LiteSR: Literature-Guided Agentic Retrieval for Symbolic Regression

Zalish Mahmud[1], **Anantaa Kotal**[1], **Lixin Jin**[2], **Anthony Darrouzet-Nardi**[3], **Aritran Piplai**[1], **Nan Jiang**[1*]
University of Texas at El Paso, TX, USA
[1]Department of Computer Science, [2]Department of Geological Sciences, [3]Department of Biological Sciences
zmahmud@miners.utep.edu, {akotal,ljin2,apiplai,ajdarrouzetnardi,njiang}@utep.edu

Abstract

Discovering governing equations from experimental data is a central goal of AI-driven scientific discovery, but existing symbolic regression methods often struggle with complex dynamical systems involving many interacting variables and nonlinear mechanisms. This work leverages recent progress in literature agents to extract structured scientific knowledge from papers and use it to guide symbolic regression. We propose **LiteSR**, a literature-guided agentic retrieval framework that couples a *literature agent* with an *equation-discovery agent*. The literature agent retrieves relevant scientific papers, extracts mechanistic knowledge, identifies variable semantics, candidate mathematical forms, physical constraints, dimensional relationships, and related hypotheses, and synthesizes them into a structured solver prompt. The symbolic solver agent then uses this prompt within an LLM-based symbolic regression pipeline, treating equations as executable programs and combining language-model priors with numerical optimization and iterative symbolic search. By transforming literature evidence into structured priors for equation discovery, LiteSR reduces implausible candidates while preserving the flexibility to discover equations that differ from those explicitly reported in prior work. Experiments across chemical science, soil science, and materials science show that LiteSR improves both numerical fitting and structural recovery compared with LLM-SR, especially for complex sub-expressions. Code implementation is available at: https://github.com/zalishmahmud/LiteSR.

1 Introduction

Uncovering the governing principles of dynamical systems from experimental data is a central task in AI-driven scientific discovery [24, 36, 32]. Symbolic regression addresses this goal by searching for closed-form mathematical expressions, particularly symbolic ordinary differential equations (ODEs), that explain observed data. Recent works have explored a broad range of search and learning paradigms, including genetic programming [9], sparse regression [2, 7], Monte Carlo tree search [27], Transformers [20], deep reinforcement learning [10], and large language models [25, 35]. Despite this progress, existing techniques still struggle to process complex systems involving many interacting variables and nonlinear mechanisms.

One promising direction for accelerating symbolic regression is to incorporate domain knowledge to guide the search process. Prior work has considered equation-structure constraints [13], equation-equality constraints [12], implicit knowledge from large language models [28], and physics-inspired heuristics such as symmetry and separability [29]. While effective in specific settings, these mechanisms do not actively exploit the vast body of domain knowledge accumulated in scientific literature,

*Corresponding authors: Anantaa Kotal and Nan Jiang.

40th ACM Conference on AI and Agentic Systems (CAIS) 2026. Workshop: AI Agents for Discovery in the Wild.

including known equations, modeling assumptions, approximations, physical constraints, and mechanistic interpretations. How to extract such knowledge from literature and use it to guide symbolic search remains underexplored.

Recent progress in agentic AI makes it much easier to extract domain knowledge directly from scientific papers. Methods such as ReAct [33], Toolformer [23], and Self-RAG [1] address the common problem of moving language models beyond static prompting by enabling reasoning-action loops, tool invocation, and retrieval-aware self-reflection. In parallel, scientific literature QA benchmarks such as LitQA2 [26] and ResearchQA [34] evaluate retrieval-grounded answer accuracy and citation support across scientific domains. We leave a detailed related work discussion to the appendix A. However, these works primarily focus on answering scientific questions rather than transforming retrieved literature into structured priors for symbolic search. From our experiment, we find existing retrieval-augmented LLM systems can access papers and generate natural-language summaries, but they often stop at narrative synthesis rather than converting literature evidence into executable scientific hypotheses. As a result, how to extract and embed literature-derived knowledge into the symbolic regression process remains an interesting problem to explore.

We present **LiteSR**, a literature-guided framework for symbolic regression. It decomposes the task into two coupled components: (1) The *literature agent* retrieves relevant papers and identifies variable meanings, candidate mathematical forms, physical constraints, dimensional relationships, and related hypotheses. (2) *The symbolic solver agent* then uses this information to guide an LLM-based symbolic regression pipeline, where candidate equations are generated, numerically fitted, and evaluated against data. By using literature-derived knowledge to guide the search, LiteSR reduces implausible candidate equations while still allowing the model to discover equations that differ from those reported in prior work.

In experiments, we consider datasets spanning chemical science, soil science, and materials science. We compare LiteSR against a recent baseline, LLM-SR [25], using two metrics: fitness to the observed data and the percentage of matched sub-expressions *w.r.t.* the ground-truth equation. Compared with LLM-SR, LiteSR discovers more accurate expressions with fewer learning epochs and recovers a higher percentage of matched sub-expressions, especially for complex sub-terms.

2 Preliminaries

Ordinary Differential Equations (ODEs) describe the continuous-time evolution of dynamical systems. Let $\mathbf{x}(t) = (x_1(t), \ldots, x_n(t)) \in \mathbb{R}^n$ denote the state vector of a system at time t. The temporal evolution of each state variable is governed by its time derivative, denoted by $\frac{dx_i}{dt}$. A general ODE system can be written as

$$\frac{dx_i}{dt} = f_i(\mathbf{x}(t), \mathbf{c}), \qquad \text{for } i = 1, \ldots, n$$

where f_i specifies the governing equation for the i-th state variable and $\mathbf{c}$ denotes unknown or known coefficients. Each function f_i is typically represented symbolically using a subset of state variables in $\mathbf{x}$, coefficients in $\mathbf{c}$, and mathematical operators such as addition, multiplication and division.

For example, consider a simple chemical system involving ammonium and nitrate concentrations. Let x_0 denote the ammonium (NO_3^-) concentration and x_1 denote the nitrate (NO_3^-) concentration, i.e., $x_0(t) = [\mathrm{NH_4}^+](t)$ and $x_1(t) = [\mathrm{NO_3}^-](t)$. A typical dynamical system following [3] is

$$\frac{d[\mathrm{NH_4}^+]}{dt} = -k_1[\mathrm{NH_4}^+], \qquad \frac{d[\mathrm{NO_3}^-]}{dt} = k_1[\mathrm{NH_4}^+] - k_2[\mathrm{NO_3}^-].$$

The first equation models ammonium consumption, while the second equation models nitrate production from ammonium together with nitrate loss.

Problem Formulation Following prior work [8, 27], we assume access to an experimental dataset $\mathcal{D}$ and a predefined operator set, such as $\{+, -, \times, \div, \sin, \ldots\}$. The objective is to identify a symbolic expression that best explains the observed dynamics. Existing approaches typically formulate this objective in one of two ways: (1) *gradient matching* [25], where approximated time derivatives are used as supervision labels to learn each component function f_i independently; or (2) *trajectory matching* [6, 11], where candidate ODEs are integrated forward and optimized by minimizing the discrepancy between predicted and observed trajectories.

In this work, we adopt a gradient-matching formulation and use an LLM-based search procedure to generate candidate ODEs as executable Python programs. Given a dynamical system with state variables $\mathbf{x}(t)$ evolving according to an unknown ODE system $(f_1, \ldots, f_n)$, our goal is to recover each component function f_i from derivative-labeled observations $\mathcal{D} = \{(\mathbf{x}(t_k), \dot{\mathbf{x}}(t_k))\}_{k=1}^{T}$, where $\dot{\mathbf{x}}(t_k)$ denotes the time derivative at observation time t_k. Each f_i is represented as a Python program skeleton of the form `def ODE(x, c): return x[0] * c[0], x[1]* c[1] - x[0]`, where "`c`" denotes a list of coefficients. Once the program structure is fixed, these coefficients are optimized against the dataset $\mathcal{D}$. We seek equations that not only fit the observed dynamics, but also contain sub-expressions that are consistent with relevant historical scientific literature.

3 Methodology

The proposed LiteSR framework consists of two specialized agents: a *literature agent* $\mathcal{A}_L$ and a *symbolic solver agent* $\mathcal{A}_S$. Given a problem prompt $\mathcal{T}$ described in natural language, $\mathcal{A}_L$ constructs a structured literature context $\mathcal{C}$ from a pre-indexed corpus of scientific documents. Conditioned on this context, $\mathcal{A}_S$ iteratively searches for the optimal equation. The main pipeline is illustrated in Figure 4.

3.1 Literature agent

The literature agent $\mathcal{A}_L$ operates in two phases: an offline indexing phase that constructs a semantically queryable knowledge base from a domain-specific corpus, and an online retrieval phase that recovers and synthesizes relevant mechanistic context for a problem prompt $\mathcal{T}$.

Literature Indexing. Given a corpus of scientific literature artifacts $\{l_1, \ldots, l_K\}$, the retrieval agent $\mathcal{A}_L$ processes each artifact through an LLM-based extraction pipeline. Formally, for each artifact l_k, an LLM is prompted with a structured extraction instruction to identify all equations and return each as a normalized record $r = (e, \ell, d, \tau)$, where e is the raw equation string, $\ell \in \{$`ODE`, `conservation`, `equilibrium`, `definition`, `asymptotic`, `dimensionless`$\}$ is a category label, d is a natural-language description of the scientific domain and the physical quantity governed by e, and τ is the interpretation of each variable and sub-expression in e. Records are deduplicated on e before being stored in a vector index $\mathcal{I}$, where each record is embedded as a dense vector over d. At retrieval time, d is matched against the problem context and τ is matched against the input variables of $\mathcal{T}$. Example 1 shows that the literature indexing step converts each extracted equation into a structured record.

Example 1: Literature Indexing on Nitrification

Equation: $2\,\mathrm{NH_4}^+ + 3\,\mathrm{O_2} \xrightarrow{\mathrm{AMO/HAO}} 2\,\mathrm{NO_2}^- + 2\,\mathrm{H_2O} + 4\,\mathrm{H}^+$
Category label: definition
Description: First step of nitrification converting ammonium to nitrite in nitrogen biogeochemistry.
Terms: $2\,\mathrm{NH_4}^+$ = two ammonium ions as substrate,
$3\,\mathrm{O_2}$ = three oxygen molecules as oxidant,
$2\,\mathrm{NO_2}^-$ = two nitrite ions produced,
$4\,\mathrm{H}^+$ = four protons released.

Context Retrieval. Given problem $\mathcal{T}$, agent $\mathcal{A}_L$ retrieves and synthesizes relevant mechanistic context from the literature index $\mathcal{I}$ through the following procedures:

Query Decomposition. Given the natural-language task prompt $\mathbf{p}_{\mathcal{T}}$, the retrieval agent $\mathcal{A}_L$ prompts an LLM to decompose the task into a set of targeted sub-queries $\mathcal{Q}$. Each sub-query $q_j \in \mathcal{Q}$ targets a distinct physical law, named formula, or mechanistic dependency relevant to the target problem $\mathcal{T}$. To avoid redundant retrieval, the decomposition groups variables or processes that share the same mathematical form into a single sub-query. Each sub-query is then issued independently against the vector index $\mathcal{I}$ via semantic search over the description field d, yielding a set of candidate records. Example 2 illustrates how a task description is decomposed into mechanism-specific sub-queries before retrieval. In this example, multiple substrate-limitation terms governed by Michaelis–Menten kinetics are represented by one shared query rather than several variable-specific queries.

Example 2: Query Decomposition for Nitrification

Raw Parsed Description:
Find the mathematical function skeleton that represents the rate of change of nitrate concentration ($d[\mathrm{NO_3^-}]/dt$) in an aquatic biogeochemical system. The system involves nitrification (source: $\mathrm{NH_4^+}$ oxidized to $\mathrm{NO_3^-}$), denitrification (sink: $\mathrm{NO_3^-}$ reduced to $\mathrm{N_2}$), and DNRA (sink: $\mathrm{NO_3^-}$ reduced to $\mathrm{NH_4^+}$). Substrate limitation follows Michaelis–Menten kinetics, and oxygen inhibition follows an anoxic switch.

Decomposed Sub-queries:
[1] Michaelis–Menten nitrification rate constant for ammonium
[2] Nitrification rate-constant dependence on temperature
[3] Oxygen inhibition rate constant for denitrification and DNRA
[4] Denitrification rate coefficient substrate-limitation term
[5] DNRA rate coefficient substrate-limitation term

Relevance Score. Retrieved candidate records are reranked using a relevance score that combines task-level contextual relevance with variable-level alignment. Let r denote a retrieved literature record, with description field d_r and extracted variable descriptions τ_r. Let $\mathcal{U}_\mathcal{T}$ denote the set of input-variable descriptions in the target task $\mathcal{T}$. We define

$$s(r, \mathcal{T}) = \alpha \cdot \mathrm{sim}(d_r, \mathbf{p}_\mathcal{T}) + (1 - \alpha) \cdot \frac{1}{|\tau_r|} \sum_{v \in \tau_r} \max_{u \in \mathcal{U}_\mathcal{T}} \mathrm{sim}(v, u), \tag{1}$$

where $\mathrm{sim}(d_r, \mathbf{p}_\mathcal{T})$ measures the semantic similarity between the record description d_r and the task prompt $\mathbf{p}_\mathcal{T}$. The second term measures how well the variables described in the retrieved record match the input variables of the target task: for each retrieved variable description $v \in \tau_r$, we take its best semantic match among the task-variable descriptions $u \in \mathcal{U}_\mathcal{T}$. The coefficient $\alpha \in [0, 1]$ controls the trade-off between overall contextual relevance and variable-level alignment.

Iterative Query Refinement. The top-ranked score of each sub-query is assessed against a threshold δ: sub-queries exceeding δ are labeled *high-relevance* and their results retained; those falling below δ are rephrased by an LLM using high-relevance sub-queries as terminology references and re-issued. If a rephrased sub-query still fails to exceed δ, it is further decomposed into more targeted sub-queries, which are added to $\mathcal{Q}$ and retrieved in the subsequent round. This process repeats until all sub-queries in $\mathcal{Q}$ are labeled high-relevance or a maximum number of refinement rounds is reached. Example 3 shows that low-relevance retrieval queries are iteratively rephrased to better match literature-indexed equation descriptions.

Example 3: Iterative Query Refinement for Nitrification

Round 1: Retrieval Assessment (2 of 5 shown)

[1] Michaelis–Menten nitrification rate constant for ammonium — score=0.749 ✓
[2] Nitrification rate-constant dependence on temperature — score=0.506 ✗
... 3 additional sub-queries assessed ...

Round 2: Refinement (1 weak query rephrased)

[2] ~~Nitrification rate constant dependency on temperature~~
→ Temperature dependence of the Michaelis–Menten nitrification rate constant — score=0.741 ✓

Ranking and Query Synthesis. Retrieved records from all high-relevance sub-queries are then pooled, deduplicated on e, and filtered by ℓ using a two-tier strategy: the first tier retains records with $\ell \in \{\mathrm{ODE}, \mathrm{definition}, \mathrm{equilibrium}, \mathrm{conservation}\}$; if fewer than a minimum threshold of records qualify, a fallback tier retains all records except those with $\ell \in \{\mathrm{asymptotic}, \mathrm{dimensionless}\}$. This tiering reflects the observation that governing equations, differential equations, equilibrium expressions, and conservation laws, directly constrain the functional form of the target equation, whereas asymptotic approximations and dimensionless scalings are valid only under restrictive conditions and may mislead the generation model into proposing structurally incorrect hypotheses. The surviving records are synthesised by $\mathcal{A}_L$ into a structured context $\mathcal{C}$, which identifies each equation by name, presents its mathematical form, and explains the physical role of each term. $\mathcal{C}$ is injected into the generation agent's prompt at each iteration.

3.2 Symbolic Solver Agent

The context-guided symbolic solver agent $\mathcal{A}_S$ is instantiated as a pre-trained LLM π_θ that iteratively generates and refines equation-program hypotheses for a target problem $\mathcal{T}$. At iteration t, $\mathcal{A}_S$ samples a batch of b candidate equation-program skeletons $\{f_i\}_{i=1}^{b}$ from π_θ, conditioned on a structured prompt p_t.

The prompt p_t consists of four components: (i) the *Problem Specification Prompt*, which describes the target problem in natural language, including the key variables, known constraints, and desired output; (ii) the *Literature Context*, which contains the structured mechanistic summary produced by the literature agent $\mathcal{A}_L$; (iii) the *Evaluator*, which specifies the objective used to score each proposed skeleton after parameter optimization; and (iv) the *Best Discovered Equations*, which provides in-context demonstrations of high-performing equation skeletons selected from the experience buffer. As shown in Example 4, the solver agent receives a structured prompt containing the task specification, literature context, evaluation function, experience demonstrations, and the target function to complete.

Example 4: Symbolic Solver Agent for Subsurface Nitrification

Problem Specification Prompt $\mathcal{T}$

Find a mathematical function skeleton that represents the rate of change of nitrate concentration, $d[NO_3^-]/dt$, in an aquatic biogeochemical system. Complete the `equation` *function below by considering the physical meaning of the input variables and their mechanistic relationships.*

```
def equation(NO3, NH4, O2, T, params):  "Improved version of equation_v0"
```

Literature Context $\mathcal{C}$

[First Step of Nitrification] $2\,NH_4^+ + 3\,O_2 \xrightarrow{\text{AMO/HAO}} 2\,NO_2^- + 2\,H_2O + 4\,H^+$; Michaelis–Menten substrate limitation on NH_4^+ and O_2; *... two additional equations in $\mathcal{C}$ ...*

Evaluator

def evaluate(equation, data): ...

Best Discovered Equations

```
def equation_v0(NO3, NH4, O2, T, params):  ...
def equation_v1(NO3, NH4, O2, T, params):  ...
def equation_v2(NO3, NH4, O2, T, params):  ...
```

Parameter Optimization and Scoring. Each valid skeleton f_i is represented as an executable equation program of the form `def f(x, params): ... return y`, where `params` denotes a vector of numeric placeholders. For each skeleton, we fit these parameters on the dataset $\mathcal{D}$ (such as BFGS or Adam optimizers) to obtain the optimized values $\texttt{params}^*$. After parameter fitting, the fitness of each hypothesis is evaluated by its prediction error:

$$\texttt{Fitness}_{\mathcal{T}}(f, \mathcal{D}) = -\texttt{MSE}(\hat{y}, y), \qquad \hat{y} = f(\mathbf{x}, \texttt{params}^*).$$

Thus, higher scores correspond to lower prediction error. Skeletons that fail to execute or exceed a predefined runtime limit are discarded.

4 Experiment

In this section, we compare our LiteSR with a recent popular baseline LLM=SR on three challenging symbolic ODE settings, spanning chemical science (in section 4.1), soil science (in section 4.2), and materials science (in section 4.3). Experiment setting are available in appendix B.1.

4.1 Case Study 1: Discovering Bicarbonate Dynamics in CO_2 Dissolution

Problem setting. Bicarbonate (HCO_3^-) is a central intermediate in aqueous carbonate chemistry, arising when injected CO_2 dissolves into saline aquifers and participates in a cascade of acid-base reactions. Dissolved CO_2 forms carbonic acid, which dissociates to produce HCO_3^-; bicarbonate is in turn consumed through further dissociation to carbonate (CO_3^{2-}), while reverse recombination reactions regenerate carbonic acid and bicarbonate depending on the local proton concentration

Table 1: Structural comparison of LiteSR and LLM-SR discovered equations against the target $\mathrm{d}[\mathrm{HCO_3}^-]/\mathrm{d}t$ equation. ✓ = correct match; ≈ = partial match (correct form, wrong detail); ✗ = missing or incorrect.

Subcomponent	**Ground-truth**	**LiteSR**		**LLM-SR** [25]	
H_2CO_3 dissociation	$+k_3(T)\cdot[\mathrm{H_2CO_3}]$	$+k_3(T)\cdot[\mathrm{H_2CO_3}]$	✓	$+k_1(T)\cdot[\mathrm{H_2CO_3}]$	✓
HCO_3^- recombination	$-k_{-3}(T)\cdot Y\cdot[\mathrm{HCO_3}^-]$	$-k_{-3}(T)\cdot Y\cdot[\mathrm{HCO_3}^-]$	✓	$-k_{r1}(T)\cdot Y\cdot[\mathrm{HCO_3}^-]$	✓
HCO_3^- dissociation	$-k_4(T)\cdot[\mathrm{HCO_3}^-]$	$-k_4(T)\cdot[\mathrm{HCO_3}^-]$	✓	$-k_2(T)\cdot[\mathrm{HCO_3}^-]$	✓
CO_3^{2-} recombination	$+k_{-4}(T)\cdot Y\cdot[\mathrm{CO_3}^{2-}]$	$+k_{-4}(T)\cdot Y\cdot[\mathrm{CO_3}^{2-}]$	✓	$+k_{r2}(T)\cdot Y\cdot[\mathrm{CO_3}^{2-}]$	✓
Arrhenius exponent form	$-\frac{E_a}{R}\left(\frac{1}{T}-\frac{1}{T^\circ}\right)$	$-\frac{E_a}{R\cdot T}$	≈	$-\frac{E_a}{T}$	✗
Separate E_a per reaction	4 independent	4 independent	✓	shared per forward/reverse pair	✗
Exact match (%)		**5/6 (83.3%)**		**4/6 (66.7%)**	

[16]. Temperature modulates all reaction rates through Arrhenius-type scaling with reaction-specific activation energies [19]. This benchmark therefore tests whether LiteSR can recover a compact mechanistic equation that combines the source-sink structure of carbonate chemistry with temperature-dependent kinetics, a system requiring the joint discovery of multiple coupled Arrhenius terms from a single observed trajectory.

The input variables are $\mathbf{x} = [\mathrm{H_2CO_3},\ \mathrm{HCO_3}^-,\ \mathrm{CO_3}^{2-},\ T,\ Y]$, where Y denotes an auxiliary proton concentration derived from charge conservation, and the target variable is $y = \mathrm{d}[\mathrm{HCO_3}^-]/\mathrm{d}t$. The ground-truth governing equation is:

$$\begin{aligned}\frac{\mathrm{d}[\mathrm{HCO_3}^-]}{\mathrm{d}t} = &\underbrace{k_3^\Theta e^{-\frac{E_{a,3}}{R}\left(\frac{1}{T}-\frac{1}{T^\Theta}\right)}[\mathrm{H_2CO_3}]}_{\text{carbonic acid dissociation}} - \underbrace{k_{-3}^\Theta e^{-\frac{E_{a,-3}}{R}\left(\frac{1}{T}-\frac{1}{T^\Theta}\right)}Y[\mathrm{HCO_3}^-]}_{\text{carbonic acid recombination}} \\ &- \underbrace{k_4^\Theta e^{-\frac{E_{a,4}}{R}\left(\frac{1}{T}-\frac{1}{T^\Theta}\right)}[\mathrm{HCO_3}^-]}_{\text{bicarbonate dissociation}} + \underbrace{k_{-4}^\Theta e^{-\frac{E_{a,-4}}{R}\left(\frac{1}{T}-\frac{1}{T^\Theta}\right)}Y[\mathrm{CO_3}^{2-}]}_{\text{carbonate recombination}}\end{aligned} \tag{2}$$

where $\boldsymbol{\theta} = [k_*^\Theta,\ E_{a,*},\ R,\ T^\Theta]$ denotes the model parameters: k_i^Θ are reaction-rate constants at reference temperature T^Θ, $E_{a,i}$ are reaction-specific activation energies, R is the gas constant.

Result and Analysis. Table 1 shows that both LiteSR and LLM-SR successfully recover the common carbonate kinetic structure, including the four source and sink terms involving H_2CO_3, HCO_3^-, and CO_3^{2-} [16]. Both methods therefore achieve perfect scores on the common kinetic components. However, the two methods differ on the Arrhenius temperature-dependence structure [19]. LiteSR correctly recovers four independent activation energies, one per reaction, but misses the reference-temperature offset $1/T^\circ$ in the Arrhenius exponent, discovering $-E_a/(R\cdot T)$ rather than the full $-(E_a/R)(1/T - 1/T^\circ)$ form, yielding an overall score of 5/6 (83.3%). LLM-SR performs worse on both Arrhenius subcomponents: it omits the gas constant R and the reference-temperature offset, and additionally collapses the activation energies into a shared value per forward/reverse pair rather than treating them independently, resulting in a score of 4/6 (66.7%).

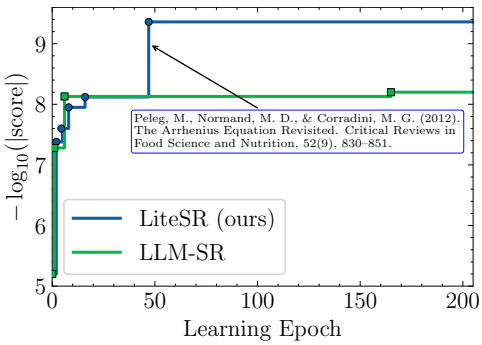

Figure 1: Comparison of $d[\mathrm{HCO_3^-}]/dt$ equation best score over epochs for LiteSR and LLM-SR. The y-axis shows $-\log_{10}(|\text{score}|)$, where higher values indicate better performance. The score is defined in appendix B.1.

4.2 Case Study 2: Equation of State under Extreme Compression

Problem Setting. Under static high-pressure conditions, the pressure–volume (P–V) relationship of a material is governed by its equation of state (EoS), which encodes the bulk elastic response of the crystal lattice to compression. Standard formulations such as the Rydberg–Vinet [30] and third-order Birch-Murnaghan EoS [17] describe the P–V curve in terms of the zero-pressure bulk modulus K_0 and its pressure derivative K_0', but diverge under extrapolation to extreme pressures due to differing

Table 2: Structural comparison of LiteSR and LLM-SR discovered equations against the Vinet (Rose–Smith) Equation of State. Note: ✓ = correct match; ≈ = partial match (correct form, wrong detail); ✗ = missing or incorrect.

Subcomponent	**Ground-truth**	**LiteSR (ours)**	**LLM-SR** [25]
Outer prefactor	$3K_0\left(\frac{V}{V_0}\right)^{-K'_\infty}$	$3K_0\left(\frac{V}{V_0}\right)^{-K'_\infty}$ ✓	$\frac{3K_0\left[1-(V/V_0)^{1/3}\right]}{(V/V_0)^{2/3}}$ ✗
Compression variable	$(V/V_0)^{1/3}$	$(V/V_0)^{1/3}$ ✓	$(V/V_0)^{1/3}$ ✓
Strain $= 1-(V/V_0)^{1/3}$	$1-(V/V_0)^{1/3}$	$1-(V/V_0)^{1/3}$ ✓	$1-(V/V_0)^{1/3}$ ✓
Linear strain bracket	$[\text{strain}]$	$[\text{strain}]$ ✓	$[\text{strain}]$ ✓
Exponential envelope	$\exp\{\cdots\}$	$\exp\{\cdots\}$ ✓	$\exp\{\cdots\}$ ✓
Higher-order correction factor	absent	absent ✓	$1+\alpha\left[1-(V/V_0)^{1/3}\right]^2$ ✗
Coefficient of K'_0 in exponent	$\frac{3}{2}K'_0$	$\frac{3}{2}K'_0$ ✓	η (single free param, no K'_0) ✗
Coefficient of K'_∞ in exponent	$-3K'_\infty$	$-3K'_\infty$ ✓	absorbed into η ✗
Constant offset in exponent	$+\frac{1}{2}$	$+\frac{1}{2}$ ✓	absent ✗
Exponent multiplied by strain	$\left(\frac{3}{2}K'_0-3K'_\infty+\frac{1}{2}\right)\cdot$ strain	$\left(\frac{3}{2}K'_0-3K'_\infty+\frac{1}{2}\right)\cdot\text{strain}$ ✓	$\left(\frac{3}{2}K'_0-3K'_\infty+\frac{1}{2}\right)\cdot\text{strain}$ ✓
Separate K'_0 and K'_∞	$K'_0 \neq K'_\infty$, independent	`K0_prime` ≠ `Kinf_prime`, independent ✓	merged into η ≈
Exact match (%)		**11/11 (100%)**	**6/11 (54.5%)**

assumptions about the high-pressure limit of K' [21]. The generalized Rydberg-Stacey (R-S) EoS addresses this by introducing K'_∞, the pressure derivative of the bulk modulus at infinite pressure, as an explicit parameter recovering the Rydberg-Vinet, and Birch-Murnaghan models as special cases. This benchmark therefore tests whether LiteSR can recover the functional form of the R-S EoS from volumetric compression data alone, a task requiring identification of the nested exponential P–V structure without prior knowledge of the equation form. The ground-truth equation is

$$P = 3K_0\left(\frac{V}{V_0}\right)^{-K'_\infty}\left[1-\left(\frac{V}{V_0}\right)^{\frac{1}{3}}\right]\exp\left\{\left(\frac{3}{2}K'_0-3K'_\infty+\frac{1}{2}\right)\left[1-\left(\frac{V}{V_0}\right)^{\frac{1}{3}}\right]\right\} \tag{3}$$

The parameters in this equation are denoted collectively by $\boldsymbol{\theta} = [P,\ K_0,\ K'_0,\ K'_\infty,\ V,\ V_0]$, where, P denotes the pressure. K_0 is the bulk modulus at zero pressure. K'_0 is the first pressure derivative of the bulk modulus at zero pressure. K'_∞ is the first pressure derivative of the bulk modulus at infinite pressure, and V_0 is the reference (zero pressure) volume.

Result and Analysis. Table 2 shows that LiteSR successfully recovers all eleven structural components of the Equation of State under Extreme Compression, achieving a perfect score of $11/11$ (100%). LLM-SR, by contrast, matches only $6/11$ components (54.5%). Both methods correctly identify the compression variable $(V/V_0)^{1/3}$, the strain $1-(V/V_0)^{1/3}$, the linear strain bracket, and the exponential envelope $\exp\{\cdots\}$, as well as the full exponent multiplied by strain. However, LLM-SR fails to recover the correct outer prefactor, replacing $3K_0(V/V_0)^{-K'_\infty}$ with the standard Vinet form $3K_0[1-(V/V_0)^{1/3}]/(V/V_0)^{2/3}$, and additionally introduces a spurious higher-order correction factor $1+\alpha[1-(V/V_0)^{1/3}]^2$ absent from the ground truth. LLM-SR also merges the physically distinct K'_0 and K'_∞ parameters into a single free parameter η, omitting the explicit $\frac{3}{2}K'_0$ and $-3K'_\infty$ coefficients and the constant offset $+\frac{1}{2}$ in the exponent, only partially recovering the separation of the two parameters (≈). In contrast, LiteSR correctly

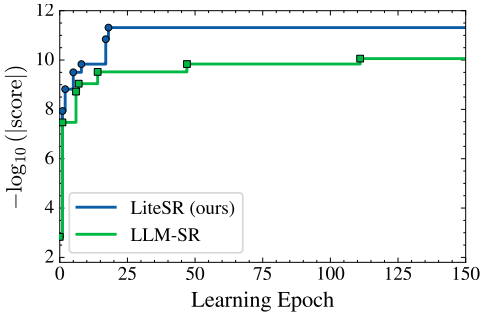

Figure 2: Best-score comparison between LiteSR and LLM-SR across learning epochs on the Equation-of-State task under extreme compression. The y-axis reports $-\log_{10}(|\text{score}|)$, where higher values indicate a better fit.

retains the $\frac{3}{2}K_0'$ and $-3K_\infty'$ coefficients, the $+\frac{1}{2}$ offset, the absence of any spurious correction factor, and independent `K0_prime` and `Kinf_prime` parameters, suggesting that literature-guided retrieval substantially improves the recovery of domain-specific equation components that are unlikely to be inferred from generic symbolic search alone. In Figure 2, LiteSR also achieves a significantly better numerical result within the first 150 epochs. Specifically, LiteSR reaches a best residual of 4.85×10^{-12} at epoch 18, compared to 8.76×10^{-11} for LLM-SR at epoch 111, corresponding to an approximately $\sim 18\times$ improvement in score. This indicates that while LLM-SR converges to a reasonable approximate structure, LiteSR ultimately recovers a more accurate and mechanistically faithful equation in far fewer iterations.

4.3 Case Study 3: Discovering Nitrate Dynamics in Soil Nitrogen Cycling

Problem Setting. Within soil pore water, nitrate (NO_3^-) is an important inorganic nutrient that can change rapidly in concentration, often exhibiting spatial hot spot patterns across the landscape [5]. A major internal source of NO_3^- is microbial nitrification, in which NO_3^- is produced from ammonium (NH_4^+) by nitrifying bacteria and archaea that oxidize NH_4^+ for net energy gain [18]. Nitrate is highly mobile in soils because it binds weakly to the soil matrix, which is also negatively charged. Thus, it can readily diffuse or move by advection to plant and microbial uptake transporters [15], as well as be leached out of the soil by infiltrating rain water. Due to this high mobility, and to variation in NH_4^+ supply from organic N mineralization, microbial turnover, and biological uptake, nitrate concentrations can vary by orders of magnitude. Another main loss pathway is denitrification, a microbially driven process that occurs under low-oxygen conditions, opposite to nitrification. In these conditions, NO_3^- is used as a terminal electron acceptor instead of O_2; it also requires a carbon substrate as an energy source. Denitrification results in gaseous loss of N from the system, mostly as N_2 [31]. Temperature modulates enzymatic rates across the microbial pathways. Due to high variation in concentrations and rapid dynamics, this soil nutrient pool is not often modeled on its own. However, by considering its key proximate source to soil pore water—nitrification—and several typical sinks, including *denitrification*, *leaching*, and *biotic uptake*, a simple model for soil pore water nitrate dynamics can be constructed.

The input variables are $\mathbf{x} = \left[\mathrm{NH_4}^+,\ \mathrm{NO_3}^-,\ \mathrm{O_2},\ \mathrm{CH_2O},\ T,\ \mathrm{pH},\ q\right]$, and the output variable is $y = \mathrm{d}[\mathrm{NO_3}^-]/\mathrm{d}t$. Here, q denotes an effective drainage or flushing rate controlling nitrate leaching losses from the soil profile. The ground-truth equation is

$$\begin{aligned}\frac{\mathrm{d}[\mathrm{NO_3}^-]}{\mathrm{d}t} = &\underbrace{k_{\mathrm{nit}} \cdot \frac{[\mathrm{NH_4}^+]}{K_{\mathrm{NH_4}} + [\mathrm{NH_4}^+]} \cdot \frac{[\mathrm{O_2}]}{K_{\mathrm{O_2}}^{\mathrm{nit}} + [\mathrm{O_2}]} \cdot e^{\theta_{\mathrm{nit}}(T - T_{\mathrm{ref}})}}_{\text{nitrification}} - \underbrace{k_{\mathrm{upt}} \cdot \frac{[\mathrm{NO_3}^-]}{K_{\mathrm{upt}} + [\mathrm{NO_3}^-]}}_{\text{biotic uptake}} \\ &\underbrace{- q \cdot [\mathrm{NO_3}^-]}_{\text{leaching}} \underbrace{- k_{\mathrm{den}} \cdot \frac{[\mathrm{NO_3}^-]}{K_{\mathrm{NO_3}} + [\mathrm{NO_3}^-]} \cdot \frac{[\mathrm{CH_2O}]}{K_C + [\mathrm{CH_2O}]} \cdot \frac{K_{\mathrm{O_2}}^{\mathrm{inh}}}{K_{\mathrm{O_2}}^{\mathrm{inh}} + [\mathrm{O_2}]} \cdot e^{\theta_{\mathrm{den}}(T - T_{\mathrm{ref}})}}_{\text{denitrification}}.\end{aligned} \tag{4}$$

The remaining quantities are model parameters, denoted collectively by $\boldsymbol{\theta} = [k_*,\ K_*,\ \theta_*,\ T_{\mathrm{ref}}]$. The k parameters represent effective maximum process rates for nitrification, denitrification, and biotic uptake. The K parameters denote half-saturation constants, except for $K_{\mathrm{O_2}}^{\mathrm{inh}}$, which acts as a half-inhibition constant for oxygen-limited denitrification.

Results and Analysis. Table 3 shows that LiteSR discovers a more biologically and structurally faithful nitrate dynamics equation than the LLM-SR baseline. In particular, LiteSR correctly recovers the Monod-type substrate dependencies for the key nitrogen and carbon species, including NH_4^+, NO_3^-, and CH_2O, and correctly identifies denitrification as a nitrate-driven sink. In contrast, LLM-SR introduces several structurally and biologically inconsistent terms, including an incorrect NO_2^- dependence in the denitrification pathway and non-Monod substrate forms involving $\min(\cdot)$ and Hill-like exponents.

Both methods fail to fully recover the oxygen Monod term in nitrification and use a shared temperature-

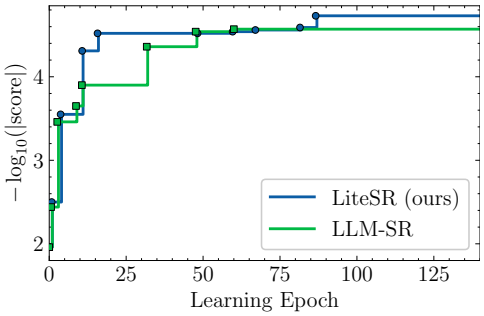

Figure 3: Comparison of the best discovered $d[\mathrm{NO_3}^-]/dt$ equation score over epochs for LiteSR and LLM-SR.

Table 3: Structural comparison of LiteSR and LLM-SR discovered equations against the target $\mathrm{d}[\mathrm{NO_3}^-]/\mathrm{d}t$ equation. Note: ✓ = correct match; ≈ = partial match (correct form, wrong detail); ✗ = missing or incorrect.

Subcomponent	**Ground-truth**	**LiteSR (ours)**		**LLM-SR** [25]	
$\mathrm{NH_4}^+$ Monod kinetics	$\frac{[\mathrm{NH_4}^+]}{K_{\mathrm{NH_4}} + [\mathrm{NH_4}^+]}$	$\frac{p_0 \cdot \mathrm{NH_4}^+}{p_5 + \mathrm{NH_4}^+}$	✓	$\frac{\mathrm{NH_4}^+}{p_7 + [\mathrm{NH_4}^+]^\alpha}$	≈
$\mathrm{NO_3}^-$ Monod kinetics	$\frac{[\mathrm{NO_3}^-]}{K_{\mathrm{NO_3}} + [\mathrm{NO_3}^-]}$	$\frac{p_1 \cdot \mathrm{NO_3}^-}{p_6 + \mathrm{NO_3}^-}$	✓	$\frac{\min(\mathrm{NO_3}^-,\, K_s + [\mathrm{NO_3}^-]^\alpha)}{(K_s + [\mathrm{NO_3}^-]^\alpha)^2}$	✗
$\mathrm{CH_2O}$ Monod kinetics	$\frac{[\mathrm{CH_2O}]}{K_C + [\mathrm{CH_2O}]}$	$\frac{p_2 \cdot \mathrm{CH_2O}}{p_7 + \mathrm{CH_2O}}$	✓	$\frac{\min(\mathrm{CH_2O},\, K_s + [\mathrm{NO_2}^-]^\alpha)}{(K_s + [\mathrm{NO_2}^-]^\alpha)^2}$	✗
$\mathrm{O_2}$ Monod in nitrification	$\frac{[\mathrm{O_2}]}{K^{\mathrm{nit}}_{\mathrm{O_2}} + [\mathrm{O_2}]}$	absent	✗	absent	✗
$\mathrm{O_2}$ inhibition of denitrification	$\frac{K^{\mathrm{inh}}_{\mathrm{O_2}}}{K^{\mathrm{inh}}_{\mathrm{O_2}} + [\mathrm{O_2}]}$	$\frac{p_8}{p_9 + \mathrm{O_2}}$	✓	$\left(\frac{p_4}{p_4 + \mathrm{O_2}}\right)^2$	≈
Arrhenius temperature scaling	$e^{\theta_{\mathrm{nit}}(T - T_{\mathrm{ref}})}$ and $e^{\theta_{\mathrm{den}}(T - T_{\mathrm{ref}})}$ independently	single shared $e^{p_4(T - p_{13})}$	≈	single shared $e^{p_5(T - p_6)}$	≈
Nitrification as source	$+k_{\mathrm{nit}} \cdot f(\mathrm{NH_4}^+,\, \mathrm{O_2})$	$+p_0 \cdot f(\mathrm{NH_4}^+)$	✓	$+k_{\mathrm{nit}} \cdot f(\mathrm{NH_4}^+)$	✓
Denitrification substrate	$\mathrm{NO_3}^-$	$\mathrm{NO_3}^-$	✓	$\mathrm{NO_2}^-$	✗
Denitrification as sink	$-k_{\mathrm{den}} \cdot f(\mathrm{NO_3}^-,\, \mathrm{CH_2O})$	$-f(\mathrm{NO_3}^-,\, \mathrm{CH_2O})$	✓	$-f(\mathrm{NO_2}^-,\, \mathrm{CH_2O})$	≈
$\mathrm{O_2}$ inhibition form	pure Monod: $\frac{K^{\mathrm{inh}}_{\mathrm{O_2}}}{K^{\mathrm{inh}}_{\mathrm{O_2}} + [\mathrm{O_2}]}$	Monod + $e^{-p_3 \mathrm{O_2}}$ hybrid	✗	Monod squared	≈
DNRA pathway	absent	explicit $\frac{p_1 \cdot \mathrm{NO_3}^-}{p_6 + \mathrm{NO_3}^-}$ sink	✗	explicit sink	✗
Separate θ per process	$\theta_{\mathrm{nit}} \neq \theta_{\mathrm{den}}$	single θ	✗	single θ	✗
pH dependence	absent	absent	✓	absent	✓
$\mathrm{NO_2}^-$ dependence	absent	absent	✓	denitrification substrate	✗
Exact match (%)		**9/14 (64.3%)**		**6/14 (42.9%)**	

scaling factor rather than separate temperature dependencies for nitrification and denitrification. Nevertheless, LiteSR better preserves the core source–sink structure of nitrate cycling and avoids the biologically incorrect substrate substitution produced by LLM-SR. This suggests that literature-guided retrieval provides useful mechanistic constraints that guide symbolic regression toward more interpretable and scientifically plausible equations.

Figure 3 further shows that LiteSR converges more rapidly than LLM-SR. In particular, LiteSR crosses the 10^{-5} residual threshold at epoch 11, whereas LLM-SR reaches the same threshold only at epoch 32. This faster convergence indicates that the retrieved scientific context not only improves structural fidelity but also helps narrow the search space toward high-quality candidate equations earlier in the optimization process.

5 Conclusion

In this paper, we presented LiteSR, a literature-guided agentic retrieval framework for symbolic regression. LiteSR retrieves, filters, and synthesizes equation-level knowledge from scientific literature, and uses the resulting mechanistic context to guide the search for candidate symbolic expressions. This design moves beyond purely data-driven search by incorporating domain-relevant source–sink structures, kinetic forms, and physical constraints into the equation discovery process.

Across three case studies, LiteSR consistently produced equations with higher structural fidelity to the target dynamics and faster convergence than the baseline. These results show that literature-derived equation context can provide effective mechanistic guidance for symbolic regression, improving both interpretability and search efficiency. More broadly, our findings suggest that coupling scientific retrieval with symbolic regression is a practical path toward reliable, knowledge-grounded equation discovery in complex scientific domains.

Acknowledgment

This work was supported by the Center for Healthy and Efficient Mobility (CHEM) under grant 69A3552348329. Nan Jiang acknowledges support from the Texas Advanced Computing Center (TACC) under award CCR25054.

References

[1] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *ICLR*. OpenReview.net, 2024.

[2] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.*, 113(15):3932–3937, 2016.

[3] S. C. Chapra. *Surface Water-Quality Modeling*. McGraw-Hill, New York, 1997. ISBN 9780070113640.

[4] M. Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl. *arXiv preprint arXiv:2305.01582*, 2023.

[5] A. Darrouzet-Nardi and W. D. Bowman. Hot spots of inorganic nitrogen availability in an alpine-subalpine ecosystem, colorado front range. *Ecosystems*, 14:848–863, 2011.

[6] S. d'Ascoli, S. Becker, A. Mathis, P. Schwaller, and N. Kilbertus. Odeformer: Symbolic regression of dynamical systems with transformers. In *ICLR*. OpenReview.net, 2024.

[7] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A*, 478(2260):20210904, 2022.

[8] B. Gec, N. Omejc, J. Brence, S. Dzeroski, and L. Todorovski. Discovery of differential equations using probabilistic grammars. In *DS*, volume 13601, pages 22–31. Springer, 2022.

[9] B. He, Q. Lu, Q. Yang, J. Luo, and Z. Wang. Taylor genetic programming for symbolic regression. In *GECCO*, pages 946–954, 2022.

[10] N. Jiang, M. Nasim, and Y. Xue. Vertical symbolic regression via deep policy gradient. In *IJCAI*, pages 5891–5899. ijcai.org, 2024.

[11] N. Jiang, M. Nasim, and Y. Xue. Active symbolic discovery of ordinary differential equations via phase portrait sketching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17626–17634, 2025.

[12] N. Jiang, Z. Wang, and Y. Xue. EGG-SR: Embedding symbolic equivalence into symbolic regression via equality graph. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=oh9ChF7Pv0`.

[13] L. S. Keren, A. Liberzon, and T. Lazebnik. A computational framework for physics-informed symbolic regression with straightforward integration of domain knowledge. *Scientific Reports*, 13(1):1249, 2023.

[14] J. Lála, O. O'Donoghue, A. Shtedritski, S. Cox, S. G. Rodriques, and A. D. White. Paperqa: Retrieval-augmented generative agent for scientific research. *CoRR*, abs/2312.07559, 2023.

[15] A. J. Miller, X. Fan, M. Orsel, S. J. Smith, and D. M. Wells. Nitrate transport and signalling. *Journal of Experimental Botany*, 58:2297–2306, 2007.

[16] M. J. Mitchell, O. E. Jensen, K. A. Cliffe, and M. M. Maroto-Valer. A model of carbon dioxide dissolution and mineral carbonation kinetics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2117):1265–1290, 2010.

[17] F. D. Murnaghan. Finite deformations of an elastic solid. *American Journal of Mathematics*, 59 (2):235–260, 1937.

[18] J. M. Norton and J. M. Stark. Chapter fifteen - regulation and measurement of nitrification in terrestrial systems. In M. G. Klotz, editor, *Methods in Enzymology*, pages 343–368. Academic Press, 2011.

[19] M. Peleg, M. D. Normand, and M. G. Corradini. The arrhenius equation revisited. *Critical reviews in food science and nutrition*, 52(9):830–851, 2012.

[20] Z. Qian, K. Kacprzyk, and M. van der Schaar. D-CODE: discovering closed-form odes from observed trajectories. In *ICLR*. OpenReview.net, 2022.

[21] T. Sakai, H. Kadobayashi, Y. Nakamoto, H. Dekura, N. Ishimatsu, S. Kawaguchi-Imada, Y. Seto, O. Sekizawa, K. Nitta, and K. Shimizu. The equations of state of nine materials up to 0.43 tpa for extreme pressure science. *Communications Materials*, 6(1):68, 2025.

[22] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*, 2023.

[23] T. Schick, J. Dwivedi-Yu, R. Dessi, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

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

[25] P. Shojaee, K. Meidani, S. Gupta, A. B. Farimani, and C. K. Reddy. LLM-SR: scientific equation discovery via programming with large language models. In *ICLR*, 2025.

[26] M. D. Skarlinski, S. Cox, J. M. Laurent, J. D. Braza, M. M. Hinks, M. J. Hammerling, M. Ponnapati, S. G. Rodriques, and A. D. White. Language agents achieve superhuman synthesis of scientific knowledge. *CoRR*, abs/2409.13740, 2024.

[27] F. Sun, Y. Liu, J. Wang, and H. Sun. Symbolic physics learner: Discovering governing equations via monte carlo tree search. In *ICLR*, 2023.

[28] B. Taskin, W. Xie, and T. Lazebnik. Knowledge integration for physics-informed symbolic regression using pre-trained large language models. *Scientific Reports*, 16(1):1614, 2026.

[29] S.-M. Udrescu and M. Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16), 2020.

[30] P. Vinet, J. R. Smith, J. Ferrante, and J. H. Rose. Temperature effects on the universal equation of state of solids. *Physical Review B*, 35(4):1945, 1987.

[31] K. L. Weier, J. W. Doran, J. F. Power, and D. T. Walters. Denitrification and the dinitrogen/nitrous oxide ratio as affected by soil water, available carbon, and nitrate. *Soil Science Society of America Journal*, 57:66–72, 1993.

[32] T. Wu and M. Tegmark. Toward an artificial intelligence physicist for unsupervised learning. *Phys. Rev. E*, 100:033311, Sep 2019.

[33] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. In *ICLR*. OpenReview.net, 2023.

[34] L. S. Yifei, A. Chang, C. Malaviya, and M. Yatskar. ResearchQA: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *Transactions of the Association for Computational Linguistics*, 2026.

[35] H. Zhang, Q. Chen, B. XUE, W. Banzhaf, and M. Zhang. RAG-SR: Retrieval-augmented generation for neural symbolic regression. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] S. Zhang and G. Lin. Robust data-driven discovery of governing physical laws with error bars. *Proc Math Phys Eng Sci.*, 474(2217), 2018.

A Related Work

Agentic Language Models and Tool Use. Recent work has moved beyond static prompting toward *agentic* language models that interleave reasoning with actions. ReAct [33] showed that alternating reasoning traces with tool invocation improves multi-step reasoning. Toolformer [22] demonstrated that language models can learn, through self-supervision, when to call external APIs during generation. Self-RAG further incorporated retrieval-aware self-reflection, allowing models to critique and revise retrieved evidence before producing an answer.

However, most agentic frameworks use retrieved information as unstructured text appended to the prompt. The dominant pattern is still *context injection*, rather than conversion into typed intermediate representations. Our work departs from this design by enforcing a structured interface between literature retrieval and downstream symbolic reasoning.

Scientific Literature Retrieval and Synthesis. Retrieval-augmented systems have recently been adapted to scientific domains. PaperQA [14] introduced full-text scientific retrieval with citation-grounded answer synthesis, showing that literature-aware agents can improve scholarly question answering. Later systems extended this direction with stronger retrieval, citation-graph traversal, and attribution-aware synthesis. Large-scale scientific synthesis models further incorporate structured outlining, contradiction detection, and rubric-based evaluation. Benchmarks such as LitQA2 [26] and ResearchQA [34] evaluate literature-grounded question answering across disciplines, with an emphasis on citation precision and factual support.

Despite this progress, literature agents are mainly evaluated for narrative correctness, coverage, and attribution. They are rarely assessed by whether they can produce executable scientific hypotheses or structured constraints for downstream modeling. Their outputs are typically textual summaries, not programmatic artifacts. In contrast, our framework treats literature retrieval as prior acquisition for symbolic search. Retrieved content is converted into structured constraints, including operator libraries, invariances, and variable schemas, rather than used as free-form context.

Symbolic Regression and Equation Discovery. Symbolic regression aims to discover interpretable functional forms that explain observed data. Classical approaches rely on genetic programming and evolutionary search. AI Feynman [29] introduced physics-inspired heuristics, such as dimensional analysis and separability detection, to reduce the search space. PySR [4] provides a high-performance evolutionary symbolic regression system with modern optimization backends.

Neural approaches further guide equation discovery with learned priors. Transformer-based methods such as ODEFormer [6] formulate equation generation as sequence prediction, enabling neural-guided discovery for dynamical systems. Deep symbolic regression methods similarly combine neural proposal models with symbolic search or evolutionary refinement.

LLM-SR [25] reframed symbolic regression as program synthesis with large language models. Equations are represented as executable programs with placeholder parameters: the LLM proposes candidate structures, numerical optimization fits coefficients, and an experience buffer guides iterative refinement. This hybrid of generative priors and numerical validation marks an important shift in symbolic regression.

However, existing LLM-based symbolic regression methods rely mainly on manually written problem descriptions. They do not systematically extract domain-specific priors from scientific literature before search. As a result, the search process remains largely data-driven, with domain knowledge entering only implicitly through the pretrained model or manually specified prompts.

B Implementation of LiteSR

Agents Configuration. Our evolutionary search framework uses Claude (claude-sonnet-4-6) as the primary LLM backend for hypothesis generation and refinement.

Retrieval Agent. To ground hypothesis generation in domain literature, we employ a pipeline using nomic-embed-text via Ollama for local embeddings, with vector database Milvus.

Literature Sources. Papers were sourced through keyword searches across Nature Communications, Royal Society Publishing, APS, and arXiv. Domain experts additionally helped collect relevant literature from these sources.

Hardware and Runtime. No specialized GPU hardware is required, as all generation is handled via the Anthropic API. A standard CPU machine with Ollama installed for local embeddings is sufficient. The search runs for at most 150 epochs with a practical runtime of approximately 2–3 hours.

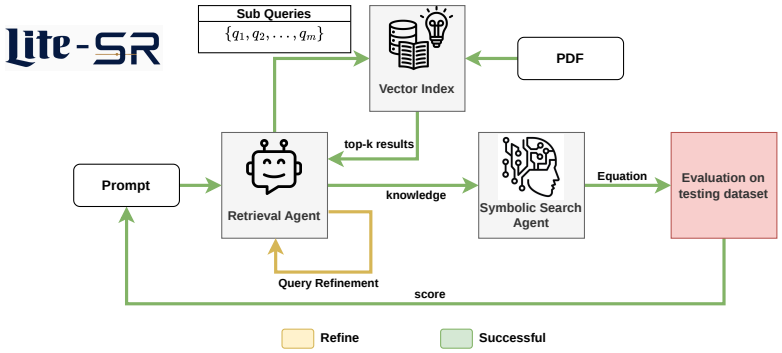

Figure 4: The pipeline of the proposed LiteSR.

As shown in Figure 4, to efficiently navigate the search landscape and avoid local minima, LiteSR adopts the islands-based experience buffer of LLM-SR [25]. A dynamic experience buffer, $\mathcal{P}_t = \{\mathcal{P}_t^i\}_{i=1}^m$ of m independently evolving islands is maintained, each storing pairs (f, s) of equation skeletons and their fitness scores. At each iteration t, newly evaluated hypotheses $\mathcal{F}_t$ are added to their source island $\mathcal{P}_t^i$ only if they improve upon its current best score s_{best}^i. The worst-performing $m/2$ islands are periodically reset and reseeded from surviving islands. Each island shares a common literature context $\mathcal{C}$, which is injected into every prompt. Upon island reset, or when score improvement across all islands stagnates, $\mathcal{C}$ is refreshed by triggering a new retrieval cycle in $\mathcal{A}_R$, ensuring that the literature context remains aligned with the evolving hypothesis space.

B.1 Evaluation Metric

We evaluate each discovered equation using a numerical residual score computed against the target dynamics on sampled input states. Specifically, for a candidate equation $\hat{f}(\mathbf{x})$ and the ground-truth right-hand side $f(\mathbf{x})$, we compute

$$\texttt{fitness}(\hat{f}, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i) \right|, \tag{5}$$

where $\{\mathbf{x}_i\}_{i=1}^N$ are sampled evaluation points. We report $-\log_{10}(\texttt{fitness})$ so that larger values correspond to smaller residual error and therefore better agreement with the target equation.