EQUATION DISCOVERY THROUGH COLLECTIVELY EVOLVING EFFECTIVE HYPOTHESIS GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deriving extrapolatable symbolic laws from empirical observations remains a central bottleneck in AI-driven scientific discovery. We propose collective reasoning intelligence for symbolic optimization (CRISO), a multi-agent framework for autonomous equation discovery without expert feedback, task-specific finetuning, or external scientific tools and knowledge bases. CRISO distills collectively selected best hypothesis into reusable scientific statements, accumulates them as shared context, and broadcasts to a population of reasoning agents, thereby implementing a collectively evolving symbolic regression. Consequently, CRISO evolves the hypothesis search toward promising regions without additional finetuning of the backbone LLM. Across ten nonlinear, stochastic, or previously uncharacterized scientific systems, CRISO outperformed or matched state-of-the-art combinatorial and recent LLM-based symbolic regression methods, including under out-of-distribution evaluation. On an in-house chemical reactor whose dynamics lie beyond the backbone LLM's prior knowledge, CRISO derived a symbolic equation with 38.02% lower generalization error than all baselines, demonstrating its scientific discovery capability beyond static LLM priors.

1 INTRODUCTION

Discovering compact mathematical relations from observations is a central goal of scientific modeling. Such relations expose how variables interact and provide hypotheses that can be inspected, analytically manipulated, and tested beyond the observed domain. Symbolic regression addresses this goal by jointly searching over equation structures and numerical coefficients (Dong & Zhong, 2025). However, the space of possible expressions grows combinatorially with the number of variables, operators, and their compositions. Effective discovery therefore depends on manually-selected operation sets, heuristic prior rules, or external scientific tools (Stephens, 2023; Wang et al., 2026b).

Large language models (LLMs) offer an alternative approach to equation discovery by incorporating mathematical and scientific prior knowledge. However, many existing methods rely on task-specific finetuning, expert feedback, or external resources, limiting their applicability in open-ended discovery settings (Wang et al., 2026a; Li et al., 2024; Pang et al., 2026). This has motivated inference-time multi-agent methods that invoke frozen LLMs only during symbolic regression and orchestrate their information exchange. LLM-SR represents an equation as an executable program and prompts the LLM to generate and revise programs conditioned on high-performing demonstrations (Shojaee et al., 2025a), whereas FunctionEvolve (FuncEv) represents equations as abstract syntax trees (ASTs) and prompts the LLM to modify candidate ASTs guided by the search trajectory (Xia et al., 2026). STRIDE combines program generation with AST-based evaluation, with inter-role propagation between its critic and executor agents (Su et al., 2026).

However, existing LLM-based methods, whether island-based or multi-agent face a central challenge in exchanging progress between agents. Because agents hold different hypotheses, each agent must decide both how to communicate its own progress and how to incorporate progress received from other agents into its ongoing hypothesis. Sharing a whole equation as a demonstration is a direct option. However, a demonstration mixes reusable relations with details that belong only to that candidate. The receiving agent must then decide which parts to keep, imposing a subsequent challenge to extract essential knowledge from source codes. Useful shared feedback should therefore expose essential variable–operator relations, because leaving each agent to analyze shared hypotheses and adapt them individually to its own equation is costly and risks biasing the shared knowledge toward that agent's current candidate.

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.

We propose *Collective Reasoning Intelligence for Symbolic Optimization* (CRISO), an evolutionary multi-agent symbolic regression in which the agent population collectively evolves not only its global hypotheses but also the local hypothesis generator of each agent, by distilling essential variable–operator relations from the population-level collective knowledge. Agents generate executable programs, whose equations are represented as ASTs and evaluated prediction error together with AST-induced equation complexity. At each cycle, CRISO selects the collective-best equation and uses a separate LLM analyst to distill its variable–operator relations into reusable scientific statements encoded in natural-language. This guidance is accumulated as collective knowledge in a shared global memory and shared across the population. At the next hypothesis generation stage, the accumulated collective knowledge is injected into each agent. Consequently, although the backbone LLM remains frozen, the distribution from which each agent samples its next hypothesis shifts continually toward regions of the hypothesis space where promising structures have been discovered anywhere in the population, yielding a collectively evolving hypothesis generator. CRISO implements collectively evolving symbolic regression without expert feedback, task-specific finetuning, or external scientific tools and knowledge bases.

Across ten scientific systems with deterministic, stochastic, and previously uncharacterized governing laws, CRISO autonomously recovered governing equations and matched or outperformed all evaluated baselines, including state-of-the-art LLM-based symbolic regression methods. In further out-of-distribution (OOD) evaluations, CRISO outperformed LLM-driven symbolic regression methods, and CRISO-derived equations reduced extrapolation error by up to six orders of magnitude compared with DNNs, while compressing 0.2–1 million parameters into only 5–40 equation coefficients. Notably, on our in-house chemical reactor benchmark, whose dynamics are not readily identifiable from backbone LLM priors alone, CRISO produced a compact equation with substantially lower generalization error than all baselines. The ablation study demonstrates the benefits of the dual-space strategy for hypothesis search and propagation of distilled collective knowledge.

2 Related Work

PySR leverages an evolutionary algorithm to optimize computational trees that represent the relations between the input and output variables (Cranmer, 2023). In addition to conventional search method, deep learning methods alternatively employed for symbolic regression based on expression-generation policy optimization (Mundhenk et al., 2021), transformer-based equation skeleton generation (Kamienny et al., 2022), and symbolic hypothesis generative models (Li et al., 2026). However, conventional methods generally obtain their search priors either from manually specified operators or from pre-defined resources. LaSR combines genetic programming with an LLM-based exploration mechanism. It abstracts promising equations and stores them in a concept library, and an LLM referring to this library replaces genetic programming operators with low probability (Grayeli et al., 2024). Because LaSR is fundamentally a framework that performs symbolic regression through genetic programming while invoking an LLM only as an auxiliary tool, it still inherits the dependence on a human-specified operator set that bounds the hypothesis space in advance. This prior selection problem in conventional methods naturally motivates LLM-based symbolic regression.

Recent methods leverage LLMs either as inference-time search operators or as task-adapted symbolic regressors. ICSR performs single-agent in-context refinement (Merler et al., 2024), MLLM-SR relies on multimodal instruction tuning and user feedback (Li et al., 2024), and Symbolic-R1 requires task-specific instruction and reinforcement fine-tuning (Hua et al., 2025). Several LLM-based methods instead perform inference-time evolutionary search without additional finetuning. PiT-PO combines LLM-based symbolic regression with reinforcement learning (Wang et al., 2026a). MOT-SR employs an LLM that invokes external data-analysis tools and passes the resulting guidance to a separate equation-generating LLM at every iteration, so its knowledge transfer depends on human-specified tools and on the full set of front-ranking candidates rather than on a single selected structure (Wang et al., 2026b). DE maintains a reflective memory in which an LLM periodically summarizes search trajectories, which are combined with tool-based residual diagnostics to produce adopted search strategy rather than accumulating the structure of the best hypothesis (Pang et al., 2026). However, they are not a direct baseline of this work because they rely on task-specific LLM finetuning or human-specified external tools which may limit their applicability to open-ended scientific problems for which appropriate tools are unavailable.

LLM-SR, FunctionEvolve (FuncEv), and STRIDE instead perform inference-time LLM-based evolutionary search without expert feedback, task-specific finetuning, or human-specified external tools (Shojaee et al., 2025a; Xia et al., 2026; Su et al., 2026). LLM-SR generates executable programs

Table 1: Comparison of LLM-based symbolic regression methods and CRISO.

Method	AST-Based Evaluation	Inter-Role Propagation	Hypothesis Distillation	Collective Evolution
LLM-SR (Shojaee et al., 2025a)	✗	✗	✗	✗
FuncEv (Xia et al., 2026)	✓	✓	✗	✗
STRIDE (Su et al., 2026)	✓	✓	✗	✗
CRISO (Ours)	✓	✓	✓	✓

and reuses high-performing programs as demonstrations for subsequent program-level revision, but measures complexity using code length, which remains sensitive to code-specific artifacts (Shojaee et al., 2025a). FuncEv instead employs AST as a representation of the equations and performs hypothesis generation on the AST-based representation (Xia et al., 2026). STRIDE combines program generation with AST-based evaluation, with inter-role propagation between its critic and executor agents (Su et al., 2026). However, none of the three methods propagates reusable collective knowledge across a population of search agents. In all three, what is transferred is bound to individual candidates or to the initial problem context, and no representation-independent knowledge extracted from the selected hypothesis is accumulated and shared across a population of search agents.

Table 1 compares CRISO with the baseline LLM symbolic regression methods. MOT-SR and DE are omitted because their search guidance depends on human-specified analysis tools and is therefore not comparable under the tool-free setting shared by the listed methods. AST-based evaluation denotes scoring hypotheses on their parsed tree structure rather than on raw program code. Inter-role propagation denotes transfer of search information between distinct LLM roles, such as a selector and a mutator or a critic and an executor. Hypothesis distillation denotes extracting reusable variable–operator relations from the globally selected hypothesis into reusable scientific statements. Collective evolution denotes accumulating such distilled knowledge in a shared context and propagating it to a population of agents, thereby adapting each agent's hypothesis generator. FuncEv and STRIDE propagate information between roles, but what they pass on is a particular candidate to expand or an instruction to repair it, rather than reusable scientific knowledge that other agents can adapt for collective hypothesis evolution. Together, hypothesis distillation and collective evolution across a population of agents realize collective evolutionary symbolic regression in CRISO.

3 METHOD

3.1 ASTS AS STRUCTURED REPRESENTATIONS OF SCIENTIFIC KNOWLEDGE

CRISO separately represents a scientific equation as an executable program in hypothesis generation and an AST hypothesis evaluation, thereby enabling hypothesis generation and evaluation in their respective specialized representation spaces. In particular, AST-based representation further provides a mathematically grounded proxy for the explainability of discovered equations by quantifying their logical complexity as the depth of corresponding ASTs. Formally, CRISO measures the explainability of the discovered equations as a value inversely proportional to the tree depth of its corresponding ASTs, thereby prioritizing equations with lower logical complexity as knowledge with high explainability. For a set of observations $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, we define a discovery score that measures both the prediction accuracy and explainability of the equations in AST space as:

$$f(A, \mathcal{D}) = -\left\{ \sum_{(\mathbf{x},y)\in\mathcal{D}} (y - A(\mathbf{x}))^2 + d + v \right\}, \tag{1}$$

where d is the depth of the AST A associated with the equation, and v denotes the number of free parameters in A. When candidate hypotheses exhibit large differences in prediction error, the prediction term dominates the discovery score. As the search reaches hypotheses with comparable prediction errors, the complexity penalties d and v become more influential in their relative ranking, favoring simpler equations. Importantly, the score is used to rank hypotheses within each problem; it is not intended to define a universally calibrated accuracy–complexity trade-off across datasets.

3.2 CRISO: COLLECTIVE REASONING INTELLIGENCE FOR SYMBOLIC OPTIMIZATION

Fig. 1 illustrates the overall symbolic regression process of CRISO. It orchestrates K reasoning agents to collectively evolve, critique, and consolidate their hypothetical equations across program

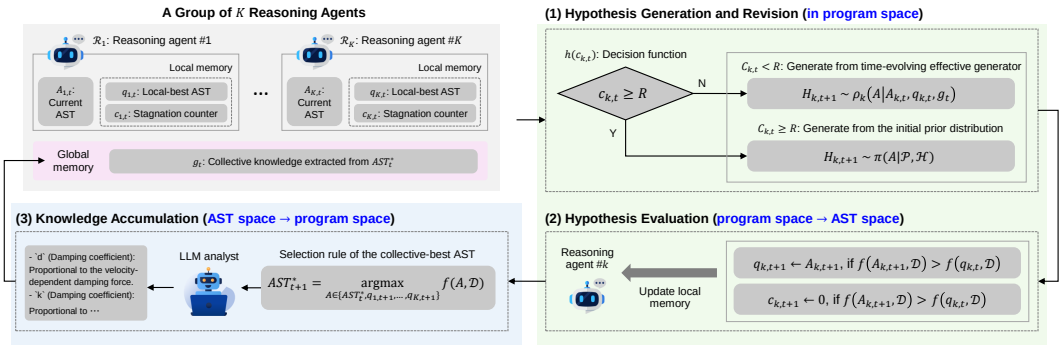

Figure 1: The overall process of the LLM-induced state transition from S_t to S_{t+1}. Steps (1) and (2), shown in green, are performed independently within each agent, whereas step (3), shown in blue, is performed globally using information aggregated from all agents.

and AST spaces. Briefly, each evolutionary cycle consists of three stages: (1) Each agent independently generates an equation hypothesis in the form of executable code in program space. (2) A parser converts each candidate program into its corresponding AST, and each agent's local memory is updated according to the discovery score in Eq. 1 in AST space. (3) An isolated LLM distills essential variable-operator relations of the globally best-performing AST into actionable scientific statements, which are accumulated in a shared global memory. In the next cycle, accumulated collective knowledge is incorporated into each agent's context to guide subsequent hypothesis generation.

3.2.1 Augmented States for Multi-Agent Symbolic Regression

The symbolic regression process of CRISO is represented as a stochastic decision process to maximize the discovery score in Eq. 1. Let $\mathcal{H}_t = (H_{1,t}, \cdots, H_{K,t})$ be a tuple of equation programs generated by K agents at time t. Subsequently, $\mathcal{Q}_t = (q_{1,t}, \cdots, q_{K,t})$ is defined as a tuple of the individual-best (local-best) ASTs up to t, and g_t is collective knowledge extracted from the best-performing AST across the agents at t. Additionally, we introduce $C_t = (c_{1,t}, \cdots, c_{K,t})$, where $c_{k,t} \in \mathbb{N}$ is the number of consecutive iterations in which the discovery score of the k-th agent has not increased. Across the symbolic regression process, $\mathcal{Q}_t$ and g_t serve as individual local memory and group-level shared memory, respectively. CRISO defines the symbolic regression as a stochastic process on a memory-augmented state $S_t = (\mathcal{H}_t, \mathcal{Q}_t, g_t, C_t)$.

3.2.2 Collectively Evolving Hypothesis Generation in Program Space

Given S_t, each agent generates or updates equation program $H_{k,t+1}$ through a mixture of context-induced distributions ρ_k and LLM's initial prior π as:

$$P(H_{k,t+1} = H|S_t) = (1 - h(c_{k,t}))\rho_k(H|H_{k,t}, q_{k,t}, g_t) + h(c_{k,t})\pi(H|\mathcal{P}), \tag{2}$$

where $h(c_{k,t}) = \mathbf{1}\{c_{k,t} \geq R\}$ is a function that determines whether the k-th agent generates an equation from a state-conditioned distribution $\rho_k(\cdot|H_{k,t}, q_{k,t}, g_t)$ or an LLM prior $\pi(\cdot|\mathcal{P})$ induced by an initial problem specification $\mathcal{P}$, and $R \in \mathbb{N}$ is a pre-defined maximum stagnation counter value. CRISO implements $\rho_k(\cdot|H_{k,t}, q_{k,t}, g_t)$ as an LLM-driven programming on the current program $H_{k,t}$ and reasoning context $(q_{k,t}, g_t)$. However, despite fixed backbone parameters, each agent's effective hypothesis generator evolves over time as its reasoning context $(q_{k,t}, g_t)$ is updated. Detailed specification of $\mathcal{P}$ and a prompt for operating $\rho_k(\cdot|H_{k,t}, q_{k,t}, g_t)$ is provided in Appendix C.

The hypothesis generation of the next $H_{k,t+1}$ in Eq. 2 adaptively performs exploitation and exploration through $\rho_k(\cdot|H_{k,t}, q_{k,t}, g_t)$ and $\pi(\cdot|\mathcal{P})$, respectively. When the k-th agent exhibits an improvement of the discovery score within the last R transitions, it performs exploitation through $\rho_k(\cdot|H_{k,t}, q_{k,t}, g_t)$ to reinforce its current hypothesis. Otherwise, it performs exploration to search unexplored regions of the hypothesis space through the initial prior $\pi(\cdot|\mathcal{P})$.

3.2.3 Hypothesis Evaluation and Local Memory Update in AST Space

After generating $H_{k,t+1}$, an AST parser calculates $A_{k,t+1}$ corresponding to $H_{k,t+1}$, i.e., the hypothesis in program space is projected into AST space for hypothesis evaluation. Then, the agent-specific

Table 2: Benchmark symbolic regression problems.

Domain	Problem Name (Target System)	Input Variables	Output Variable	Governing Equation
Physical science	**Chi2PDF** (Greenwood & Nikulin, 1996) (PDF of chi-squared distributions)	• x: Data point • k: Degree of freedom	Probability density	Eq. 6 in Appendix D
	NDO (Shojaee et al., 2025a) (Nonlinear damped oscillator)	• t: Time • x: Position • v: Velocity	Acceleration	Eq. 9 in Appendix D
	NNN (LeCun et al., 2015) (Nonlinear neural networks)	• x_1: 1st model input • x_2: 2nd model input	Model output	Eq. 10 in Appendix D
	MSB (Aakash et al., 2019) (Strain-stress relation in a metallic material)	• T: Temperature • ϵ: Strain	Mechanical stress	Unknown
Chemical science	**FHST** (Flory, 1942) (Flory–Huggins equation for polymer mixing)	• ϕ: Volume fraction • N: Chain length • χ: Interaction param.	Total change of Gibbs free energy	Eq. 7 in Appendix D
	BDC (Saha & Goebel, 2007) (Lithium-ion battery)	Six measurement parameters (cycle, voltage, etc.)	Discharge capacity	Unknown
	SFL (Agrawal et al., 2014) (Fatigue limit of alloy compositions)	Weight Percents of elements and tempering temperature	Fatigue limit	Unknown
	NOMC (Chemical reactor for non-oxidative methane conversion)	Six engineering parameters (temperature, pressure, etc.)	C_2 conversion yield	Unknown
Biology	**ECBG** (Shojaee et al., 2025a) (Differential equation for bacterial growth rate of Escherichia coli)	• b: Population density • s: Substrate conc. • T: Temperature • pH: pH level	Bacterial growth rate	Eq. 13 in Appendix D
	HHM (Hodgkin & Huxley, 1952) (Differential equation for membrane potential in Hodgkin-Huxley model)	Five biophysical parameters (membrane potential, gating probability, etc.)	Rate of potential change	Eq. 8 in Appendix D

local information $q_{k,t}$ and $c_{k,t}$ are updated based on the discovery score of $H_{k,t+1}$ as:

$$q_{k,t+1} = \begin{cases} A_{k,t+1} & \text{if } f(A_{k,t+1}, \mathcal{D}) > f(q_{k,t}, \mathcal{D}) \\ q_{k,t} & \text{otherwise} \end{cases}, \tag{3}$$

$$c_{k,t+1} = \begin{cases} 0 & \text{if } f(A_{k,t+1}, \mathcal{D}) > f(q_{k,t}, \mathcal{D}) \\ \min(c_{k,t}+1, R) & \text{otherwise} \end{cases}. \tag{4}$$

Therefore, each agent updates its local memory only when a newly generated hypothesis achieves a higher discovery score than its current local-best hypothesis.

3.2.4 Accumulation of Collective Knowledge for Program-Space Reasoning

After the evaluation and local memory update procedures, CRISO selects the current best-performing AST across all agents. Formally, the current best AST^*_{t+1} is selected as the AST that yields the highest discovery score up to t as:

$$AST^*_{t+1} = \underset{A \in \{AST^*_t, q_{1,t+1}, \cdots, q_{K,t+1}\}}{\arg\max} f(A, \mathcal{D}) \tag{5}$$

If no candidate achieves a discovery score strictly greater than $f(AST^*_t, \mathcal{D})$, the current AST^*_t is retained, thereby making the collective knowledge set absorbing. Then, an isolated LLM distills the essential variable–operator relations of AST^*_{t+1} into actionable scientific statements g_{t+1} encoded in natural language. Hence, essential discovery from AST^*_{t+1} is projected into program space. After generating g_{t+1}, CRISO accumulates it in the global memory shared by all agents. Consequently, CRISO generates a time-evolving context shared globally across all agents, yielding collectively evolving effective hypothesis generators $\rho_k(\cdot|H_{k,t+1}, q_{k,t+1}, g_{t+1})$.

4 Experiments

Benchmark Symbolic Regression Problems. For rigorous assessments in real-world scientific discovery, we conducted experimental evaluations on benchmark symbolic regression problems across physics, chemistry, and biology. These benchmark problems were selected to cover a broad range of functional forms and varying levels of uncertainty in governing equations. Table 2 summarizes the problem specifications of the ten benchmark problems. Detailed descriptions of each benchmark

Table 3: WMAPE values of CRISO and the competitor methods on the benchmark datasets. N/A indicates an execution failure caused by unresolved runtime errors of the original implementations.

Problem	PySR	GenSR	LLM-SR	FuncEv	STRIDE	CRISO
Chi2PDF	0.4587 (±0.0199)	0.1752 (±0.0113)	0.0289 (±0.0013)	0.0897 (±0.0061)	1.05E-5 (±4.48E-6)	**5.67E-6 (±1.87E-6)**
FHST	0.0783 (±0.0091)	0.0403 (±0.0012)	0.0147 (±0.0010)	0.0117 (±0.0008)	0.0375 (±0.0010)	**9.40E-8 (±1.62E-8)**
HHM	0.8869 (±0.0129)	0.1062 (±0.0069)	1.85E-6 (±2.13E-6)	0.0834 (±0.0051)	3.51E-7 (±1.26E-7)	**7.93E-8 (±4.79E-8)**
NDO	0.1713 (±0.1741)	0.0661 (±0.0036)	0.0289 (±0.0014)	0.0010 (±0.0002)	N/A	**6.92E-5 (±0.0001)**
NNN	N/A	0.0650 (±0.0041)	0.0050 (±0.0002)	0.0059 (±0.0003)	0.0030 (±0.0002)	**0.0016 (±0.0001)**
ECBG	1.4973 (±0.0001)	0.8028 (±0.0320)	0.3315 (±0.0133)	0.9499 (±0.0654)	0.4979 (±0.0358)	**0.0742 (±0.0079)**
MSB	0.1613 (±0.0101)	0.0449 (±0.0058)	0.0418 (±0.0018)	0.0402 (±0.0013)	0.0887 (±0.0049)	**0.0299 (±0.0028)**
BDC	0.0149 (±0.0035)	0.0143 (±0.0023)	0.0128 (±0.0003)	**0.0098 (±0.0004)**	0.0124 (±0.0003)	**0.0092 (±0.0011)**
SFL	0.1179 (±0.0016)	0.0585 (±0.0045)	0.1078 (±0.0050)	0.0545 (±0.0022)	0.1925 (±0.0046)	**0.0431 (±0.0011)**
NOMC	0.2031 (±0.0001)	0.1269 (±0.0158)	0.1409 (±0.0081)	0.1252 (±0.0135)	0.1735 (±0.0156)	**0.0776 (±0.0154)**

Table 4: L2RE values of CRISO and the competitor methods.

Problem	PySR	GenSR	LLM-SR	FuncEv	STRIDE	CRISO
Chi2PDF	0.2934 (±4.55E-5)	0.0998 (±0.0072)	0.0185 (±0.0005)	0.0565 (±0.0014)	1.04E-5 (±3.26E-6)	**1.44E-7 (±2.46E-7)**
FHST	0.0983 (±0.0080)	0.0505 (±0.0021)	0.0095 (±0.0008)	0.0148 (±0.0009)	0.0522 (±0.0026)	**1.29E-7 (±2.22E-7)**
HHM	0.5847 (±0.0692)	0.0714 (±0.0105)	2.42E-7 (±1.56E-8)	0.0823 (±0.0050)	1.28E-7 (±8.69E-8)	**4.89E-8 (±2.32E-8)**
NDO	0.0864 (±0.0315)	0.0728 (±0.0063)	0.0285 (±0.0010)	0.0011 (±0.0001)	N/A	**8.18E-5 (±0.0001)**
NNN	N/A	0.0771 (±0.0122)	0.0052 (±0.0018)	0.0071 (±0.0002)	0.0036 (±0.0002)	**0.0019 (±9.35E-5)**
ECBG	0.7573 (±0.0008)	0.4034 (±0.0519)	0.3320 (±0.0138)	0.5180 (±0.0356)	0.4784 (±0.0131)	**0.1065 (±0.0895)**
MSB	0.2020 (±0.0062)	0.0734 (±0.0249)	0.0608 (±0.0018)	0.0619 (±0.0043)	0.1016 (±0.0069)	**0.0535 (±0.0028)**
BDC	0.0201 (±0.0038)	0.0182 (±0.0015)	0.0163 (±0.0019)	**0.0128 (±0.0008)**	0.0175 (±0.0012)	**0.0136 (±0.0026)**
SFL	0.1522 (±0.0070)	0.0788 (±0.0029)	0.1071 (±0.0042)	0.0715 (±0.0032)	0.2248 (±0.0083)	**0.0585 (±0.0014)**
NOMC	0.2626 (±7.13E-7)	0.3769 (±0.0176)	0.1755 (±0.0073)	0.1730 (±0.0202)	0.2188 (±0.0288)	**0.1165 (±0.0215)**

and its governing equation are provided in Appendix D. A well-known LLM-SRBench primarily isolates memorization and data-driven reasoning through transformed known equations and synthetically constructed discovery tasks (Shojaee et al., 2025b). Our evaluation instead focuses on equation discovery in physically grounded systems under different levels of backbone LLM's prior knowledge. The benchmark problems were categorized into the systems within LLM prior knowledge (Chi2PDF, FHST, HHM), with unknown or randomly generated coefficients (NDO, NNN, ECBG), or with previously uncharacterized forward mechanism (MSB, BDC, SFL, NOMC).

Baselines. We compared CRISO with representative baselines spanning three symbolic regression paradigms: (1) PySR (Cranmer, 2023) from symbolic combinatorial optimization; (2) GenSR (Li et al., 2026) from equation generative models; (3) LLM-SR (Shojaee et al., 2025a), FuncEv (Xia et al., 2026), and STRIDE (Su et al., 2026) from LLM-based symbolic regression. Equation coefficients of LLM-SR were optimized by BFGS. However, three relevant methods, PiT-PO, MOT-SR, and DE, were not employed as direct baselines due to their requirements on additional LLM finetuning or manually-specified external tools. To ensure a fair comparison on experimental evaluations, all LLM-based methods used the same backbone LLM, Mixtral:8x7b (Jiang et al., 2024), under comparable wall-clock time budgets, as detailed in Appendix B. In addition to existing baselines, we implemented three ablation variants CRISO$_{ind}$, CRISO$_{txt}$, and CRISO$_{eq}$ for an ablation study in Section 4.4. CRISO$_{ind}$ runs K independent agents and returns the highest-scoring local-best equation. CRISO$_{txt}$ disables the AST-depth-based guidance Eq. 1, whereas CRISOeq disables the distillation of AST_t^* into actionable scientific statements. Implementation details and hyperparameter setting are provided in Appendix B. Evaluation results of CRISO with a different backbone LLM are reported in Appendix G. All source code of this work are publicly available at `https://anonymous.4open.science/r/criso-8DE7`.

Evaluation. The generalization error of the discovered equations is quantified by a weighted mean absolute percentage error (WMAPE) (De Myttenaere et al., 2016) and a standard L_2 relative error (L2RE) (Messenger & Bortz, 2021). We primarily use WMAPE, because it directly calculates a scaled percentage error of the discovered equations. All experiments were independently repeated five times while keeping the training and evaluation datasets and inference budget fixed.

4.1 Symbolic Regression Accuracy

Quantitative Results. Tables 3 and 4 exhibit the measured WMAPE and L2RE values on the benchmark problems, respectively. GenSR substantially reduced WMAPE for all benchmark problems over PySR. LLM-SR and FuncEv further improved the symbolic regression accuracy, but their WMAPE values still exceeded 0.1 on the ECBG and NOMC problems, which contain randomized components and experimental uncertainty, respectively. STRIDE also exhibited high WMAPE val-

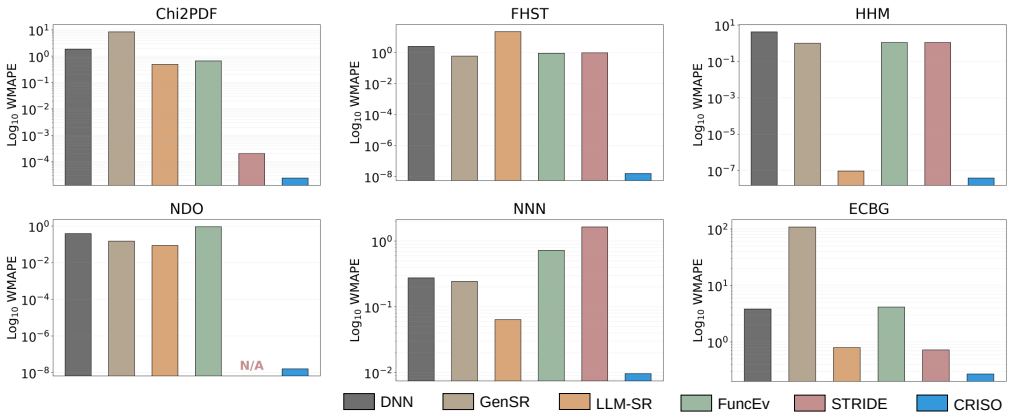

Figure 2: WMAPE values of DNN and symbolic regression methods under OOD conditions.

ues than those of CRISO for all benchmark problems. In contrast, CRISO achieved WMAPE values below 0.1 and outperformed or matched all state-of-the-art baselines. In the results on L2RE, CRISO exhibited state-of-the-art symbolic regression accuracy on nine benchmark problems.

Equation Discovery Beyond Static LLM Prior. For the benchmark systems with uncharacterized forward mechanisms with experimental uncertainty (MSB, BDC, SFL, and NOMC), CRISO also showed the lowest WMAPE values. The results on the NOMC problem are especially noteworthy because the target scientific system is an in-house chemical reactor with extensive chemical uncertainty. LLM-SR, FuncEv, and STRIDE showed WMAPE values of 0.1409, 0.1252, and 0.1735 on the NOMC problem, respectively. However, CRISO achieved a substantially lower WMAPE value of 0.0776 and reduced 38.02% generalization error compared to the WMAPE value of FuncEv. This result demonstrates the potential of CRISO for previously unexplored scientific problems.

4.2 Out-of-Distribution Generalization

To assess extrapolation beyond the observed input ranges, we generated strictly isolated ground-truth OOD samples for six benchmark problems with known governing equations: Chi2PDF, FHST, HHM, NDO, NNN, and ECBG. We constructed OOD inputs outside the ranges used for equation discovery and computed their target outputs using the true governing equations and coefficients as in the training data. Thus, the input distribution severely changes while the underlying input–output relation remains unchanged. We evaluate each discovered equation with its structure and fitted coefficients fixed, without using OOD samples for equation selection or parameter fitting.

Fig. 2 compares the OOD WMAPE of CRISO with those of GenSR, LLM-SR, FuncEv, STRIDE, and DNNs with 0.2–1 million learnable parameters. Although the competing symbolic regression methods often achieved low in-distribution errors, their OOD performance varied substantially across problems. In particular, their larger errors on FHST, whose governing relation is a well-known scientific equation, show that access to relevant LLM priors does not necessarily lead to the recovery of the underlying governing equation. In this experiment, CRISO achieved low OOD errors across all six problems, including NDO, NNN, and ECBG, which contain randomized components. These results suggest that CRISO recovers or approximates relations that remain predictive beyond the observed input ranges, rather than only fitting the training domain. Appendix F further illustrates this behavior by plotting prediction errors inside and outside the in-distribution ranges.

4.3 Qualitative Analysis on Discovered Equations

Based on Eq. 1, CRISO prioritizes simpler equations with simpler mathematical structures and fewer free parameters. We qualitatively assessed the explainability and scientific usability of the discovered equations by comparing them with their ground-truth counterparts. This qualitative analysis covered three systems with distinct discovery conditions: (1) Chi2PDF within the prior knowledge of the backbone LLM; (2) ECBG with randomized coefficients and mathematical terms; and (3) BDC under previously uncharacterized equation skeletons and experimental uncertainty.

Fig. 3 shows the true and discovered equations on the three benchmark problems. For the Chi2PDF problem, CRISO recovered the defining analytical structure of the true probability density, whereas

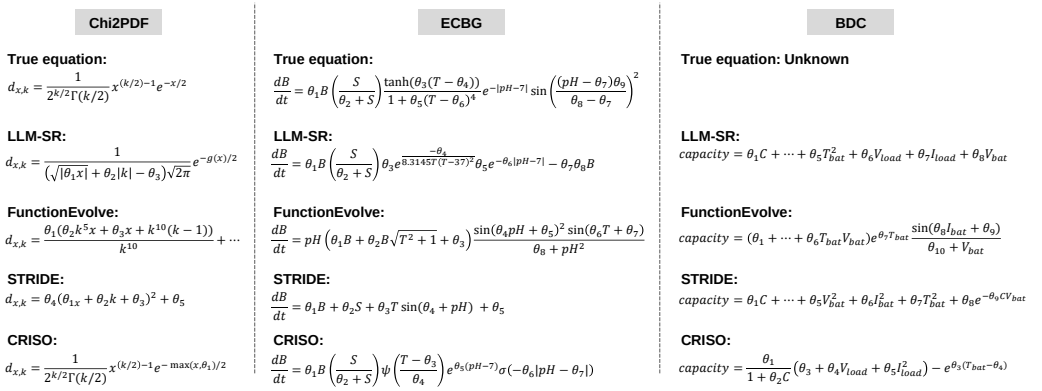

Figure 3: Qualitative evaluation of discovered equations on Chi2PDF, ECBG, and BDC problems.

Table 5: WMAPE values of CRISO and its ablation variants CRISO$_{ind}$, CRISO$_{txt}$, and CRISO$_{eq}$.

Method	NDO	NNN	ECBG	MSB	BDC	SFL	NOMC
CRISO$_{ind}$	0.0432 (±0.0055)	0.0324 (±0.0052)	0.2732 (±0.0122)	0.0438 (±0.0043)	0.0108 (±0.0033)	0.0732 (±0.0052)	0.1538 (±0.0120)
CRISO$_{eq}$	0.0556 (±0.0062)	0.0062 (±0.0004)	0.1117 (±0.0128)	0.0429 (±0.0053)	0.0114 (±0.0042)	0.0713 (±0.0025)	0.0842 (±0.0158)
CRISO$_{txt}$	0.0040 (±0.0031)	0.0043 (±0.0015)	0.1357 (±0.0157)	0.0323 (±0.0047)	0.0238 (±0.0076)	0.8278 (±0.2513)	1.3671 (±0.0356)
CRISO	6.92E-5 (±0.0001)	0.0016 (±0.0001)	0.0742 (±0.0079)	0.0299 (±0.0028)	0.0092 (±0.0011)	0.0431 (±0.0011)	0.0776 (±0.0154)

the competing methods omitted essential components such as $\Gamma(k/2)$ and $x^{(k/2)-1}$ or replaced the true structure with overfitted polynomial approximations. For the ECBG problem, CRISO preserved the multiplicative structure among biomass growth and substrate saturation effects. CRISO approximated the complex dynamics induced by $\tanh$ as a single smooth threshold behavior derived from cumulative standard normal distribution ψ within the observed data range. Additionally, it approximated the $\sin$-induced transition as a logistic term σ. For the BDC problem under unknown true equation, CRISO nevertheless produced a concise equation that separates cycle-dependent capacity decay and current effects under temperature-dependent degradation. However, LLM-SR and STRIDE mainly returned additive polynomial expressions. FuncEv produced a more deeply nested expression with oscillatory and exponential terms. This qualitative analysis shows that CRISO performed deeper symbolic reasoning to discover physically-grounded equations of each system. This capability underlies the substantially improved performances of CRISO in the OOD evaluations.

4.4 Ablation Study

Because the accuracy on the in-prior problems are mainly dependent on backbone LLM's prior knowledge rather than the framework's reasoning capability, we focused on ablation study for the problems with randomized components or previously uncharacterized forward mechanisms. Table 5 presents WMAPE values of the three ablation variants (CRISO$_{ind}$, CRISO$_{txt}$, and CRISO$_{eq}$).

Actionable Collective Intelligence. CRISO$_{ind}$ replaces collective knowledge sharing in CRISO with independent multi-agent search. CRISO$_{eq}$ retains the architecture of CRISO but disables the distillation process of the accumulated collective knowledge into actionable scientific statements. All variants used the same backbone LLM and total reasoning budget. CRISO$_{ind}$ showed substantially higher WMAPE values, demonstrating the benefit of collective knowledge sharing beyond independent parallel search. CRISO$_{eq}$ also showed substantially degraded symbolic regression performance, even though the group-level best-performing knowledge is accumulated through the same collective procedure. This result on CRISO$_{eq}$ directly demonstrates the importance of the distillation of the best-performing knowledge into actionable scientific statements. However, these ablations do not isolate the independent contribution of long-term knowledge accumulation, since previously shared guidance can already be embedded in subsequent hypotheses. Removing earlier guidance from the explicit context therefore does not cleanly eliminate its persistent influence.

Task-Specialized Working Spaces. CRISO separately assigns hypothesis generation to program space and hypothesis evaluation to AST space. It then translates the best-performing knowledge selected in AST space into actionable knowledge, which is incorporated into each agent's context for subsequent hypothesis generation in program space. This bidirectional transfer is a key feature that distinguishes CRISO from LLM-SR and FuncEv. CRISO$_{txt}$ retains the overall architecture and reasoning budget of CRISO but performs both hypothesis generation and complexity evaluation in program space by defining the complexity measure as a normalized code length. As shown in Table

Table 6: Comparison of AST depths between the true and discovered equations.

Method	Chi2PDF	FHST	HHM	NDO	NNN	ECBG	MSB	BDC	SFL	NOMC
True equation	10	8	13	12	9	10	Unknown	Unknown	Unknown	Unknown
LLM-SR	10	15	13	13	18	11	17	10	25	21
FuncEv	17	12	16	14	17	16	13	15	22	26
STRIDE	9	13	12	N/A	13	11	10	15	16	13
CRISO	10	9	8	11	9	9	15	8	15	20

Table 7: WMAPE values of STRIDE and CRISO for different backbone LLMs on the problems with randomized components or previously uncharacterized forward mechanisms.

Dataset	Llama3.1 8B		Mixtral 8x7B		GPT-5.6 Luna	
	STRIDE	CRISO	STRIDE	CRISO	STRIDE	CRISO
NDO	N/A	3.71E-6 (±2.32E-6)	N/A	6.92E-5 (±0.0001)	N/A	5.58E-6 (±2.58E-6)
NNN	0.0136 (±0.0053)	0.0023 (±0.0007)	0.0030 (±0.0002)	0.0016 (±0.0001)	0.0048 (±0.0009)	0.0017 (±0.0002)
ECBG	1.6835 (±0.0693)	1.0336 (±0.0436)	0.4979 (±0.0358)	0.0742 (±0.0079)	0.8790 (±0.1711)	0.1058 (±0.0102)
MSB	0.1085 (±0.0082)	0.0407 (±0.0104)	0.0887 (±0.0049)	0.0299 (±0.0028)	0.0424 (±0.0036)	0.0688 (±0.0046)
BDC	0.0144 (±0.0068)	0.0128 (±0.0039)	0.0124 (±0.0003)	0.0092 (±0.0011)	0.0199 (±0.0122)	0.0125 (±0.0019)
SFL	0.1203 (±0.0161)	0.0483 (±0.0073)	0.1925 (±0.0046)	0.0431 (±0.0011)	0.0847 (±0.0156)	0.0428 (±0.0022)
NOMC	0.1885 (±0.0113)	0.1008 (±0.0153)	0.1735 (±0.0156)	0.0776 (±0.0154)	0.1801 (±0.0079)	0.0836 (±0.0085)

5, CRISO consistently outperformed CRISO$_{txt}$ across all benchmark problems, demonstrating the benefit of assigning hypothesis generation and evaluation to their respective specialized spaces. In particular, CRISO$_{txt}$ showed severe performance degradation on the SFL and NOMC problems. These experimental results show the importance of separating the hypothesis and evaluation spaces to leverage both LLM's flexible programming capability and structure-guided symbolic regression.

4.5 AST Depths of Discovered Equations

As described in Eq. 1, we quantified the explainability of an equation as an inversely proportional value of its corresponding AST. Table 6 compares the AST depths of the equations discovered by LLM-based baselines and CRISO. The AST depths of the equations discovered by CRISO were close to those of the ground-truth equations for five of the six problems with known governing equations. LLM-SR and FuncEv generally produced equations with greater AST depths than the corresponding true equations. STRIDE produced shallower ASTs than CRISO on some problems, but exhibited substantially larger prediction errors, indicating its underfitting. This result shows that minimizing AST depth alone can produce simple but inaccurate equations. CRISO maintained AST depths close to those of the true equations while achieving low prediction errors.

4.6 Impact of Backbone LLM Capability

To evaluate CRISO across different backbone LLMs, we evaluated CRISO with three backbone LLMs of increasing model capability: Llama3.1 8B, Mixtral 8x7B, and GPT-5.6 Luna. We also evaluated STRIDE, which is a state-of-the-art method implementing program-based equation generation, AST-based evaluation, and inter-role propagation. Table 7 reports WMAPE values on the seven problems with randomized components or previously uncharacterized mechanisms. First, STRIDE consistently failed on the NDO problem regardless of its backbone LLMs, indicating an inherent limitation of its framework, whereas CRISO successfully executed for all problems and backbone LLMs. Second, CRISO with Mixtral 8x7B consistently outperformed STRIDE with a stronger backbone LLM (GPT-5.6 Luna), and this result shows that the improvements by CRISO stem primarily from its collective framework rather than from a combination with a specific backbone LLM. Third, CRISO with Mixtral 8x7B performed comparably to CRISO with GPT-5.6 Luna on the NOMC problem, an in-house reactor system with no established theoretical model. This result indicates that a stronger backbone alone does not guarantee better equation discovery on unresolved scientific problems, and the collective reasoning and knowledge distillation remain essential.

5 Conclusion

This paper proposed CRISO for fully autonomous governing equation discovery. CRISO assigns hypothesis generation and evaluation to their respective specialized spaces. Then, CRISO distills the best-performing AST into actionable scientific statements and accumulates them as shared collective knowledge, thereby continuously adapting each agent's effective hypothesis generator and yielding a time-evolving search context. CRISO achieved state-of-the-art accuracy on nine of ten systems. Extensive ablations validated the dual-space strategy and collective hypothesis context evolution of CRISO, demonstrating its potential for scientific discovery.

ETHICS STATEMENT

This work does not involve human subjects, sensitive personal data, or other ethical issues requiring specific consideration.

REPRODUCIBILITY STATEMENT

Source code for CRISO, including the framework, hyperparameter settings, and experiment scripts, is available at `https://anonymous.4open.science/r/criso-8DE7`. Implementation details are provided in Appendix B. Specifications of the LLM prompts for executing CRISO are presented in Appendix C.

AI USE STATEMENT

In this work, we used generative AI tools for revising the manuscript, limited to improving the tone, grammar, and readability of author-written text. We have not used generative AI tools for any other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we did not use generative AI tools for any other tasks with recommended disclosure. We have reviewed all AI-assisted work and verified that the revisions preserved the intended scientific meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

BS Aakash, JohnPatrick Connors, and Michael D Shields. Stress-strain data for aluminum 6061-t651 from 9 lots at 6 temperatures under uniaxial and plane strain tension. *Data in brief*, 25: 104085, 2019.

Ankit Agrawal, Parijat D Deshpande, Ahmet Cecen, Gautham P Basavarsu, Alok N Choudhary, and Surya R Kalidindi. Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. *Integrating materials and manufacturing innovation*, 3 (1):90–108, 2014.

Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint arXiv:2305.01582*, 2023.

Arnaud De Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016.

Junlan Dong and Jinghui Zhong. Recent advances in symbolic regression. *ACM Computing Surveys*, 57(11):1–37, 2025.

Paul J Flory. Thermodynamics of high polymer solutions. *The Journal of chemical physics*, 10(1): 51–61, 1942.

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/file/4ec3ddc465c6d650c9c419fb91f1c00a-Paper-Conference.pdf`.

Priscilla E Greenwood and Michael S Nikulin. *A guide to chi-squared testing*. John Wiley & Sons, 1996.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

Yingfan Hua, Ruikun Li, Jun Yao, Guohang Zhuang, Shixiang Tang, Bin Liu, Wanli Ouyang, and Yan Lu. Finetuning large language model as an effective symbolic regressor. *arXiv preprint arXiv:2508.09897*, 2025. doi: 10.48550/arXiv.2508.09897. URL `https://doi.org/10.48550/arXiv.2508.09897`.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Ignacio Julian, Heidy Ramirez, Jose L Hueso, Reyes Mallada, and Jesus Santamaria. Non-oxidative methane conversion in microwave-assisted structured reactors. *Chemical Engineering Journal*, 377:119764, 2019.

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

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Qian Li, Yuxiao Hu, Juncheng Liu, and Yuntian Chen. GenSR: Symbolic regression based on equation generative space. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yanjie Li, Weijun Li, Lina Yu, Min Wu, Jingyi Liu, Wenqiang Li, Shu Wei, and Yusong Deng. MLLM-SR: Conversational symbolic regression base multi-modal large language models. *CoRR*, abs/2406.05410, 2024. doi: 10.48550/arXiv.2406.05410. URL https://doi.org/10.48550/arXiv.2406.05410.

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, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-srw.49. URL https://aclanthology.org/2024.acl-srw.49/.

Daniel A Messenger and David M Bortz. Weak sindy for partial differential equations. *Journal of computational physics*, 443:110525, 2021.

Terrell Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P Santiago, Brenden K Petersen, et al. Symbolic regression via deep reinforcement learning enhanced genetic programming seeding. *Advances in Neural Information Processing Systems*, 34:24912–24923, 2021.

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 *Forty-third International Conference on Machine Learning*, 2026.

Bhaskar Saha and Kai Goebel. Battery data set. *NASA AMES prognostics data repository*, 2007.

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. *arXiv preprint arXiv:2504.10415*, 2025b.

Trevor Stephens. gplearn documentation. *Release 0.4*, 2, 2023.

Jiarui Su, Songjun Tu, Bei Sun, and Xiaojun Liang. Stride: A self-reflective agent framework for reliable automatic equation discovery. *arXiv preprint arXiv:2605.17790*, 2026.

Boxiao Wang, Kai Li, Tianyi Liu, Chen Li, Junzhe Wang, Yifan Zhang, and Jian Cheng. Llm-based scientific equation discovery via physics-informed token-regularized policy optimization. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 12195–12206, 2026a.

Boxiao Wang, Runxiang Wang, Kai Li, Chongming Li, Zhiwei Chen, Yifan Zhang, and Jian Cheng. Mot-sr: Multi-objective tool-augmented scientific equation discovery with large language models. *arXiv preprint arXiv:2607.29561*, 2026b.

Zeyu Xia, Jun Zhu, and Dong Yan. Functionevolve: Structure-guided symbolic regression with llms, 2026. URL https://arxiv.org/abs/2606.07704.

A FORMAL DESCRIPTION OF CRISO

Algorithm 1 describes the overall symbolic regression process of CRISO through the decision processes over LLM-induced augmented states and transitions.

Algorithm 1: The overall symbolic regression process of CRISO

Input : $\mathcal{P}$: Problem specification, $\mathcal{D}$: A set of observations,
$K \in \mathbb{N}$: Number of reasoning agents, $L \in \mathbb{N}$: Chain length (Maximum process time).
Output: AST_L^*: group-level best AST (or equation) at the final iteration.

$\mathcal{R} = \{\mathcal{R}_1, \cdots, \mathcal{R}_K\}$ // A group of K reasoning agents
// Initialization
for $k = 1; k \leq |\mathcal{R}|; k = k + 1$ **do**
  $H_{k,0} \sim \pi_k(\cdot|\mathcal{P})$
  $q_{k,0} = H_{k,0}$
  $c_{k,0} = 0$
end
// Symbolic regression process
for $t = 0; t < L; t = t + 1$ **do**
  for $k = 1; k \leq |\mathcal{R}|; k = k + 1$ **do**
    $H_{k,t+1} \sim (1 - h(c_{k,t}))\rho_k(\cdot|H_{k,t}, q_{k,t}, g_t) + h(c_{k,t})\pi(\cdot|\mathcal{P})$
    `Update` $q_{k,t+1}$ through Eq. 3
    `Update` $c_{k,t+1}$ through Eq. 4
  end
  $AST_{t+1}^* = \arg\max_{A \in \{AST_t^*, q_{1,t+1}, \cdots, q_{K,t+1}\}} f(A, \mathcal{D})$
  $g_{t+1} = \texttt{LLM}(AST_{t+1}^*)$
end
return AST_L^*

B IMPLEMENTATION DETAILS

In the implementation of CRISO and LLm baselines, Mixtral:8x7b (Jiang et al., 2024) was used as a backbone LLM. All implementations and experiments were conducted in a machine with Intel Xeon Gold 6526Y CPU, 64GB 5600MHz DDR5 memory, and NVIDIA RTX PRO 6000 Blackwell Max-Q GPU. Table 8 presents the hyperparameter setting of CRISO for each benchmark symbolic regression problem. The domain name is also provided to LLM-SR for writing initial instructions. In calculating the discovery score in Eq. 1, the tree depth d is normalized by $(d - d_{min})/(d_{max} - d_{min})$, where $d_{min} = 5$ and $d_{max} = 20$. Similarly, the number of free parameters v is normalized by $(v - v_{max})/v_{max}$, where $v_{max} = 10$. If $v \leq v_{max}$, then the penalty on v becomes zero.

Table 8: Hyperparameter setting of CRISO for each benchmark symbolic regression problem. The domain name is used to write an initial instruction for each benchmark problem. The domain name is a one-word or two-word problem descriptor supplied identically to all LLM-based baselines. K and L denote the number of reasoning agents and the process length, respectively.

Problem	Domain Name	K	L
Chi2PDF	statistics	50	100
NDO	physics	50	100
NNN	machine learning	50	100
MSB	materials science	50	100
FHST	polymer science	100	50
BDC	materials science	50	100
SFL	materials science	50	100
NOMC	organic chemistry	50	200
ECBG	mathematical biology	50	200
HHM	mathematical biology	50	100

Table 9 shows the implementations of the LLM-based methods and their LLM calls with the expected number of tokens. Although FunctionEvolve and STRIDE issue one to two orders of mag-

Table 9: LLM call statistics and runtime bottleneck of the LLM baselines and CRISO.

Method	Total LLM calls	Time per LLM call	Time per hypothesis generation	Main bottleneck	Final wall-clock time
LLM-SR	10,000	$\approx$ 2.7 s	$\approx$ 2.7 s	LLM programming	448 min
FuncEv	$\approx$ 230	$\approx$ 140 s	$\approx$ 12 min	LLM-based AST modification	606 min
STRIDE	$\approx$ 1,000	$\approx$ 25.6 s	$\approx$ 32 s	Long critic/refine generations	426 min
CRISO	10,200	$\approx$ 2.7 s	$\approx$ 2.7 s	LLM programming and knowledge distillation	463 min

Initial program generation prompt

<role>
You are a helpful assistant tasked with discovering mathematical function structures for scientific systems.
</role>
<context>
Problem specification and initial hypothesis
</context>
<task>
Complete the equation function, considering the physical meaning and relationships between inputs variables.
</task>

Program analysis prompt

<role>
You are an expert in *DOMAIN*.
</role>
<context>
Collective-best AST
</context>
<task>
Analyze the relationships between the input and output variables of the equation function, and explain them from the perspective of *DOMAIN*. Summarize your analysis results into compact statements.
</task>

AST update prompt

<role>
You are a helpful assistant tasked with discovering mathematical function structures for scientific systems.
</role>
<context>
<message> Below is some prior knowledge about a function showing relatively low errors. You can leverage when updating the current equation function.</message>
Collective knowledge
</context>
<task>
<message>Update the current function below to fix the *UPDATE_DIRECTION*.</message>
Current AST
</task>

Figure 4: Structured LLM prompts for the initialization, AST analysis, and AST generation. *DOMAIN* specifies a problem domain, and *UPDATE_DIRECTION* indicates a variable assigned to either overestimation or underestimation based on the evaluation results of the current equation.

nitude fewer LLM calls than LLM-SR and CRISO, their wall-clock times are comparable or higher. This is because, under local serving, wall-clock time is governed not by the number of calls but by the total number of decoded tokens. LLM inference consists of two phases with asymmetric costs: prompt processing (prefill) is parallel and fast on GPU, whereas generation (decode) is sequential and proceeds at only tens of tokens per second. Since all requests are served serially by a single Ollama instance, total LLM time is approximately the sum of decoded tokens.

The four methods simply distribute a similar total decoding budget differently. LLM-SR and CRISO issue $\approx$ 10,000 short-form calls (a single function skeleton each, $\approx$ 2.7 s per call). STRIDE issues $\approx$ 1,000 calls, but each is $\approx 10\times$heavier by design: critic calls generate structured critiques over the full program and refine calls regenerate improved functions under a 4,800-token cap, so the $10\times$reduction in call count is offset almost exactly by the 10× higher per-call cost (426 min vs. 448 min). FunctionEvolve issues the fewest calls ($\approx$ 230), but each requests a list of twenty structured mutation suggestions, producing thousands of tokens per call ($\approx$ 140 s); moreover, under mixtral, most responses fail the framework's validity checks, triggering up to five retries per parent. These

retries account for roughly 180 of the 230 calls and dominate its runtime ($\approx 90\%$ of each search step is spent waiting on the LLM), explaining its 1.35× overhead.

C Structured LLM Prompts for Symbolic Equation Discovery

The symbolic reasoning process of CRISO includes three LLM-induced reasoning processes for: (1) initial program generation, (2) program analysis, and (3) AST update. Fig 4 presents structured LLM prompts used in each LLM reasoning process. The initial program generation prompt instructs the reasoning agents to a scientific equation written in Python from the problem specification and initial hypothesis. The program analysis prompt executes the domain-specialized reasoning agent to analyze logical structures and physical interpretations of the current best equation f_{best}, thereby generating collective knowledge across the reasoning agents. Finally, the AST update prompt directs the reasoning agents to revise their current equation based on the collective knowledge.

D Specifications of Benchmark Symbolic Regression Problems

D.1 PDF of Chi-Squared Distributions (Chi2PDF)

The chi-squared distribution with k degree of freedom is the distribution of a sum of the squares of k independent standard normal random variables (Greenwood & Nikulin, 1996). It underpins a wide range of statistical procedures, including variance estimation and independence test. We defined a benchmark problem, Chi2PDF, which aims to recover the PDF of the chi-squared distribution for an arbitrary degree of freedom k. The ground-truth PDF of the chi-squared distribution at $x \geq 0$ is given by:

$$f(x;k) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{(k/2)-1} e^{-x/2}, \tag{6}$$

where Γ is the gamma function. We randomly generated 1,000 (x, k) pairs for symbolic regression and 200 non-overlapping (x, k) pairs for an in-distribution evaluation.

D.2 Flory-Huggins Solution Theory (FHST)

Flory-Huggins solution theory (Flory, 1942) characterizes the thermodynamic landscape of the polymer system based on a mean-field lattice model. The Gibbs free energy of mixing per lattice site, ΔG_m, is defined as:

$$\Delta G_m = RT \left[\frac{\phi}{N} \ln \phi + (1-\phi)\ln(1-\phi) + \chi\phi(1-\phi) \right], \tag{7}$$

where ϕ denotes the volume fraction of the polymer, N represents the chain length of the polymer, and χ is the dimensionless Flory-Huggins interaction parameter accounting for the enthalpy of mixing. R and T were manually determined as 8.314 and 300. This benchmark contains several singularity points and manually selected constants, yielding a fundamental challenge in symbolic regression. Based on the ground-truth equation in Eq. 7, we generated 10,000, 2,000, and 2,000 non-overlapping samples for training, in-distribution testing, and OOD evaluation, respectively.

D.3 Hodgkin-Huxley Model (HHM)

The Hodgkin-Huxley model is a foundational conductance-based framework to represent the electrophysiological dynamics of the system (Hodgkin & Huxley, 1952). This model describes the generation and propagation of action potentials through a set of coupled non-linear ordinary differential equations. The lipid bilayer is treated as a parallel capacitor, where the total membrane current is the sum of the capacitive current and ionic currents:

$$C_m \frac{dV}{dt} = \bar{g}_{Na} m^3 n (V_{Na} - V) + \bar{g}_K h^4 (V_K - V) + \bar{g}_L (V_L - V) + I_{ext}, \tag{8}$$

where V denotes membrane potential, m is sodium activation gating probability, n is sodium inactivation gating probability, h is potassium activation gating probability, and I_{ext} represents external current. C_m was fixed to 1.0, and other constants $(\bar{g}_{Na}, \bar{g}_K, \bar{g}_L, V_{Na}, V_K, V_L)$ are determined manually within physically reasonable ranges. Based on the ground-truth equation in Eq. 8, we generated 10,000, 2,000, and 2,000 non-overlapping samples for training, in-distribution testing, and OOD evaluation, respectively.

D.4 NONLINEAR DAMPED OSCILLATOR (NDO)

A nonlinear damped oscillator is an oscillatory system in which the restoring force and/or the damping force depends nonlinearly on displacement and velocity. Nonlinear damped oscillators frequently appears in a wide range of real-world scientific and engineering systems, such as aerospace and automotive vibration controllers, earthquake engineering, and biological self-oscillators. We defined the NDO problem by directly adopting the problem setting and ground-truth observations introduced in LLM-SR (Shojaee et al., 2025a). The training and evaluation datasets each contain 10,000 ground-truth observations (t, x, v, a), which represent time, position, velocity, and acceleration, respectively. The ground-truth acceleration was calculated through the governing equation below.

$$a = F\sin(\omega t) - \alpha v^3 - \beta xv - \delta x\exp(\gamma x), \tag{9}$$

where the constants $F, \omega, \alpha, \beta, \delta, \gamma$ were set to 0.3, 1.0, 0.5, 1.0, 5.0, 0.5, respectively.

D.5 NONLINEAR NEURAL NETWORK (NNN)

Neural networks are a fundamental building-block of current AI (LeCun et al., 2015). The NNN problem aims to reveal a neural network architecture behind empirically observed model inputs and outputs. We implemented a fully-connected neural network with the sigmoid activation, which its output y is formally given by:

$$y = w_1^{(2)} z_1 + w_2^{(2)} z_2 + b^{(2)}, \tag{10}$$

where

$$z_1 = \sigma(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + b_1^{(1)}), \tag{11}$$

$$z_2 = \sigma(w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + b_2^{(1)}), \tag{12}$$

and $[x_1, x_2]$ is a 2-dimensional input vector, $w_{ij}^{(k)}$ (or $w_i^{(2)}$) are learnable weights, $b_i^{(k)}$ (or $b^{(2)}$) are learnable biases, and σ is the sigmoid function for nonlinear activation. We randomly generated 10,000 (x_1, x_2, y) pairs for symbolic regression and 2,000 non-overlapping (x_1, x_2, y) pairs for an in-distribution evaluation.

D.6 ESCHERICHIA COLI BACTERIAL GROWTH (ECBG)

Quantifying the growth dynamics of Escherichia coli bacteria is a cornerstone of bacterial toxicity studies and food safety analytics. Typically, the bacterial population growth rate has been modeled using nonlinear differential equations. The ECBG problem, introduced in LLM-SR, aims to discover a nonlinear differential equation governing the bacterial growth rate. The ground-truth governing equation is given by:

$$\frac{dB}{dt} = \mu_{max} B \left(\frac{S}{K_s + S}\right) \frac{\tanh(k(T - x_0))}{1 + c(T - x_{decay})^4} f_{pH}, \tag{13}$$

where the pH dependency f_{pH} is given by:

$$f_{pH} = \exp(-|pH - pH_{opt}|) \sin\left(\frac{(pH - pH_{min})\pi}{pH_{max} - pH_{min}}\right)^2, \tag{14}$$

B denotes bacterial population density, S is substrate concentration, T is temperature, and pH is the acidity or alkalinity of the growth medium. We employed the training and test datasets from LLM-SR, which contain 7,500 and 2,500 observations, respectively. The full implementation details of the ground-truth differential equation are described in the original LLM-SR paper (Shojaee et al., 2025a).

D.7 MATERIAL STRESS BEHAVIOR (MSB)

The MSB problem, introduced in LLM-SR, is a symbolic regression task that aims to infer the stress–strain behavior of Aluminium 6061-T651 at a specific temperature (Shojaee et al., 2025a; Aakash et al., 2019). Given strain and temperature as inputs, symbolic regression methods are tasked with discovering a governing equation for the corresponding stress. The MSB problem has two inherent challenges: (1) the ground-truth observations are obtained from real chemical experiments and therefore contain measurement noise and experimental error. (2) there is no predetermined theoretical model to represent the stress-strain relationships, making the symbolic regression problem intrinsically hard for LLMs.

D.8 BATTERY DISCHARGE CAPACITY (BDC)

To make this symbolic regression benchmark, we employed a well-known lithium-ion battery aging dataset extracted from a battery database provided by the NASA Ames Prognostics Center of Excellence (PCoE). A refined version of the dataset is available at `https://github.com/alg-x/Battery-Capacity-Prediction-Using-Regression`. The dataset contains experimental records of 18650-type Li-ion cell subjected to repetitive charge, discharge, and electrochemical impedance spectroscopy cycles at 24 Celsius. This dataset provides experimentally measured capacity for six measurement input variables: cycle index, battery voltage, battery current, battery temperature, voltage load, and current load. We used 334 and 83 instances for training and evaluating the symbolic regression methods, respectively.

D.9 STEEL FATIGUE LIMIT (SFL)

The steel fatigue strength dataset, sourced from the National Institute for Materials Science (NIMS) database, consists of 437 experimental records correlating manufacturing parameters with the fatigue limits of product alloys (Agrawal et al., 2014). We identified ten primary features for each record: the weight percentages (wt%) of nine chemical elements (C, Si, Mn, P, S, Ni, Cr, Cu, and Mo) and the temperature of the final heat treatment process. For rigorous accuracy evaluation, the dataset was partitioned into a training dataset of 350 records and a non-overlapping evaluation dataset of 87 records.

D.10 NON-OXIDATIVE METHANE CONVERSION (NOMC)

Non-oxidative methane conversion (NOMC) is a promising technology for mitigating greenhouse-gas emissions from industrial applications (Julian et al., 2019). We built an in-house NOMC reactor and measured conversion performance as a function of six experiment parameters: pressure, temperature, flow rate, H_2 feed amount, reactor length, and reactor diameter. We use 200 observations for symbolic regression and reserve a non-overlapping set of 51 observations for testing. The NOMC problem can be categorized as one of the most challenging symbolic regression problems due to three reasons: (1) the input-output relations are strongly nonlinear and contain inherent uncertainty from chemical reactions. (2) theoretical models to calculate the NOMC performances of chemical reactors has not been established. (3) Because the ground-truth observations were obtained from our in-house chemical reactor, the backbone LLM is unlikely to possess sufficient prior knowledge of the target system, thereby imposing the challenging requirement of scientific discovery beyond its pretrained knowledge.

E EFFICIENCY COMPARISON

We measured the wall-clock time of the LLM-based baselines and CRISO to compare their efficiency. LLM-SR, FuncEv, and STRIDE required approximately 448, 606, and 426 minutes in their original implementations, respectively. CRISO required approximately 463 minutes under 50 agents ($K = 50$) and 200 iterations ($L = 200$). LLM-SR has the most similar inference structure to CRISO, with both methods requiring one LLM call for each candidate equation generation. LLM-SR makes 10,000 LLM calls to generate a total of 10,000 candidate equations, whereas CRISO likewise makes 10,000 LLM calls, with 50 agents each performing 200 candidate equation revisions. However, CRISO requires one additional LLM call per iteration to extract collective knowledge, resulting in 200 additional LLM calls in total. This additional computational overhead is consistent with the measured wall-clock time, for which CRISO required 3.3% more time than LLM-SR. Despite requiring only 3.3%–8.7% more computational time than LLM-SR and STRIDE, CRISO achieved state-of-the-art symbolic regression accuracy on nine of the ten problems, with 6.1%–93.1% WMAPE reductions compared with the WMAPE values of the respective best baselines on the problems involving randomized components or previously uncharacterized forward mechanisms.

F QUALITATIVE ANALYSIS ON OOD GENERALIZATION

Fig. 5 shows the absolute errors of the equations discovered by LLM-SR (black) and CRISO (red) on the Chi2PDF, NDO, NNN, and FHST problems, for which the governing equations are analytically given. The absolute error is formally defined as the absolute difference between the ground-truth

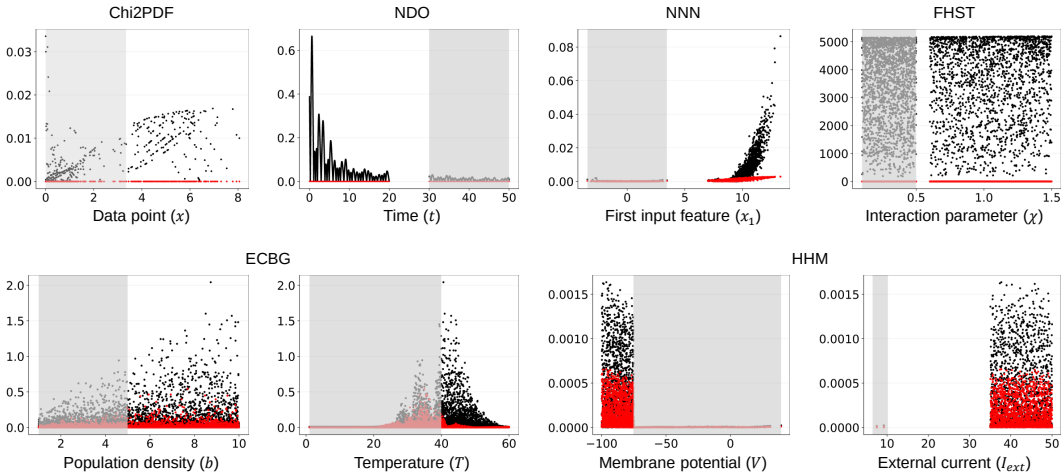

Figure 5: Prediction errors (MAEs) of the equations discovered by LLM-SR (black) and CRISO (red). X-axis: Input variable of the equation. Y-axis: The absolute difference between ground-truth and predicted outputs. Gray-shaded area: In-distribution range of the input variable.

Table 10: WMAPE values of CRISO with different backbone LLMs.

Problem	CRISO (Llama3.1 8B)	CRISO (Gemma4 e4B)	CRISO (Mixtral 8x7B)
Chi2PDF	9.35E-9 (±2.41E-10)	9.11E-9 (±1.05E-9)	5.67E-6 (±1.87E-6)
FHST	6.23E-9 (±2.13E-10)	2.58E-12 (±1.60E-12)	9.40E-8 (±1.62E-8)
HHM	1.27E-7 (±6.79E-8)	4.42E-8 (±1.89E-8)	7.93E-8 (±4.79E-8)
NDO	3.71E-6 (±2.321E-6)	0.0284 (±0.0072)	6.92E-5 (±0.0001)
NNN	0.0023 (±0.0007)	0.0081 (±0.0005)	0.0016 (±0.0001)
ECBG	1.0336 (±0.0436)	0.2083 (±0.0190)	0.0742 (±0.0079)
MSB	0.0407 (±0.0104)	0.0416 (±0.0023)	0.0299 (±0.0028)
BDC	0.0128 (±0.0039)	0.0142 (±0.0018)	0.0092 (±0.0011)
SFL	0.0483 (±0.0073)	0.0456 (±0.0010)	0.0431 (±0.0011)
NOMC	0.1008 (±0.0153)	0.1562 (±0.0109)	0.0776 (±0.0154)

y and predicted outputs $\hat{y}$, which is given by $|y - \hat{y}|$. For a comparative analysis, we presented a gray-shaded area to indicate the in-distribution range of the input variable.

Our experiments identified two critical limitations of the state-of-the-art symbolic regression method, LLM-SR, in OOD generalization: (1) Despite the averaged percentage errors below 0.1 on the NDO, NNN, and HHM problems, LLM-SR produced large absolute errors for many OOD input values. (2) While LLM-SR yields accurate predictions on in-distribution conditions for the NDO, NNN, ECBG, and HHM problems, its absolute errors escalated rapidly under OOD conditions, making a failure in extrapolation. In contrast, CRISO exhibited consistent generalization errors across the in-distribution and OOD conditions on the Chi2PDF, NDO, NNN, FHST, and ECBG problems. In addition, even when OOD inputs were generated from extreme ranges ($-100 < V < -75$ and $35 < I_{ext} < 50$), CRISO maintained the absolute error below 5×10^{-4} in most cases, reducing about 90% WMAPE compared to LLM baselines under the OOD condition. Overall, the experiment results in Fig. 5 demonstrates the superior prediction accuracy and robustness of CRISO under OOD conditions.

G Symbolic Regression Accuracy of CRISO for Different Backbone LLMs

To assess whether CRISO depends on a particular backbone LLM, we repeated all experiments with three open-weight backbones of comparable scale: Llama3.1 8B, Gemma4 e4B, and Mixtral 8x7B, while keeping all hyperparameters, reasoning budgets, and evaluation protocols identical. Table 10 reports the resulting WMAPE values. CRISO completed equation discovery on all ten problems

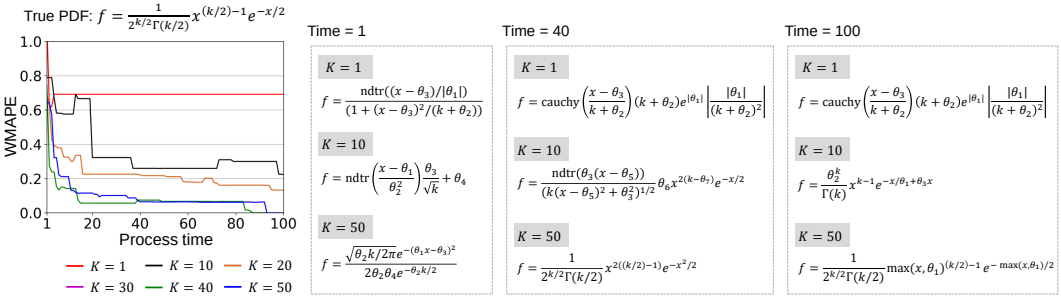

Figure 6: Discovered equations and their in-distribution WMAPE values on the Chi2PDF problem for different number of reasoning agents in CRISO.

with every backbone, without any execution failure or degenerated solution. On the three problems within the LLMs' prior knowledge (Chi2PDF, FHST, and HHM), all backbones accurately recovered governing equations. On the seven problems with randomized or previously uncharacterized mechanisms, CRISO with Gemma4 e4B and Mixtral 8x7B showed WMAPE values less than 0.21. However, CRISO with Llama3.1 8B failed on the ECBG problem, which contains severely randomized coefficients and mathematical terms. However, the results on the moderate backbone LLMs indicate that the accuracy of CRISO stems from its collective symbolic regression framework rather than from the idiosyncrasies of a specific backbone LLM.

H Population Size and Symbolic Regression Capability

We measured in-distribution WMAPE of the equations discovered by CRISO on the Chi2PDF problem for different numbers of reasoning agents $K \in \{1, 10, 20, 30, 40, 50\}$ but under the same LLM calls. Process time is a hypothetical concept used to indicate the point at which each model with a different K generated the same number of equations. Formally, process time is defined as $50/K$-th iteration, and non-integer iteration was rounded to the nearest integer-valued process time. As shown in Fig. 6, collective intelligence comprising a larger ensemble of the reasoning agents consistently yielded the equations with lower WMAPE. While the WMAPE of the single intelligence ($K = 1$) plateaued at 0.6 without further reduction, the WMAPE of the collective intelligence with $K = 40$ and 50 converged to near-zero. Furthermore, the qualitative analysis on the equations identified at the time 1, 40, and 100 reveals that increasing K enhances the capacity of the collective intelligence to accurately recover the essential structure of the target equation.

I Limitations and Future Work

I.1 Absorbing Collective Knowledge and Population Diversity

By construction, the collective knowledge g_t is distilled only from the single best-performing AST. This absorbing property guarantees monotone accumulation of useful structure, but it may also bias the population toward a single lineage of hypotheses and reduce structural diversity, particularly on problems where several competing equation skeletons achieve similar discovery scores. Exploration in CRISO currently relies on the stagnation-triggered search to the initial prior $\pi(\cdot|\mathcal{P})$. A natural extension is to distill knowledge from a Pareto set of accuracy–complexity trade-offs rather than a single maximization problem. We expect such diversity-aware distillation to further improve robustness on problems with multimodal hypothesis landscapes.

I.2 Dependence on Hyperparameters and Backbone LLM's Capability

CRISO requires setting the number of agents K and the process length L. Table 8 shows that several problems (FHST, NOMC, ECBG) benefited from problem-specific settings. Adaptive hyperparameter setting is required as future work. In addition, while CRISO with a moderate backbone outperformed STRIDE with a stronger backbone, the framework is not immune to backbone limitations, as whon in the failture of CRISO with Llama3.1 8B on the ECBG problem. Combining

CRISO with stronger open-weight reasoning models, or assigning the analyst and generator roles to different models, is a straightforward direction.

I.3 Computational Cost and Parallel Implementation

CRISO issues approximately 10,200 LLM calls per problem, and all LLM-based methods in our study were served serially by a single local instance. Although the overhead relative to LLM-SR is only 3.3%, absolute wall-clock time remains several hours per problem. Because the K agents are independent within each iteration, hypothesis generation is embarrassingly parallel, and batched or distributed serving together with early stopping is expected to reduce wall-clock time substantially without changing the algorithm.