AUTOSR: AUTOMATIC SYMBOLIC REGRESSION BY SEARCHING RESEARCH STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Automatic Symbolic Regression (AutoSR), a fully automated system that instantiates Research-Space Symbolic Regression by searching persistent scientific investigations rather than isolated equations. Finite, noisy data often yield numerically competitive expressions that imply very different behavior outside the observed regime, making numerical fit and syntactic complexity insufficient measures of scientific credibility. Recent LLM-based and agentic methods add tools, diagnostics, and memory, but they mainly use this history to propose the next equation; the search still compares candidate equations rather than the scientific record behind them, such as motivations, failed attempts, and criticism. AutoSR preserves this record in a **Research State**, coupling each candidate equation with the reasoning, computational evidence, and independent review developed along its branch. Proposer–reviewer agents develop these states under progressive-widening Monte Carlo tree search (PW-MCTS), which allocates computation across competing investigations, while the accumulated research record is ultimately synthesized into a final report that explains the leading relation and the basis for its selection. Across nine selected challenges from two benchmark suites, AutoSR recovers algebraically equivalent relations in every case, including six structurally diverse LSR-Transform problems and three cp3-bench problems that none of the twelve systems in the original cp3-bench evaluation recovered. Overall, AutoSR extends symbolic regression from equation-level search toward automated scientific investigation, allowing scientific knowledge and accumulated evidence to shape both what is explored and how the resulting equation is justified.

1 INTRODUCTION

Scientific discovery from data seeks a relation that accounts for the phenomenon and remains useful beyond the observed sample rather than merely a formula that reproduces the observations. Finite data underdetermine the generating function: infinitely many distinct functions can interpolate the same finite dataset (Bongard & Lipson, 2007; Cornelio et al., 2023). In the presence of noise, structurally different expressions can attain nearly indistinguishable errors over the observed data while differing sharply in scientific implications such as extrapolation, dimensions, monotonicity, limiting behavior, and mechanism (La Cava et al., 2021; Matsubara et al., 2024). Conventional symbolic regression (SR) makes this underdetermined problem of equation discovery from finite observations tractable by searching for explicit mathematical expressions and ranking them by numerical fit and generic measures of syntactic complexity (Koza, 1992; Schmidt & Lipson, 2009; Cranmer, 2023). These criteria favor concise candidate laws, but they do not by themselves establish scientific credibility. Scientists use *scientific priors*—including general knowledge of the physical world, domain theory, and problem-specific requirements—to judge which hypotheses are plausible. Some requirements, such as dimensional consistency, monotonicity, or a boundary value, can be compiled into specialized constraints or search spaces (Kronberger et al., 2022; Tenachi et al., 2023; Cornelio et al., 2023), whereas other forms of scientific knowledge, including causal expectations, ontology, and mechanistic plausibility, are difficult to express in a fixed grammar or scalar objective. Moreover, scientific knowledge guides not only which equation to accept but also what to investigate next. For example, a suspicious asymptote may call for a numerical limit test, whereas an unexpected functional form may motivate comparison with a theoretical model. Recent LLM-based and agentic SR methods use contextual knowledge, computational tools, diagnostics, and reusable

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.

memories to improve equation proposals (Shojaee et al., 2025a; Xia et al., 2026; Li & Pan, 2026; Pang et al., 2026; Zhao et al., 2026). Their advances motivate a complementary question: how can the complete motivation, experiments, failed alternatives, diagnostics, and criticism associated with each candidate become persistent objects of global search rather than transient context for producing the next equation? To address this gap, we introduce **Automatic Symbolic Regression (AutoSR)**, a fully automated agentic system that searches over persistent research records rather than isolated expressions. AutoSR applies scientific priors during hypothesis proposal, testing, and criticism, and synthesizes the investigation into an auditable report of the leading equation.

AutoSR targets the discovery of a scientifically credible mathematical relation under a given problem specification and computation budget. The problem specification combines the observed data with the scientific priors used to judge which relations are plausible, including any explicit requirements that an acceptable relation should satisfy. To preserve the broader investigation, AutoSR represents each research attempt as a **Research State**, a persistent search unit comprising a candidate equation and the broader research record of the attempt that produced it. Research States form a branching search space in which a new state can either initiate an investigation or extend an existing one by inheriting its accumulated findings and reviews, while sibling branches pursue competing explanations independently. With the Research State as the search unit, AutoSR must repeatedly determine which investigation to initiate or extend next from the accumulated scientific evidence and remaining computation budget. We formulate this sequential decision problem as **Research-State Search**. AutoSR performs Research-State Search using Monte Carlo tree search (MCTS) (Coulom, 2007; Kocsis & Szepesvári, 2006; Browne et al., 2012), which allocates the computation budget across the branching search space by balancing the extension of promising investigations with the exploration of alternative branches. Searching over complete scientific investigations rather than isolated expressions defines the broader paradigm of **Research-Space Symbolic Regression**, which AutoSR instantiates through Research-State Search. This design provides five practical benefits:

- **Knowledge-guided investigation.** Scientific priors supplied through the problem specification guide not only which candidate equations are accepted, but also how hypotheses are proposed, computationally tested, and criticized throughout the search.
- **Accumulated research memory.** Each Research State preserves its findings, failed attempts, and reviewer feedback, allowing descendant states along the same branch to build on accumulated evidence instead of restarting from only an equation and its score.
- **Independent alternatives.** By organizing Research States into separate branches, AutoSR allows alternative explanations to develop independently without being prematurely reduced to a single expression.
- **Fully autonomous investigation.** By combining Research-State Search with automated hypothesis generation, computational testing, and independent review, AutoSR can repeatedly select and carry out the next investigation. Once the problem specification and computation budget are provided, the system can sustain this complete process without continuous human management.
- **Auditable scientific results.** Because the investigation behind each candidate is preserved, the final report can connect the leading equation to its supporting evidence, limitations, research history, and credible alternatives.

In this way, AutoSR allows accumulated scientific evidence to determine what the system investigates next, rather than allowing numerical fit alone to control the search.

AutoSR is designed around the premise that scientific equation discovery can become more reliable when a system operationalizes the functional cycle of scientific inquiry used by human scientists. Scientists do not judge a candidate equation by its numerical fit alone; they use scientific priors to propose hypotheses, design experiments or computations to test their implications, criticize the resulting evidence, and allow what they learn to determine the next investigation. For example, a credible black-body radiation law had to account for the measured spectrum while remaining consistent with displacement and total-radiation laws and the limiting forms then available (Wien, 1893; 1896; Stefan, 1879; Boltzmann, 1884; Rayleigh, 1900). The investigation did not end with a fitted formula: the search for a physical account led Planck to introduce energy elements proportional to frequency (Planck, 1901), an idea that subsequently supported Einstein's interpretation of light quanta (Einstein, 1905). Scientific inquiry can succeed where numerical fit alone cannot because each additional

empirical or theoretical test can eliminate expressions that remain numerically competitive but violate scientific evidence or requirements, while the accumulated findings guide which hypothesis should be investigated next. AutoSR operationalizes this causal process at a computational scale: scientific priors constrain plausible equations, Research States preserve branch-specific research records, independent review challenges unsupported conclusions, and Research-State Search directs further investigation. In this functional sense, AutoSR mimics the exploratory cycle of human scientists without attempting to reproduce the reasoning of any individual scientist. Through this cycle, AutoSR is designed to distinguish scientifically credible equations from expressions that are merely accurate over the observed data.

We evaluate AutoSR through exact equation recovery on selected challenge problems from two benchmark suites. On all nine selected problems, AutoSR recovers relations algebraically equivalent to the ground truth. These include three cp3-bench equations that were not recovered by any of twelve published SR systems (Thing & Koksbang, 2025) and six structurally diverse LSR-Transform equations (Shojaee et al., 2025b), all recovered using the same core search procedure and agent roles. The present evidence establishes feasibility on these selected challenge cases; repeated-run reliability, component ablations, scheduling efficiency, and domain validation remain subjects for future evaluation.

Our contributions are threefold. First, we introduce Research-Space Symbolic Regression and, to our knowledge, the first formulation of Research-State Search as the sequential decision problem underlying this paradigm. Second, we develop AutoSR as a fully automated implementation that combines persistent Research States, independent scientific review, autonomous computational investigation, and MCTS-based search. Third, we demonstrate exact recovery on nine selected challenge equations spanning two benchmark suites. The remainder of the paper provides the related work review, defines the AutoSR method, presents the selected benchmark evaluation, and discusses the implications, limitations, and future evaluation of treating symbolic regression as automated scientific inquiry.

2 Related Work

Expression search and scientific priors. Symbolic regression searches for explicit expressions through genetic programming, sparse or compressed-sensing model construction, probabilistic inference, physics-informed decomposition, and specialized optimizers (Koza, 1992; Brunton et al., 2016; Ouyang et al., 2018; Jin et al., 2019; Udrescu & Tegmark, 2020; Cranmer, 2023; Jiang et al., 2023). SRBench, SRSD, and LLM-SRBench distinguish predictive accuracy from structural recovery and emphasize the role of evaluation domains and scientific sampling (La Cava et al., 2021; Matsubara et al., 2024; Shojaee et al., 2025b). Prior-informed methods incorporate shape restrictions, dimensions, structural hypotheses, or background axioms (Kronberger et al., 2022; Tenachi et al., 2023; Bendinelli et al., 2023; Cornelio et al., 2023; Cory-Wright et al., 2024). AutoSR admits these formal requirements alongside scientific context expressed in natural language, with their interpretation recorded as part of an investigation.

Learned proposals and tree search. Neural proposal models learn distributions over expressions from synthetic data or search experience (Petersen et al., 2021; Biggio et al., 2021; Kamienny et al., 2022; Landajuela et al., 2022). MCTS already guides expression construction, mutation, and decoding in SR (Sun et al., 2023; Kamienny et al., 2023; Shojaee et al., 2023; Xu et al., 2024; Yu et al., 2025). AutoSR uses established MCTS machinery; its distinction is that an expansion produces a persistent proposer–experiment–review record whose ancestry and evaluation guide subsequent investigations.

LLM-guided and agentic SR. LLM-SR combines scientific context with programmatic equation proposals and iterative search (Shojaee et al., 2025a); other systems couple language models to coefficient optimization, simulation, conceptual libraries, or retrieval (Merler et al., 2024; Ma et al., 2024; Grayeli et al., 2024; Zhang et al., 2025). Recent agents already use tools, diagnostics, experience buffers, reflective memory, and multiple roles (Xia et al., 2026; Li & Pan, 2026; Pang et al., 2026; Zhao et al., 2026). AutoSR therefore does not claim the first use of these components. It specifies how a complete investigation becomes a search node, how its record is inherited along one branch, how a separate review is attached before reward backup, and how the resulting tree

supports final reporting. Searching richer work products also appears in AIDE and AI Scientist-v2 (Jiang et al., 2025; Yamada et al., 2025); AutoSR specializes this organization to scientific equation discovery. Appendix A provides the expanded comparison.

3 METHOD

3.1 PROBLEM FORMULATION AND SYSTEM OVERVIEW

AutoSR receives a problem specification $\mathcal{P}$ and a computation budget B. The specification defines the scientific question, the observed data, the meanings of the variables, the available supporting evidence, the evaluation criterion, and any requirements that an acceptable relation should satisfy. Given $(\mathcal{P}, B)$, the system returns not only a leading relation $\hat{f}$, but a final report $\mathcal{R}_{\text{final}}$ that connects this relation to its evidence, limitations, research history, and credible alternatives:

$$\text{AutoSR}(\mathcal{P}, B) \longrightarrow (\hat{f}, \mathcal{R}_{\text{final}}). \tag{1}$$

The task is therefore broader than minimizing a fixed loss over an expression space. AutoSR must use the available scientific information to produce, test, criticize, and compare candidate relations while deciding how to allocate the remaining budget.

The concrete interface to $\mathcal{P}$ consists of `problem.md`, data, and supporting files. The supporting material may include background knowledge, figures, documents, existing code, and explicit scientific requirements. Requirements such as dimensional consistency, monotonicity, convexity, asymptotic behavior, special values, and known theoretical limits can often be checked directly (Kronberger et al., 2022; Tenachi et al., 2023; Cornelio et al., 2023). Other priors, including causal expectations, ontology, and mechanistic plausibility, may remain expressed in scientific language rather than in a fixed grammar or penalty. AutoSR supplies the complete specification to both its proposal and review roles, allowing formal requirements and broader scientific knowledge to influence hypothesis generation, computational testing, and criticism.

Starting from a root that contains $\mathcal{P}$, AutoSR builds a tree of Research States. Each tree expansion executes an automated proposer–reviewer cycle, and the search controller selects which branch to initiate or extend next. Progressive-widening Monte Carlo tree search (PW-MCTS) (Browne et al., 2012; Couëtoux et al., 2011) balances new lines of inquiry against continuations of promising branches, while a pending-aware UCT rule (Kocsis & Szepesvári, 2006; Liu et al., 2020) permits multiple long-running expansions to proceed asynchronously. A complementary loop applies established symbolic-regression packages, and the final synthesis compares the strongest candidates from both processes. Figure 1 summarizes this workflow.

3.2 RESEARCH STATES AND BRANCH-SPECIFIC MEMORY

A Research State is the persistent record produced by one research attempt. For a non-root node i, write

$$\mathcal{S}_i = (f_i, m_i, \mathcal{A}_i, \rho_i, s_i; p_i), \tag{2}$$

where f_i is the candidate relation, m_i is its scientific motivation and reasoning, $\mathcal{A}_i$ contains the computational artifacts and findings, ρ_i is the review record, s_i is the reviewer score, and p_i identifies the parent Research State. The artifacts may include code, execution results, fitted parameters, figures, residual analyses, diagnostics, and unsuccessful tests. The scalar s_i supports search control, whereas the full record $(m_i, \mathcal{A}_i, \rho_i)$ preserves the evidence needed to interpret that score.

The central artifact in each state is `ansatz.md`. It serves as both summary and index: it states the current hypothesis, condenses the main argument and results, records the review, and points to the supporting files. The local contents of $\mathcal{S}_i$ record attempt i, while the ordered path from the root to i forms the memory of the corresponding line of inquiry. When a proposer extends state i, it receives the `ansatz.md` files on this ancestral path in order. It does not receive summaries from sibling branches. A child of the root therefore begins an independent investigation, whereas a deeper child can build on the useful ideas, failed trials, and criticism accumulated by its ancestors. This separation allows competing explanations to develop independently without discarding the research history within any one branch.

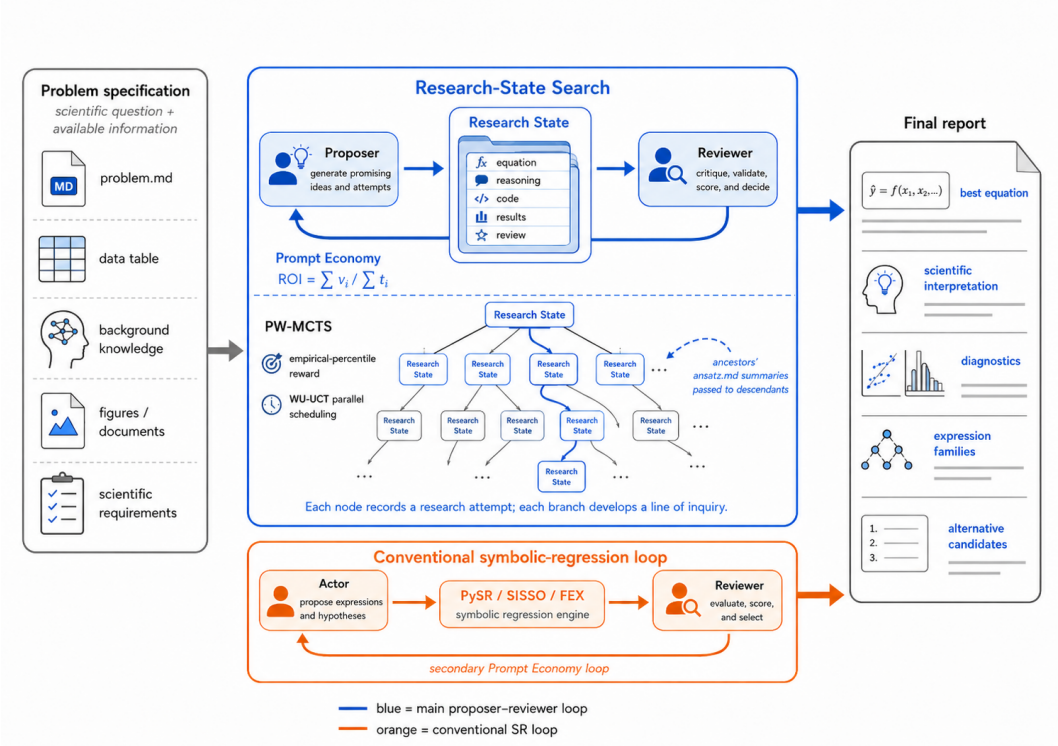


Figure 1: AutoSR uses a proposer–reviewer loop to build a tree of Research States with progressive-widening MCTS. The problem specification supplies data, scientific context, and explicit requirements to the investigation. A secondary actor–reviewer loop advances conventional symbolic regression. The Prompt Economy annotation uses the AutoSR-specific reuse metric defined in Equation 3. The final report combines the strongest candidates, their supporting evidence, and the resulting search landscape.

3.3 Automated Investigation and Independent Review

The proposer receives $\mathcal{P}$ together with the branch memory selected by the search controller. It develops an ansatz, writes and executes code, fits parameters, studies plots and numerical behavior, and revises the candidate in response to the resulting evidence. The proposer may change transformations, objectives, optimizers, or computational methods when the current formulation is unstable or scientifically misleading, and it may call symbolic-regression packages or other scientific tools when useful. These decisions are not prescribed as a fixed pipeline: they are part of the investigation recorded in $(m_i, \mathcal{A}_i)$.

After the proposal is complete, a separate reviewer role reads the problem specification, `ansatz.md`, and the supporting artifacts. It reruns key computations, checks the stated evaluation criterion, examines residuals, and searches for leakage, hidden parameters, violated requirements, or unsupported conclusions. Held-out data and hidden evaluation instructions can be supplied through a reviewer-only channel. When feedback from this channel influences descendants or branch selection, these data function as validation data; final test evaluation requires data untouched by that adaptive process. The reviewer records its diagnosis ρ_i and score s_i in `ansatz.md`. The reviewer is instructed to treat a mandatory scientific requirement as an acceptance condition rather than a tradeable improvement in numerical fit. The scalar score is used to allocate search effort; it does not replace the reviewer's evidence or the scientific comparison preserved in the final report. Appending this review completes $\mathcal{S}_i$, after which its descendants can inherit both the proposal and its criticism.

Producing a Research State can require repeated reasoning, coding, execution, and revision. AutoSR draws on Prompt Economy, which treats the maintenance of role prompts and handoff protocols as a bounded engineering cost and favors reusing a stable prompt surface across many useful invocations

(Sun et al., 2026b;a). We operationalize this principle for AutoSR with a time-based, system-specific reuse metric. Let t_i be the time spent writing or revising prompt i, and let v_i be the number of times that prompt is executed:

$$\mathrm{ROI}_{\mathrm{AutoSR}} = \frac{\sum_i v_i}{\sum_i t_i}. \tag{3}$$

This AutoSR-specific quantity adapts the Prompt Economy principle rather than reproducing the ROI formula in the cited work; it is a system-design criterion, not the scientific objective in Equation 1. Reusing the same proposer and reviewer roles at every expansion increases the number of autonomous investigations without requiring a new prompt for each candidate or domain.

3.4 Research-State Search

Research-State Search is the sequential decision problem of choosing which investigation to initiate or extend under the remaining budget. At scheduling step t, let $\mathcal{T}_t$ denote the tree of created Research States, including completed states and expansions still in flight. Each completed proposer–reviewer cycle supplies one new node and one observed reward; in this sense, a complete research attempt plays the role of an MCTS evaluation (Coulom, 2007; Browne et al., 2012). AutoSR allocates these evaluations with progressive-widening MCTS (PW-MCTS; Couëtoux et al., 2011).

For node i, let C_i be its set of created children, N_i the number of completed evaluations whose selected paths pass through i, and O_i the corresponding number of evaluations that have been dispatched but have not yet returned. The controller may create another child of i when

$$|C_i| < k_{\mathrm{pw}} \max(N_i + O_i, 1)^{\beta}. \tag{4}$$

Here $k_{\mathrm{pw}} > 0$ controls the overall width and $0 < \beta < 1$ makes the allowed branching factor grow sublinearly with the number of completed and pending evaluations. If the condition holds, the proposer creates a new child from the branch memory of i; otherwise, the controller continues through an existing child according to the tree policy below. Progressive widening therefore admits more independent or descendant investigations as evidence accumulates without allowing early branching to consume the entire budget.

The reviewer score can depend on the evaluation criterion, scientific requirements, and the supporting analysis, and its raw scale may differ across problems. Let $\mathcal{V}_t$ be the set of Research States completed by scheduling step t, with larger s_i indicating a stronger review. AutoSR converts each score to its empirical percentile within the current completed set:

$$r_i(t) = 10 \times \frac{1}{|\mathcal{V}_t|} \sum_{\ell \in \mathcal{V}_t} \mathbf{1}[s_\ell \le s_i]. \tag{5}$$

Percentile rewards lie in $(0, 10]$ and are interpreted at the current scheduling step because their reference set grows as reviews return. This common rank scale prevents the magnitude or units of a task-specific evaluator from changing the exploration pressure. For a child j, let $Q_j(t)$ be the mean percentile reward of the completed Research States in the subtree rooted at j. The tree policy ranks only children whose own evaluation has completed, so $N_j \ge 1$ and $Q_j(t)$ is always defined.

Because a proposer–reviewer cycle may run for minutes or hours, scheduling evaluations only after earlier calls finish would leave substantial parallel capacity unused. AutoSR therefore adapts Watch the Unobserved in UCT (WU-UCT; Liu et al., 2020) to Research-State Search. Among existing children, the controller selects according to

$$U_t(i, j) = Q_j(t) + c\sqrt{\frac{\log \max(N_i + O_i, 1)}{N_j + O_j}}. \tag{6}$$

The in-flight counts allow dispatched work to affect both selection and progressive widening before its reviewer score becomes observable. They discourage redundant simultaneous expansion of the same branch while retaining the exploitation term supplied by completed reviews.

One asynchronous search iteration consists of four operations. First, the controller traverses the tree from the root, applying Equation 4 to decide whether to create a child and Equation 6 when it must

continue through existing children. Second, before dispatching the selected proposer–reviewer cycle, it increments O_i along the selected path so that later scheduling decisions observe the pending work. Third, the proposer and reviewer produce the new Research State without blocking other dispatched evaluations. Finally, when the review returns, the controller decrements the path's in-flight counts, increments its completed counts, adds the new score to $\mathcal{V}_t$, and updates the percentile rewards and subtree means used by subsequent selections. This cycle continues until budget B is exhausted. The controller then stops dispatching new work, resolves the remaining in-flight evaluations, and collects the completed Research States for final synthesis.

3.5 Conventional Symbolic-Regression Loop

A secondary Prompt Economy loop applies conventional symbolic regression to the same problem specification. This loop complements the flexible investigations in the Research-State tree with mature expression-search engines and their specialized optimizers. Its actor configures and runs methods such as PySR (Cranmer, 2023), SISSO (Ouyang et al., 2018), and FEX (Jiang et al., 2023), then studies their equations, residuals, and failure modes. Its reviewer examines the search space and returned candidates, identifies overlooked transformations or settings, and directs the next package run. One package execution is therefore an experiment within a sustained actor–reviewer process rather than the complete conventional-SR contribution. The loop passes its strongest equations, diagnostics, and reviews to the same final synthesis used for Research-State candidates.

3.6 Final Selection and Report Synthesis

After both search processes end, AutoSR ranks completed candidates by reviewer score while retaining the corresponding requirement checks and review records. A candidate that violates a requirement identified by the problem specification cannot be presented as an acceptable scientific relation solely because it attains a strong numerical fit. The leading eligible candidate becomes $\hat{f}$ in Equation 1, but ranking determines only which result the report presents first; it does not discard competitive alternatives or their evidence.

The first part of $\mathcal{R}_{\text{final}}$ explains the leading result through its formula, measured performance, scientific interpretation, diagnostics, and limitations. The second reconstructs the Research-State search landscape: it presents a report-ready tree, groups states into expression families, summarizes the evidence associated with each family, and traces clear evolution paths through the parent relation in Equation 2. An appendix presents the next four candidates as standalone alternatives. When several relations remain competitive, the report compares their empirical and scientific implications instead of forcing an unsupported single conclusion. The equation remains the central result, while the preserved artifacts and reviewer records make the investigation behind that result available for audit.

4 Experiments

We report nine selected AutoSR recovery trajectories and compare them with newly run PySR, FEX, LLM-SR, and Direct controls on the same tasks. The comparison describes these recorded outputs; it does not estimate benchmark-wide AutoSR accuracy or isolate the contribution of Research-State Search.

4.1 Evaluation Scope and Protocol

Tasks. The three cp3-bench tasks, C3g, C5f, and C6c, were not recovered by any of the twelve systems in the original benchmark evaluation (Thing & Koksbang, 2025). They involve dependent inputs, cusp/core density profiles, and an oscillatory gravitational-wave relation. The six LSR-Transform tasks come from LLM-SRBench (Shojaee et al., 2025b) and span radicals, a trigonometric relation solved for an unusual variable, fractional powers, paired exponentials, rational structure, and logarithmic inversion. These selected tasks are not a representative sample of either suite. The controls use the source 80/20 split with random state 42 for cp3-bench and the source split at stored precision for LSR-Transform; both test splits are in-distribution.

AutoSR runs. Each AutoSR trajectory uses the search procedure and role prompts of Section 3. The proposer uses `gpt-5.6-sol` with reasoning effort `xhigh`, and the reviewer uses Claude Opus 5 with effort `xhigh`. For LSR-Transform, the proposer receives only the training split, and the reviewer computes the reported NRMSE on the held-out split, which is withheld from the proposer. For cp3-bench, the problem specification supplies the full dataset and scores full-data MSE; the recovered C3g expression has no fitted constants. No run has Internet access, and the ground-truth equation is used only in the post-run equivalence check. We report one recovered trajectory per task. In all nine tasks, the reported equation came from a Research State in the proposer–reviewer tree. The total number of attempted runs and the total compute have not yet been reconstructed from the run records. We therefore treat the AutoSR values as reported results, not as test errors matched to the controls.

Controls. PySR 1.5.10 (Julia 1.11.5) uses maximum expression size 60, 15 populations of 33, and 160 iterations, and selects its output on the training data (Cranmer, 2023). FEX uses depth 3, 1,000 search steps with batch size 10, and refinement of the top 10 candidates for 20,000 steps (Jiang et al., 2023). LLM-SR keeps the upstream feedback buffer and selector, with an adapted model and task interface (Shojaee et al., 2025a). Direct sends the same prompt repeatedly without history, fits coefficients on the training data, and keeps the program with the best training error. Both LLM controls use `gpt-5.6-sol` with reasoning effort `max` and 40 completed responses per run; LLM-SR obtains them as ten rounds of four responses. Direct is therefore a best-of-40 control, not a single response or a search over Research States. Control prompts use the original benchmark descriptions and exclude AutoSR-specific background, requirements, and outputs.

Runs and reporting. Each control run selects a program and fits its coefficients on the training split only; we then fix both and evaluate them once on the test split. FEX has five runs per task, and PySR has five except on II.35.18 and II.36.38, where the measurement harness refused two runs and one run, respectively. The LLM controls have two or three completed runs per task. Table 1 reports the median test error over runs, and Table S2 gives the minimum and maximum. The runs were executed at different times, so we do not treat them as paired samples across methods. Every control output gives a finite prediction for every test point.

Metrics and structural checks. We report MSE for cp3-bench and NRMSE for LSR-Transform, where NRMSE is the square root of the MSE divided by the variance of the test labels; for controls, it is computed per run before aggregation. A low error does not by itself show that the correct structure was recovered, so we check the structure of control outputs separately. A *coefficient check* tests whether the returned program, with its fitted coefficients, equals the target. A *structure check* tests whether some finite real coefficients make the returned program equal to the target on the whole evaluation domain, with all operations defined; this domain is the training input box restricted to the points where the target is defined. A check that cannot be completed is reported as unresolved, not as a failure. The reported AutoSR recoveries rely on a separate algebraic-equivalence assessment, so we do not pool structural success rates across AutoSR and the controls.

4.2 Selected Benchmark Results

On C3g, both LLM controls reach a test MSE of about 8.62×10^{-19} and pass the structure check; AutoSR reports 9.0×10^{-19}. The non-recovery result of the original benchmark therefore does not hold for the current LLM controls. The largest differences in Table 1 occur on C5f and C6c, where AutoSR reports MSEs of 6.2×10^{-32} and 6.1×10^{-30}, and the best control medians are PySR's 7.22×10^{-5} and 8.76×10^{-2}. These two contrasts are subject to the evaluation caveat above. The structure checks leave the LLM outputs on C5f unresolved, and C6c has no complete target expression for the check; thus neither task shows that every control failed structurally.

On LSR-Transform, AutoSR reports NRMSEs between 7.2×10^{-8} and 3.6×10^{-7}. Both LLM controls return at least one output that passes the structure check on every task, and PySR passes it on II.35.18 and II.36.38. Several methods reach errors of similar small size, and differences in the last digits do not rank their scientific quality. On II.36.38, both Direct outputs pass the structure check, yet their NRMSEs are about 1.34×10^{-6} and 2.99×10^{-6}.

Table 1: Numerical results on nine selected tasks. The cp3-bench tasks use MSE, and the LSR-Transform tasks use NRMSE; lower is better. Control entries are median test errors over runs (ranges in Table S2). AutoSR entries are the values reported by one recovered trajectory per task; their evaluation rows are not verified to match the control test splits.

Task	PySR	FEX	LLM-SR	Direct	AutoSR
cp3-bench: MSE					
`C3g`	1.85×10^{-5}	3.19×10^{-5}	8.62×10^{-19}	8.62×10^{-19}	9.0×10^{-19}
`C5f`	7.22×10^{-5}	1.05×10^{-3}	1.32×10^{-2}	4.20×10^{-2}	6.2×10^{-32}
`C6c`	8.76×10^{-2}	3.57×10^{-1}	5.96×10^{-1}	8.53×10^{-1}	6.1×10^{-30}
LSR-Transform: NRMSE					
`I.32.17_4_3`	7.34×10^{-2}	1.51×10^{-1}	1.80×10^{-7}	1.80×10^{-7}	1.8×10^{-7}
`I.29.16_0_0`	6.05×10^{-2}	1.67×10^{-1}	1.65×10^{-7}	1.65×10^{-7}	1.2×10^{-7}
`II.6.15a_2_0`	2.36×10^{-2}	1.11×10^{-1}	1.02×10^{-7}	1.02×10^{-7}	1.0×10^{-7}
`II.35.18_0_0`	3.94×10^{-7}	1.08×10^{-1}	4.60×10^{-7}	4.60×10^{-7}	3.6×10^{-7}
`II.36.38_3_0`	8.00×10^{-8}	1.61×10^{-1}	8.88×10^{-8}	2.16×10^{-6}	8.0×10^{-8}
`III.4.33_3_0`	1.23×10^{-4}	4.34×10^{-2}	7.22×10^{-8}	7.22×10^{-8}	7.2×10^{-8}

4.3 Run-to-Run Variation and Scope of the Evidence

Medians hide variation between runs. On II.35.18, LLM-SR's three test NRMSEs are about 4.60×10^{-7}, 1.59×10^{-2}, and 4.60×10^{-7}. The poor run returns the family $p_0 n\exp(p_1 z)$, whereas the target contains $n[\exp(z)+\exp(-z)]$; no coefficient values make a single exponential equal this sum. This example does not identify the cause of the poor run or estimate AutoSR's repeatability.

The nine AutoSR trajectories show recovery on selected tasks under one workflow. The controls show that simpler procedures also reach several of these structures; they do not isolate the effect of review, branch memory, or tree search. Appendix B lists the successful proposer session times, which exclude total system cost.

5 Discussion and Limitations

AutoSR's methodological contribution is the explicit organization of each investigation as a persistent search state: a candidate is retained with its motivation, computations, failed tests, and review, and that record is inherited along its branch. This design supports the use of accumulated evidence in subsequent investigations and in final reporting. The selected recoveries demonstrate capability under a common workflow, while the strong simple controls delimit what recovery alone establishes about the architecture. The quality of the research records, alternative comparisons, and final reports has not been separately evaluated.

The comparison remains limited by selected tasks, one reported successful AutoSR trajectory per task, and incomplete alignment of AutoSR's original evaluation and resource records. Control repetitions do not measure AutoSR's probability of success or its total cost. Matched-context and matched-budget evaluations would be needed to distinguish the contributions of a single tool-using agent, proposer–reviewer iteration, branch memory, tree search, and the conventional-SR loop. The current Direct control contributes useful context but does not replace these interventions. Scheduler comparisons would also be needed to assess percentile rewards, pending-aware selection, worker utilization, and time to a specified quality.

Exact recovery on established benchmarks does not establish scientific credibility under noise, extrapolation, incomplete theory, or conflicting requirements. Pretraining familiarity cannot be ruled out, and separate proposer/reviewer roles do not guarantee independent judgment (Matsubara et al., 2024; Shojaee et al., 2025b). New or private domain problems, external checks, and domain-expert assessment remain necessary to evaluate the broader scientific-discovery objective. In this setting, fully automated means that the system can continue investigating after a problem specification and budget are supplied; scientists still define the problem and interpret the result.

AI USE STATEMENT

AutoSR itself uses generative AI: language-model agents propose and refine candidate equations, write and execute analysis code, review candidate results, and synthesize reports (Section 3); the LLM-SR and Direct controls also query a language model (Section 4). We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Deaglan J. Bartlett, Harry Desmond, and Pedro G. Ferreira. Priors for Symbolic Regression. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, GECCO '23 Companion, pp. 2402–2411, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3583133.3596327.

Tommaso Bendinelli, Luca Biggio, and Pierre-Alexandre Kamienny. Controllable Neural Symbolic Regression. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2063–2077. PMLR, 2023. URL https://proceedings.mlr.press/v202/bendinelli23a.html.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural Symbolic Regression that Scales. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 936–945. PMLR, 2021. URL https://proceedings.mlr.press/v139/biggio21a.html.

Ludwig Boltzmann. Ableitung des Stefan'schen Gesetzes, betreffend die Abhängigkeit der Wärmestrahlung von der Temperatur aus der electromagnetischen Lichttheorie. *Annalen der Physik*, 258 (6):291–294, 1884. doi:10.1002/andp.18842580616.

Josh Bongard and Hod Lipson. Automated Reverse Engineering of Nonlinear Dynamical Systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007. doi:10.1073/pnas.0609476104.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi:10.1109/TCIAIG.2012.2186810.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi:10.1073/pnas.1517384113.

Cristina Cornelio, Sanjeeb Dash, Vernon Austel, Tyler R. Josephson, Joao Goncalves, Kenneth L. Clarkson, Nimrod Megiddo, Bachir El Khadir, and Lior Horesh. Combining Data and Theory for Derivable Scientific Discovery with AI-Descartes. *Nature Communications*, 14:1777, 2023. doi:10.1038/s41467-023-37236-y.

Ryan Cory-Wright, Cristina Cornelio, Sanjeeb Dash, Bachir El Khadir, and Lior Horesh. Evolving Scientific Discovery by Unifying Data and Background Knowledge with AI Hilbert. *Nature Communications*, 15:5922, 2024. doi:10.1038/s41467-024-50074-w.

Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. Continuous Upper Confidence Trees. In *Learning and Intelligent Optimization*, volume 6683 of *Lecture Notes in Computer Science*, pp. 433–445. Springer, 2011. doi:10.1007/978-3-642-25566-3_32.

Rémi Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In *Computers and Games*, volume 4630 of *Lecture Notes in Computer Science*, pp. 72–83. Springer Berlin Heidelberg, 2007. doi:10.1007/978-3-540-75538-8_7.

Miles Cranmer. Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl. arXiv preprint arXiv:2305.01582, 2023. URL https://arxiv.org/abs/2305.01582.

Mengge Du, Yuntian Chen, Zhongzheng Wang, Longfeng Nie, and Dongxiao Zhang. Large language models for automatic equation discovery of nonlinear dynamics. *Physics of Fluids*, 36(9): 097121, 2024. doi:10.1063/5.0224297.

Albert Einstein. Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. *Annalen der Physik*, 322(6):132–148, 1905. doi:10.1002/andp.19053220607.

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. Curran Associates, Inc., 2024. doi:10.52202/079017-1419. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4ec3ddc465c6d650c9c419fb91f1c00a-Abstract-Conference.html.

Zhengyao Huang, Daniel Huang, Tiannan Xiao, Dina Ma, Zhenyu Ming, Hao Shi, and Yuanhui Wen. Improving Monte Carlo Tree Search for Symbolic Regression. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40819–40856. Curran Associates, Inc., 2025. doi:10.52202/085713-1363. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3a28dd36daed2e3d66a3a143ab8b53cc-Abstract-Conference.html.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-Driven Exploration in the Space of Code, 2025. URL https://arxiv.org/abs/2502.13138.

Zhongyi Jiang, Chunmei Wang, and Haizhao Yang. Finite Expression Methods for Discovering Physical Laws from Data. arXiv preprint arXiv:2305.08342, 2023. URL https://arxiv.org/abs/2305.08342.

Ying Jin, Weilin Fu, Jian Kang, Jiadong Guo, and Jian Guo. Bayesian Symbolic Regression, 2019. URL https://arxiv.org/abs/1910.08892.

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. Curran Associates, Inc., 2022. doi:10.52202/068431-0746. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/42eb37cdbefd7abae0835f4b67548c39-Abstract-Conference.html.

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.

Levente Kocsis and Csaba Szepesvári. Bandit Based Monte-Carlo Planning. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pp. 282–293. Springer, 2006. doi:10.1007/11871842_29.

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

Gabriel Kronberger, Fabrício Olivetti de França, Bogdan Burlacu, Christian Haider, and Michael Kommenda. Shape-Constrained Symbolic Regression—Improving Extrapolation with Prior Knowledge. *Evolutionary Computation*, 30(1):75–98, 2022. doi:10.1162/evco_a_00294.

Jiří Kubalík, Erik Derner, and Robert Babuška. Symbolic Regression Driven by Training Data and Prior Knowledge. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, pp. 958–966, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3377930.3390152.

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 Shing Lee, Jiachen Yang, Ruben Glatt, Claudio P. Santiago, Ignacio Aravena, Terrell 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. Curran Associates, Inc., 2022. doi:10.52202/068431-2463. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/dbca58f35bddc6e4003b2dd80e42f838-Abstract-Conference.html.

Yunlun Li and Sinno Jialin Pan. Robust Equation Structure Learning with Adaptive Refinement. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=z9TKJhLVKj.

Anji Liu, Jianshu Chen, Mingze Yu, Yu Zhai, Xuewen Zhou, and Ji Liu. Watch the Unobserved: A Simple Approach to Parallelizing Monte Carlo Tree Search. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1810.11755.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B. Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. LLM and Simulation as Bilevel Optimizers: A New Paradigm to Advance Physical Scientific Discovery. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33940–33962. PMLR, 2024. URL https://proceedings.mlr.press/v235/ma24m.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.

Trent McConaghy. FFX: Fast, Scalable, Deterministic Symbolic Regression Technology. In *Genetic Programming Theory and Practice IX*, pp. 235–260. Springer, 2011. doi:10.1007/978-1-4614-1770-5_13.

Matteo Merler, Katsiaryna Haitsiukevich, Nicola Dainese, and Pekka Marttinen. In-Context Symbolic Regression: Leveraging Large Language Models for Function Discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 427–444. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.acl-srw.49. URL https://aclanthology.org/2024.acl-srw.49/.

Runhai Ouyang, Stefano Curtarolo, Emre Ahmetcik, Matthias Scheffler, and Luca M. Ghiringhelli. SISSO: A Compressed-Sensing Method for Identifying the Best Low-Dimensional Descriptor in an Immensity of Offered Candidates. *Physical Review Materials*, 2(8):083802, 2018. doi:10.1103/PhysRevMaterials.2.083802.

Xinyu Pang, Zhanke Zhou, Xuan Li, Fangrui Lv, Shanshan Wei, Sen Cui, Bo Han, and Changshui Zhang. Deliberate Evolution: Agentic Reasoning for Sample-Efficient Symbolic Regression with LLMs. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=5JUrQ67tYL.

Brenden K. Petersen, Mikel Landajuela Larma, T. Nathan Mundhenk, Claudio P. Santiago, Soo K. 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.

Max Planck. Ueber das Gesetz der Energieverteilung im Normalspectrum. *Annalen der Physik*, 309 (3):553–563, 1901. doi:10.1002/andp.19013090310.

Lord Rayleigh. LIII. Remarks upon the Law of Complete Radiation. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 49(301):539–540, 1900. doi:10.1080/14786440009463878.

Kai Ruan, Yilong Xu, Ze-Feng Gao, Yang Liu, Yike Guo, Ji-Rong Wen, and Hao Sun. Discovering Physical Laws with Parallel Symbolic Enumeration. *Nature Computational Science*, 6:53–66, 2026. doi:10.1038/s43588-025-00904-8.

Evgeny S. Saveliev, Samuel Holt, Nabeel Seedat, David L. Bentley, Jim Weatherall, and Mihaela van der Schaar. Influence-Guided Symbolic Regression: Scientific Discovery via LLM-Driven Equation Search with Granular Feedback. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ujiUua1gGV.

Michael Schmidt and Hod Lipson. Distilling Free-Form Natural Laws from Experimental Data. *Science*, 324(5923):81–85, 2009. doi:10.1126/science.1165893.

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. Curran Associates, Inc., 2023. doi:10.52202/075280-1990. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8ffb4e3118280a66b192b6f06e0e2596-Abstract-Conference.html.

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 *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=m2nmp8P5in.

Parshin Shojaee, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D. Doan, and Chandan K. Reddy. LLM-SRBench: A New Benchmark for Scientific Equation Discovery with Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55325–55359. PMLR, 2025b. URL https://proceedings.mlr.press/v267/shojaee25a.html.

Zhuo-Yang Song, Zeyu Cai, Shutao Zhang, Jiashen Wei, Jichen Pan, Shi Qiu, Qing-Hong Cao, Tie-Jiun Hou, Xiaohui Liu, Ming-xing Luo, and Hua Xing Zhu. Iterated Agent for Symbolic Regression, 2025. URL https://arxiv.org/abs/2510.08317.

Josef Stefan. Ueber die Beziehung zwischen der Wärmestrahlung und der Temperatur. *Sitzungsberichte der Kaiserlichen Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Classe*, 79:391–428, 1879.

Fangzheng Sun, Yang Liu, Jian-Xun Wang, and Hao Sun. Symbolic Physics Learner: Discovering Governing Equations via Monte Carlo Tree Search. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ZTK3SefE8_Z.

Youran Sun, Xingyu Ren, Chugang Yi, Jiaxuan Guo, Kejia Zhang, Jianda Du, and Haizhao Yang. Agon: An Autonomous Large-Scale Omnidisciplinary Research System Built on Prompt Economy. arXiv preprint arXiv:2606.24177, 2026a. URL https://arxiv.org/abs/2606.24177.

Youran Sun, Xingyu Ren, Kejia Zhang, Xinpeng Liu, and Jiaxuan Guo. PerspectiveGap: A Benchmark for Multi-Agent Orchestration Prompting. arXiv preprint arXiv:2606.08878, 2026b. URL https://arxiv.org/abs/2606.08878.

Wassim Tenachi, Rodrigo Ibata, and Foivos I. Diakogiannis. Deep Symbolic Regression for Physics Guided by Units Constraints: Toward the Automated Discovery of Physical Laws. *The Astrophysical Journal*, 959(2):99, 2023. doi:10.3847/1538-4357/ad014c.

Mattias E. Thing and Sofie M. Koksbang. cp3-bench: A tool for benchmarking symbolic regression algorithms demonstrated with cosmology. *Journal of Cosmology and Astroparticle Physics*, 2025 (01):040, 2025. doi:10.1088/1475-7516/2025/01/040.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A Physics-Inspired Method for Symbolic Regression. *Science Advances*, 6(16):eaay2631, 2020. doi:10.1126/sciadv.aay2631.

Willy Wien. Eine neue Beziehung der Strahlung schwarzer Körper zum zweiten Hauptsatz der Wärmetheorie. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, pp. 55–62, 1893.

Willy Wien. Ueber die Energievertheilung im Emissionsspectrum eines schwarzen Körpers. *Annalen der Physik*, 294(8):662–669, 1896. doi:10.1002/andp.18962940803.

Shijie Xia, Yuhan Sun, and Pengfei Liu. SR-Scientist: Scientific Equation Discovery With Agentic AI. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KBN6oUx5uL.

Ziyu Xiang, Kenna Ashen, Xiaofeng Qian, and Xiaoning Qian. Graph-based Symbolic Regression with Invariance and Constraint Encoding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 6882–6914. Curran Associates, Inc., 2025. doi:10.52202/085713-0237. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0a4e065fb5ee13441c90bb4b4d6072d0-Abstract-Conference.html.

Zikai Xie, Wenmei Li, Man Luo, Jun Jiang, and Linjiang Chen. Language models guide symbolic equation discovery by controlling search, 2026. URL https://arxiv.org/abs/2607.04156.

Yilong Xu, Yang Liu, and Hao Sun. Reinforcement Symbolic Regression Machine. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ef3a55fa15aa5fe39b7a2617b3a5d06e-Abstract-Conference.html.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search, 2025. URL https://arxiv.org/abs/2504.08066.

Zihan Yu, Jingtao Ding, Yong Li, and Depeng Jin. Symbolic Regression via MDLformer-Guided Search: From Minimizing Prediction Error to Minimizing Description Length. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a402493de088886740b5939f666a6e56-Abstract-Conference.html.

Hengzhe Zhang, Qi 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.

Wenxiao Zhao, Dong Liu, Kaiyi Xu, Feng Liu, Zhen Zhao, Fei Ben, Shu Wang, Wenhao Li, Ying Nian Wu, Fenghua Ling, Haobo Li, and Lei Bai. A-SR: Self-Evolving Agentic LLMs for Symbolic Regression via Hierarchical Coordination, 2026. URL https://arxiv.org/abs/2608.04872.

A Additional Related-Work Context

A.1 Conventional and Prior-Informed Symbolic Regression

Symbolic regression jointly searches for the structure and parameters of an explicit mathematical expression. Genetic programming established expression trees, evolutionary variation, and fitness-based selection as a general formulation of this search (Koza, 1992). Early systems for scientific discovery extended expression search with active probing of dynamical systems (Bongard & Lipson, 2007) and searches for invariant relations in experimental data (Schmidt & Lipson, 2009). Subsequent approaches span deterministic model building such as FFX (McConaghy, 2011), sparse identification over a prescribed function library in SINDy (Brunton et al., 2016), compressed-sensing feature construction in SISSO (Ouyang et al., 2018), probabilistic search over expression trees in Bayesian SR (Jin et al., 2019), and physics-inspired decomposition by symmetry, separability, and dimensional analysis in AI Feynman (Udrescu & Tegmark, 2020). Modern systems such as PySR and FEX combine expressive search spaces with specialized optimization (Cranmer, 2023; Jiang et al., 2023), while parallel symbolic enumeration shares subtree computations to evaluate very large collections of expressions efficiently (Ruan et al., 2026).

Across these algorithmic families, candidates are ordinarily selected by predictive error together with parsimony, sparsity, a structural prior, or a related model-selection criterion. SRBench compares contemporary methods on both known-ground-truth and black-box tasks and shows that predictive accuracy, symbolic recovery, and expression complexity expose different aspects of performance (La Cava et al., 2021). SRSD further emphasizes realistic sampling ranges, irrelevant variables, and normalized tree-edit distance for scientific-discovery evaluation (Matsubara et al., 2024), while LLM-SRBench adds transformed and synthetic scientific equations together with safeguards against equation memorization (Shojaee et al., 2025b). These benchmarks make clear that syntactic size is a useful operational proxy, but neither low error nor a short expression alone establishes that a recovered relation has the intended scientific meaning.

Prior knowledge can narrow this ambiguity when it can be formalized. It has been incorporated through data generated from known constraints (Kubalík et al., 2020), bounds and derivative-shape restrictions such as monotonicity or convexity (Kronberger et al., 2022), dimensional consistency enforced during expression construction (Tenachi et al., 2023), user-specified structural hypotheses for neural SR (Bendinelli et al., 2023), and probabilistic priors over functions, structures, and parameters (Bartlett et al., 2023). AI-Descartes evaluates candidate formulas against logical background axioms (Cornelio et al., 2023), and AI-Hilbert jointly searches for polynomial laws consistent with data and background theory while producing formal certificates (Cory-Wright et al., 2024). AutoSR treats these formal constraints as valuable special cases: its problem specification may include them, but also admits scientific knowledge that remains in natural language and must influence hypothesis formation, computational testing, and criticism rather than a single fixed objective.

A.2 Learned and Tree-Search-Guided Symbolic Regression

Learned SR methods replace or augment hand-designed proposal distributions with neural priors. Deep Symbolic Regression trains a recurrent policy with a risk-seeking objective (Petersen et al., 2021), whereas NeSymReS and end-to-end Transformer systems pretrain on synthetic equations and infer expressions from sets of numerical observations (Biggio et al., 2021; Kamienny et al., 2022). A unified deep-SR framework combines recursive simplification, neural-guided search, pretraining, genetic programming, and linear models, illustrating that learned generation and explicit search are complementary rather than mutually exclusive (Landajuela et al., 2022).

MCTS provides a general mechanism for allocating simulations in a tree by repeatedly selecting, expanding, evaluating, and backing up nodes (Coulom, 2007; Browne et al., 2012). UCT applies an upper-confidence bandit rule at each internal node to balance exploration and exploitation (Kocsis & Szepesvári, 2006), and progressive widening extends the approach to spaces in which enumerating every action at a node is impractical (Couëtoux et al., 2011). In SR, Symbolic Physics Learner uses MCTS to construct mathematical expression trees (Sun et al., 2023); DGSR-MCTS uses a pretrained and online-refined neural mutation model (Kamienny et al., 2023); and TPSR adds MCTS lookahead to Transformer decoding with accuracy–complexity feedback (Shojaee et al., 2023). RSRM combines expression-tree MCTS with double Q-learning and learned subtree operators (Xu et al.,

2024), while SR4MDL searches subformulas using a learned minimum-description-length objective (Yu et al., 2025). More recent methods introduce extreme-bandit allocation and nonlocal mutation or crossover actions (Huang et al., 2025), or search permutation-invariant expression graphs with neural and constraint guidance (Xiang et al., 2025). Thus MCTS itself, and its use in SR, are established; the distinction in AutoSR is the state being searched and evaluated. Previous SR applications assign nodes to tokens, subexpressions, formulas, or expression graphs, whereas one AutoSR expansion produces a persistent proposer–experiment–review record whose value and ancestry determine the allocation of subsequent investigations.

A.3 LLM-Guided and Agentic Symbolic Regression

LLMs introduce scientific language and program synthesis as additional proposal mechanisms. In-Context Symbolic Regression iteratively refines LLM-generated formulas with external coefficient optimization (Merler et al., 2024), and LLM4ED uses LLMs as black-box optimizers and evolutionary operators for equation discovery (Du et al., 2024). The Scientific Generative Agent couples LLM proposals with simulation and continuous optimization in a bilevel loop (Ma et al., 2024). LaSR evolves a natural-language concept library alongside symbolic hypotheses (Grayeli et al., 2024), while LLM-SR generates programmatic equation skeletons inside a multi-island evolutionary search using scientific context (Shojaee et al., 2025a). RAG-SR instead trains an online language model and retrieves previously searched expressions to guide feature construction without large-scale pretraining (Zhang et al., 2025).

Recent work moves from LLM proposal components toward longer-horizon agents. SR-Scientist gives an agent tools for data analysis and equation evaluation and preserves strong equations in an experience buffer (Xia et al., 2026). RESTART uses residual diagnostics for short-term structural correction and distills successful refinements into a long-term structure library (Li & Pan, 2026); Deliberate Evolution separates equation generation from search guidance through adaptive operators, diagnostic tools, and reflective trajectory memory (Pang et al., 2026); and the Iterated Agent uses natural-language rationales as semantic operators in an evolutionary search (Song et al., 2025). Other concurrent preprints let an LLM control a conventional SR engine (Xie et al., 2026), combine granular influence feedback with term generation and MCTS (Saveliev et al., 2026), or coordinate Generator, Analyst, Simplifier, and Reviewer roles through process memory (Zhao et al., 2026). These systems already go substantially beyond a formula–score loop, so AutoSR does not claim the first use of tools, diagnostics, memory, multiple roles, or iterative scientific reasoning in SR. Its narrower contribution is to make each complete, artifact-rich investigation a persistent node of global search, keep its memory local to an ancestral branch, append a separate review to the node before reward backup, and synthesize the resulting tree as auditable evidence rather than using the trace only to improve subsequent equation proposals.

Tree search over richer work products also appears in general research agents: AIDE searches code-solution states (Jiang et al., 2025), and AI Scientist-v2 searches branching experimental workflows (Yamada et al., 2025). AutoSR specializes this broader direction to scientific equation discovery by defining the contents, inheritance, independent review, and search semantics of a Research State.

B Reported AutoSR Proposer Times

Table S1: Successful-proposer session times for the nine selected AutoSR trajectories. These values do not measure end-to-end system cost.

Task	Reported successful-proposer time
C3g	8 min 58 s
C5f	15 min 44 s
C6c	33 min 1 s
I.32.17_4_3	16 min 13 s
I.29.16_0_0	12 min 32 s
II.6.15a_2_0	7 min 47 s
II.35.18_0_0	6 min 26 s
II.36.38_3_0	9 min 7 s
III.4.33_3_0	8 min 9 s

These values exclude other explored branches, reviewers, parallel work, conventional-SR runs, and orchestration overhead. Complete AutoSR calls, tokens, total wall time, and attempted-run counts are unavailable here.

C Control Run Ranges

Table S2 gives the median, minimum, and maximum test error of every control method on each task, over the same runs as Table 1.

Table S2: Observed control-run ranges on the nine selected tasks. The minimum and maximum describe the observed runs; they are not confidence intervals. With $n = 2$, the median is the arithmetic mean of the two observations. Each metric is aggregated on its own scale; NRMSE is transformed per run before aggregation.

Task	Method	n	Median	Minimum	Maximum
cp3-bench: test MSE					
C3g	PySR	5	1.85×10^{-5}	7.42×10^{-6}	6.61×10^{-5}
	FEX	5	3.19×10^{-5}	1.51×10^{-5}	1.41×10^{-4}
	LLM-SR	2	8.62×10^{-19}	8.62×10^{-19}	8.62×10^{-19}
	Direct	2	8.62×10^{-19}	8.62×10^{-19}	8.62×10^{-19}
C5f	PySR	5	7.22×10^{-5}	1.42×10^{-5}	4.58×10^{-4}
	FEX	5	1.05×10^{-3}	2.84×10^{-4}	1.20×10^{-3}
	LLM-SR	3	1.32×10^{-2}	1.71×10^{-3}	1.32×10^{-2}
	Direct	3	4.20×10^{-2}	1.32×10^{-2}	5.86×10^{-2}
C6c	PySR	5	8.76×10^{-2}	4.09×10^{-3}	4.57×10^{-1}
	FEX	5	3.57×10^{-1}	3.06×10^{-1}	5.42×10^{-1}
	LLM-SR	3	5.96×10^{-1}	4.97×10^{-1}	7.77×10^{-1}
	Direct	3	8.53×10^{-1}	7.06×10^{-1}	8.76×10^{-1}
LSR-Transform: test NRMSE					
I.32.17_4_3	PySR	5	7.34×10^{-2}	8.22×10^{-3}	9.40×10^{-2}
	FEX	5	1.51×10^{-1}	1.33×10^{-1}	1.97×10^{-1}
	LLM-SR	2	1.80×10^{-7}	1.80×10^{-7}	1.80×10^{-7}
	Direct	2	1.80×10^{-7}	1.80×10^{-7}	1.80×10^{-7}
I.29.16_0_0	PySR	5	6.05×10^{-2}	1.66×10^{-2}	8.06×10^{-2}
	FEX	5	1.67×10^{-1}	1.39×10^{-1}	3.12×10^{-1}
	LLM-SR	3	1.65×10^{-7}	1.59×10^{-7}	1.67×10^{-7}
	Direct	3	1.65×10^{-7}	1.56×10^{-7}	1.66×10^{-7}
II.6.15a_2_0	PySR	5	2.36×10^{-2}	1.52×10^{-2}	3.14×10^{-2}
	FEX	5	1.11×10^{-1}	5.53×10^{-2}	1.28×10^{-1}
	LLM-SR	2	1.02×10^{-7}	1.02×10^{-7}	1.02×10^{-7}
	Direct	2	1.02×10^{-7}	1.02×10^{-7}	1.02×10^{-7}
II.35.18_0_0	PySR	3	3.94×10^{-7}	3.65×10^{-7}	4.81×10^{-7}
	FEX	5	1.08×10^{-1}	7.44×10^{-2}	1.50×10^{-1}
	LLM-SR	3	4.60×10^{-7}	4.60×10^{-7}	1.59×10^{-2}
	Direct	3	4.60×10^{-7}	4.60×10^{-7}	4.60×10^{-7}
II.36.38_3_0	PySR	4	8.00×10^{-8}	8.00×10^{-8}	8.01×10^{-8}
	FEX	5	1.61×10^{-1}	1.20×10^{-1}	3.30×10^{-1}
	LLM-SR	2	8.88×10^{-8}	8.88×10^{-8}	8.88×10^{-8}
	Direct	2	2.16×10^{-6}	1.34×10^{-6}	2.99×10^{-6}
III.4.33_3_0	PySR	5	1.23×10^{-4}	2.27×10^{-6}	2.33×10^{-4}
	FEX	5	4.34×10^{-2}	3.33×10^{-2}	1.03×10^{-1}
	LLM-SR	2	7.22×10^{-8}	7.22×10^{-8}	7.22×10^{-8}
	Direct	2	7.22×10^{-8}	7.22×10^{-8}	7.22×10^{-8}