Set*, keeping calibration usability separate from final task selection.

Calibration usability. Let $\mathcal{D}_{\text{cal}}$ contain the $N = 200$ calibration tasks, and let $e^s_{i,a}$ denote the NMSE for algorithm a on task i and split s. The tasks with jointly available errors are

$$\mathcal{Q}_a = \left\{ i \in \mathcal{D}_{\text{cal}} : e^s_{i,a} \in \mathbb{R}_{\geq 0} \text{ for all } s \in \{\text{train}, \text{ID}, \text{OOD}\} \right\}.$$

Membership in $\mathbb{R}_{\geq 0}$ requires a finite and nonnegative value. The finite result rate is

$$q(a) = \frac{|\mathcal{Q}_a|}{N}.$$

For a positive threshold τ, define the explosion tasks as

$$\mathcal{B}_{a,\tau} = \left\{ i \in \mathcal{D}_{\text{cal}} : \exists s \in \{\text{ID}, \text{OOD}\}, e^s_{i,a} \in \mathbb{R}_{\geq 0}, e^s_{i,a} > \tau \right\}.$$

The explosion rate is

$$b_\tau(a) = \frac{|\mathcal{B}_{a,\tau}|}{N}.$$

Both rates use all 200 tasks as the denominator. An unavailable error on one split and an explosion on another can affect both rates. Reaching the time limit does not reduce usability if evaluable outputs remain available.

Calibration usability is

$$H_\alpha(a;\tau) = \alpha q(a) + (1-\alpha)\left[1 - b_\tau(a)\right], \qquad 0 \leq \alpha \leq 1.$$

The default values are $\alpha = 0.5$ and $\tau = 100$. The score combines numerical availability and resistance to numerical explosion.

For visualization, $q(a) \geq 0.9$ and $b_{100}(a) \leq 0.3$ define a reference region. Figure 6 visualizes the calibration usability of all 12 candidates at the default setting. All 12 algorithms remain eligible in the subsequent subset search.

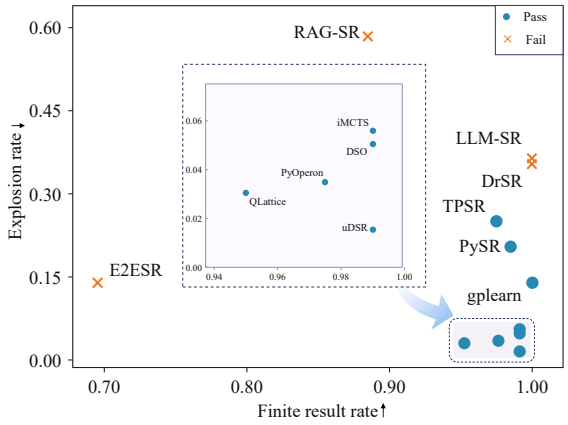


Figure 6: **Calibration usability of candidate algorithms.** Points show finite result rate $q(a)$ and explosion rate $b_{100}(a)$ on *Calibration Set*. The reference region marks high availability and low explosion rates. All 12 algorithms remain eligible for probe selection.

Probe objective and paradigm coverage. Let $\mathcal{A}_{12}$ denote the calibration algorithms and $g(a)$ denote the primary paradigm label used in Table 2. DSO and uDSR belong to Neural and Policy Search. iMCTS and QLattice belong to Structured Search. The labels represent primary categories and do not imply mutually exclusive mechanisms.

For a minimum of m primary paradigms, feasible subsets are

$$\mathcal{P}_m = \{P \subseteq \mathcal{A}_{12} : |P| = 4, \quad |\{g(a) : a \in P\}| \geq m\}.$$

Subset usability is

$$U(P;\alpha,\tau) = \frac{1}{|P|} \sum_{a \in P} H_\alpha(a;\tau).$$

The default rule selects

$$\mathcal{A}_4 \in \arg\max_{P \in \mathcal{P}_3} U(P; 0.5, 100).$$

All $\binom{12}{4} = 495$ subsets are enumerated, and 468 satisfy the three paradigm requirement. Exact rational arithmetic preserves tied scores. A four algorithm subset covering at least three paradigms contains at most two members from one primary paradigm.

Table 5: **Highest scoring feasible four probe combinations.** Mean calibration usability uses $\alpha = 0.5$ and $\tau = 100$. Each reported subset covers three primary paradigms.

Rank	Candidate combination	Mean usability
1	**DSO, PyOperon, iMCTS, uDSR**	**0.971250**
2	DSO, PyOperon, QLattice, uDSR	0.969375
3	PyOperon, iMCTS, QLattice, uDSR	0.968750

Table 5 gives the unique optimum

$$\mathcal{A}_4 = \{\mathrm{DSO}, \mathrm{PyOperon}, \mathrm{iMCTS}, \mathrm{uDSR}\}.$$

The selected subset covers three primary paradigms. Sensitivity analysis appears in Appendix F.

C.3 *Core50* CONSTRUCTION

Core50 selects 50 tasks from *Full Task Set* using probe responses and task descriptors. The same eligibility rules apply to every source family.

Construction records and aggregation. Each record contains ID and OOD NMSE, output validity, and run status. Valid expressions remain evaluable after timeout, while missing or invalid outputs retain explicit annotations.

Finite nonnegative NMSE values use

$$\ell(x) = \mathrm{clip}\left(\log_{10}(\max(x, 10^{-12})), -12, 12\right).$$

For each task i and probe p, let $\mathcal{S}_{i,p}^{\mathrm{valid}}$ contain seeds with valid evaluable outputs. For split $a \in \{\mathrm{id}, \mathrm{ood}\}$, the median log error and seed variability are

$$m_{i,p}^{a} = \mathrm{median}_{s \in \mathcal{S}_{i,p}^{\mathrm{valid}}} \ell(\mathrm{NMSE}_{i,p,s}^{a}), \qquad q_{i,p}^{a} = \mathrm{IQR}_{s \in \mathcal{S}_{i,p}^{\mathrm{valid}}} \ell(\mathrm{NMSE}_{i,p,s}^{a}).$$

The valid fraction $v_{i,p}$ uses all $K = 3$ scheduled seeds. An empty valid seed set leaves the corresponding median and IQR unavailable.

Task descriptors and information. Structural descriptors encode task properties and redundancy information. Response descriptors encode probe performance, validity, and seed variation.

Let $\mathcal{N}$ denote normalization to $[0, 1]$, and let H_i measure the entropy of the valid output pattern across probes. The discrimination score is

$$\begin{aligned}\mathrm{Disc}_i = {} & 0.45\,\mathcal{N}\left(\mathrm{Var}_p(m_{i,p}^{\mathrm{id}})\right) + 0.45\,\mathcal{N}\left(\mathrm{Var}_p(m_{i,p}^{\mathrm{ood}})\right) \\ & + 0.10\,\mathcal{N}(H_i).\end{aligned}$$

Construction instability and stability are

$$u_i = \mathrm{Mean}_p\left[q_{i,p}^{\mathrm{id}} + q_{i,p}^{\mathrm{ood}} + 1 - v_{i,p}\right], \qquad \mathrm{Stab}_i = 1 - \mathcal{N}(u_i).$$

Task information combines discrimination and stability

$$\mathrm{Info}_i = \mathrm{Disc}_i\,\mathrm{Stab}_i.$$

A high information score requires probe separation and consistency across seeds. Construction stability is distinct from the final evaluation axis ***STAB***.

Difficulty, failure mode, and eligibility. Tasks with unusable probe responses are excluded. Partially failed or high variance tasks remain eligible subject to category limits. A timeout alone does not determine eligibility if evaluable outputs remain available.

Difficulty is

$$\text{Difficulty}_i = \text{Mean}_p \left[\frac{m_{i,p}^{\text{id}} + m_{i,p}^{\text{ood}}}{2} \right].$$

Difficulty thresholds define easy, medium, hard, and extreme tasks. Validity and ID to OOD degradation define failure annotations.

***Core50* objective and constraints.** Coverage and mean task information are

$$\text{Coverage}(S) = 0.6\,\text{StructuralCoverage}(S) + 0.4\,\text{ResponseCoverage}(S),$$
$$\text{MeanInfo}(S) = \frac{1}{|S|} \sum_{i \in S} \text{Info}_i.$$

For feasible subsets satisfying $|S| = 50$, the objective combines coverage, mean task information, and distributional balance

$$\boxed{J(S) = 0.45\,\text{Coverage}(S) + 0.35\,\text{MeanInfo}(S) + 0.20\,\text{Balance}(S)}.$$

The feasible set requires

- exactly 50 tasks
- at most one task per semantic duplicate group
- at most one task per basename
- at most 6 tasks per subgroup
- family counts within the shared quota formula
- 4 to 10 easy tasks, 14 to 24 medium tasks, 12 to 22 hard tasks, and 3 to 6 extreme tasks
- at most 8 limited quota tasks
- at most 2 one sided tasks
- at most 5 unstable tasks
- at most 5 all struggle tasks
- 3 to 20 tasks per winning probe

Let $n = 50$, $N = 664$, and F denote the number of source families. Let n_f denote the number of *Full Task Set* tasks from family f. Family targets use

$$t_f = n \left(\beta \frac{n_f}{N} + (1 - \beta) \frac{1}{F} \right), \qquad \beta = 0.6,$$

with bounds

$$q_f^{\min} = \max(1, \lfloor t_f - 1 \rfloor), \qquad q_f^{\max} = \lceil t_f + 2 \rceil.$$

The same quota rule and eligibility criteria apply to every family.

Search and freezing. Greedy initialization constructs feasible subsets using coverage and task information. Feasible single task swaps refine the objective across multiple initializations. The highest observed feasible solution is frozen as *Core50* before fidelity validation.

C.4 Selection Baselines and Fidelity Validation

We compare the frozen *Core50* with six alternative selectors that each produce 50 tasks.

- **random-50** selects 50 distinct dataset IDs uniformly from all 664 tasks.

- **family-stratified random-50** samples distinct dataset IDs within source families using the smoothed targets t_f defined above. Largest remainder rounding assigns integer quotas totaling 50 tasks.
- **metadata-diverse-50** selects diverse tasks using metadata only. Inputs include source family, feature count, and formula complexity. Sample count is also included.
- **response-space K-medoids-50** selects medoids in the response space of *four probes*. The response representation contains ID and OOD behavior, validity, and stability.
- **top-information-50** selects the 50 tasks with the largest Info_i values.
- **difficulty-balanced-50** applies quotas across difficulty bins and uses task information to break ties.

Aggregate score fidelity. For task i and probe p, let K be the number of scheduled runs and $f_{i,p}$ the number of runs with no valid evaluable output. For a task and probe pair with at least one valid run,

$$\bar{m}_{i,p} = \frac{m_{i,p}^{\mathrm{id}} + m_{i,p}^{\mathrm{ood}}}{2}.$$

The task score is

$$z_{i,p} = -\bar{m}_{i,p} - \lambda \frac{f_{i,p}}{K}, \qquad f_{i,p} < K.$$

The evaluation uses $K = 3$ and $\lambda = 2$. A task and probe pair with no valid run receives $z_{i,p} = -12$.

Probe scores are averaged across all tasks in S

$$A_p(S) = \frac{1}{|S|} \sum_{i \in S} z_{i,p}.$$

Aggregate fidelity is measured by

$$\boxed{\mathrm{MAE}(S, \mathcal{D}_{664}) = \frac{1}{|\mathcal{A}_4|} \sum_{p \in \mathcal{A}_4} \left| A_p(S) - A_p(\mathcal{D}_{664}) \right|, \qquad |\mathcal{A}_4| = 4}.$$

All selectors use the same construction records and scoring rule. NMAE divides MAE by the range of the four aggregate probe scores on $\mathcal{D}_{664}$. At the default setting, *Core50* achieves an MAE of 0.1388 and an NMAE of 2.33%.

Penalty sensitivity is evaluated at $\lambda \in \{0, 0.5, 1, 2, 4\}$. *Core50* membership and the -12 assignment for zero valid runs remain fixed. All five settings preserve the probe ordering on *Full Task Set*. NMAE ranges from 2.29% to 2.55%.

Repeated random selection. Each random selector is evaluated on 100 independently sampled subsets of 50 tasks. MAE is computed separately for each draw using fixed probe responses. Table 6 reports the mean and sample standard deviation. Empirical 2.5th and 97.5th percentiles describe variation across sampled subsets.

Two of the 100 uniform draws obtain lower MAE than the fixed *Core50*. No draw from the smoothed family stratified selector reaches the *Core50* value in the current sample. A proportional family quota check yields a mean MAE of 0.4931 and a standard deviation of 0.1959. Complete memberships and per draw results accompany the supplementary data.

The remaining four selectors are reported as fixed point estimates.

Ranking validation. Fidelity is also assessed using algorithm rankings. Validation considers Spearman and Kendall correlations and pairwise order agreement. The four construction probes define six unordered pairs. Validation algorithms excluded from benchmark construction provide additional evidence beyond *four probes*. Ranking results are reported in the main experiments.

Table 6: **Aggregate score fidelity of 50 task selectors.** Random selectors report mean MAE $\pm$ standard deviation and empirical 95% ranges across 100 draws from *Full Task Set*. Other selectors report point estimates. *Core50* is fixed, and the final row gives a proportional family quota check.

Selector	MAE	Empirical 95% range
Uniform random	0.5298 ± 0.2550	$[0.1605, 1.1819]$
Smoothed family stratified	0.5438 ± 0.2072	$[0.2257, 1.0102]$
Metadata diverse	0.5060	—
Response K-medoids	0.6044	—
Top information	1.0339	—
Difficulty balanced	1.0430	—
Core50	**0.1388**	—
Proportional family quota check	0.4931 ± 0.1959	$[0.2459, 0.9728]$

C.5 *Core50* MANIFEST

Table 7 presents the *Core50* membership and ground truth expressions. *Core50* includes 16 LLM-SRBench tasks, 15 SRSD tasks, 9 SRBench1.0 tasks, 3 Nguyen tasks, 2 each from SRBench2025, Korns, and Keijzer, and 1 Vladislavleva task.

Table 7: ***Core50* ground truth formula manifest.** Tasks are grouped by source family. Variables follow the recorded order, and constants use at most four decimal places. Exact expressions and task identifiers are provided in the released metadata.

Idx	Task	Variables	Ground Truth
SRSD	15 tasks		
1	Feynman II.27.18	x_0, x_1, x_2	$8.854 \times 10^{-12} x_0^2$
2	Feynman I.43.31	x_0, x_1	$1.3806 \times 10^{-23} x_0 x_1$
3	Feynman II.34.11	$x_0, \ldots, x_6$	$\frac{x_0 x_2 x_4}{2x_5}$
4	Feynman I.18.16	$x_0, \ldots, x_5$	$x_0 x_1 x_3 \sin x_4$
5	Feynman II.34.2a	$x_0, \ldots, x_4$	$\frac{x_1 x_3}{2\pi x_4}$
6	Feynman I.39.22	$x_0, \ldots, x_4$	$\frac{1.3806 \times 10^{-23} x_1 x_3}{x_4}$
7	Feynman I.11.19	$x_0, \ldots, x_7$	$x_0 x_1 + x_2 x_3 + x_5 x_6$
8	Feynman II.4.23	x_0, x_1	$2.824 \times 10^{10} \frac{x_0}{\pi x_1}$
9	Feynman III.7.38	x_0, x_1	$6.0368 \times 10^{33} \pi x_0 x_1$
10	Feynman I.13.12	$x_0, \ldots, x_3$	$6.6743 \times 10^{-11} x_0 x_1 \left(\frac{1}{x_2} - \frac{1}{x_3}\right)$
11	Feynman III.10.19	$x_0, \ldots, x_5$	$x_2 \sqrt{x_3^2 + x_4^2 + x_5^2}$
12	Feynman I.32.5	x_0, x_1	$\frac{6.9862 \times 10^{-16} x_0^2 x_1^2}{\pi}$
13	Feynman I.29.16	$x_0, \ldots, x_3$	$\sqrt{x_0^2 + 2x_0 x_1 \cos(x_2 - x_3) + x_1^2}$
14	Feynman I.15.3t	$x_0, \ldots, x_4$	$\frac{x_1 - 1.1126 \times 10^{-17} x_3 x_4}{\sqrt{1 - 1.1126 \times 10^{-17} x_3^2}}$
15	Feynman Bonus 20	$x_0, \ldots, x_5$	$\frac{7.8370 \times 10^{-29} x_2^2 \left(\frac{x_2}{x_3} - \sin^2 x_5 + \frac{x_3}{x_2}\right)}{\pi x_3^2}$
SRBench 1.0	9 tasks		
16	Feynman I.27.6	d_1, d_2, n	$\frac{1}{\frac{1}{d_1} + \frac{n}{d_2}}$
17	Feynman II.15.4	mom, B, θ	$-mom\, B \cos\theta$
18	Strogatz Shear Flow 1	x, y	$\frac{\cos x}{\tan y}$
19	Strogatz Predator Prey 2	x, y	$y\left(\frac{x}{1+x} - 0.075y\right)$
20	Strogatz Bar Mag 2	x, y	$0.5 \sin(y - x) - \sin y$
21	Strogatz Bar Mag 1	x, y	$0.5 \sin(x - y) - \sin x$

Table 7: ***Core50* ground truth formula manifest (continued).**

Idx	Task	Variables	Ground Truth
22	Feynman I.12.11	q, E_f, B, v, θ	$q\,(E_f + Bv\sin\theta)$
23	Strogatz Van der Pol 1	x, y	$10\left(y - \frac{1}{3}x^3 + \frac{1}{3}x\right)$
24	Feynman Test 15	c, v, ω, θ	$\frac{\sqrt{1-v^2/c^2}\,\omega}{1+\frac{v}{c}\cos\theta}$
SRBench 2025	2 tasks		
25	Hubble Law	D	$73.3\,D$
26	Leavitt Law	$\log P$	$-2.084\log P + 15.65$
LLM-SRBench	16 tasks		
27	II.34.2	$mom,\ q,\ r$	$\frac{2\,mom}{q\,r}$
28	III.21.20	$j, \rho_{c,0}, q, A_{\text{vec}}$	$-\frac{A_{\text{vec}}\,q\,\rho_{c,0}}{j}$
29	I.11.19	$A, x_1, x_2, x_3, y_1, y_3$	$\frac{-A+x_1y_1-x_3y_3}{x_2}$
30	II.11.27	$P_{ol}, n, \varepsilon, E_f$	$\frac{3P_{ol}}{n(3E_f\varepsilon+P_{ol})}$
31	Chemical Reaction 11	t, A	$-0.8817A^2 + 0.8817\sin\sqrt{A}$
32	Chemical Reaction 10	t, A	$-0.1749A^2 + 0.1749\sin\ln(A+1)$
33	Chemical Reaction 34	t, A	$-0.1689A^2 + 0.1689A^{0.3333}$
34	Physical Oscillation 19	x, t, v	$-0.2v - x\,e^{-\lvert x\rvert}$
35	Chemical Reaction 0	t, A	$-0.1899A^2 + \frac{0.1899A^2}{0.7497A^4+1}$
36	Physical Oscillation 41	x, t, v	$-0.3254(1-x^2)v - 0.3333^2x\,e^{-\lvert x\rvert}$
37	Chemical Reaction 33	t, A	$-0.1184\sqrt{A} - 0.1184A^2 + 0.1184\,t\sin\ln(A+1)$
38	II.6.15b	$E_f, \varepsilon, p_d, \theta$	$\frac{6^{1/3}}{2\pi^{1/3}}\left(\frac{p_d\sin\theta\cos\theta}{E_f\varepsilon}\right)^{1/3}$
39	Bio Population Growth 5	t, P	$0.9198P\left(1-\frac{P}{84.0283}\right) + \frac{0.9198P^2}{7.5314P+1}$
40	Materials Science 21	ε, T	$28.5804\varepsilon^2 - 0.2857(T-281.8702) + 4.2784\varepsilon^3(T-281.8702)$
41	Bio Population Growth 3	t, P	$0.8450P\left(\frac{P}{5.1153}-1\right)\left(1-\frac{P}{34.4388}\right) + 0.8450P\left(1-e^{-0.0968P}\right)$
42	Bio Population Growth 9	t, P	$0.1699P\left(\frac{P}{1.0472}-1\right)\left(1-\frac{P}{10.2105}\right) + 0.1699P\left(1-\frac{P}{10.2105}\right) + 0.1699P\left(1-e^{-0.0970P}\right)$
Nguyen	3 tasks		
43	Nguyen 12	x_1, x_2	$x_1^4 - x_1^3 + \frac{x_2^2}{2} - x_2$
44	Nguyen 6	x_1	$\sin(x_1) + \sin(x_1 + x_1^2)$
45	Nguyen 9	x_1, x_2	$\sin(x_1) + \sin(x_2^2)$
Korns	2 tasks		
46	Korns 4	$x_1, \ldots, x_5$	$0.13\sin(x_3) - 2.3$
47	Korns 2	$x_1, \ldots, x_5$	$0.23 + 14.2\ \text{pdiv}(x_4 + x_2, 3x_5)$
Keijzer	2 tasks		
48	Keijzer 11	x_1, x_2	$x_1x_2 + \sin\big((x_1-1)(x_2-1)\big)$
49	Keijzer 2	x_1	$0.3\,x_1\sin(2\pi x_1)$
Vladislavleva	1 task		
50	Vladislavleva 4	$x_1, \ldots, x_5$	$\frac{10}{5+\sum_{i=1}^{5}(x_i-3)^2}$

For Korns 2, protected division is defined as $\text{pdiv}(a,b) = a/b$ for $|b| > 10^{-3}$ and 1 otherwise.

D UNIFIED EXECUTION PROTOCOL

The protocol provides a shared contract for heterogeneous SR algorithms. Each run specifies an algorithm, task, seed, and time budget. Algorithms receive training data and permitted task information. Fixed ID and OOD test splits and ground truth expressions remain reserved for evaluation. Algorithm specific configurations are reported separately.

D.1 WRAPPER AND EXECUTION ARCHITECTURE

Figure 7 presents the execution architecture of *SymbolicArena*. All entry points share one dataset contract and evaluation interface. Algorithm dependencies remain isolated. Wrappers expose common fitting, prediction, and expression retrieval operations. Native search procedures and model selection rules remain unchanged.

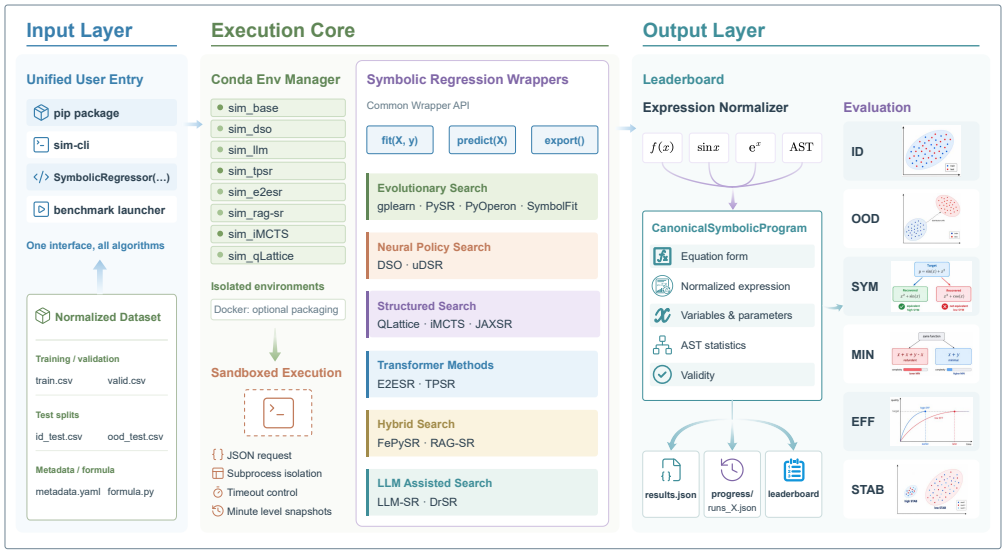


Figure 7: ***SymbolicArena* execution architecture.** A shared task contract connects isolated algorithm wrappers to standardized evaluation artifacts.

The execution core maps each algorithm to an isolated environment. Conda provides isolated environments, and Docker supports optional packaging. A structured request specifies task inputs and run configuration. Subprocess supervision isolates execution, enforces the run budget, and records termination status.

Algorithm 1 defines the common adapter interface. Each wrapper maps native operations to shared fitting, prediction, and expression retrieval functions.

Algorithm 1 Unified SR adapter

Require: Native method A, identifier a, environment E_a
1: $W_a \leftarrow \textsc{CreateAdapter}(A)$
2: $\textsc{Bind}(W_a.\texttt{fit}, A.\texttt{fit})$
3: $\textsc{Bind}(W_a.\texttt{predict}, A.\texttt{predict})$
4: $\textsc{Bind}(W_a.\texttt{get_optimal_equation}, A)$
5: $\textsc{Bind}(W_a.\texttt{get_total_equations}, A)$
6: $\textsc{Register}(a, E_a, W_a)$

Native outputs are converted to a shared symbolic representation before metric computation. The representation binds expression content to variables and fitted parameters. Structural statistics and validity information support symbolic evaluation and output checks. Raw outputs are retained for auditing. Symbolic simplification and adjudication follow Appendix E.

D.2 Unified Run Procedure

Algorithm 2 specifies one run defined by an algorithm, task, seed, and evaluation condition.

Algorithm 2 Unified symbolic regression run procedure

Require: Algorithm a, dataset d, seed s, condition c, budget B, checkpoint interval Δ
$D \leftarrow \textsc{LoadAndValidate}(d)$
$\theta \leftarrow \textsc{BuildConfig}(a, s, B, \mathrm{Schema}(D))$
$\widetilde{D}_{\mathrm{tr}} \leftarrow \textsc{PerturbTrainingLabels}(D_{\mathrm{tr}}, c, s)$
$w \leftarrow \textsc{StartIsolatedFit}(a, \widetilde{D}_{\mathrm{tr}}, \theta)$
$\mathcal{T} \leftarrow \textsc{EmptyTrajectory}$; $t \leftarrow 1$
while $\textsc{Running}(w) \wedge t\Delta \leq B$ **do**
  $\textsc{WaitUntilCheckpoint}(t\Delta)$
  $(f_t, u_t, e_t) \leftarrow \textsc{ReadNativeIncumbent}(w)$
  if e_t provides an auditable candidate **then**
    $p_t \leftarrow \textsc{ExportProgram}(f_t)$
    $m_t \leftarrow \textsc{EvaluateNumeric}(p_t, D_{\mathrm{ID}}, D_{\mathrm{OOD}})$
    $\mathcal{T}[t] \leftarrow (p_t, u_t, m_t, e_t)$
  else
    $\mathcal{T}[t] \leftarrow \textsc{Unavailable}(e_t)$
  end if
  $t \leftarrow t + 1$
end while
$\mathcal{T} \leftarrow \textsc{CompleteTrajectory}(\mathcal{T}, B, \Delta)$
$(\hat{f}, \sigma) \leftarrow \textsc{FinalizeOrRecover}(w, \mathcal{T})$
if $\textsc{Valid}(\hat{f})$ **then**
  $p \leftarrow \textsc{ExportProgram}(\hat{f})$
  $M \leftarrow \textsc{EvaluateNumeric}(p, D_{\mathrm{ID}}, D_{\mathrm{OOD}})$
else
  $p \leftarrow$ Unavailable; $M \leftarrow$ NA
end if
$\textsc{Record}(a, d, s, c, p, \mathcal{T}, M, \sigma)$

Let $D = (D_{\mathrm{tr}}, D_{\mathrm{val}}, D_{\mathrm{ID}}, D_{\mathrm{OOD}})$, and let θ denote the algorithm specific configuration. At each checkpoint, the wrapper records the native incumbent and its provenance. An auditable incumbent is exported to the shared representation and evaluated on the fixed ID and OOD splits.

ID and OOD scores remain external to search and candidate selection. Training label noise modifies D_{tr} only. Failed or unavailable outputs retain explicit status labels.

D.3 Recorded Results and Search Trajectories

Each run produces a final artifact and search trajectory. The final artifact stores the selected expression, evaluation results, and run status. Trajectory records store checkpoint expressions and provenance.

Formal evaluation uses 180 minutes with one checkpoint per minute. Each checkpoint stores the native best so far expression supported by auditable evidence. The latest auditable expression is carried forward between updates and after early termination. A checkpoint lacking sufficient evidence remains marked as `unavailable`.

Recorded artifacts support Multi Axis Evaluation. Final expressions support numerical and symbolic evaluation. Search trajectories provide temporal evidence for EFF, and repeated seed runs provide the records required by STAB.

D.4 Execution Isolation and Reproducibility Boundary

Separate environments and subprocess isolation reduce dependency conflicts across algorithms. Frozen task inputs, execution rules, environment specifications, and recorded artifacts define the reproducibility boundary. Additional boundaries for language model assisted search and symbolic postprocessing are described in Appendix I.

E MULTI AXIS EVALUATION DETAILS

The six axes are fixed independently of the evaluated algorithm set. ID and OOD measure numerical quality, SYM and MIN measure symbolic quality, and EFF and STAB characterize search behavior.

Formal evaluation uses 50 tasks and three random seeds

$$\mathcal{S} = \{520, 521, 522\}.$$

All axes except STAB are computed per task and seed and averaged across 150 runs. STAB is computed per task across the three seeds and averaged across 50 tasks. Unless stated otherwise, $\mathbb{E}[\cdot]$ denotes the empirical average associated with a metric. Metrics use the $[0, 1]$ scale and the 0 to 100 display scale.

E.1 NUMERICAL QUALITY MAPPING

ID and OOD use the same bounded transformation of NMSE. EFF and the numerical component of STAB reuse the mapping. For a valid NMSE value x, the clipped log error is

$$r(x) = \mathrm{clip}(\log_{10}(\max(x, \epsilon)), \ell_{\min}, \ell_{\max}) .$$

Numerical quality is

$$\phi(x) = \frac{\ell_{\max} - r(x)}{\ell_{\max} - \ell_{\min}}.$$

The bounds are $\ell_{\min} = -12$ and $\ell_{\max} = 2$, with $\epsilon = 10^{-12}$. The mapping satisfies $\phi(x) \in [0, 1]$. Lower error yields higher quality. Invalid or unevaluable outputs receive quality zero.

For a final expression

$$q^{\mathrm{ID}} = \phi(\mathrm{NMSE}^{\mathrm{ID}}), \qquad q^{\mathrm{OOD}} = \phi(\mathrm{NMSE}^{\mathrm{OOD}}).$$

E.2 NUMERICAL QUALITY

ID and OOD measure final numerical quality on the fixed evaluation splits

$$\boxed{\mathrm{Score}^{\mathrm{ID}} = \mathbb{E}[q^{\mathrm{ID}}], \qquad \mathrm{Score}^{\mathrm{OOD}} = \mathbb{E}[q^{\mathrm{OOD}}]}.$$

ID to OOD degradation is reported separately for diagnostic analysis

$$\log_{10} \frac{\mathrm{NMSE}^{\mathrm{OOD}} + \epsilon}{\mathrm{NMSE}^{\mathrm{ID}} + \epsilon}.$$

The OOD score excludes the degradation measure.

E.3 OPUS ASSISTED SYMBOLIC POSTPROCESSING

Opus 5 assists symbolic postprocessing following search completion. Each request contains the selected final expression, fixed variable mappings, and coefficients. Domain and operator assumptions are also supplied. Ground truth expressions are processed separately and remain fixed across algorithms. Model assisted processing remains external to search and candidate selection.

Simplification and validation. The model returns a structured JSON record with outcome `simplified`, `unchanged`, or `unable`. An optional expression and brief justification are also returned. The prompt requires preservation of mathematical meaning and excludes coefficient refitting, variable renaming, and additional domain assumptions.

Responses must satisfy a fixed schema. Candidate expressions pass syntax and variable checks. Symbolic comparisons and numerical probes provide diagnostic evidence. An `unable` outcome retains the original expression and an explicit fallback flag.

Ordinary expressions use normalized expression trees implemented with SymPy (Meurer et al., 2017). gplearn retains protected operators as typed prefix nodes and uses its native evaluator near guard boundaries. Protected operators remain distinct from ordinary division, logarithms, and square roots.

Adjudication. Separate requests compare each processed prediction with its reference for SYM. Seed pairs from the same algorithm, task, and condition are compared for the structural component of STAB. Ground truth is excluded from the seed pair comparison.

Structural judgments distinguish mathematical equivalence, agreement after abstraction of numerical constants, and different or undetermined structure. MIN and partial SYM use deterministic expression tree statistics. A positive model judgment provides evaluation evidence and does not constitute formal mathematical proof.

Freezing and reuse. Processed expressions and judgments are frozen before metric aggregation. Artifacts retain original and processed expressions, input and response hashes, and model identifiers. Prompt and schema versions are also retained. Cached judgments are reused only after matching frozen expression dependencies and run provenance. Audit corrections remain linked to original responses. Metric aggregation requires no additional model calls.

E.4 SYMBOLIC QUALITY

SYM. SYM measures agreement between the recovered expression and the underlying symbolic relation. Ground truth and predicted expressions use the same simplification procedure

$$f^{\mathrm{ref}} = \mathcal{S}(f^{\mathrm{gt}}), \qquad \widetilde{f} = \mathcal{S}(\widehat{f}).$$

Let d_{eq} denote the evaluator's equivalence judgment. The equivalence indicator is

$$Eq = \mathbf{1}\left[d_{\mathrm{eq}} = \texttt{equivalent}\right].$$

For predictions lacking a positive equivalence judgment, partial recovery uses expression structure and symbolic content. Tree similarity is

$$S_{\mathrm{tree}} = 1 - \mathrm{NED}\left(\widetilde{f}, f^{\mathrm{ref}}\right),$$

where NED denotes normalized expression tree distance.

Variable and operator recovery are

$$F_v = \mathrm{F1}_{\mathrm{variable}}, \qquad F_o = \mathrm{F1}_{\mathrm{operator}},$$

where both scores compare the corresponding symbolic sets of $\widetilde{f}$ and f^{ref}.

For an invalid or unparsable expression

$$Eq = S_{\mathrm{tree}} = F_v = F_o = 0.$$

The symbolic score for one task and seed is

$$\boxed{m^{\mathrm{SYM}} = \begin{cases} 1, & Eq = 1, \\ 0.5\left(S_{\mathrm{tree}} F_v F_o\right)^{1/3}, & Eq = 0. \end{cases}}$$

The final SYM score is

$$\boxed{\mathrm{Score}^{\mathrm{SYM}} = \mathbb{E}[m^{\mathrm{SYM}}]}.$$

MIN. MIN measures expression complexity relative to the simplified reference used by SYM. Complexity is the number of nodes in the expression tree

$$C^{\mathrm{ref}} = C(f^{\mathrm{ref}}), \qquad C^{\mathrm{pred}} = C(\widetilde{f}).$$

The minimality score for one task and seed is

$$\boxed{m^{\mathrm{MIN}} = \min\left(1, \frac{C^{\mathrm{ref}}}{C^{\mathrm{pred}}}\right)}.$$

An invalid or unparsable final expression receives

$$m^{\mathrm{MIN}} = 0.$$

The final MIN score is

$$\boxed{\mathrm{Score}^{\mathrm{MIN}} = \mathbb{E}[m^{\mathrm{MIN}}]}.$$

A larger MIN score indicates lower expression complexity relative to the reference.

E.5 Search Behavior

EFF. EFF measures search progress relative to the best numerical quality reached in each run. One numerical evaluation is recorded per minute. Symbolic judgments remain external to EFF.

For

$$t \in \{1, \ldots, T\}, \qquad T = 180,$$

the evaluator measures the ID and OOD quality of the expression selected by the algorithm's native objective or model selection rule. Test results remain external to candidate selection. Numerical quality at minute t is

$$q(t) = \frac{q^{\mathrm{ID}}(t) + q^{\mathrm{OOD}}(t)}{2},$$

where

$$q^{\mathrm{ID}}(t) = \phi\left(\mathrm{NMSE}^{\mathrm{ID}}(t)\right), \qquad q^{\mathrm{OOD}}(t) = \phi\left(\mathrm{NMSE}^{\mathrm{OOD}}(t)\right).$$

The maximum observed quality is

$$q^* = \max_{1 \le t \le T} q(t).$$

For a complete auditable trajectory, the run score is

$$\boxed{m^{\mathrm{EFF}} = \frac{1}{T} \sum_{t=1}^{T} \frac{q(t)}{q^*}}.$$

A run with $q^* = 0$ receives

$$m^{\mathrm{EFF}} = 0.$$

The final EFF score is

$$\boxed{\mathrm{Score}^{\mathrm{EFF}} = \mathbb{E}[m^{\mathrm{EFF}}]}.$$

The trajectory $q(t)$ may decrease after expression updates. The value q^* is computed at run end and used only for normalization. Unavailable checkpoints follow the trajectory completion rule in Appendix D.3.

STAB. STAB measures consistency across repeated random seed runs on the same task. For $K = |\mathcal{S}|$ runs, consistency is evaluated across $\binom{K}{2}$ seed pairs. The experiment uses $K = 3$. STAB combines numerical consistency, output validity, and structural consistency.

Numerical consistency. For seeds i and j

$$\delta_{ij}^{\mathrm{num}} = \frac{\left|q_i^{\mathrm{ID}} - q_j^{\mathrm{ID}}\right| + \left|q_i^{\mathrm{OOD}} - q_j^{\mathrm{OOD}}\right|}{2}.$$

Numerical consistency for one task is

$$N = 1 - \frac{1}{\binom{|\mathcal{S}|}{2}} \sum_{i<j} \delta_{ij}^{\mathrm{num}}.$$

Output validity. The valid run fraction for one task is

$$V = \frac{\#\{\text{valid seeds}\}}{|\mathcal{S}|}.$$

Structural consistency. The evaluator judges every scheduled pair of frozen final expressions. A valid pair receives a positive judgment for mathematical equivalence or matching structure after abstraction of numerical constants. The indicator is

$$I_{ij}^{\text{struct}} = \begin{cases} 1, & \text{valid pair with a positive symbolic judgment,} \\ 0, & \text{otherwise.} \end{cases}$$

Invalid pairs and judgments labeled `undetermined` or `non_applicable` receive zero. Structural consistency is

$$C = \frac{1}{\binom{K}{2}} \sum_{i<j} I_{ij}^{\text{struct}}.$$

The stability score for one task is

$$\boxed{m^{\text{STAB}} = (NVC)^{1/3}}.$$

The final STAB score is

$$\boxed{\text{Score}^{\text{STAB}} = \mathbb{E}[m^{\text{STAB}}]}.$$

The geometric mean requires all three components to remain high.

Adjudication limitations. An audit found inconsistent pairwise symbolic judgments for 3 clean, 2 one percent noise, and 3 five percent noise algorithm task groups. The cases limit interpretation of the structural component of STAB.

E.6 Bootstrap Confidence Intervals

Confidence intervals use 1000 task bootstrap replicates. Each replicate samples 50 tasks from *Core50* with replacement and retains the three recorded runs for each sampled task. The same task draws are used across algorithms, axes, and evaluation conditions.

For the five axes defined per run, each replicate averages the sampled run scores. STAB averages the sampled task scores. The 2.5th and 97.5th percentiles define the 95% interval on the 0 to 100 display scale.

Evaluation outputs and symbolic judgments remain fixed during resampling. The intervals quantify task resampling variability within the empirical *Core50* sample and are not interpreted as significance tests.

E.7 Supplementary Training Noise Robustness

Training noise robustness is reported separately from the six formal evaluation axes. Two supplementary conditions are evaluated

$$\sigma \in \{0.01, 0.05\}.$$

Training labels are perturbed according to

$$y' = y + \sigma \,\text{std}(y)\epsilon, \qquad \epsilon \sim \mathcal{N}(0, 1).$$

The values of σ set the Gaussian noise standard deviation to 1% and 5% of the training target standard deviation. ID and OOD test sets remain clean.

Clean and noisy numerical quality is reported on both evaluation splits. Performance degradation is reported for each noise condition.

F SENSITIVITY ANALYSIS

The analysis examines two construction decisions in *SymbolicArena*: Probe4 selection and *Core50* construction. Each decision is re-evaluated across controlled perturbations.

F.1 PROBE4 SELECTION STABILITY AND REPLACEMENT STRUCTURE

All 12 calibration algorithms remain eligible for probe selection. The observed selection mass concentrates on a small competitive set, motivating an audit of panel stability and replacement structure. We examine the default panel

$$\mathcal{A}_4 = \{\mathrm{DSO}, \mathrm{PyOperon}, \mathrm{iMCTS}, \mathrm{uDSR}\}$$

across variation in the calibration sample and selection rule, followed by direct member removal tests. All analyses reuse the 2,400 Calib200 records from Appendix C.2 and require no additional SR runs.

Calibration resampling. We perform 1,000 family stratified task bootstrap replicates. Each replicate samples task blocks with replacement and preserves the original source family counts. The 12 algorithm records associated with each sampled task move as one block. Probe selection is repeated at the default setting $\alpha = 0.5$, $\tau = 100$, and $m = 3$, with all tied optima retained. The default Probe4 is the unique optimum in 63.4% of replicates and belongs to the optimal set in 73.4%. Its tie averaged selection mass is 68.3%, and its median competition rank is 1 with an interquartile range of 1–2.

Table 8 shows a concentrated competitive set. uDSR and PyOperon receive nearly all bootstrap inclusion mass. DSO and iMCTS remain frequent members, and QLattice is the principal alternative. gplearn receives limited mass. The other candidates receive none. The pattern emerges from the complete 12 algorithm selection procedure and is not imposed by prior filtering.

Table 8: Calibration usability and bootstrap inclusion of the 12 eligible probe candidates. Inclusion frequency averages tied optima uniformly.

Algorithm	Probe4	H	Inclusion (%)	Any optimum (%)
uDSR	Yes	0.9875	100.0	100.0
PyOperon	Yes	0.9600	98.8	99.3
DSO	Yes	0.9700	88.1	91.5
iMCTS	Yes	0.9675	81.4	85.4
QLattice		0.9600	30.5	35.0
gplearn		0.9300	1.2	1.9
PySR		0.8750	0.0	0.0
TPSR		0.8625	0.0	0.0
DrSR		0.8225	0.0	0.0
LLM-SR		0.8175	0.0	0.0
E2ESR		0.7775	0.0	0.0
RAG-SR		0.6500	0.0	0.0

Selection rule perturbation. We vary α from 0.30 to 0.70 in increments of 0.05 and evaluate $\tau \in \{10, 30, 100, 300, 1000\}$ with $m = 3$ fixed. Across 45 settings, Probe4 is the unique optimum in 31 and belongs to the optimal set in 33. Its competition rank remains within the top three for every tested setting. The result indicates stable competition across moderate changes in the usability definition.

Member removal. Each Probe4 member is removed in turn, and the complete selection procedure is rerun on the remaining 11 algorithms. QLattice replaces DSO, iMCTS, or uDSR, and gplearn replaces PyOperon. Every fallback panel preserves three paradigms. Table 9 reports the resulting replacement panels and usability losses. The largest usability loss is 0.007500, corresponding to 0.75 percentage points on the $[0, 1]$ usability scale. The result shows that the selection procedure admits structured fallback choices with limited loss in calibration usability.

Table 9: Probe4 replacement analysis. U denotes panel usability, and ΔU measures the loss from the default Probe4 score $U = 0.971250$.

Removed	Replacement	Best U	ΔU	Paradigms
iMCTS	QLattice	0.969375	0.001875	3
DSO	QLattice	0.968750	0.002500	3
uDSR	QLattice	0.964375	0.006875	3
PyOperon	gplearn	0.963750	0.007500	3

Paradigm coverage. The minimum paradigm requirement isolates the role of search diversity in probe selection. At $m = 2$, the default panel shares the optimum with a second panel at the same best usability of 0.971250. Requiring three paradigms removes the tie and preserves the same best usability. At $m = 4$, the default panel becomes infeasible, and the best feasible panel changes to

$$\{\mathrm{PyOperon}, \mathrm{iMCTS}, \mathrm{TPSR}, \mathrm{uDSR}\},$$

with usability 0.944375. The three paradigm setting resolves the ambiguity at $m = 2$ with no reduction in maximum usability.

Overall, all 12 candidates enter the selection procedure, yet the evidence consistently concentrates on a small competitive set. Probe4 remains dominant or near dominant across calibration resampling and rule perturbations. Member removal produces structured fallback panels with small usability loss and preserved paradigm coverage. The analysis supports the stability of the selection process and exposes its principal alternatives.

F.2 *Core50* CONSTRUCTION

The *Core50* selection objective is

$$S^* = \arg \max_{S \subset \mathcal{D}_{664},\ |S|=50} \left[w_C \,\mathrm{Coverage}(S) + w_I \,\mathrm{MeanInfo}(S) + w_B \,\mathrm{Balance}(S) \right],$$

where Coverage measures task coverage in the structural and probe response spaces, MeanInfo measures task-level discriminative power across algorithms and stability, and Balance controls distributional shifts in difficulty and failure modes. The default weights are $\boldsymbol{w}_0 = (0.45, 0.35, 0.20)$.

Single-parameter weight sensitivity. Each weight w_j is adjusted to a given value v, and the other two weights are renormalized to sum to one. The examined ranges are $w_C \in [0.40, 0.50]$, $w_I \in [0.30, 0.40]$, and $w_B \in [0.15, 0.25]$.

For each setting, 30 independent random restarts are performed, yielding 450 selection results in total. The full-ranking rate is the frequency with which the complete ordering $\mathrm{uDSR} > \mathrm{iMCTS} > \mathrm{DSO} > \mathrm{PyOperon}$ is preserved.

Table 10: **Single-parameter weight perturbation results.** Jaccard similarity and overlapping tasks are measured against the default-weight selection under 30 random restarts per setting. Within-group Jaccard similarity compares the 30 restarts of one setting.

Perturbed weight	Weight range	Median Jaccard vs. default	Overlapping tasks	Median within-group Jaccard	Full-ranking rate
Coverage	$[0.40, 0.50]$	0.724	42	0.667	100%
MeanInfo	$[0.30, 0.40]$	0.710	41.5	0.667	100%
Balance	$[0.15, 0.25]$	0.710	41.5	0.667	100%

Table 10 shows that weight perturbations mainly affect tasks near the selection boundary. The median Jaccard similarity between the reselected 50 task sets and the default set is approximately 0.710, corresponding to 41 to 42 overlapping tasks. The median within-group Jaccard similarity across the 30 restarts under the same weight setting is 0.667, indicating that approximately 40 high-quality tasks are selected consistently, and the remaining positions are occupied by interchangeable marginal tasks. These results demonstrate that the *Core50* objective is robust to moderate perturbations of the default weights.

Joint random weight analysis. To examine joint variations in multiple weights, the sampling region is

$$\mathcal{W}_{\text{box}} = \left\{ \boldsymbol{w} \;\middle|\; \begin{array}{l} w_C \in [0.40, 0.50], \\ w_I \in [0.30, 0.40], \\ w_B = 1 - w_C - w_I \in [0.15, 0.25] \end{array} \right\}.$$

5,000 weight vectors are sampled uniformly from $\mathcal{W}_{\text{box}}$. For each sampled vector, all candidate 50 task sets are rescored, and the highest-scoring set is retained. Table 11 shows that four distinct optimal sets are obtained across the 5,000 samples, with the default-weight optimal set accounting for 40.24% of the selections.

Table 11: **Distribution of optimal sets under joint weight sampling.** Selections report the number of the 5,000 sampled weight vectors that produce each optimal set. S0 denotes the default-weight selection.

Optimal set	Selections	Share	Jaccard vs. default
S0 (*Core50*)	2012	40.24%	1.000
S1	1803	36.06%	0.724
S2	1173	23.46%	0.786
S3	12	0.24%	0.754

Table 12 reports a mean Jaccard similarity to the default set of 0.850, corresponding to approximately 46 overlapping tasks on average. All 5,000 selected sets preserve the ranking of the four probes on *Full Task Set*.

Task inclusion frequencies further reveal a stable core: 37 tasks are selected in all 5,000 runs, 38 appear in at least 90% of the runs, and 42 appear in at least 75%.

Table 12: **Joint random weight analysis results.** Statistics summarize the 5,000 sampled weight vectors.

Statistic	Result
Mean Jaccard vs. default	0.850
Minimum overlapping tasks	42 / 50
Full-ranking rate	100%
Tasks selected in all runs	37
Tasks in $\geq 90\%$ of runs	38
Tasks in $\geq 75\%$ of runs	42

Under joint variations of all three weights, the optimal sets continue to overlap substantially with *Core50*, and the resulting algorithm ranking remains unchanged.

G Additional Experimental Results

The appendix provides supporting evidence for Multi Axis Evaluation in *SymbolicArena*. The analyses examine redundancy among evaluation axes and quantify uncertainty from task sampling. Additional results assess robustness to training noise and examine a representative gap between numerical approximation and symbolic recovery.

G.1 Correlation Among Evaluation Axes

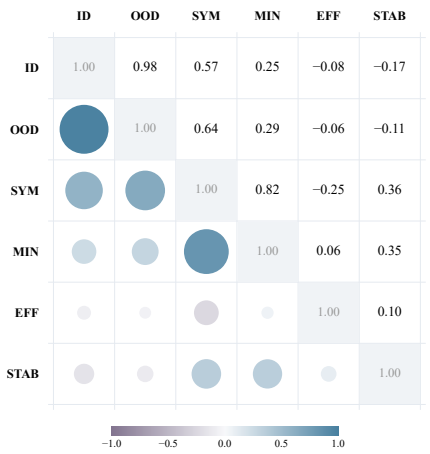


Figure 8: Spearman correlations among the six evaluation axes across 15 algorithms on clean *Core50*.

Figure 8 shows distinct association patterns among the six axes. ID and OOD are strongly correlated ($\rho = 0.98$), indicating similar numerical rankings across the two evaluation distributions. SYM and MIN also show a strong association ($\rho = 0.82$). Numerical quality has moderate correlation with SYM ($\rho = 0.57$ to 0.64) and weaker correlation with MIN ($\rho = 0.25$ to 0.29).

EFF shows little association with final numerical quality, with correlations of -0.08 for ID and -0.06 for OOD. STAB also has weak association with numerical quality. The observed structure provides empirical support for retaining separate dimensions for numerical quality, symbolic quality, and search behavior.

G.2 Uncertainty in Six Axis Scores

Figure 9 quantifies task sampling uncertainty for all six axes. Interval widths differ across algorithms and metrics, indicating different sensitivity to the sampled tasks. Small gaps between point estimates should consequently be interpreted alongside their bootstrap intervals. The analysis complements the six axis profiles reported in the main text.

G.3 Robustness to Training Noise

Training label perturbations extend the clean evaluation to corrupted training signals.

Table 13 shows substantial variation in numerical robustness across algorithms. High clean OOD performance does not consistently correspond to small degradation under noise. iMCTS retains the highest OOD score across all evaluated conditions, whereas DSO shows a smaller absolute drop from its clean score. FePySR remains competitive at both noise levels. The pattern distinguishes clean numerical quality from robustness to training corruption.

Tables 14 and 15 report the complete six axis profiles at $\sigma = 0.01$ and $\sigma = 0.05$. The tables extend the numerical robustness analysis to symbolic quality and search behavior, using the same metric definitions as the clean evaluation. Clean profiles are reported in Table 2.

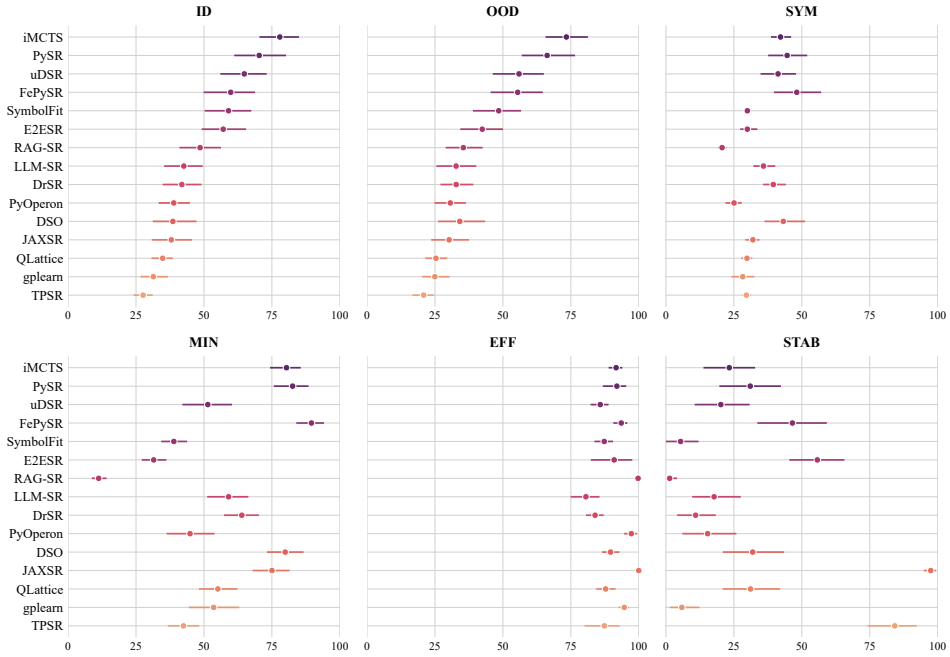


Figure 9: **Mean scores and 95% task bootstrap intervals on clean *Core50*.** Points denote mean scores and bars denote percentile intervals from 1,000 resamples of 50 task blocks. Each sampled block retains the three recorded runs. Scores use the 0 to 100 display scale.

Table 13: Numerical robustness of 15 symbolic regression methods under training label noise. ID and OOD scores are reported under clean, 1%, and 5% noise conditions. $\Delta_{5\%}$ denotes the score change from clean to 5% noise.

		ID				OOD			
Paradigm	**Algorithm**	**Clean ↑**	**1% ↑**	**5% ↑**	**$\Delta_{5\%}$ ↑**	**Clean ↑**	**1% ↑**	**5% ↑**	**$\Delta_{5\%}$ ↑**
Structured Search	**iMCTS**	**77.91**	**55.08**	**46.69**	-31.22	**73.35**	**49.40**	**40.73**	−32.62
	QLattice	34.79	32.36	29.18	−5.61	25.37	23.09	20.27	−5.10
	JAXSR	37.99	32.62	31.00	−6.99	30.21	25.91	24.56	−5.65
Evolutionary Search	**PySR**	70.34	48.42	41.05	−29.29	66.28	41.50	35.53	−30.75
	PyOperon	38.89	36.29	32.97	−5.92	30.64	27.85	24.42	−6.22
	gplearn	31.37	30.14	29.93	−1.44	24.96	23.97	23.84	**−1.12**
	SymbolFit	59.04	44.38	38.42	−20.62	48.46	33.87	28.26	−20.20
Neural Policy Search	**uDSR**	64.84	44.61	38.87	−25.97	55.94	32.86	27.44	−28.50
	DSO	38.56	37.27	37.61	**−0.95**	34.14	32.95	32.84	−1.30
Hybrid Search	**FePySR**	59.84	46.69	41.69	−18.15	55.45	42.77	36.92	−18.53
	RAG-SR	48.58	39.95	36.33	−12.25	35.45	27.26	24.35	−11.10
Transformer Methods	**E2ESR**	57.09	43.64	36.31	−20.78	42.40	31.36	25.97	−16.43
	TPSR	27.60	26.83	24.48	−3.12	20.84	19.81	18.05	−2.79
LLM Assisted Search	**DrSR**	41.86	37.67	31.76	−10.10	32.82	28.55	23.44	−9.38
	LLM-SR	42.59	35.12	33.36	−9.23	32.78	26.08	23.00	−9.78

Display scale: 0–100. $\Delta_{5\%} = \text{Score}_{5\%} - \text{Score}_{clean}$.

Table 14: Six-axis profiles of 15 symbolic regression methods on *Core50* at 1% training-label noise ($\sigma = 0.01$). Methods are grouped by their primary search paradigm. Higher values denote better performance on each axis.

		Numerical		Symbolic		Search	
Paradigm	**Algorithm**	**ID ↑**	**OOD ↑**	**SYM ↑**	**MIN ↑**	**EFF ↑**	**STAB ↑**
Structured Search	**iMCTS**	**55.08**	**49.40**	**37.22**	72.88	89.34	12.82
	QLattice	32.36	23.09	29.59	54.60	90.87	38.32
	JAXSR	32.62	25.91	30.88	76.16	**99.95**	**91.33**
Evolutionary Search	**PySR**	**48.42**	41.50	30.53	60.26	86.08	6.02
	PyOperon	36.29	27.85	25.14	43.76	97.30	15.92
	gplearn	30.14	23.97	28.03	53.30	94.24	7.77
	SymbolFit	44.38	33.87	29.48	36.93	84.49	2.74
Neural Policy Search	**uDSR**	44.61	32.86	27.79	32.64	83.80	4.14
	DSO	37.27	32.95	**43.32**	**81.81**	90.21	32.25
Hybrid Search	**FePySR**	46.69	**42.77**	35.82	**83.67**	89.49	**43.75**
	RAG-SR	39.95	27.26	19.35	9.64	**99.35**	0.00
Transformer Methods	**E2ESR**	43.64	31.36	27.65	28.21	89.44	4.09
	TPSR	26.83	19.81	29.36	41.17	87.37	8.13
LLM Assisted Search	**DrSR**	37.67	28.55	34.65	64.07	84.74	11.73
	LLM-SR	35.12	26.08	30.85	50.86	81.69	8.23

Display scale 0–100. Best (1st) Second-best (2nd)

Table 15: Six-axis profiles of 15 symbolic regression methods on *Core50* at 5% training-label noise ($\sigma = 0.05$). Methods are grouped by their primary search paradigm. Higher values denote better performance on each axis.

		Numerical		Symbolic		Search	
Paradigm	**Algorithm**	**ID ↑**	**OOD ↑**	**SYM ↑**	**MIN ↑**	**EFF ↑**	**STAB ↑**
Structured Search	**iMCTS**	**46.69**	**40.73**	36.43	66.26	86.11	5.24
	QLattice	29.18	20.27	29.19	54.99	85.95	29.66
	JAXSR	31.00	24.56	31.29	77.47	**99.95**	**91.95**
Evolutionary Search	**PySR**	41.05	35.53	30.66	59.16	85.28	6.73
	PyOperon	32.97	24.42	23.60	39.83	97.27	15.33
	gplearn	29.93	23.84	27.30	53.52	93.24	3.90
	SymbolFit	38.42	28.26	29.23	36.38	83.24	2.77
Neural Policy Search	**uDSR**	38.87	27.44	27.88	32.53	82.47	7.05
	DSO	37.61	32.84	**42.83**	**80.35**	90.29	29.73
Hybrid Search	**FePySR**	**41.69**	**36.92**	**36.44**	**84.89**	88.83	**40.14**
	RAG-SR	36.33	24.35	18.79	8.66	**99.47**	0.00
Transformer Methods	**E2ESR**	36.31	25.97	27.30	26.51	89.18	1.99
	TPSR	24.48	18.05	28.62	42.08	88.48	2.75
LLM Assisted Search	**DrSR**	31.76	23.44	33.36	61.38	83.43	12.44
	LLM-SR	33.36	23.00	31.26	51.32	80.96	4.05

Display scale 0–100. Best (1st) Second-best (2nd)

G.4 Numerical Approximation and Symbolic Recovery

A clean `Nguyen-9` run from PySR illustrates the distinction between numerical approximation and symbolic recovery. For seed 520, the reference expression and processed prediction are

$$f^{\mathrm{gt}}(x_1, x_2) = \sin(x_1) + \sin(x_2^2),$$
$$\widetilde{f}(x_1, x_2) = f^{\mathrm{gt}}(x_1, x_2) + 1.1685188 \times 10^{-8}.$$

The recorded NMSE values are 4.54×10^{-15} on the fixed ID split and 8.28×10^{-17} on the fixed OOD split. Despite the near exact numerical fit, the evaluator assigns `not_equivalent`, producing $m^{\mathrm{SYM}} = 0.4886$.

The prediction preserves the principal functional structure and differs from the reference by a small additive residual. Exact symbolic equivalence is lost despite negligible numerical error. The example provides a concrete motivation for evaluating numerical quality and symbolic recovery separately in *SymbolicArena*.

H EVALUATED ALGORITHMS AND EXPERIMENTAL SETTINGS

The study evaluates 15 symbolic regression algorithms across six primary search paradigms. FePySR (Yu et al., 2026), SymbolFit (Tsoi et al., 2025), and JAXSR (Kitchin, 2024) are excluded from benchmark construction and serve as validation algorithms. All methods are evaluated on the frozen *Core50* benchmark.

Algorithm panel. Table 16 characterizes the algorithmic coverage of the evaluation by grouping the 15 methods according to their primary search paradigms and search mechanisms.

Table 16: **Symbolic regression algorithms evaluated in *SymbolicArena*.** Bold marks validation algorithms.

Paradigm	Method	Description
Structured Search	iMCTS (Huang et al., 2025)	Monte Carlo tree search for guided expression exploration.
	QLattice (Broløs et al., 2021)	Probabilistic graph search for compact symbolic models.
	JAXSR (Kitchin, 2024)	Structured symbolic model construction with optimization in JAX.
Neural Policy Search	uDSR (Landajuela et al., 2022)	Neural symbolic search with evolutionary refinement.
	DSO (Petersen et al., 2021)	Policy search using risk seeking optimization.
Evolutionary Search	PySR (Cranmer, 2023)	Population search with complexity aware model selection.
	PyOperon (Burlacu et al., 2020)	Evolutionary tree search with constant optimization.
	gplearn (Stephens, 2016)	Genetic programming with protected operators and parsimony control.
	SymbolFit (Tsoi et al., 2025)	Hybrid symbolic search with numerical parameter optimization.
Transformer Methods	E2ESR (Kamienny et al., 2022)	Pretrained neural generation of expressions from numerical data.
	TPSR (Shojaee et al., 2023)	Pretrained expression generation guided by planning.
Hybrid Search	**FePySR** (Yu et al., 2026)	PySR based symbolic regression with feature enhancement.
	RAG-SR (Zhang et al., 2025)	Retrieval guided symbolic search with evolutionary optimization.
LLM Assisted Search	DrSR (Wang et al., 2025)	LLM symbolic regression with iterative candidate refinement.
	LLM-SR (Shojaee et al., 2025a)	LLM equation proposals with numerical parameter refinement.

Evaluation configurations. Table 17 records the configurations used in the formal *Core50* evaluation. The evaluated uDSR configuration uses the fixed *uDSR trunk variant*, which combines a DSO style controller, the LINEAR poly token, and GP meld.

Table 17: **Algorithm specific configurations for the formal *Core50* evaluation.** Gray rows mark validation algorithms.

Algorithm	Search setting	Operators and basis	Fixed settings
gplearn	Population 1000.	add, sub, mul, div, sqrt, log, sin, cos	1 job, tournament 20, initial depth 2 to 6, MAE metric, parsimony 0.001, crossover 0.9, subtree, hoist, and point mutation 0.01.
PyOperon	Population 500, pool 500.	add, sub, mul, div, aq, exp, log, sin, cos, tanh, sqrt, square, constants, variables	4 threads, max length 50, max depth 10, tournament 5, keep best reinsertion, LM optimizer, local search probability 1.0.
PySR	8 populations of size 64, 500 cycles per iteration.	Binary $+, -, \times, /$. Unary square, cube, exp, log, sin, cos.	Serial execution with 1 process, max size 30, max depth 10, parsimony 0.001, deterministic mode, precision 32, best model selection.

Table 17: **Algorithm specific configurations for the formal *Core50* evaluation (continued).**

Algorithm	Search setting	Operators and basis	Fixed settings
QLattice	BIC criterion.	QLattice/Feyn internal symbolic basis	4 threads, regression mode, BIC model selection, 4 significant digits.
DSO	Batch size 1000.	add, sub, mul, div, sin, cos, exp, log	4 batch cores, reward inv NRMSE, $\epsilon = 0.05$, learning rate 5×10^{-4}, entropy weight 0.03, expression length 4 to 64.
uDSR	Batch size 1000, GP meld population 100 for 20 generations.	add, sub, mul, div, sin, cos, exp, log, sqrt, 1.0, constants, poly token	1 batch core, reward inv NRMSE, $\epsilon = 0.05$, learning rate 5×10^{-4}, entropy weight 0.03, expression length 4 to 100, GP meld and linear poly enabled.
iMCTS	$K = 500$, max depth 6.	$+, -, \times, /$, sin, cos, exp, log, real constant token	Exploration constant 4.0, $\gamma = 0.5$, GP rate 0.2, mutation 0.1, exploration 0.2, max constants 10, LN Nelder–Mead optimization.
E2ESR	200 input points, 10 bags, refine 10 trees, stop refinement after 1.	Pretrained E2E decoder vocabulary with arithmetic, inverse and powers, log and exp, trigonometric and inverse trigonometric functions.	Single Torch thread, rescaling and execution forced to CPU, top k feature selection and relabeling, missing variables filled with zero.
TPSR	200 input points, 10 bags, refine 10 trees, beam size 10, width 3, rollout 3, horizon 200.	Pretrained E2E decoder vocabulary. Unsupported variables projected to zero.	4 CPU threads and 1 interop thread, E2E backbone, sampling beam, one beam, $\lambda = 0.1$, no value training, reward sample limit 2048, prefix cache disabled, CPU mode.
LLM-SR	4 samples per iteration, max 10 fitted parameters.	Python expression skeletons from an LLM. Constants are fitted.	Semantic prompt injection and prompt variables enabled, sample persistence disabled, DeepInfra Llama-3.1-8B-Instruct-Turbo, max 1024 tokens, temperature 0.6, top p 0.3, top k 30.
DrSR	4 samples per iteration, 400000 proposals, max 10 fitted parameters.	Python expression skeletons from an LLM. Constants are fitted.	Same prompt contract as LLM-SR, one sampler and one evaluator, sample persistence disabled, DeepInfra Llama-3.1-8B-Instruct-Turbo, max 1024 tokens, temperature 0.6, top p 0.3, top k 30.
RAG-SR	Population 200, gene number 10.	Add, Sub, Mul, AQ, Sqrt, AbsLog, Abs, Square, RSin, RCos, Max, Min, Neg	4 CPU threads, automatic lexicase selection, crossover 0.9, mutation 0.1, max height 10, RidgeCV, MinMax normalization, R^2 score, external archive enabled, `number_of_invokes`=0.
SymbolFit	Max size and complexity 25.	Binary $+, -, \times, /$. Unary sin, cos, exp, log.	Serial deterministic search with one process, model selection based on accuracy, input rescaling, mean target scaling, max stderr 20, fixed target uncertainty 1, no uncertainty fitting.
FePySR	8 experiments, FMN 30 epochs with batch size 64, PySR uses 4 populations of 40 and 100 cycles per iteration.	Binary $+, -, \times, /$. Unary sin, cos, exp, log.	4 workers, FMN learning rate 0.1, 10 extracted features, 6 PySR features, max size 20, max depth 8.
JAXSR	Max 5 terms, CV with 5 folds, greedy forward selection.	Constant and linear bases, polynomials of degree 3, and interactions of order 2. No transcendental or ratio terms.	BIC model selection, no regularization, constant and linear bases enabled, polynomial and interaction bases enabled.

I LLM USAGE AND INFORMATION BOUNDARIES

SymbolicArena uses language models in two separate roles. Llama participates in equation search for LLM-SR and DrSR. Opus 5 performs symbolic postprocessing after search completion. Search models cannot access ground truth expressions or ID and OOD evaluation targets. Appendix D defines the execution boundary. The appendix specifies model roles, prompt settings, and information access for language model components.

Table 18 summarizes the model roles and their relation to search.

Table 18: **Language model roles in *SymbolicArena*.** Opus 5 operates only during evaluator side postprocessing.

Model	Stage	Primary role	Affects search
Opus 5	Postprocessing	Symbolic processing and judgment	No
Llama	LLM-SR search	Equation structure generation	Yes
Llama	DrSR search	Generation and diagnostic feedback	Yes

I.1 LLAMA IN LLM-SR AND DRSR

The formal *Core50* evaluation uses Llama 3.1 8B Instruct (Grattafiori et al., 2024), served as `meta-llama/Meta-Llama-3.1-8B-Instruct-Turbo` by DeepInfra[1] for LLM-SR and DrSR. Generation uses `temperature=0.6`, `top_p=0.3`, and `max_tokens=1024`. Model weights remain fixed. Candidate selection and coefficient fitting use training objectives. ID and OOD splits remain reserved for evaluation.

LLM-SR. LLM-SR iteratively requests equation structures from Llama. Prompts contain task context and variable descriptions. Historical candidates provide additional search context. Multiple islands maintain separate candidate histories. Each generation prompt includes up to two functions selected by training score. Each iteration requests four new structures.

For each structure, BFGS optimizes coefficients in `params` using training MSE. All coefficients are initialized to one. Candidates with higher training scores can enter the history pool.

The shared generation instruction is:

```
You are a helpful assistant tasked with discovering mathematical
function structures for scientific systems.
Complete the 'equation' function below, considering the physical
meaning and relationships of inputs.
```

The LLM-SR user message follows:

```
[fixed generation instruction]

[task background]
[variable descriptions]

[code preamble]

def equation_v0(...):
    ...

def equation_v1(...):
    """Improved version of `equation_v0`."""
```

[1] https://deepinfra.com/meta-llama/Meta-Llama-3.1-8B-Instruct-Turbo

Historical functions come from candidates recorded during the current run. The final empty `equation_vN` function requests the next structure. Candidate evaluation occurs outside the Llama request.

DrSR. DrSR uses the same equation generation mechanism and candidate pool design. Additional Llama calls provide experience feedback and residual analysis.

Experience feedback. Candidate outcomes are categorized using training score changes or execution failures. Selected outcomes are converted into compact guidance for later generation.

Residual analysis. Improved candidates can trigger a residual analysis request containing training data, the current expression, and prediction residuals. The numerical matrix is

$$[X,\ y,\ y - y_{\text{pred}}],$$

and displayed values use three decimal places. The generated analysis can enter later generation prompts.

DrSR uses the task header:

```
Find the mathematical function skeleton that represents
{dependent}, given data on {independent}.
```

DrSR adds a single line output requirement to the shared generation instruction:

```
Write the final formula as a single-line return statement only.
Example: return params[0] + params[1] * x0 + params[2] * x1
```

A generation message follows:

```
[task header]

[optional residual analysis]

[optional previous experience]

[fixed generation instruction]

Variables:
{variables_block}

Background:
{background_text}

[candidate functions]

def equation_vN(...):
    ...
```

Candidate coefficients are evaluated locally. The formal DrSR evaluator uses five $\text{Uniform}(-1, 1)$ initializations for BFGS and retains the solution with the lowest training MSE. Optimization uses `maxiter=200`, `gtol=1e-10`, and `eps=1e-12`.

I.2 Opus 5 in Symbolic Evaluation

Claude Opus 5 (Anthropic, 2026) operates only on completed runs. Requests use `claude-opus-5`, contain no conversation history, and process final expressions only. The configuration uses adaptive thinking, extra high effort, and `max_tokens=65536`. Minute level trajectories used by EFF never invoke Opus 5.

Detailed simplification and adjudication procedures appear in Appendix E.3. Opus 5 supports expression simplification, equivalence assessment, and pairwise structural assessment. Ground truth

and predicted expressions are processed independently before comparison. Protected operator semantics and declared domain assumptions remain fixed.

Opus 5 does not directly compute evaluation scores. Deterministic code aggregates frozen expressions and stored judgments. Table 19 records its contribution to each axis.

Table 19: **Role of Opus 5 in Multi Axis Evaluation.**

Axis	Role of Opus 5
ID	None
OOD	None
SYM	Simplification and equivalence judgment
MIN	Simplification before deterministic tree size computation
EFF	None
STAB	Pairwise structural judgment of final expressions

I.3 Information Isolation and Leakage Control

Search models cannot access ground truth expressions or ID and OOD evaluation targets. Formal runs map variable names to $x_0, x_1, \ldots$ and the target to y. Declared semantic descriptions can be supplied as algorithm inputs. Such descriptions remain separate from ground truth formulas and evaluation targets.

For tasks lacking semantic metadata, the generic background is:

```
Find the mathematical function skeleton that represents
{target_desc}, given data on {feature_text}.
```

For SRSD tasks containing distractor variables, semantic roles are exposed as an unordered multiset. The association between roles and individual variables remains hidden:

```
There are {n_total} candidate variables in an unknown order.
The unordered semantic-role multiset is: {role_text}.
The mapping from semantic roles to variable names is intentionally
hidden; do not assume which x_i corresponds to which semantic role.
```

The masking prevents direct identification of the numerical column associated with a semantic role.

Benchmark construction. The initial LLM-SR probe used for *Calibration Set* mining disables semantic prompt injection and uses a generic symbolic regression background. The final *four probes* consists of DSO, PyOperon, iMCTS, and uDSR. All four construction probes operate without language models. Final *Full Task Set* responses used for *Core50* selection therefore contain no LLM generated probe responses.

Evaluator separation. Opus 5 receives ground truth only during symbolic processing of completed runs. Its output cannot alter candidate generation or recorded search trajectories. Processed expressions and judgments are frozen before metric aggregation. Model identifiers and prompt hashes are retained for auditing.