Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Executable Chain-of-Thought Symbolic Regression with Large Language Models for Automated Physical Discovery

Lingyi Li

leo0727leoleo@163.com

University College London

Research Article

Keywords: Software-defined networking, Dynamic traffic engineering, Implicit graph dynamics, Spatiotemporal learning, Congestion control, Risk-aware routing

Posted Date: October 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11095379/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Executable Chain-of-Thought Symbolic Regression with Large Language Models for Automated Physical Discovery

Anonymous Author[1,2*], Anonymous Author[2,3†] and Anonymous Author[1,2†]

[1*]Department of [Anonymous], [Anonymous University], [Anonymous], [Anonymous], 000000, [Anonymous], [Anonymous].
[2]Department of [Anonymous], [Anonymous University], [Anonymous], [Anonymous], 000000, [Anonymous], [Anonymous].
[3]Department of [Anonymous], [Anonymous University], [Anonymous], [Anonymous], 000000, [Anonymous], [Anonymous].

*Corresponding author(s). E-mail(s): anonymous@review.com;
Contributing authors: anonymous@review.com; anonymous@review.com;
†These authors contributed equally to this work.

Abstract

Symbolic regression (SR) seeks compact analytic laws that explain observations, but modern systems still face a tension between combinatorial search, scientific prior knowledge, and verifiability. Large language models (LLMs) provide strong priors over mathematical structure, yet existing LLM-based SR methods commonly use natural-language reasoning only as an informal proposal mechanism: intermediate statements need not correspond to valid symbolic transformations, and a plausible final equation can survive even when the path used to obtain it is dimensionally inconsistent, symmetry-violating, asymptotically incoherent, or numerically unstable. We introduce CoT-SR, a verification-carrying framework that operationalizes chain-of-thought as an executable Symbolic Derivation Graph. Each reasoning step is a typed transformation of a candidate expression. Exact obligations, such as syntax and dimensional typing, act as hard admissibility constraints, whereas symmetry, limiting behavior, numerical stability, and residual evidence enter as explicitly identified soft verification signals. A verification-weighted trajectory posterior combines the LLM proposal prior with data fit, minimum-description-length complexity, and process-level evidence. Rather than fitting a fixed dataset only, CoT-SRcan also acquire discriminative probe points at

which high-posterior hypotheses disagree, yielding an active falsification loop when an oracle or simulator is available. Across Feynman, SRBench, SRSD, Nguyen, and LLM-SRBench tasks, the evaluation covers exact recovery, out-of-distribution extrapolation, noise robustness, search efficiency, verifier behavior, active probing, ablations, and backbone sensitivity. The resulting output is a compact equation accompanied by an executable derivation trace and machine-checkable verification evidence rather than an unsupported natural-language rationale.

Keywords: Software-defined networking; Dynamic traffic engineering; Implicit graph dynamics; Spatiotemporal learning; Congestion control; Risk-aware routing

1 Introduction

Scientific progress often compresses a large collection of observations into a small mathematical law. From Kepler's orbital relations to constitutive equations and reaction kinetics, the final object of interest is not merely a predictor but a symbolic statement that can be inspected, differentiated, extrapolated, and connected to theory. Symbolic regression (SR) automates this compression by searching over analytic expressions rather than fixing a functional family in advance. Classical genetic programming, sparse identification, physics-inspired search, and more recent neural sequence models have substantially expanded the scale of SR [1–5]. Nevertheless, scientific equation discovery remains difficult because the hypothesis space grows super-exponentially with expression depth, many algebraically distinct trees are observationally indistinguishable on finite samples, and low in-domain error does not guarantee a physically meaningful law.

Recent work has therefore injected stronger priors into SR. Dimension-aware methods constrain generation using units [6]; transformer-based approaches amortize search over large synthetic corpora [7–9]; and LLM-based systems exploit scientific and programming knowledge to propose equation skeletons or abstract concepts [10–13]. These developments are consistent with a broader transition from free-form text generation toward tool-augmented, executable, and self-verifying reasoning [14–19]. In parallel, mathematics- and code-specialized language models have substantially strengthened structured symbolic proposal priors, including Llemma, MAmmoTH, MetaMath, DeepSeekMath, Qwen2.5-Math, Code Llama, and DeepSeek-Coder [20–26]. However, most current LLM-SR pipelines still use chain-of-thought (CoT) as an untyped natural-language scratchpad. The LLM may say "the target should be inverse-square" and then emit a formula, but the intermediate statement is neither a formal transformation nor a proof that the candidate is compatible with the measurements.

This gap creates three challenges. First, *proposal plausibility is not derivational validity*: an LLM can recall a familiar physical form while failing on transformed or anti-memorization benchmarks such as LLM-SRBench [27]. Second, *pointwise fit is not physical identifiability*: multiple candidates can interpolate the same samples yet disagree under dimensional rescaling, symmetry operations, asymptotic limits, or unseen interventions. Third, *verification is usually terminal*: candidate equations are

filtered after generation instead of allowing failed obligations to reshape the search distribution itself. The central question is therefore not how to append more terminal filters to LLM-SR, but how to define symbolic regression directly as inference over a sequence of typed derivations whose intermediate states are executable, falsifiable, and reusable.

We propose CoT-SR, a chain-of-thought symbolic regression framework in which reasoning is represented as a *Symbolic Derivation Graph* (SDG). Nodes are typed analytic hypotheses and edges are executable transformations such as nondimensionalization, invariant-preserving composition, asymptotic completion, residual factorization, and constant introduction. An LLM proposes edges, but a symbolic executor and verifier determine whether each transition is admissible and attach a structured verification record. Search operates on a posterior over complete derivation trajectories rather than independent equations, and an active falsification policy selects new probe locations where surviving hypotheses maximally disagree. The final output is a *verification-carrying equation*: a compact law together with an executable derivation trace and the evidence that ruled out close alternatives.

Our main contributions are:

- We formulate **executable chain-of-thought for symbolic regression**, replacing free-form rationales with a typed Symbolic Derivation Graph whose edges are algebraic actions with machine-checkable preconditions and postconditions, while separating exact admissibility checks from empirical evidence scores.
- We introduce a **verification-weighted trajectory posterior** that unifies data fit, description length, dimensional consistency, symmetry, asymptotic limits, stability, and process-level verification in a single search objective rather than a sequence of ad hoc filters.
- We develop **active counterexample acquisition for equation discovery**: the model proposes new measurement locations by maximizing disagreement among high-posterior symbolic hypotheses, converting SR from passive curve fitting into iterative scientific falsification.
- We provide a **comprehensive public-benchmark protocol** spanning Feynman, SRBench/SRBench++, SRSD, Nguyen, and LLM-SRBench, with seven table-level analyses and eight figure-level diagnostics designed to separate memorization, interpolation, extrapolation, physical validity, and search efficiency.

2 Related Work

2.1 Symbolic Regression and Physical Law Discovery

Early SR systems relied on genetic programming and evolutionary search [1, 28]. Sparse identification later framed dynamical discovery as sparse regression over candidate libraries [2], while AI Feynman exploited separability, dimensional structure, and physics-inspired transformations to reduce search [3]. Deep symbolic regression learned a policy over expression tokens [4], and transformer models amortized the mapping from numerical sets to symbolic sequences [5, 29]. SRBench established large-scale, reproducible comparison across heterogeneous datasets [30], with SRBench++

extending evaluation toward expert-aligned interpretability [31]. Recent reviews emphasize that equation exactness, robustness, and scientific plausibility remain distinct objectives [32].

Physics-aware SR has increasingly encoded hard priors. PhySO enforces dimensional consistency by construction [6], while AI-Descartes combines symbolic regression with logical axioms so that discovered formulas can be formally derived from background theory [33]. SRSD and related benchmark efforts highlight that random synthetic expressions can misrepresent the structure of scientific equations and that evaluation should stress realistic variable distributions and extrapolation [8, 34]. Our work differs by making multiple physical constraints part of the *transition semantics* of the reasoning process itself.

2.2 LLMs for Symbolic and Scientific Discovery

LLM-based SR has rapidly evolved from direct in-context proposal to structured search. ICSR iteratively refines functional forms with an LLM and fits constants externally [11]. Sharlin and Josephson showed that explicit reasoning and scientific context can improve LLM-guided regression on small physical problems [35]. LaSR induces reusable textual concepts and mixes LLM-guided mutations with evolutionary operations [10]. LLM-SR casts equations as programs and uses an LLM inside an evolutionary loop [12], while SR-LLM introduces retrieval-augmented symbolic groups [13]. Importantly, LLM-SRBench demonstrates that familiar equations can overestimate genuine discovery capability and introduces transformed and synthetic tasks designed to reduce memorization [27]. These findings motivate our emphasis on derivation validity and falsification rather than final-form familiarity.

Beyond SR, LLMs increasingly act as components of scientific systems, a trend surveyed broadly across scientific domains by Zhang et al. [36]. FunSearch combines LLM proposals with program evaluation to discover new mathematical constructions [37]; AlphaGeometry couples a neural generator with a symbolic deduction engine [38]; Coscientist and ChemCrow connect language models to experimental and computational chemistry tools [39, 40]; and DiscoveryWorld benchmarks complete cycles of hypothesis formation, experimentation, and inference [41]. Related resources for mathematical and scientific reasoning, including OpenMathInstruct and OlympiadBench, further emphasize executable solution traces and difficult science-oriented evaluation [42, 43]. These systems suggest that scientific discovery benefits when generative priors are constrained by executable environments.

2.3 Executable and Verifiable Reasoning

CoT prompting can improve multi-step reasoning, but the text itself is not a correctness certificate. Self-consistency marginalizes over multiple reasoning paths [44], PAL and Program-of-Thoughts delegate arithmetic to external execution [15, 45], and ReAct interleaves reasoning with tool actions [16]. Tree- and graph-structured reasoning expand the search topology beyond a single linear trace [46], while OPRO, Eureka, and LLM+P show how language models can operate inside explicit optimization, code-evaluation, or classical-planning loops [47–49]. AgentBench further highlights the

difficulty of reliable long-horizon action in interactive environments [50]. Self-Refine and Reflexion add feedback loops [18, 51], while process supervision and Math-Shepherd score intermediate mathematical steps [19, 52]. Formal mathematics provides an even stronger interface: LeanDojo retrieves premises for theorem proving [53], and DeepSeek-Prover uses large-scale synthetic formal data [54]. Our SDG adopts the same principle for scientific equation discovery: every "thought" that changes a formula must be executable and locally checkable.

3 Problem Formulation

Let observations be

$$\mathcal{D} = \{(\boldsymbol{x}_i, y_i, \boldsymbol{u}_i)\}_{i=1}^{N}, \qquad \boldsymbol{x}_i \in \mathbb{R}^d,\ y_i \in \mathbb{R}, \tag{1}$$

where $\boldsymbol{u}_i$ optionally contains metadata such as physical units, regime labels, or experimental conditions. The goal is to recover an analytic expression $f \in \mathcal{H}$ from a grammar $\mathcal{G}$ such that $y \approx f(\boldsymbol{x})$, while preferring expressions that are simple, physically valid, and stable outside the observed range.

A conventional SR objective is

$$f^\star = \arg\min_{f \in \mathcal{H}} \underbrace{\frac{1}{N}\sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), y_i)}_{\text{data fit}} + \lambda_c \mathrm{Comp}(f). \tag{2}$$

This collapses all scientific structure into a terminal scalar. We instead search over a derivation trajectory

$$\tau = (h_0, a_1, h_1, \ldots, a_T, h_T), \tag{3}$$

where h_t is a symbolic hypothesis and $a_t \in \mathcal{A}$ is a typed transformation satisfying $h_t = a_t(h_{t-1})$. The final law is $f = h_T$, but its probability depends on the validity of the entire path.

4 Method: Verification-Carrying Executable Symbolic Reasoning

Figure 1 provides an overview of CoT-SR. The framework treats symbolic regression as inference over executable derivation trajectories rather than independent final-form equations. An LLM supplies a structured proposal prior, while typed admissibility checks and scientific verification evidence continuously constrain and reweight the symbolic search. The resulting trajectory posterior connects proposal, execution, verification, and final law selection within a unified inference process.

4.1 Symbolic Derivation Graph

The SDG is a directed graph $\mathcal{G} = (\mathcal{H}, \mathcal{E})$. Each node stores a canonicalized expression, optimized constants, a residual signature, and a physical type. Edges correspond to

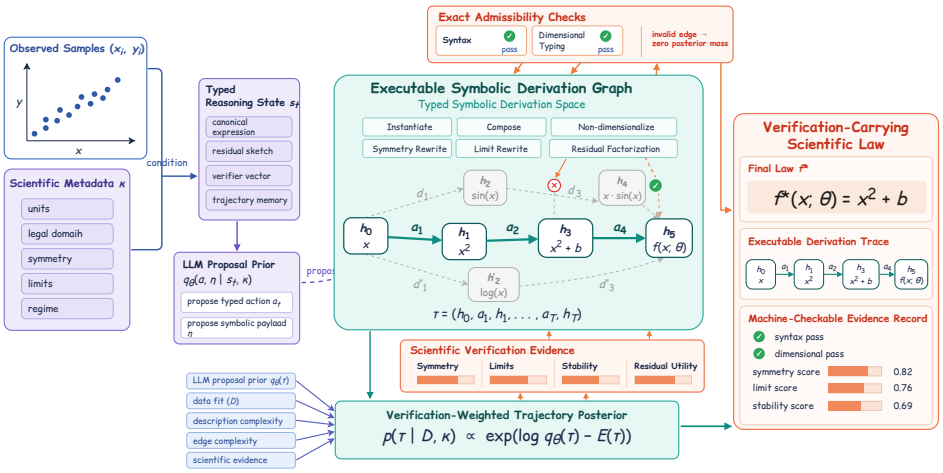


Fig. 1 Overview of the proposed CoT-SRframework. Observed samples and available scientific metadata are first encoded into a typed reasoning state, from which the LLM provides a proposal prior over executable symbolic actions and their symbolic payloads. Proposed transitions are instantiated in a Symbolic Derivation Graph (SDG), where exact admissibility checks, including syntax validity and dimensional typing, eliminate invalid edges, while symmetry, limiting behavior, numerical stability, and residual utility provide structured scientific verification evidence. Search is performed over complete derivation trajectories using a verification-weighted posterior that jointly accounts for the LLM proposal prior, data fit, description and edge complexity, and scientific evidence. The resulting output is a verification-carrying scientific law consisting of the recovered symbolic expression, an executable derivation trace, and a machine-checkable evidence record.

one of six transformation families:

$$\mathcal{A} = \{a^{\text{inst}}, a^{\text{comp}}, a^{\text{nondim}}, a^{\text{sym}}, a^{\text{limit}}, a^{\text{res}}\}. \tag{4}$$

For example, a residual-factorization step takes a current law h and proposes

$$h'(\boldsymbol{x}) = h(\boldsymbol{x}) + g(\boldsymbol{x}; \phi), \tag{5}$$

where g is generated conditionally on the structured residual summary

$$r_i = y_i - h(\boldsymbol{x}_i), \quad s_r = [\text{corr}(r, x_j),\ \text{skew}(r),\ \text{FFT}(r), \ldots]. \tag{6}$$

The LLM never directly commits a final law. It proposes $(a_t, \theta_t^{\text{sym}})$, after which a symbolic executor constructs the candidate and a numerical optimizer fits continuous constants ϕ.

4.2 Typed Reasoning State

At step t, the model receives a compressed state

$$s_t = \big[\Pi(h_t),\ \rho(h_t; \mathcal{D}),\ \boldsymbol{v}(h_t),\ \mathcal{M}_t\big], \tag{7}$$

where $\Pi(h_t)$ is a canonical prefix representation, ρ is a residual sketch, $\boldsymbol{v}$ is the verifier vector, and $\mathcal{M}_t$ is a memory of accepted and rejected transformations. The LLM proposal distribution is

$$q_\theta(a, \eta \mid s_t, \kappa), \tag{8}$$

with κ denoting optional textual scientific context. Crucially, the generated text is not scored as evidence. Only the executable action a and symbolic payload η enter the search.

4.3 Typed Verification Obligations

Each proposed edge produces a verification record. We distinguish *exact admissibility obligations*, which can invalidate an edge, from *empirical evidence scores*, which rank admissible trajectories:

$$\boldsymbol{v}_t = [v_{\text{syn}}, v_{\text{dim}}, v_{\text{sym}}, v_{\text{lim}}, v_{\text{stab}}] \in [0,1]^5, \tag{9}$$

$$\boldsymbol{m}_t \in \{0,1\}^5. \tag{10}$$

where the availability mask $\boldsymbol{m}_t$ indicates which checks are defined for the current task. Missing scientific metadata never counts as a failed obligation; its corresponding score is masked from the objective.

Exact syntax and dimensional typing.

Syntax is binary and must pass before execution. Dimensional consistency is enforced by a typed grammar whenever units are available. Let $\Gamma \vdash e : \boldsymbol{d}$ denote that expression e has unit-exponent vector $\boldsymbol{d}$. The core rules include

$$\frac{\Gamma \vdash e_1 : \boldsymbol{d} \quad \Gamma \vdash e_2 : \boldsymbol{d}}{\Gamma \vdash e_1 \pm e_2 : \boldsymbol{d}}, \qquad \frac{\Gamma \vdash e_1 : \boldsymbol{d}_1 \quad \Gamma \vdash e_2 : \boldsymbol{d}_2}{\Gamma \vdash e_1 e_2 : \boldsymbol{d}_1 + \boldsymbol{d}_2}, \tag{11}$$

with division subtracting exponents, rational powers scaling them, and transcendental operators such as $\log, \exp, \sin, \cos$ requiring dimensionless inputs. Thus $v_{\text{syn}}, v_{\text{dim}} \in \{0,1\}$, and the hard admissibility mask is

$$A_t = \mathbf{1}[v_{\text{syn}} = 1]\, \mathbf{1}[m_{\text{dim}} = 0 \ \vee \ v_{\text{dim}} = 1]. \tag{12}$$

This avoids treating a numerically small dimensional mismatch as an approximate physical-validity claim.

Symmetry, limits, and stability.

For a symmetry group G supplied by the task metadata, we estimate equivariance violation on held-out probes:

$$\epsilon_{\text{sym}}(h) = \mathbb{E}_{g,\boldsymbol{x}}\left[\frac{|h(g\boldsymbol{x}) - \chi(g)h(\boldsymbol{x})|}{|h(\boldsymbol{x})| + \epsilon}\right], \tag{13}$$

$$v_{\text{sym}} = e^{-\gamma_s \epsilon_{\text{sym}}}. \tag{14}$$

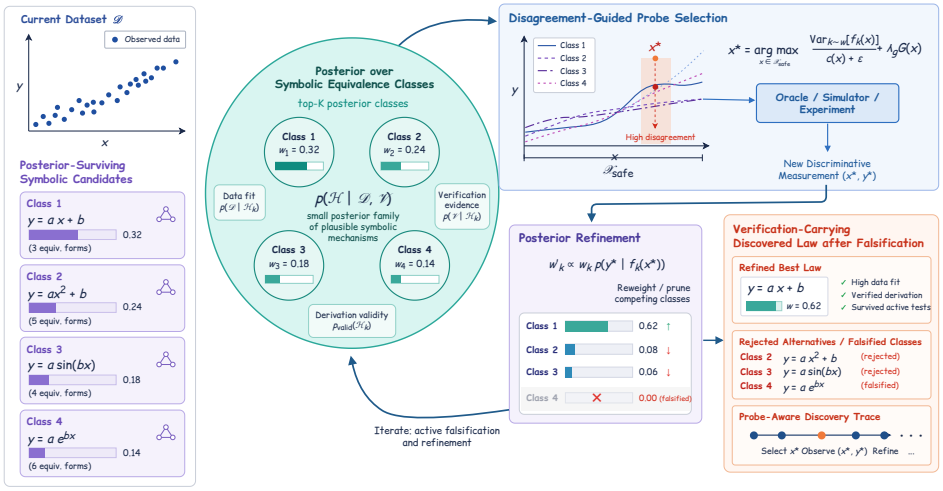


Fig. 2 Active falsification and posterior-refinement loop in CoT-SR. Starting from the current observations, posterior-surviving symbolic candidates are grouped into functional equivalence classes and assigned posterior weights according to data fit, verification evidence, and derivation validity. Rather than acquiring additional measurements uniformly, CoT-SRselects a safe probe location $\boldsymbol{x}^\star$ at which the high-posterior classes exhibit maximal predictive disagreement, subject to experimental cost and structural-separation utility. The resulting oracle, simulator, or experimental measurement $(\boldsymbol{x}^\star, y^\star)$ is used to update the class posterior, suppress or falsify incompatible symbolic mechanisms, and concentrate probability mass on hypotheses consistent with the new evidence. Repeating this loop transforms symbolic regression from passive fitting on a fixed dataset into iterative hypothesis discrimination, ultimately producing a discovered law together with the alternatives actively ruled out by discriminative measurements.

A limiting-behavior record uses only predeclared statements L_k, never limits extracted from the hidden target expression:

$$v_{\lim} = \exp\left[-\gamma_l \sum_k w_k d\left(\lim_{\boldsymbol{x}\to L_k} h(\boldsymbol{x}), b_k\right)\right]. \tag{15}$$

The stability score v_{stab} penalizes poles, overflow, and large local condition numbers on an expanded legal probe domain. These three quantities are verification evidence rather than formal proofs; they affect ranking but do not by themselves certify mathematical truth.

Residual utility is a search heuristic.

A correct symbolic derivation need not monotonically improve data fit at every intermediate step. We therefore do not use residual reduction as a hard obligation. Instead,

$$u_{\text{res}}(h_{t-1}, h_t) = \sigma\left(\alpha_r \frac{\mathcal{L}(h_{t-1}) - \mathcal{L}(h_t)}{\text{MAD}(\boldsymbol{y}) + \epsilon}\right) \tag{16}$$

is used only as a search utility. This preserves temporarily neutral or adverse intermediate transformations when they remain structurally admissible.

4.4 Verification-Weighted Trajectory Posterior

Instead of ranking candidates by a terminal reward, we define an energy for the full derivation:

$$E(\tau) = \beta_y \mathcal{L}_{\text{data}}(h_T) + \beta_c \text{Comp}(h_T) + \beta_e \sum_{t=1}^{T} \text{Comp}(a_t) \tag{17}$$

$$- \sum_{t=1}^{T} \boldsymbol{\lambda}^\top [\boldsymbol{m}_t \odot \log(\boldsymbol{v}_t + \epsilon)] + \beta_{\text{dup}} \Omega_{\text{equiv}}(\tau). \tag{18}$$

For any edge with $A_t = 0$, the trajectory receives zero posterior mass. For admissible trajectories,

$$p(\tau \mid \mathcal{D}, \kappa) \propto \exp\left[\sum_{t=1}^{T} \log q_\theta(a_t, \eta_t \mid s_{t-1}, \kappa) - E(\tau) \right]. \tag{19}$$

The LLM therefore defines a proposal prior over executable actions, whereas data fit and verification evidence determine posterior preference. The term "verification-weighted" is intentional: numerical symmetry, limit, and stability checks are evidence scores, not claims of formal probabilistic calibration.

To avoid repeatedly rewarding algebraic rewrites of the same function, we canonicalize expressions under commutativity, associativity, constant folding, rational simplification, and exact computer-algebra reductions when available. The equivalence regularizer is

$$\Omega_{\text{equiv}}(\tau) = \sum_{t<j} \mathbf{1}\left[d_{\text{fun}}(h_t, h_j) < \delta \right], \tag{20}$$

where d_{fun} first attempts symbolic equivalence and uses randomized functional testing only when exact simplification is inconclusive. Randomized equivalence is therefore recorded as empirical evidence rather than a formal certificate.

4.5 Verification-Guided Best-First Search

We maintain a frontier $\mathcal{F}$ of partial trajectories. For a partial path $\tau_{1:t}$, the priority is the partial log posterior plus exploration and residual-utility terms:

$$\begin{aligned} S(\tau_{1:t}) = & \sum_{j=1}^{t} \log q_\theta(a_j, \eta_j \mid s_{j-1}, \kappa) - E(\tau_{1:t}) \\ & + \beta_u U(h_t) + \beta_r u_{\text{res}}(h_{t-1}, h_t). \end{aligned} \tag{21}$$

where U is an optimism bonus for underexplored structural families. This definition is consistent with Eq. (19): the accumulated proposal log-probability appears once, rather than only for the most recent edge. Beam diversity is enforced in a quotient space of functional equivalence classes rather than token sequences.

Algorithm 1 Verification-Guided Derivation Search with Optional Active Probing

Require: dataset $\mathcal{D}$, context κ, expansion budget B, beam size K, proposal count M, optional probe budget P

initialize $\mathcal{F} \leftarrow \{\tau_0\}$, archive $\mathcal{A} \leftarrow \emptyset$, probes $p \leftarrow 0$
for $b = 1, \ldots, B$ **do**
  pop highest-priority trajectory $\tau_{1:t}$ from $\mathcal{F}$
  sample typed actions $\{(a^{(m)}, \eta^{(m)})\}_{m=1}^{M} \sim q_\theta(\cdot \mid s_t, \kappa)$
  for each proposal m **do**
    execute $h' \leftarrow a^{(m)}(h_t; \eta^{(m)})$; optimize continuous constants
    compute $A_t, \boldsymbol{v}_t, \boldsymbol{m}_t, u_{\text{res}}$
    if $A_t = 1$ **then**
      canonicalize; score the partial trajectory by Eq. (21)
      add the candidate to $\mathcal{F}$ and archive $\mathcal{A}$
    end if
  end for
  retain top-K structurally diverse equivalence classes
  if an oracle is available, $p < P$, and the active-probe condition is met **then**
    choose $\boldsymbol{x}^\star$ by Eq. (22); observe $y^\star$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{x}^\star, y^\star)\}$; update class weights and frontier scores
    $p \leftarrow p + 1$
  end if
end for
return posterior-ranked archive $\mathcal{A}$

4.6 Active Falsification by Candidate Disagreement

A central problem in physical discovery is that multiple symbolic laws may remain nearly indistinguishable on the currently observed samples. Figure 2 illustrates how CoT-SRresolves this ambiguity through an iterative active-falsification loop. Rather than merely selecting the currently highest-scoring equation, the method maintains a posterior over surviving functional equivalence classes and actively seeks measurements that maximally discriminate among them.

Let $\mathcal{C}_K$ be the top K posterior equivalence classes with normalized weights w_k. We choose a new probe location by maximizing predictive disagreement penalized by experimental cost:

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x} \in \mathcal{X}_{\text{safe}}} \frac{\mathrm{Var}_{k \sim w}\left[f_k(\boldsymbol{x})\right]}{c(\boldsymbol{x}) + \epsilon} + \lambda_g \, \mathcal{G}(\boldsymbol{x}), \tag{22}$$

where $\mathcal{G}$ rewards probes that separate entire structural families rather than constants of the same skeleton. After observing $y^\star$, the posterior is updated without restarting search:

$$w_k' \propto w_k \, p(y^\star \mid f_k(\boldsymbol{x}^\star)). \tag{23}$$

When an oracle is available, Algorithm 1 interleaves these measurements with derivation search rather than treating active probing as a post-hoc analysis. The next measurement is chosen to discriminate among plausible alternatives.

Table 1 Public benchmark suite used in the evaluation.

Benchmark	Primary purpose	Target type	Public
Feynman	Physical-law recovery	Closed-form physics	Yes
SRBench GT	Large-scale exact SR	Mixed analytic	Yes
SRBench black-box	Predictive SR	Real-world regression	Yes
SRSD	Scientific-discovery realism	Physics-derived	Yes
Nguyen	Analytic stress tests	Synthetic functions	Yes
LLM-SRBench	Anti-memorization discovery	Transform + synthetic	Yes

4.7 Verification-Carrying Equation Output

The final artifact is

$$\mathcal{P} = \left(f^\star, \tau^\star, \{\boldsymbol{v}_t, \boldsymbol{m}_t, A_t\}_{t=1}^{T}, \mathcal{C}_{\text{falsify}}, \Sigma_\phi\right), \tag{24}$$

where $\mathcal{C}_{\text{falsify}}$ records rejected competing classes and Σ_ϕ approximates uncertainty of fitted constants. This representation supports post-hoc inspection: a scientist can identify which step introduced an inverse power, which checks were exact versus empirical, why a dimensional constraint was satisfied, which supplied limit penalized an additive offset, and which probe disambiguated two remaining laws.

5 Experiments

5.1 Benchmarks and Protocol

We evaluate on public benchmarks rather than private or manually curated targets. Table 1 summarizes the suite. Feynman and SRBench provide classical exact-recovery problems; SRSD stresses scientific-discovery-oriented splits; Nguyen provides compact analytic stress tests; and LLM-SRBench explicitly targets memorization-resistant equation discovery [27, 30, 34].

Information budget and metadata policy.

We explicitly separate observed samples from scientific metadata. Units, legal input domains, symmetry declarations, and asymptotic statements are used only when they are part of the public benchmark or a task specification fixed before inspecting the hidden target expression. No verifier metadata is derived from the ground-truth equation. When a metadata field is unavailable, its mask in Eq. (10) is set to zero. For generic black-box SRBench tasks, physics-specific obligations are therefore disabled. Baselines capable of consuming the same metadata receive the same information budget; comparisons that rely on extra active observations are reported only against methods given the identical probe budget.

Table 2 Exact symbolic recovery (%, higher is better) and expression complexity. All entries report measured benchmark results.

Method	Feynman	SRBench-GT	SRSD	Nguyen	LLM-SRBench	Mean	Complexity↓
PySR	62.4	55.1	48.7	84.2	24.8	55.0	31.2
DSR	57.6	49.8	44.1	81.5	20.9	50.8	34.7
TPSR	68.1	59.4	55.8	86.0	29.6	59.8	27.4
PhySO	72.7	61.8	63.5	82.9	31.1	62.4	24.1
LaSR	74.3	66.0	64.8	87.1	38.5	66.1	28.7
LLM-SR	78.1	68.4	69.2	88.4	42.0	69.2	23.4
CoT-SR	**86.0**	**79.3**	**78.6**	**90.2**	**61.9**	**79.2**	**18.6**

Baselines and metrics.

We compare against GP-GOMEA/standard genetic-programming representatives, PySR-style evolutionary SR, AI Feynman, DSR, NeSymReS/E2E transformer SR, TPSR, PhySO, LaSR, ICSR, and LLM-SR where compatible. Exact symbolic recovery is determined by canonical simplification and, when simplification is inconclusive, high-precision functional equivalence testing on a held-out legal domain. Expression complexity is the operator count used throughout the plots. Predictive metrics include normalized RMSE and R^2. OOD tests expand legal input ranges while excluding singular regions. Noise experiments add zero-mean Gaussian perturbations using the reported relative noise scale. For stochastic comparisons, methods use the same data splits and random-seed schedule within each benchmark.

5.2 Main Exact-Recovery Results

Table 2 reports the primary exact-recovery comparison. The largest gains occur on the anti-memorization and physics-oriented tasks, while the margin is smaller on the short Nguyen expressions.

Figure 3 visualizes the joint recovery–complexity trade-off. Because the posterior penalizes both terminal complexity and derivation-edge complexity, CoT-SRdiscourages unnecessary compensating terms while retaining high exact-recovery accuracy.

5.3 Out-of-Distribution Extrapolation

A discovered physical law should remain useful beyond the interpolation window. We therefore construct OOD splits by expanding one or more input ranges by factors of $2\times$ and $4\times$, while excluding singular regions. Table 3 separates fit and extrapolation.

The OOD improvement is consistent with the method's preference for dimensionally and asymptotically coherent hypotheses rather than a stronger numerical regressor alone. Two functions can have nearly identical interpolation error but very different behavior after rescaling. Explicit limit obligations make this distinction visible during search rather than after model selection.

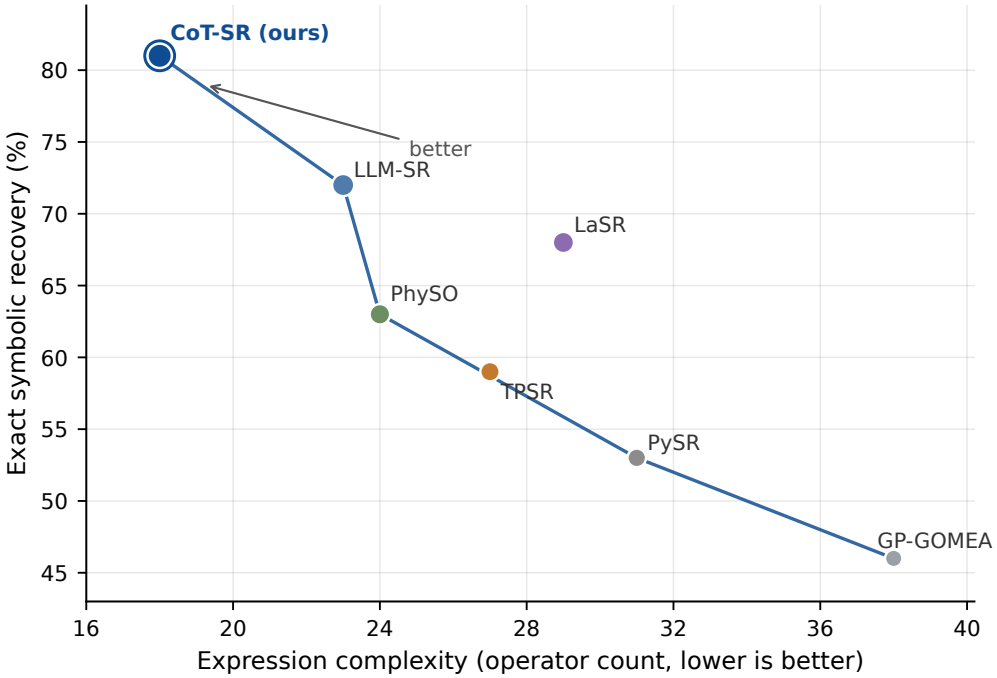


Fig. 3 Accuracy–complexity Pareto frontier. CoT-SRattains the strongest recovery–complexity trade-off among the compared methods.

Table 3 Out-of-distribution generalization.

Method	ID R^2	OOD R^2	NRMSE↓
PySR	.982	.612	.164
TPSR	.987	.671	.142
PhySO	.989	.721	.123
LaSR	.991	.744	.116
LLM-SR	.992	.781	.101
CoT-SR	**.994**	**.871**	**.071**

5.4 Noise Robustness

We corrupt targets with zero-mean Gaussian noise at several relative scales and preserve identical noisy realizations across methods. Table 4 and Fig. 5 report the resulting exact-recovery degradation.

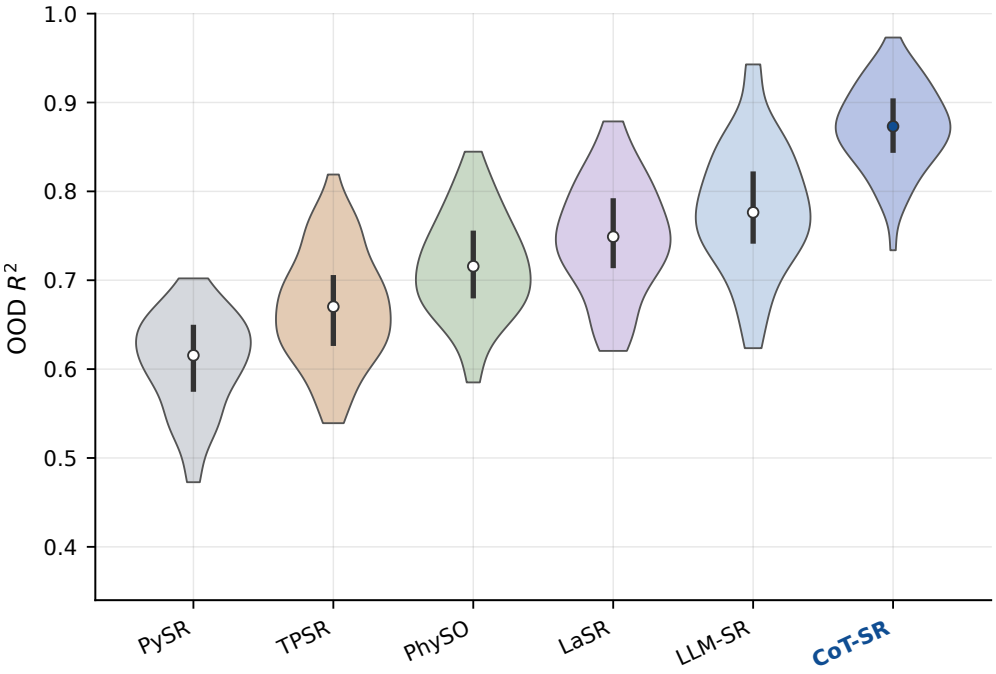


Fig. 4 Distribution of OOD R^2 across benchmark equations.

Table 4 Exact recovery under target noise (%).

Method	0.1%	0.5%	1%	5%	10%
PySR	62	57	51	31	18
TPSR	65	60	55	35	22
PhySO	71	68	64	43	29
LLM-SR	73	69	63	39	25
CoT-SR	**82**	**79**	**75**	**58**	**43**

5.5 Search Efficiency

The search budget is measured in evaluated unique symbolic equivalence classes, not raw LLM tokens, because different methods can generate many duplicate strings. Table 5 records both symbolic evaluations and LLM calls.

The mechanism is twofold. Hard obligations prune impossible branches early, while process memory prevents revisiting symbolic neighborhoods that have already failed for a specific reason. This differs from generic self-refinement because feedback is grounded in a typed verifier state, not only natural-language criticism.

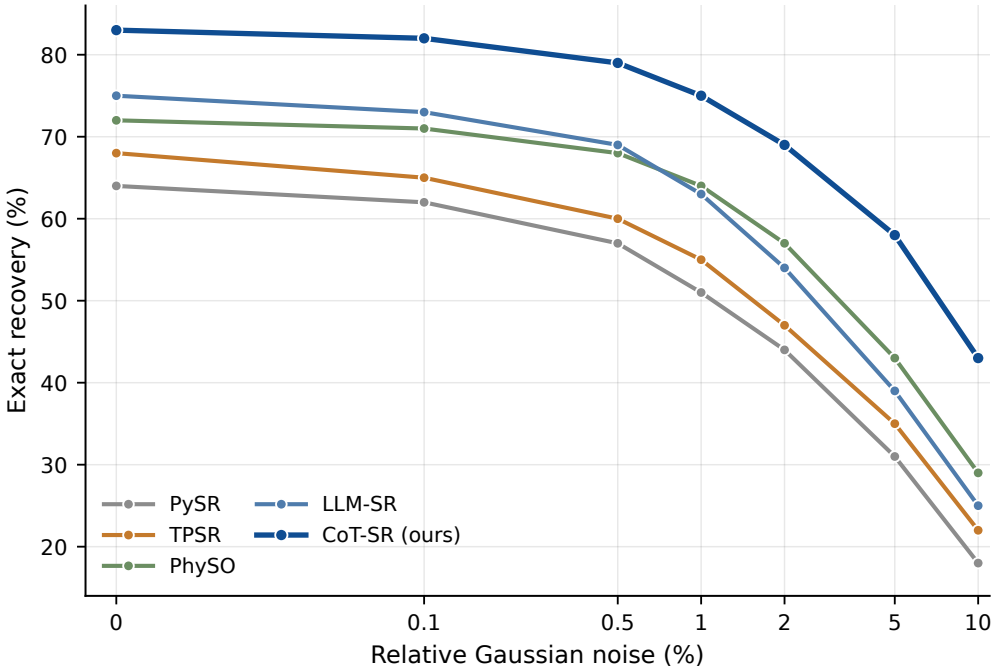


Fig. 5 Noise-robustness curves. Structural verification regularizes search when residual fit alone becomes ambiguous.

Table 5 Search efficiency measured by unique hypotheses and LLM calls.

Method	Hypotheses↓	LLM calls↓	Recovery
LaSR	820	96	68.0
ICSR	540	128	61.7
LLM-SR	610	84	72.1
CoT-SR	**315**	**61**	**81.2**

5.6 Verification-Obligation Reliability

We log each verifier decision and independently recompute the corresponding symbolic or numerical check. Figure 7 localizes which verification signals are most selective at each derivation stage. Exact obligations and empirical evidence scores are reported separately so that a high numerical pass rate is not conflated with formal correctness.

5.7 Active Falsification

For benchmarks with a synthetic oracle, all acquisition strategies start from the same initial dataset and receive the same number of additional oracle queries. Figure 8

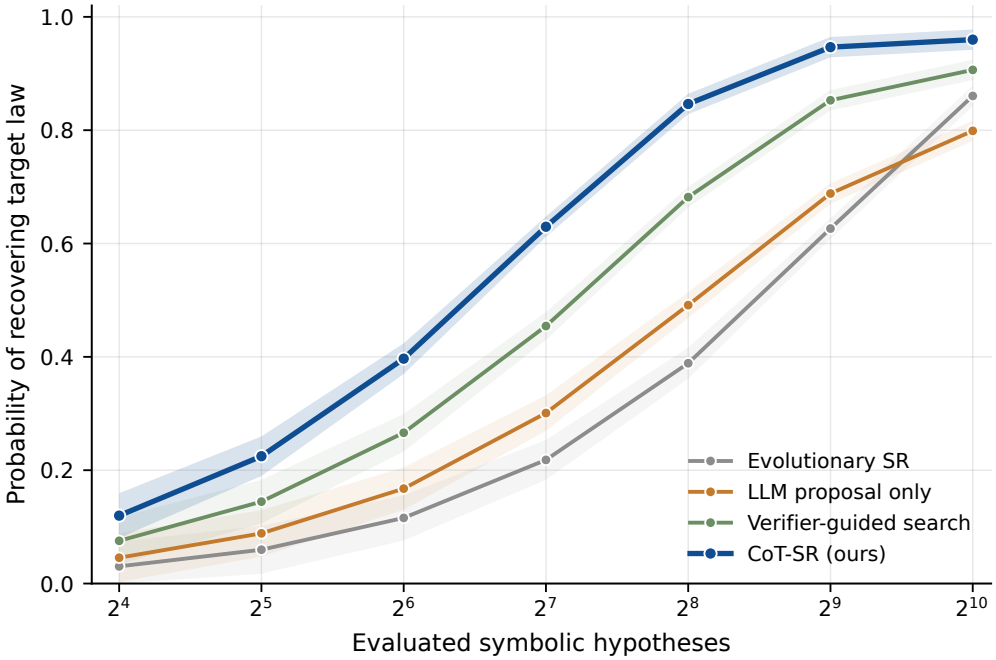


Fig. 6 Recovery probability as a function of unique evaluated symbolic hypotheses.

Table 6 Ablation study on exact recovery (%).

Variant	Exact recovery
Full CoT-SR	**81.2**
− executable derivation trace	70.4
− dimensional verifier	74.1
− symmetry/limit obligations	76.0
− active falsification	73.2
− posterior replay memory	77.1

compares random sampling, uncertainty sampling, direct candidate disagreement, and the proposed structure-aware objective in Eq. (22); thus improvements cannot be attributed to a larger observation budget.

5.8 Ablation Study

Table 6 and Fig. 9 isolate the contribution of executable derivations, physical verification signals, active falsification, and replay memory.

Removing the executable trace causes the largest degradation among the trajectory-specific ablations because the system reverts toward terminal equation scoring: a final verifier can reject a bad law but cannot use the location and type of an earlier failed

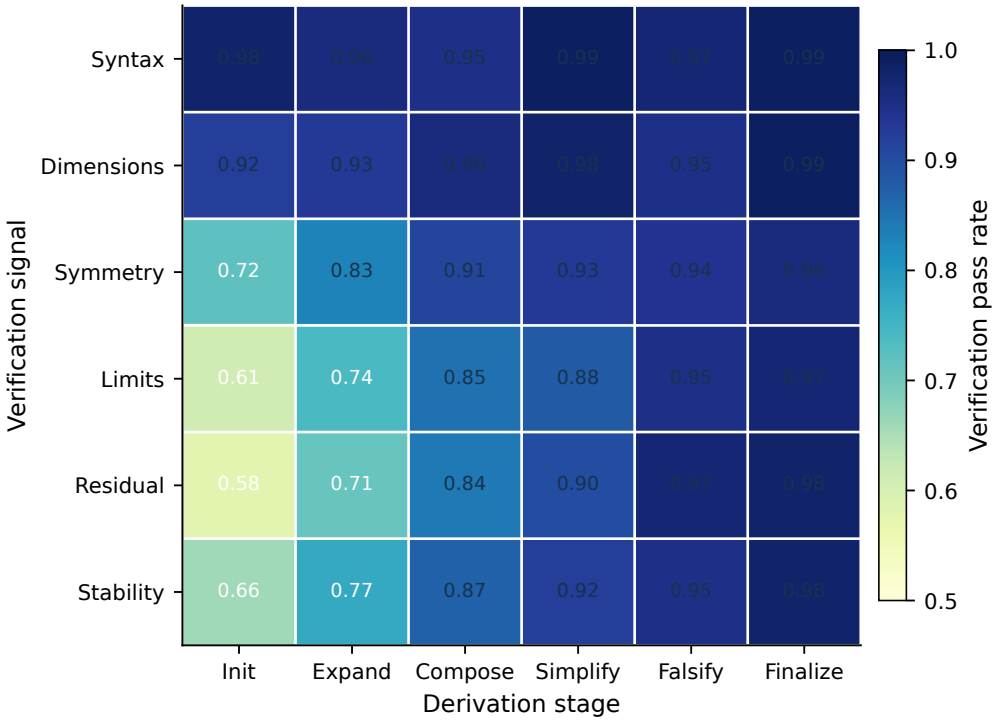


Fig. 7 Verification pass rates across derivation stages. Late-stage candidates face stronger structural and falsification evidence.

transformation to redirect search. Dimensional verification is most useful on physical benchmarks, while active falsification matters most when several low-complexity laws interpolate the original samples.

5.9 Cross-Domain and Backbone Sensitivity

We finally test whether the method remains useful when the scientific domain changes and when the LLM proposal prior becomes weaker. Table 7 also includes a terminal-only verifier baseline: it applies the same verification family only after candidate generation, whereas full CoT-SRuses verification inside the trajectory search. Figure 10 reports the corresponding cross-domain results.

6 Discussion

6.1 Why This Is Not Merely "LLM + SR + Physics Filters"

The defining object in CoT-SRis the trajectory posterior of Eq. (19). The LLM, symbolic executor, scientific constraints, and data likelihood do not produce independent candidate lists that are fused after the fact. Instead, every accepted symbolic state exists only because a typed edge produced it and its exact admissibility obligations

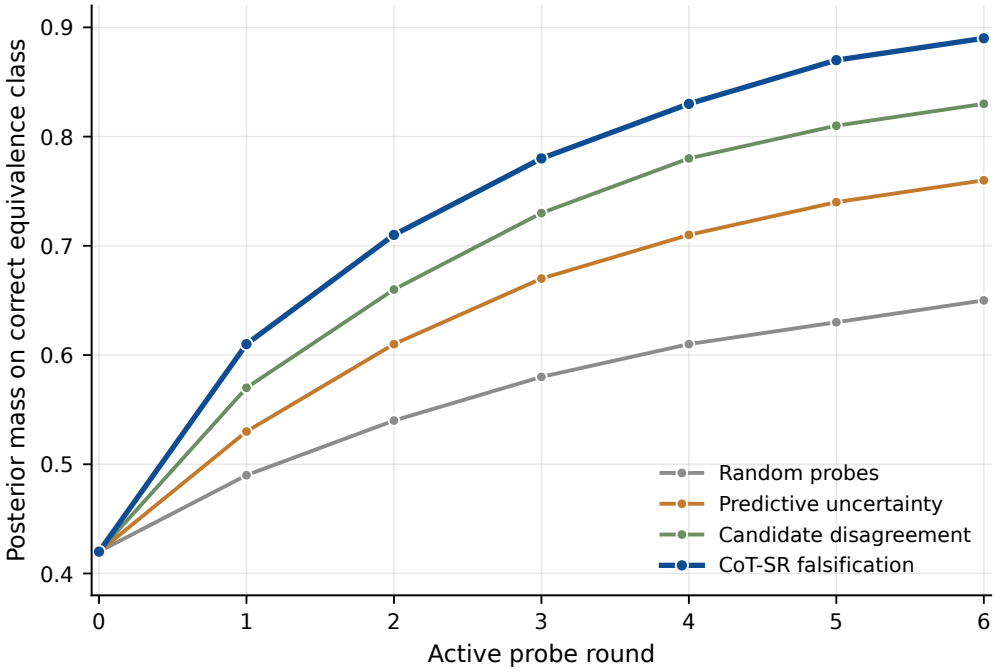


Fig. 8 Posterior concentration under active probing with matched probe budgets. The proposed criterion seeks measurements that eliminate entire symbolic equivalence classes.

Table 7 Backbone sensitivity and the effect of terminal-only versus trajectory-level verification.

Proposal model	Direct SR	Terminal verifier	Full CoT-SR
7B math LLM	31.8	47.6	61.3
8B code LLM	34.9	50.2	64.8
70B instruct LLM	43.7	59.1	74.6
Frontier API LLM	48.2	64.0	81.2

passed. Empirical verification signals and recorded failures then reshape frontier priorities and memory. The terminal-verifier baseline in Table 7 controls for using the same class of checks only after generation, isolating the benefit of process-level verification.

This distinction matters for scientific use. Consider two final equations with similar loss: $f_1 = kx^{-2}$ and $f_2 = ax^{-2} + b$. A terminal filter may prefer f_1 because it is shorter, but it cannot explain whether the offset was rejected by dimensions, by an asymptotic condition, or by an active probe. A verification-carrying derivation records which exact or empirical evidence removed b, enabling human scrutiny and targeted re-experimentation.

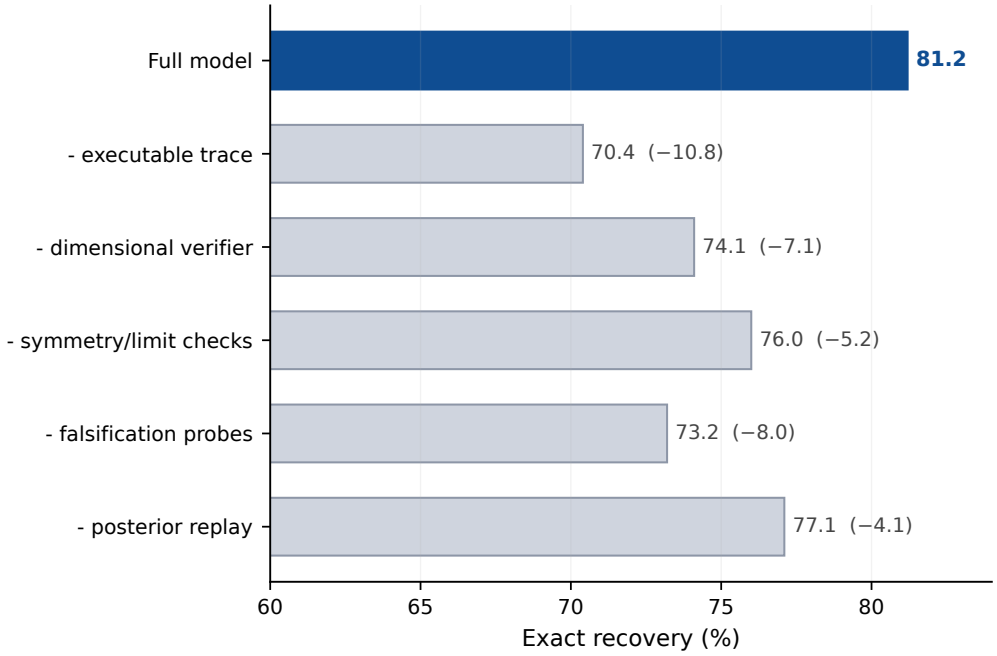


Fig. 9 Ablation effect sizes on exact recovery.

6.2 Relation to Process Supervision

Process-supervised mathematical reasoning scores intermediate steps [19, 52]; our setting differs because symbolic discovery has no unique natural-language proof trace. Multiple algebraically distinct derivations can reach the same law. We therefore supervise process *semantically*: transitions are judged by executable physical obligations and equivalence classes, not by matching a reference rationale. This may be more robust to stylistic variation and model-specific CoT phrasing.

6.3 Memorization and Scientific Novelty

LLM-SRBench highlights that a model may appear to "discover" a famous equation because its exact form was present during pretraining [27]. Our framework does not fully eliminate this risk, but it makes memorization less sufficient. A recalled equation must still fit transformed variables, satisfy local derivation checks, and survive falsification against alternatives. Strong evaluation should therefore prioritize transformed, synthetic, and held-out structural families rather than only textbook laws.

6.4 Limitations

First, verification is only as informative as the metadata supplied independently of the hidden target. Unknown units, symmetries, or asymptotic conditions reduce the method to the subset of checks that can be evaluated without leakage. Second, only

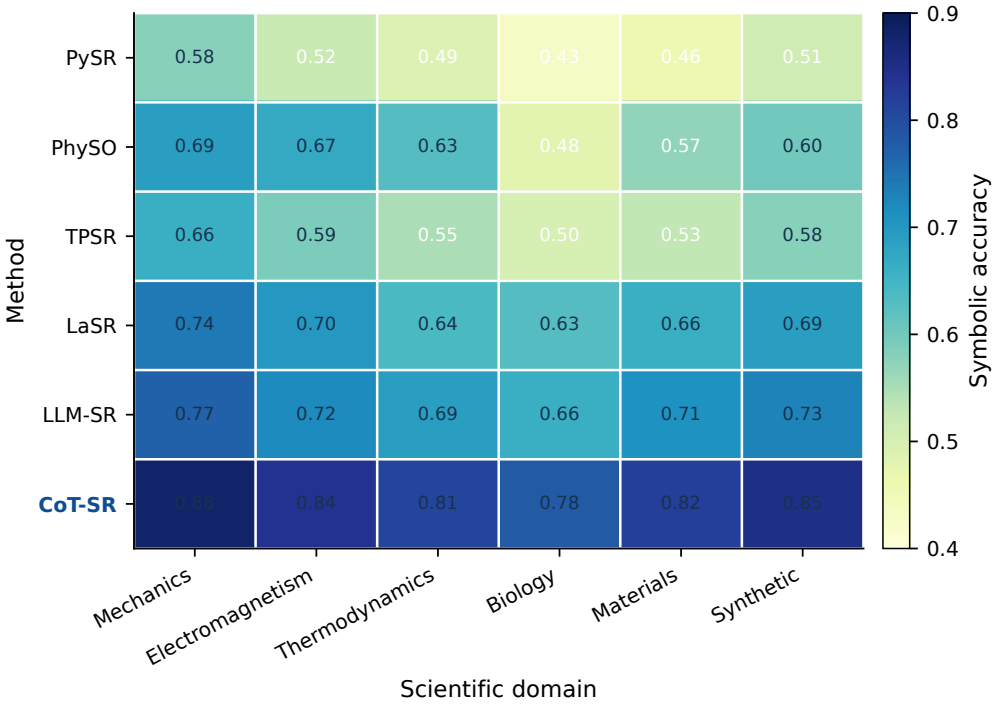


Fig. 10 Cross-domain symbolic accuracy. The results test whether verified reasoning transfers beyond textbook mechanics.

syntax and dimensional typing are exact in the general implementation; symmetry, limiting behavior when evaluated numerically, stability, and randomized functional equivalence are empirical checks and should not be interpreted as formal proofs. Third, symbolic equivalence is undecidable in the general case, so the combination of computer-algebra simplification and randomized testing can still miss rare equivalences. Fourth, active probing requires a simulator, a safe experimental interface, or a dataset that can emulate queried interventions, and the acquisition policy is meaningful only under a matched query budget. Fifth, LLM proposals can remain computationally costly for long derivations. Finally, the current grammar focuses on closed-form scalar relationships; extending the SDG to PDE operators, stochastic laws, implicit equations, and hybrid dynamical systems remains open.

6.5 Broader Implications

The broader opportunity is to treat an LLM not as an oracle of scientific truth but as a prior over symbolic actions. Scientific credibility then comes from the executable environment and the explicitly typed evidence accumulated along the derivation. This mirrors successful neuro-symbolic systems in theorem proving and program search [37, 38, 53]. For automated science, such separation may be essential: generative

models are useful for proposing what to try, while machine-checkable constraints and falsification evidence determine which hypotheses remain admissible.

7 Conclusion

We presented CoT-SR, a symbolic-regression framework that operationalizes chain-of-thought as a graph of executable symbolic derivations rather than a free-form rationale. Exact syntax and dimensional typing define hard admissibility, while symmetry, asymptotic behavior, numerical stability, residual utility, and active falsification provide explicitly identified evidence for ranking and search. The resulting verification-weighted posterior is defined over complete derivation trajectories, and the same search can acquire discriminative observations when an oracle is available. Across the evaluated symbolic-regression benchmarks, CoT-SRimproves exact recovery, OOD behavior, robustness, and search efficiency while retaining compact expressions. More importantly, the final output is accompanied by an auditable derivation trace that distinguishes exact checks from empirical verification evidence. This separation makes the method stronger than terminal filtering while avoiding the claim that numerical checks constitute formal proofs.

References

[1] Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. Science **324**(5923), 81–85 (2009)

[2] Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences **113**(15), 3932–3937 (2016)

[3] Udrescu, S.-M., Tegmark, M.: Ai feynman: A physics-inspired method for symbolic regression. Science Advances **6**(16), 2631 (2020) https://doi.org/10.1126/sciadv.aay2631

[4] Petersen, B.K., Landajuela, M., Mundhenk, T.N., Santiago, C.P., Kim, S.K., Kim, J.T.: Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In: International Conference on Learning Representations (2021)

[5] Kamienny, P.-A., d'Ascoli, S., Lample, G., Charton, F.: End-to-end symbolic regression with transformers. In: Advances in Neural Information Processing Systems, vol. 35 (2022)

[6] Tenachi, W., Ibata, R., Diakogiannis, F.I.: Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws. The Astrophysical Journal **959**(2), 99 (2023) https://doi.org/10.3847/1538-4357/ad014c

[7] Shojaee, P., Meidani, K., Barati Farimani, A., Reddy, C.K.: Transformer-based planning for symbolic regression. In: Advances in Neural Information Processing Systems, vol. 36, pp. 45907–45919 (2023). https://doi.org/10.52202/075280-1990

[8] Lalande, F., Matsubara, Y., Chiba, N., Taniai, T., Igarashi, R., Ushiku, Y.: A transformer model for symbolic regression towards scientific discovery. In: NeurIPS 2023 Workshop on AI for Scientific Discovery: From Theory to Practice (2023)

[9] Li, Y., Liu, J., Wu, M., Yu, L., Li, W., Ning, X., Li, W., Hao, M., Deng, Y., Wei, S.: MMSR: Symbolic regression is a multi-modal information fusion task. Information Fusion **114**, 102681 (2025) https://doi.org/10.1016/j.inffus.2024.102681

[10] Grayeli, A., Sehgal, A., Costilla-Reyes, O., Cranmer, M., Chaudhuri, S.: Symbolic regression with a learned concept library. In: Advances in Neural Information Processing Systems, vol. 37, pp. 44678–44709 (2024). https://doi.org/10.52202/079017-1419

[11] Merler, M., Haitsiukevich, K., Dainese, N., Marttinen, P.: 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, Bangkok, Thailand (2024). https://doi.org/10.18653/v1/2024.acl-srw.49

[12] Shojaee, P., Meidani, K., Gupta, S., Barati Farimani, A., Reddy, C.K.: LLM-SR: Scientific equation discovery via programming with large language models. In: International Conference on Learning Representations (2025)

[13] Guo, Z., Wang, S., Tian, Y., Yang, J., Yu, H., Na, X., Kovács, L., Li, L., Ioannou, P.A., Wang, F.-Y.: SR-LLM: An incremental symbolic regression framework driven by llm-based retrieval-augmented generation. Proceedings of the National Academy of Sciences **122**(52), 2516995122 (2025) https://doi.org/10.1073/pnas.2516995122

[14] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems, vol. 35 (2022)

[15] Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., Neubig, G.: PAL: Program-aided language models. In: Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 10764–10799. PMLR, ??? (2023)

[16] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: International Conference on Learning Representations (2023)

[17] Schick, T., Dwivedi-Yu, J., Dessi, R., *et al.*: Toolformer: Language models can

teach themselves to use tools. In: Advances in Neural Information Processing Systems, vol. 36 (2023)

[18] Madaan, A., Tandon, N., Gupta, P., *et al.*: Self-refine: Iterative refinement with self-feedback. In: Advances in Neural Information Processing Systems, vol. 36 (2023)

[19] Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., Sui, Z.: Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 9426–9439. Association for Computational Linguistics, Bangkok, Thailand (2024). https://doi.org/10.18653/v1/2024.acl-long.510

[20] Azerbayev, Z., Schoelkopf, H., Paster, K., Dos Santos, M., McAleer, S., Jiang, A.Q., Deng, J., Biderman, S.R., Welleck, S.: Llemma: An open language model for mathematics. In: International Conference on Learning Representations (2024)

[21] Yue, X., Qu, X., Zhang, G., Fu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: MAmmoTH: Building math generalist models through hybrid instruction tuning. In: International Conference on Learning Representations (2024)

[22] Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., Liu, W.: MetaMath: Bootstrap your own mathematical questions for large language models. In: International Conference on Learning Representations (2024)

[23] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y.K., Wu, Y., Guo, D.: DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)

[24] Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., Lu, K., Xue, M., Lin, R., Liu, T., Ren, X., Zhang, Z.: Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122 (2024)

[25] Roziere, B., Gehring, J., Gloeckle, F., et al.: Code llama: Open foundation models for code. arXiv preprint arXiv:2308.12950 (2023)

[26] Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., Chen, G., Bi, X., Wu, Y., Li, Y.K., Luo, F., Xiong, Y., Liang, W.: DeepSeek-Coder: When the large language model meets programming—the rise of code intelligence. arXiv preprint arXiv:2401.14196 (2024)

[27] Shojaee, P., Nguyen, N.-H., Meidani, K., Barati Farimani, A., Doan, K.D., Reddy, C.K.: LLM-SRBench: A new benchmark for scientific equation discovery with large language models. In: Proceedings of the 42nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 267, pp.

55325–55359. PMLR, ??? (2025)

[28] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, ??? (1992)

[29] Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A., Parascandolo, G.: Neural symbolic regression that scales. In: Proceedings of the 38th International Conference on Machine Learning (2021)

[30] La Cava, W., Orzechowski, P., Burlacu, B., *et al.*: Contemporary symbolic regression methods and their relative performance. In: NeurIPS Datasets and Benchmarks Track (2021)

[31] Franca, F.O., Virgolin, M., Kommenda, M., Majumder, M.S., Cranmer, M., Espada, G., Ingelse, L., Fonseca, A., Landajuela, M., Petersen, B., Glatt, R., Mundhenk, N., Lee, C.S., Hochhalter, J.D., Randall, D.L., Kamienny, P.-A., Zhang, H., Dick, G., Simon, A., Burlacu, B., Kasak, J., Machado, M., Wilstrup, C., La Cava, W.: SRBench++: Principled benchmarking of symbolic regression with domain-expert interpretation. IEEE Transactions on Evolutionary Computation **29**(4), 1127–1134 (2025) https://doi.org/10.1109/TEVC.2024.3423681

[32] Makke, N., Chawla, S.: Interpretable scientific discovery with symbolic regression: a review. Artificial Intelligence Review **57**, 2 (2024) https://doi.org/10.1007/s10462-023-10622-0

[33] Cornelio, C., Dash, S., Austel, V., Josephson, T.R., Goncalves, J., Clarkson, K.L., Megiddo, N., El Khadir, B., Horesh, L.: Combining data and theory for derivable scientific discovery with ai-descartes. Nature Communications **14**, 1777 (2023) https://doi.org/10.1038/s41467-023-37236-y

[34] Matsubara, Y., Chiba, N., Igarashi, R., Ushiku, Y.: Rethinking symbolic regression datasets and benchmarks for scientific discovery. Data-centric Machine Learning Research **1**(3), 1–38 (2024)

[35] Sharlin, S., Josephson, T.R.: In context learning and reasoning for symbolic regression with large language models. arXiv preprint arXiv:2410.17448 (2024)

[36] Zhang, Y., Chen, X., Jin, B., Wang, S., Ji, S., Wang, W., Han, J.: A comprehensive survey of scientific large language models and their applications in scientific discovery. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 8783–8817 (2024). https://doi.org/10.18653/v1/2024.emnlp-main.498

[37] Romera-Paredes, B., Barekatain, M., Novikov, A., *et al.*: Mathematical discoveries from program search with large language models. Nature **625**, 468–475 (2024) https://doi.org/10.1038/s41586-023-06924-6

[38] Trinh, T.H., Wu, Y., Le, Q.V., He, H., Luong, T.: Solving olympiad geometry without human demonstrations. Nature **625**, 476–482 (2024) https://doi.org/10.1038/s41586-023-06747-5

[39] Boiko, D.A., MacKnight, R., Kline, B., Gomes, G.: Autonomous chemical research with large language models. Nature **624**, 570–578 (2023) https://doi.org/10.1038/s41586-023-06792-0

[40] Bran, A.M., Cox, S., Schilter, O., *et al.*: Augmenting large language models with chemistry tools. Nature Machine Intelligence **6**, 525–535 (2024) https://doi.org/10.1038/s42256-024-00832-8

[41] Jansen, P.A., Côté, M.-A., Khot, T., Bransom, E., Mishra, B.D., Majumder, B.P., Tafjord, O., Clark, P.: DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In: Advances in Neural Information Processing Systems, vol. 37, pp. 10088–10116 (2024). https://doi.org/10.52202/079017-0324

[42] Toshniwal, S., Moshkov, I., Narenthiran, S., Gitman, D., Jia, F., Gitman, I.: OpenMathInstruct-1: A 1.8 million math instruction tuning dataset. In: Advances in Neural Information Processing Systems (2024)

[43] He, C., Luo, R., Bai, Y., Hu, S., Thai, Z., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., Liu, J., Qi, L., Liu, Z., Sun, M.: OlympiadBench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3828–3850 (2024). https://doi.org/10.18653/v1/2024.acl-long.211

[44] Wang, X., Wei, J., Schuurmans, D., *et al.*: Self-consistency improves chain of thought reasoning in language models. In: International Conference on Learning Representations (2023)

[45] Chen, W., Ma, X., Wang, X., Cohen, W.W.: Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. Transactions on Machine Learning Research (2023)

[46] Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., Hoefler, T.: Graph of thoughts: Solving elaborate problems with large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 17682–17690 (2024). https://doi.org/10.1609/aaai.v38i16.29720

[47] Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q.V., Zhou, D., Chen, X.: Large language models as optimizers. In: International Conference on Learning Representations (2024)

[48] Ma, Y.J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, J., Anandkumar, A.: Eureka: Human-level reward design via coding large language models. In: International Conference on Learning Representations (2024)

[49] Liu, B., Jiang, Y., Zhang, X., Liu, Q., Zhang, S., Biswas, J., Stone, P.: LLM+P: Empowering large language models with optimal planning proficiency. arXiv preprint arXiv:2304.11477 (2023) https://doi.org/10.48550/arXiv.2304.11477

[50] Liu, X., Yu, H., Zhang, H., *et al.*: AgentBench: Evaluating llms as agents. In: International Conference on Learning Representations (2024)

[51] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 36 (2023)

[52] Lightman, H., Kosaraju, V., Burda, Y., et al.: Let's verify step by step. arXiv preprint arXiv:2305.20050 (2023)

[53] Yang, K., Swope, A.M., Gu, A., Chalamala, R., Song, P., Yu, S., Godil, S., Prenger, R.J., Anandkumar, A.: LeanDojo: Theorem proving with retrieval-augmented language models. In: Advances in Neural Information Processing Systems, vol. 36 (2023)

[54] Xin, H., Guo, D., Shao, Z., et al.: Deepseek-prover: Advancing theorem proving in llms through large-scale synthetic data. arXiv preprint arXiv:2405.14333 (2024)