BITEMPO: ADAPTIVE TWO-TIMESCALE LATENT INFERENCE WITH LANGUAGE MODELS FOR SCIENTIFIC EQUATION DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific equation discovery requires searching over symbolic structures whose quality is revealed only after their constants are fitted to data. Language-model-based methods typically carry feedback forward as example programs or textual memory, without explicitly separating persistent structural preferences from local corrections in their search state. We propose BITEMPO, a test-time latent inference framework that accumulates numerical feedback in factorized global and local distributions over generation preferences. The global block encodes variables, operators, and composition, while the local block encodes refinement directions informed by training residuals. Sampled latents condition program generation and compatible-parent retrieval; each evaluated equation contributes evidence for distribution updates, with the local block adapting faster than the global one. A lightweight utility surrogate ranks latent proposals before language-model calls, without fine-tuning the language model. On the four LSR-Synth domains of LLM-SRBench, BITEMPO with 400 candidates improves domain-averaged Acc@0.01 over the strongest baseline from 25.79% to 52.81% with Llama-3.1-8B and from 24.58% to 34.08% with Qwen3-4B. On measured stress–strain data, it reduces ID and OOD NMSE by 48.6% and 61.0% relative to the strongest baseline. Ablations show that removing distribution updates or collapsing the two timescales lowers ID accuracy from 45.00% to 30.00% and 33.33%, respectively.

1 INTRODUCTION

Mathematical equations express scientific hypotheses in an inspectable and testable form (Schmidt & Lipson, 2009). Symbolic regression infers an equation's structure and numerical constants from observations (La Cava et al., 2021). The difficulty lies in searching a large space of structures whose predictive quality depends on parameter estimation. Evolutionary methods maintain populations of expressions and revise their structures through variation and selection, interleaved with constant optimization (Cranmer, 2023). Neural methods learn a data-to-expression mapping from synthetic equations, then use numerical optimization to estimate the generated structures' constants (Biggio et al., 2021).

Large language models expand the range of candidate equations that can be expressed as executable programs. LLM-SR combines this generation capability with numerical parameter estimation and iterative search (Shojaee et al., 2025a). Its generation–evaluation loop proposes a program, fits its constants, and uses evaluated examples to guide subsequent generation. Under a finite generation budget, a central challenge is to accumulate this feedback into a search strategy that improves as more candidates are evaluated.

An experience buffer preserves useful programs as examples. Their evaluations can also inform decisions that recur across different programs, such as which variables to combine or which structural corrections to explore. We call these decisions generation preferences. A latent representation gives these preferences shared coordinates, allowing numerical feedback to update a distribution over future generation strategies rather than only rank existing equations. Latent-guided sequence models similarly infer representations to direct generation toward desired properties (Kong et al., 2024a; 2025). For equation discovery, the evidence is the quality of fitted programs and the structure of their

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.

remaining errors. BITEMPO uses this evidence to adapt a task-specific latent distribution alongside a program archive. The controller learns within the current search run; neither offline controller training nor language-model parameter updates are required.

A latent specifies preferences, not a unique equation. Stochastic generation and numerical fitting therefore make each candidate's score a noisy observation of the strategy that produced it. Aggregating feedback across evaluated candidates shifts the sampling distribution toward preferences associated with useful equations. Structural directions and local corrections, however, serve different roles. A promising structural direction should persist while its candidates undergo refinement; residual evidence can support a local correction without justifying a change in the overall direction. We therefore separate global structural preferences from local refinement preferences. Both blocks use the same candidate history and elite-selection rule. Local preferences update frequently, while global updates require more new observations, preserving structural continuity across successive corrections.

We present BITEMPO, a **test-time latent inference framework for scientific equation discovery**. Its central idea is to accumulate equation evaluations in a factorized global–local representation: slower structural updates preserve a search direction, while faster local updates respond to opportunities for correction. Our main contributions are:

1. **Scientific equation discovery through adaptive latent inference.** We represent generation preferences as a task-specific latent distribution inferred from evaluated programs, without offline training or language-model parameter updates. This representation accumulates feedback across syntactically different candidates that share structural or refinement preferences, providing persistent search state beyond the reuse of individual programs as examples.
2. **Global–local factorization with two-timescale adaptation.** We separate persistent structural preferences from responsive local refinement preferences and update the two blocks at different rates from shared candidate evidence. Residual information further guides local corrections, allowing frequent refinement without immediately redirecting the global search distribution.
3. **Latent-guided generation, retrieval, and acquisition.** The same sampled latent guides program generation and compatible-parent retrieval, coupling generation preferences with relevant examples. A utility surrogate compares latent proposals using predicted utility and uncertainty before generation, guiding acquisition without additional language-model calls.

We examine predictive precision and latent adaptation through LLM-SRBench comparisons, component studies, and search trajectories. Structural recovery cases and measured stress–strain data assess the scientific dependencies captured by the discovered equations.

2 RELATED WORK

Symbolic regression and scientific equation discovery. Classical genetic programming evolves expression structures through selection and variation (Koza, 1992); PySR combines evolutionary search with numerical optimization (Cranmer, 2023), and AI Feynman exploits function properties to simplify certain discovery problems (Udrescu & Tegmark, 2020). Neural symbolic regression instead learns a generation policy from task rewards or a corpus of synthetic equations, e.g. risk-seeking policy gradients (Petersen et al., 2021) or end-to-end Transformers trained across generated equation datasets (Biggio et al., 2021; Kamienny et al., 2022). These approaches couple generation to execution and selection. BITEMPO uses numerical feedback to adapt a task-local distribution of generation preferences while keeping the pretrained language model fixed, with separate update schedules for structural and local refinement preferences.

Language models for symbolic program generation. LLM-SR couples language-model program proposals to numerical evaluation and an experience buffer (Shojaee et al., 2025a); BITEMPO builds on this generation-and-evaluation backbone and adds an explicit latent inference state. Agentic approaches instead organize feedback through reasoning, diagnosis, and memory: Deliberate Evolution (DE) combines adaptive search operators, structural diagnosis, and reflective memory (Pang et al., 2026), and A-SR explores role-conditioned evidence and multi-agent coordination (Zhao et al., 2026). BITEMPO contributes a compact statistical representation of generative preferences, so that numerical feedback updates explicit global and local latent distributions and structural exploration and local refinement can adapt at distinct rates. Residual-based improvement also has a broader history in gradient boosting (Friedman, 2001) and, more recently, in RESTART's combination of equation

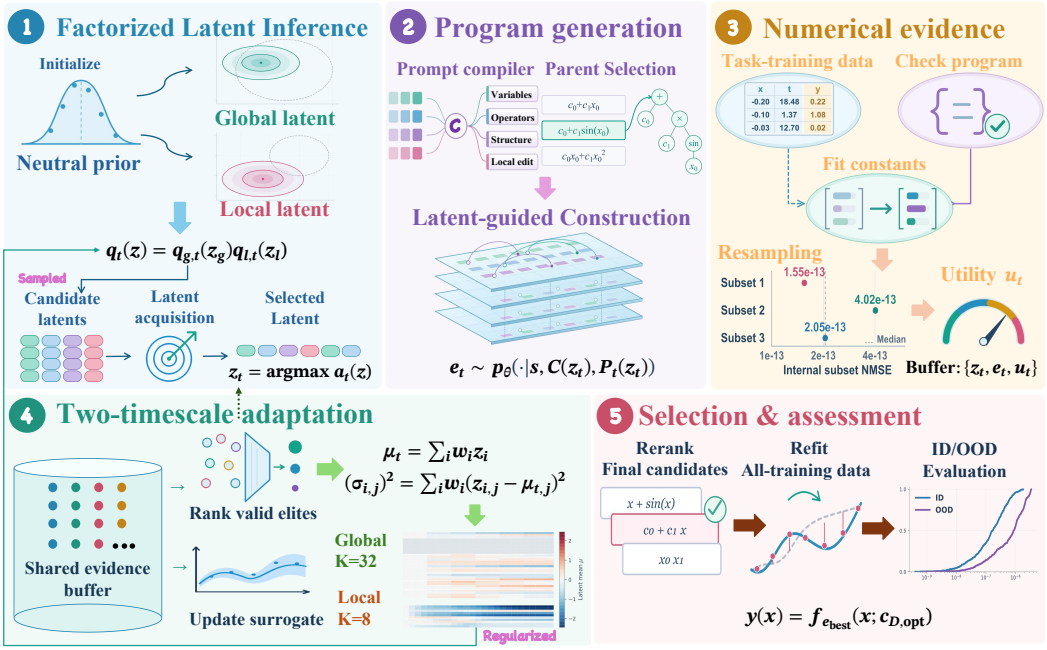

Figure 1: Overview of BITEMPO. A factorized global–local latent policy guides program generation and parent retrieval. Numerical feedback adapts the two blocks on different timescales, linking evaluated equations to subsequent proposals, followed by training-based final selection and external assessment.

refinement with reusable structural knowledge (Li & Pan, 2026); in BITEMPO residual evidence specifically guides the local distribution over refinement preferences.

3 METHOD

3.1 PROBLEM SETUP

Given observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, scientific equation discovery seeks a symbolic expression and its numerical parameters. Following LLM-SR (Shojaee et al., 2025a), we represent a candidate as an executable equation program $f_e(x; c)$, where e specifies its structure and $c \in \mathcal{C}_e$ its free constants. The fitting objective is

$$\min_{e \in \mathcal{E},\ c \in \mathcal{C}_e} \frac{1}{n} \sum_{i=1}^{n} \big(f_e(x_i; c) - y_i\big)^2, \tag{1}$$

where $\mathcal{E}$ is the set of admissible equation programs. Under a finite candidate budget, a language model proposes structures, a numerical optimizer estimates their constants, and training-based evaluation guides further search. Eq. (1) defines the underlying regression objective; candidate feedback and final selection use the resampling-based criteria in §3.3 and §3.6. Supplied external validation and OOD data are reserved for assessment.

3.2 FACTORIZED GLOBAL–LOCAL LATENT REPRESENTATION

An evaluated equation provides evidence both about a structural direction and about aspects of the response that remain unexplained. BITEMPO associates this evidence with the latent preferences that guided generation, and separates persistent structural preferences from responsive refinement preferences. A candidate may capture the broad dependence of a system while still requiring a local correction; revising that correction need not redirect the structural search. Each step samples a latent, uses it to guide generation and parent retrieval, evaluates the generated program, and updates the corresponding preference distribution. Figure 1 illustrates this feedback loop.

The state for one task comprises evaluated-program history, a parent archive, a latent sampling distribution, and a utility surrogate; each task starts from a freshly initialized state. The 40 latent coordinates are grouped as $z = (z_g, z_l)$, $z_g \in \mathbb{R}^{30}$, $z_l \in \mathbb{R}^{10}$ (dimension breakdown in Appendix C, Table 6), with diagonal distribution

$$q_t(z) = q_{g,t}(z_g)\, q_{l,t}(z_l), \qquad q_{b,t} = \mathcal{N}(\mu_{b,t}, \mathrm{diag}(\sigma^2_{b,t})), \qquad b \in \{g, l\}. \tag{2}$$

We initialize active coordinates with a standard Gaussian and bound the sampled latent vectors; core settings appear in Appendix C. A semantic compiler translates z into generation guidance $C(z)$, specifying preferred variables, operators, interactions, and refinements. These preferences guide the language model rather than prescribe an expression tree or the complete set of admissible operations. The same continuous latent scores compatibility with archived programs to retrieve parents $P_t(z)$. Consequently, latent control acts through both generation guidance and the examples supplied to the model; identical compiled guidance can still accompany different retrieved parents. Given a task specification s containing the problem description and function interface, the frozen generator produces $e \sim p_\theta(\cdot \mid s, C(z), P_t(z))$ autoregressively.

Let $u(e)$ denote the training-based utility of a fitted candidate (§3.3). The value of a latent at step t is the expected utility of the programs it leads the generator to produce, $V_t(z) = \mathbb{E}_{e\sim p_\theta(\cdot|s,C(z),P_t(z))}[u(e)]$, and a useful search distribution places more mass on high-value z. BITEMPO never estimates $V_t(z)$ for a fixed z by repeated sampling: each evaluated candidate is one noisy observation of the value of the latent that guided it. The cross-entropy method (CEM) fits a sampling distribution from exactly such observations, repeatedly sampling, evaluating, and refitting toward a selected elite set (Rubinstein, 1999; de Boer et al., 2005); §3.4 adapts it to the factorized, two-timescale setting.

3.3 CONSTANT OPTIMIZATION AND RESAMPLING-BASED EVALUATION

Each generated program undergoes interface and numerical-validity checks before fitting. The numerical backend estimates up to ten bounded constants through multi-start optimization. When a program is affine in some constants conditional on the others, variable projection solves the affine subproblem within the nonlinear fit (Golub & Pereyra, 1973; O'Leary & Rust, 2013). Training resampling then measures prediction error across internal subsets. Subset fits can use the full-task-training fit as a warm start; all feedback remains within the task-training data. Appendix C.7 specifies the normalized subset error E_k and the fitting protocol.

Let E_{med} and $E_{\max}$ be the median and maximum subset errors, and let S_E denote the standard deviation of their floored base-ten logarithms. We combine accuracy and consistency into

$$u(e) = -\log_{10} \widetilde{E}_{\mathrm{med}} - \alpha_w \log_{10} \frac{\widetilde{E}_{\max}}{\widetilde{E}_{\mathrm{med}}} - \alpha_s S_E, \qquad \widetilde{E} = \max\{E, \varepsilon\}. \tag{3}$$

The floor $\varepsilon = 10^{-15}$ prevents divergent utility at zero error; it applies only to internal utility, not reported prediction errors. Invalid or non-finite evaluations consume a budget slot, receive utility -20, and are excluded from the valid elite set. Positive weights α_w, α_s penalize a difficult subset and dispersion, respectively. Training residuals additionally provide six family summaries r, obtained from correlations with polynomial, trigonometric, saturating, and exponential probes, to inform local refinement.

3.4 TWO-TIMESCALE ADAPTATION

Let N_t count observed latent-controlled proposals and N_b the count at the last update of block $b \in \{g, l\}$. A block is eligible when $N_t - N_b \geq K_b$, with $K_g = 32$ and $K_l = 8$, provided at least six distinct latent proposals have been observed, including three with valid, finite evaluations. Both blocks use the same recent-observation pool and elite-selection rule. For elite latents $\mathcal{E}_t$ with normalized rank weights w_i, the raw targets are

$$\bar{\mu}_b = \sum_{i\in\mathcal{E}_t} w_i z_{i,b}, \qquad \bar{v}_b = \sum_{i\in\mathcal{E}_t} w_i (z_{i,b} - \bar{\mu}_b)^{\odot 2}, \qquad \sum_{i\in\mathcal{E}_t} w_i = 1, \tag{4}$$

where $\odot 2$ denotes coordinatewise squaring. These moments define the weighted Gaussian fit before residual adjustment and regularization; a variance floor preserves nonzero spread. For the local

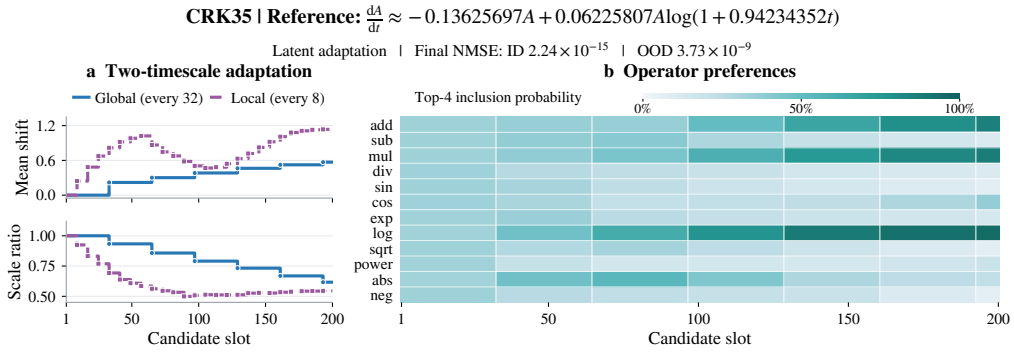

Figure 2: CRK35 latent adaptation: global/local mean and scale changes (a), and operator-inclusion probabilities (b).

residual coordinates, positive training-residual evidence adjusts the mean targets, while edit-action targets remain elite-based.

Write $\eta_b = (\mu_b, \sigma_b)$ for the distribution parameters and η_b^{tar} for the residual-adjusted target after displacement and scale-change limits. The eligible block follows

$$\eta_b^{+} = \Pi_{\mathcal{B}}\big((1-\lambda_0)\big[(1-\rho)\eta_b + \rho\eta_b^{\mathrm{tar}}\big] + \lambda_0\eta_{b,0}\big)\,, \tag{5}$$

where ρ smooths adaptation, λ_0 shrinks toward the initial parameters, and $\Pi_{\mathcal{B}}$ enforces the allowed parameter ranges. Ineligible blocks remain unchanged. This separates evidence aggregation from update regularization: local preferences can respond repeatedly while the global structural direction persists. Appendix D gives the inference interpretation and establishes the raw fitting and bounded-update properties.

3.5 Utility-Guided Latent Acquisition

A ridge-regression surrogate (Hoerl & Kennard, 1970) predicts utility from the compiler's binary feature vector $\phi(z) \in \{0,1\}^{40}$ and estimates uncertainty from the observed fitting residuals. After 16 latent observations, we draw a set of latent drafts $\mathcal{Z}_t$ and acquire

$$z_t^{\star} = \arg\max_{z \in \mathcal{Z}_t} \big[\widehat{u}_t(z) + \kappa\, \widehat{s}_t(z)\big], \tag{6}$$

where $\widehat{u}_t$ is predicted utility, $\widehat{s}_t$ an uncertainty score, and κ the exploration weight. This upper-confidence-style criterion balances predicted quality and exploration (Abbasi-Yadkori et al., 2011); it does not require a calibrated confidence bound. Only the selected latent invokes the decoder. Appendix C.6 details the surrogate fit.

3.6 Final Selection and Parameter Estimation

At candidate budget B, stronger axis-based resampling evaluates a shortlist $\mathcal{S}_B$ of up to four existing candidates. A hierarchical tolerance rule prioritizes worst-subset error, log-error dispersion, and typical error, followed by fitting discrepancy and complexity as tie-breakers. The selected structure is refitted using all task-training observations:

$$\widehat{e} = \mathrm{Select}_{\mathrm{hier}}(\mathcal{S}_B; \mathcal{D}), \qquad \widehat{c} \approx \arg\min_{c \in \mathcal{C}_{\widehat{e}}} \mathrm{MSE}_{\mathcal{D}}(\widehat{e}, c). \tag{7}$$

External ID and OOD data are used only to assess the resulting equation. Algorithm 1 summarizes the search, and Appendix C specifies fitting strengths and failure handling.

4 Experiments

Our experiments examine predictive precision across scientific domains, the role of latent-inference components, and the behavior of discovered equations on scientific data.

Table 1: ID performance on the four LSR-Synth domains. NMSE is an arithmetic mean ↓; Acc@0.01 is a percentage ↑. Method suffixes denote candidate budgets. Bold and underline indicate the best and second-best values within each backbone, including ties.

	Physics		Material		Chemistry		Biology	
Method	NMSE	Acc@0.01	NMSE	Acc@0.01	NMSE	Acc@0.01	NMSE	Acc@0.01
Llama-3.1-8B-Instruct								
LLMDirect	9.95e-3	0.00	8.16e-2	24.00	1.22e+0	2.78	1.29e-1	4.17
LLM-SR	3.00e-3	6.82	2.16e-1	60.00	5.24e-2	16.67	1.76e-2	12.50
LASR	6.07e-3	9.09	9.47e-4	32.00	1.82e-3	8.33	6.40e-3	0.00
SGA	1.55e-1	2.27	4.35e-2	12.00	4.58e-2	8.33	2.42e-1	0.00
DE	**1.01e-3**	11.36	2.89e-4	64.00	4.16e-4	11.11	1.17e-2	16.67
BiTempo-200	2.96e-3	22.73	6.53e-2	**68.00**	**6.43e-6**	69.44	8.60e-3	**41.67**
BiTempo-400	1.14e-2	**25.00**	**1.03e-4**	64.00	3.16e-4	**80.56**	**5.02e-3**	**41.67**
Qwen3-4B-Instruct-2507								
LLMDirect	5.46e-2	6.82	1.42e-3	52.00	2.66e-1	2.78	4.46e-2	8.33
LLM-SR	2.51e-3	6.82	3.55e-3	44.00	3.36e-2	13.89	1.88e-2	12.50
LASR	6.04e-3	6.82	6.21e-4	8.00	2.31e-3	2.78	9.56e-3	0.00
SGA	1.04e-1	0.00	1.02e-2	16.00	1.61e-1	2.78	1.73e-1	4.17
DE	**4.37e-4**	**15.91**	**1.47e-4**	**56.00**	**1.88e-4**	13.89	**6.69e-3**	12.50
BiTempo-400	3.07e-2	**15.91**	1.29e-1	44.00	8.45e-3	**47.22**	9.68e-3	**29.17**

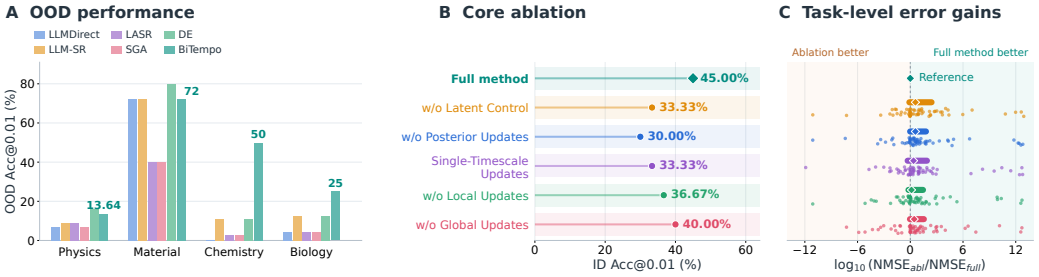

Figure 3: Performance and component analysis. A: Qwen OOD accuracy. B: Llama ID ablation accuracy. C: paired ID log-NMSE ratios; positive values favor BiTempo.

Benchmark and baselines. We evaluate on LLM-SRBench under its ID and OOD protocol (Shojaee et al., 2025b). We focus the main comparison on the four LSR-Synth domains. Prior work suggests that equation familiarity may influence performance on some LSR-Transform tasks (Xia et al., 2026); we retain LSR-Transform for equation-recovery cases. The baselines are direct LLM generation, LLMDirect (Shojaee et al., 2025b), LLM-SR (Shojaee et al., 2025a), LASR (Grayeli et al., 2024), SGA (Ma et al., 2024), and Deliberate Evolution (Pang et al., 2026). We use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen3-4B-Instruct-2507 (Yang et al., 2025; Qwen Team, 2025), abbreviated Llama and Qwen.

Implementation and evaluation. Following the experimental configuration of Deliberate Evolution (Pang et al., 2026), we use 400 candidate samples for DE and 1,000 for LLM-SR, LASR, and SGA. We evaluate BiTempo at 200 and 400 candidates, denoted BiTempo-200 and BiTempo-400. Each candidate occupies one budget slot, including invalid or timed-out proposals. For an evaluation set S, NMSE is the mean squared prediction error divided by $\max\{\mathrm{Var}_S(y), 10^{-12}\}$, and $R^2_S = 1 - \mathrm{NMSE}_S$. NMSE is computed over all evaluation points. Following the accuracy-to-tolerance convention (Biggio et al., 2021; Kamienny et al., 2022; Pang et al., 2026), Acc@0.01 measures the fraction of tasks with relative error at most 1% on at least 95% of points, with $|\hat{y} - y| \leq 10^{-12}$ required when $|y| \leq 10^{-12}$. Domain-level NMSE is the arithmetic mean across tasks; cross-domain accuracy averages assign equal weight to the four scientific domains.

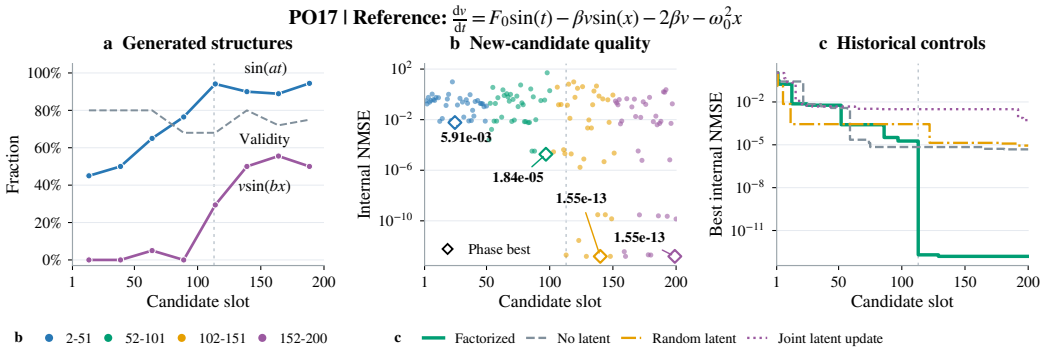

Figure 4: PO17 search dynamics: structural motifs and validity (a), candidate error (b), and best-so-far error (c).

4.1 Main Results

Table 1 demonstrates the broad accuracy gains of BiTempo across scientific domains. With Llama, BiTempo-200 outperforms all baselines in Acc@0.01 in each of the four domains, achieving a domain-average accuracy of 50.46% compared with 25.79% for DE. Increasing the budget to 400 candidates raises the average to 52.81%, with Chemistry reaching 80.56% and Physics 25.00%, while Biology maintains 41.67% at both budgets. The improvements also extend to prediction precision: BiTempo-400 obtains the lowest Llama mean NMSE in Material and Biology, and BiTempo-200 achieves the lowest in Chemistry at 6.43×10^{-6}. These results show that the method can improve both equation-level success and numerical accuracy across different scientific settings. The accuracy gains carry over to Qwen, where BiTempo reaches a domain average of 34.08% versus 24.58% for DE, leads in Chemistry and Biology, and ties the strongest baseline in Physics. Figure 3A further shows Qwen OOD accuracy of 50.00% in Chemistry and 25.00% in Biology, indicating that successful discovery extends to shifted-domain observations. Performance remains domain- and metric-dependent: DE retains lower mean NMSE in Llama Physics and across the Qwen domains, and higher Qwen OOD accuracy in Material and Physics. Thus, the strongest evidence for BiTempo is its broad improvement in equation-level accuracy, complemented by leading numerical precision in several Llama domains.

4.2 Ablation Study

Figure 3B–C compares five components on the same 60 tasks, with Llama and 200 candidates per task. *w/o Latent Control* removes latent guidance; *w/o Distribution Updates*, labeled *w/o Posterior Updates* in the figure, tests latent diversity without feedback-driven adaptation. *Single-Timescale Updates* adapts the global and local preferences jointly on one schedule. *w/o Local Updates* retains only global adaptation, while *w/o Global Updates* retains only local adaptation. The latter two correspond to Global-only and Local-only, respectively.

BiTempo reaches 45.00% ID accuracy, compared with 33.33% without latent control, 30.00% without distribution updates, and 33.33% with single-timescale updates. Global-only and Local-only achieve 36.67% and 40.00%, respectively, supporting the combination of persistent structural adaptation and responsive local refinement. Figure 3C further compares paired task errors through $\log_{10}(\mathrm{NMSE}_{\mathrm{abl}} / \mathrm{NMSE}_{\mathrm{full}})$: positive values indicate lower error for the full method, diamonds mark medians, and thick segments indicate interquartile ranges. The complete configuration achieves lower ID NMSE on 39, 43, 41, 36, and 40 of the 60 tasks against the five variants in the order above. Thus the accuracy improvement is accompanied by improvements across individual tasks, rather than only an aggregate shift.

Inference dynamics. Figure 4 examines PO17, whose reference relation is $\dot{v} = F_0 \sin(t) - \beta v \sin(x) - 2\beta v - \omega_0^2 x$. Panel a tracks two structural motifs and overall validity over four phases. Panel b resolves the error of individual candidates, with each phase's best candidate marked, while panel c compares cumulative progress. The Single-Timescale control, labeled *Joint latent update*

Table 2: Reference relations with explicit parameter values and recovered dominant forms. Displayed coefficients are rounded; NMSE uses the full-precision programs in Appendix E. A dagger marks reference parameters estimated from training data for comparison only. A dash denotes an unavailable OOD split.

Task	Reference relation and parameters	Recovered dominant form	NMSE
CRK16	$\dot{A} = k_f A^{1/3} - kA$ $k_f \simeq 0.97092945^{\dagger}$ $k \simeq 0.24581249$	$\widehat{\dot{A}} \simeq 0.97093\, A^{1/3}$ $- 0.24581\, A$	ID: 1.21×10^{-14} OOD: 2.85×10^{-7}
PO18	$\dot{v} = -\beta \sin(x) v$ $- \omega_0^2 x$ $\beta \simeq 0.58365604^{\dagger}$ $\omega_0^2 \simeq 1.81264754^{\dagger}$	$\widehat{\dot{v}} \simeq -0.58366 \sin(x) v$ $- 1.81265\, x$	ID: 9.27×10^{-16} OOD: 1.10×10^{-15}
MatSci22	$\sigma = a\epsilon^2$ $+ b\epsilon \sin(T - \phi_0)$ $a \simeq 72.33109008$ $b \simeq 7.74654609$ $\phi_0 \simeq 287.77390919$ Equivalent cosine phase: $\phi_c = (\phi_0 + \pi/2) \bmod 2\pi$ $\phi_c \simeq 0.31818139$	$\widehat{\sigma} \simeq 72.331\, \epsilon^2$ $+ 7.7465\, \epsilon$ $\cdot \cos(T - 0.31818)$	ID: 1.03×10^{-11} OOD: 2.20×10^{-10}
II.34.29b _3_0	$\mu = C \dfrac{E_n h}{B J_z g}$ $C = \dfrac{1}{2\pi}$ $C \simeq 0.1591549431$	$\widehat{\mu} \simeq 0.15915 \dfrac{E_n h}{B J_z g}$	ID: 6.23×10^{-15} OOD: –

in the plot, updates all coordinates on one shared schedule. By the end of search, two-timescale adaptation attains best internal NMSE several orders of magnitude below the no-latent, random-latent, and single-timescale controls.

Figure 2 shows the inference state evolving on CRK35. The local distribution shifts frequently and contracts early, while the global distribution changes in less frequent steps, consistent with update intervals of 8 and 32 latent observations. Panel b tracks Top-4 operator-inclusion probabilities: `log`, `mul`, and `add` gain support during search. The selected equation achieves ID NMSE 2.24×10^{-15} and OOD NMSE 3.73×10^{-9}. These trajectories connect the two update clocks to observable changes in generation preferences.

4.3 Scientific Case Studies and Real-World Application

Structural recovery. Explicit expressions permit comparison of dependencies, exponents, and coefficients against reference relations. Table 2 presents four cases spanning reaction kinetics, oscillator dynamics, material response, and multivariable physics. All four pass Acc@0.01 on their available external splits. These examples demonstrate structural agreement; Appendix E provides the full derivations and parameter comparisons.

The recovered dominant forms capture the reference dependencies, with fitted exponents and coefficients close to their expected values. MatSci22 illustrates an equivalent trigonometric representation: applying $\sin u = \cos(u - \pi/2)$ and reducing the phase modulo 2π aligns reference and recovered phases to within 6.77×10^{-7} radians. Figure 5 examines how fitting suppresses an unnecessary dependency in PO18. Panel a shows pointwise relative-error coverage on ID and OOD data, with NMSE 9.27×10^{-16} and 1.10×10^{-15}. Panel b separates damping and restoring contributions across the input range; panel c displays effective coefficients and near-zero extraneous terms. Although time is available as an input, its fitted contribution becomes nearly zero, leaving state-based damp-

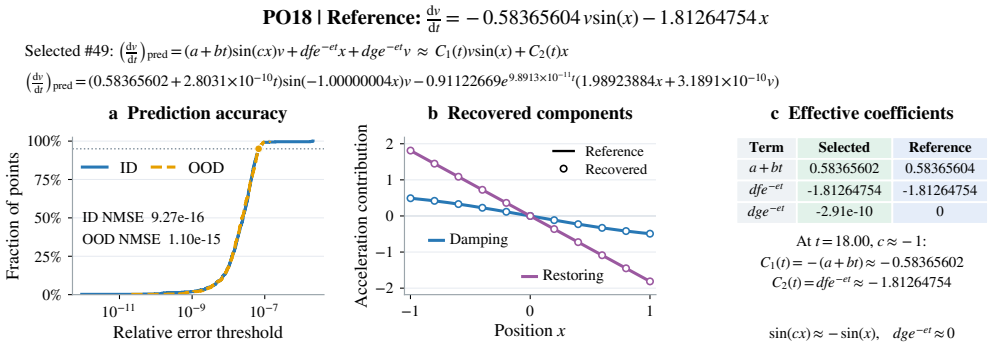

Figure 5: PO18 recovery: error coverage (a), recovered components (b), and effective coefficients (c). Reference coefficients are estimated from task-training data, marked † in Table 2.

ing and restoring terms. Table 9 compares individual parameters. Together, the cases show how flexible programs yield interpretable relations through characteristic powers, equivalent functional representations, and suppression of unnecessary dependencies.

Measured stress–strain response. We evaluate on the measured aluminum 6061-T651 stress–strain task released with LLM-SR (Shojaee et al., 2025a; Aakash et al., 2019). The inputs are engineering strain ϵ and temperature T, the target is stress σ, and 200 °C is held out for OOD assessment.

Table 3 shows lower ID and OOD NMSE for BITEMPO at both budgets. Relative to DE, BITEMPO-200 reduces ID and OOD NMSE by 34.9% and 39.1%; BITEMPO-400 reduces them by 48.6% and 61.0%. The improvements indicate accurate prediction both within the measured regime and at the held-out temperature. Appendix F analyzes the recovered expression.

Table 3: Stress–Strain ID/OOD NMSE ↓. Method suffixes denote candidate budgets.

Method	ID	OOD
LLMDirect	3.91e-1	1.20e+0
LLM-SR	1.44e-1	6.34e-1
LASR	2.52e-1	1.15e+0
SGA	3.95e+0	1.84e+0
Deliberate Evolution	1.11e-1	2.98e-1
BITEMPO-200	7.23e-2	1.81e-1
BITEMPO-400	**5.71e-2**	**1.16e-1**

5 DISCUSSION AND CONCLUSION

BITEMPO accumulates equation-evaluation feedback in global–local latent distributions, combining persistent structural preferences with responsive corrections. By maintaining an explicit distributional search state alongside previously evaluated programs, it allows evidence from syntactically different equations to reinforce shared generation preferences. The ablations indicate that both feedback-driven adaptation and timescale separation contribute to search performance. Chemistry and Biology show strong accuracy across backbones, while ID gains in Physics and Material do not consistently translate into better OOD accuracy, suggesting that training-based selection alone does not guarantee extrapolation. The recovered equations expose interpretable dependencies and parameters, and the measured stress–strain task demonstrates utility beyond synthetic equations. Accurate prediction, however, does not establish a unique physical mechanism.

Limitations and future directions. Our evaluation focuses on scalar-valued equation discovery under fixed candidate budgets. Extending the framework to coupled systems and tasks with additional scientific constraints is a natural next step. The present implementation also uses fixed update intervals for the global and local distributions; adapting these intervals to the progress of each task could provide greater flexibility. More broadly, combining latent inference with dimensional information and domain knowledge may help guide discovery toward scientifically meaningful relations.

REPRODUCIBILITY STATEMENT

Appendix C specifies the core hyperparameters, latent layout, and numerical-fitting protocol. LLM-SRBench and the aluminum 6061-T651 stress–strain dataset are publicly available (Shojaee et al., 2025b; Aakash et al., 2019). Section 4 defines NMSE, R^2, and Acc@0.01. The generator and evaluation backend build on the publicly released LLM-SR framework (Shojaee et al., 2025a). Appendix D provides the mathematical analysis.

ETHICS STATEMENT

This work concerns automated discovery of scientific equations from numerical, non-human-subjects data (public symbolic-regression benchmarks and a published materials-science dataset). We do not identify human-subjects, privacy, fairness, or dual-use concerns specific to this method or its evaluation.

AI USE STATEMENT

We used generative AI tools (including a code-assistant LLM) during this project for: debugging code; drafting and revising parts of the manuscript text, English expression, and LaTeX preparation. Generative AI tools were not used to generate synthetic experimental data, to fabricate or alter reported results, or to design the core research methodology or hypotheses independently of the authors. All AI-assisted text, code, and analysis were reviewed, verified, and edited by the authors, who take full responsibility for the accuracy of the final content, including the correctness of all reported numbers and their attribution to the referenced benchmarks and sources.

REFERENCES

B. S. 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. doi: 10.1016/j.dib.2019.104085. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC6603851/.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html.

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

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010. URL https://www.jmlr.org/papers/v11/cawley10a.html.

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

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005. doi: 10.1007/s10479-005-5724-z.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451. URL https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full.

Gene H. Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973. doi: 10.1137/0710036. URL https://epubs.siam.org/doi/10.1137/0710036.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Arya Grayeli, Atharva Sehgal, Omar Costilla-Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44678–44709, 2024. doi: 10.52202/079017-1419. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4ec3ddc465c6d650c9c419fb91f1c00a-Abstract-Conference.html.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. doi: 10.1098/rsta.2015.0202.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10269–10281, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/42eb37cdbefd7abae0835f4b67548c39-Abstract-Conference.html.

Deqian Kong, Yuhao Huang, Jianwen Xie, Edouardo Honig, Ming Xu, Shuanghong Xue, Pei Lin, Sanping Zhou, Sheng Zhong, Nanning Zheng, and Ying Nian Wu. Molecule design by latent prompt transformer. In *Advances in Neural Information Processing Systems*, volume 37, pp. 89069–89097, 2024a. doi: 10.52202/079017-2826. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a229cb89a98a84b2373496bb3cfc3570-Abstract-Conference.html.

Deqian Kong, Dehong Xu, Minglu Zhao, Bo Pang, Jianwen Xie, Andrew Lizarraga, Yuhao Huang, Sirui Xie, and Ying Nian Wu. Latent plan transformer for trajectory abstraction: Planning as latent space inference. In *Advances in Neural Information Processing Systems*, volume 37, pp. 123379–123401, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/df22a19686a558e74f038e6277a51f68-Abstract-Conference.html.

Deqian Kong, Minglu Zhao, Dehong Xu, Bo Pang, Shu Wang, Edouardo Honig, Zhangzhang Si, Chuan Li, Jianwen Xie, Sirui Xie, and Ying Nian Wu. Latent thought models with variational Bayes inference-time computation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31419–31436, 2025. URL https://proceedings.mlr.press/v267/kong25c.html.

John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992. ISBN 978-0-262-11170-6. URL https://mitpress.mit.edu/9780262527910/genetic-programming/.

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

Yunlun Li and Sinno Jialin Pan. Robust equation structure learning with adaptive refinement. In *International Conference on Learning Representations*, pp. 70029–70053, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/72416ded78a439907ff72165ac9c56e0-Abstract-Conference.html.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B. Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. LLM and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33940–33962. PMLR, 2024. URL https://proceedings.mlr.press/v235/ma24m.html.

Dianne P. O'Leary and Bert W. Rust. Variable projection for nonlinear least squares problems. *Computational Optimization and Applications*, 54(3):579–593, 2013. doi: 10.1007/s10589-012-9492-9. URL https://link.springer.com/article/10.1007/s10589-012-9492-9.

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. *arXiv preprint arXiv:2606.04360*, 2026. URL https://arxiv.org/abs/2606.04360.

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

Qwen Team. Qwen3-4B-Instruct-2507. Model card, 2025. URL https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507.

David R. Roberts, Volker Bahn, Simone Ciuti, Mark S. Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J. Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, David I. Warton, Brendan A. Wintle, Florian Hartig, and Carsten F. Dormann. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929, 2017. doi: 10.1111/ecog.02881.

Peter J. Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993. doi: 10.1080/01621459.1993.10476408.

Reuven Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143. URL https://doi.org/10.1023/A:1010091220143.

Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer, New York, 2004. doi: 10.1007/978-1-4757-4321-0. URL https://link.springer.com/book/10.1007/978-1-4757-4321-0.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi: 10.1126/science.1165893.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*, pp. 16054–16085, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/28df8e730c054c5331855fd4d5403ba9-Abstract-Conference.html.

Parshin Shojaee, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D. Doan, and Chandan K. Reddy. LLM-SRBench: A new benchmark for scientific equation discovery with large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55325–55359. PMLR, 2025b. URL https://proceedings.mlr.press/v267/shojaee25a.html.

Ilya M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. doi: 10.1016/0041-5553(67)90144-9.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974. doi: 10.1111/j.2517-6161.1974.tb00994.x. URL https://academic.oup.com/jrsssb/article/36/2/111/7027414.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. doi: 10.1126/sciadv.aay2631. URL https://arxiv.org/abs/1905.11481.

Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7:91, 2006. doi: 10.1186/1471-2105-7-91. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-91.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020. doi: 10.1038/s41592-019-0686-2. URL https://www.nature.com/articles/s41592-019-0686-2.

Shijie Xia, Yuhan Sun, and Pengfei Liu. SR-Scientist: Scientific equation discovery with agentic AI. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.11661.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Wenxiao Zhao, Dong Liu, Kaiyi Xu, Feng Liu, Zhen Zhao, Fei Ben, Shu Wang, Wenhao Li, Ying Nian Wu, Fenghua Ling, Haobo Li, and Lei Bai. A-SR: Self-evolving agentic LLMs for symbolic regression via hierarchical coordination. *arXiv preprint arXiv:2608.04872*, 2026. URL https://arxiv.org/abs/2608.04872.

A Extended Related Work

This appendix continues §2 with the latent-variable-inference and distribution-based-optimization lines of work most relevant to BiTempo's adaptation mechanism, and summarizes BiTempo's overall positioning.

Latent-variable inference for conditional generation. The Latent Prompt Transformer (LPT) family studies generation guided by latent inference across molecule design, planning, and reasoning (Kong et al., 2024a;b; 2025). BiTempo develops this latent-guided generation viewpoint for scientific equation discovery: a textual policy compiler and evaluation-driven distribution adaptation infer the representation directly within each task, so it specializes immediately to that task's variables, operators, and error patterns without requiring an offline training corpus.

Distribution-based optimization and adaptive search. The cross-entropy method (CEM) alternates sampling, evaluating, and fitting a distribution toward a selected elite set (Rubinstein, 1999; de Boer et al., 2005; Rubinstein & Kroese, 2004); in BiTempo these samples encode generation preferences, each evaluated through a proposed equation, so a single observation is noisy evidence about a policy's future usefulness. The small acquisition surrogate we use relates to sequential methods that balance predicted utility with uncertainty (Abbasi-Yadkori et al., 2011); its interpretation is heuristic because the evolving parent archive makes the generation objective history-dependent, but it provides an inexpensive mechanism for comparing latent proposals before committing an expensive decoder call.

Positioning of BITEMPO. BITEMPO brings together program-based equation discovery and adaptive latent inference. Building on the LLM-SR generation-and-experience pipeline (Shojaee et al., 2025a), it represents structural and refinement preferences as global and local latent blocks, updates them at different rates from numerical outcomes, and uses latent-compatible examples to connect sampled preferences to concrete program revisions. Table 4 summarizes how representative approaches implement the recurring cycle of proposal, numerical evaluation, feedback integration, and renewed proposal; the central distinction is the search state through which an evaluated equation influences the next generation step. CEM supplies the distribution-fitting foundation, a utility surrogate assists acquisition, and variable projection (Golub & Pereyra, 1973; O'Leary & Rust, 2013) supports numerical evaluation. The ablations in §4.2 assess latent control, adaptive versus fixed-distribution search, and the contributions of the two latent blocks within this shared backend.

Table 4: Feedback mechanisms in iterative scientific equation discovery.

Approach	State carried across iterations	How evaluation shapes the next proposal
Evolutionary search (PySR) (Cranmer, 2023)	Populations of expression structures	Fitness-based selection and variation evolve expressions, combined with simplification and constant optimization.
Deep Symbolic Regression (Petersen et al., 2021)	Neural expression-generation policy	Fitting rewards drive risk-seeking policy-gradient updates that change the expression-sampling distribution.
LLM-SR (Shojaee et al., 2025a)	Experience buffer of evaluated programs	Selected high-quality programs enter the prompt as examples for subsequent program generation.
Deliberate Evolution (Pang et al., 2026)	Search guidance, structural diagnostics, and reflective memory	Evaluation informs adaptive search operators and diagnosis; accumulated experience guides subsequent revisions.
BITEMPO	Global and local latent distributions, a utility surrogate, and evaluated-program history	Utility-ranked evidence and residual information update generation preferences on two timescales; sampled preferences guide both program generation and parent retrieval.

Note. Each row describes a feedback mechanism within the shared iterative cycle. A parent is a previously evaluated program used as a reference for proposing a new equation.

B EXTENDED MATHEMATICAL PRELIMINARIES

This appendix expands the notation summarized in §3.1.

Parameter estimation for a fixed program. For a data subset A, the numerical fitting error is $\mathrm{MSE}_A(e, c) = |A|^{-1} \sum_{(x,y)\in A} (f_e(x; c) - y)^2$. The optimizer estimates constants for a proposed structure, after which training-based resampling determines its search utility (§3.3). Both latent blocks guide program generation; the local block represents refinement preferences rather than numerical constants. Since numerical optimization has a finite budget, a poor evaluation can reflect unsuccessful parameter estimation as well as an unsuitable structure.

An expression structure e specifies operations and their composition; a vector $c \in \mathcal{C}_e$ specifies its free constants. For example, $f_e(x; c) = c_0 + c_1 \exp(c_2 x)$ has a fixed exponential-plus-offset structure and three constants; replacing the exponential by a sine changes e, not just c. To compare two generated programs' structure, each can be parsed into an abstract syntax tree (AST): Figure 6 shows the tree for $x_1 + x_2^2$. Two different ASTs may represent the same function, so a source-level duplicate check does not establish algebraic equivalence.

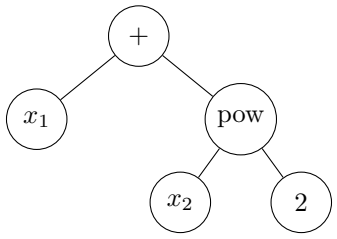

Figure 6: An expression tree for $x_1 + x_2^2$: the power node combines x_2 and 2, and the root adds the result to x_1.

Example 1 (Structure and coefficients are different unknowns)**.** *The formulas $c_0 + c_1 x$ and $c_0 + c_1 x + c_2 x^2$ are distinct structures. Within the second, changing c_2 from 0.1 to 0.2 is constant fitting; removing the quadratic term is a structural change, although setting $c_2 = 0$ makes the two functions coincide on every input.*

Linear and nonlinear parameter estimation. Linearity is defined with respect to a particular unknown. In $f(x; c) = c_0 + c_1 x^2 + c_2 \sin x$, once x is observed the columns $1, x^2, \sin x$ are known numerical vectors, so fitting c is linear least squares. In contrast, $f(x; a, b) = a \exp(bx)$ is nonlinear in b: changing b changes the basis function itself, though at fixed b the amplitude a is linear. Variable projection exploits exactly this partial structure (§3.3); because each fit has a finite computational budget, a promising expression can still receive a poor score when its constants are poorly estimated.

Error measures and identifiability. NMSE is not pointwise percentage error: a few large residuals can dominate its mean, while Acc@0.01 (§4) measures the fraction of tasks satisfying the specified pointwise-error coverage criterion. Both are informative, especially when targets span several scales. Predictive accuracy and scientific identification are also different: if observations satisfy $x_2 = 2x_1$, the formulas x_2 and $2x_1$ give identical predictions on those observations, so even exact numerical agreement cannot determine which variable relationship is causal; finite, correlated, or noisy data may support multiple formulas, which is why we report both the selected expression and its behavior outside the fitting domain (§4.3).

Prior distributions and approximate inference. In BITEMPO, the initial Gaussian p_0 treats available preference coordinates symmetrically. Evaluated programs provide evidence about useful search directions through reward-ranked distribution fitting: CEM fits a tractable Gaussian to elite latent observations, with smoothing and prior shrinkage moderating the update. This is an optimization-based adaptation rule, not an exact Bayesian posterior obtained from an explicit likelihood. It operates within a single task's generation budget without offline controller training. The prior initializes exploration and supplies a reference for regularization; Appendix D.2 explains the Gaussian fitting step.

Training resampling and external evaluation. An internal resampling scheme divides task-training observations into fitting sets A_k and evaluation sets $V_k = \mathcal{D} \setminus A_k$; the distinction between selecting a predictor and assessing its error is fundamental to cross-validation (Stone, 1974), and reusing validation estimates for selection can introduce optimism (Varma & Simon, 2006; Cawley & Talbot, 2010), while structured data require splits that respect the scientific assessment question (Roberts et al., 2017). The implementation permits a full-task-training fit to warm-start subset fitting, so we call the resulting score *warm-started training resampling* rather than independently fitted cross-validation; all such points remain task-training points, with supplied external validation and OOD labels reserved for external assessment.

C Implementation Details and Hyperparameters

This appendix summarizes the algorithm, core configuration, and numerical procedures underlying Section 3.

C.1 ALGORITHM

Algorithm 1 summarizes the full per-task loop described in §3: candidate generation guided by the current latent distribution and surrogate-assisted acquisition (Eq. 6), resampling-based evaluation (Eq. 3), and two-timescale distribution updates (Eq. 5).

Algorithm 1 BITEMPO: two-timescale latent inference for one task

Require: $\mathcal{D},\ s,\ p_\theta,\ p_0,\ B,\ K_g,\ K_l$
$(q, \mathcal{H}, \mathcal{A}, \mathcal{S}, N, N_g, N_l) \leftarrow (p_0, \varnothing, \varnothing, \varnothing, 0, 0, 0)$
Evaluate an initialization program; record into $\mathcal{H}, \mathcal{A}$
for $j = 2, \ldots, B$ **do**
  Draw M_j latents from q, or p_0 on a prior refresh (48 for eligible adapted draws, otherwise 1)
  Select z_j by Eq. (6) if $M_j > 1$, else use the single draw
  Generate $e_j \sim p_\theta(\cdot \mid s, C(z_j), P_j(z_j))$; evaluate via Eq. (3); record; $N \leftarrow N + 1$
  Refit surrogate $\mathcal{S}$ when eligible
  for $b \in \{g, l\}$ **do**
    if $N - N_b \geq K_b$ and at least 6 distinct proposals and 3 valid evaluations are available **then**
      Rank elites; update (μ_b, σ_b) via Eq. (5) (with residual-target substitution for $b = l$); $N_b \leftarrow N$
    end if
  end for
end for
Rerank top candidates; refit the selected structure on all training data; **return** equation, constants, history

A prior refresh samples from p_0 without resetting q: after at least four prior and four adapted proposals and a successful distribution update, it occurs after six consecutive adapted proposals.

C.2 CORE SETTINGS

LLM decoding uses temperature 0.8; API requests do not explicitly specify `top_p` or `max_tokens`.

Table 5: Core settings for the evaluated configuration.

Setting	Value
Latent size	40, partitioned into 30 global and 10 local
Operator guidance width	4 of 12 operator labels
Variable/residual guidance widths	Up to 3 variables; 2 residual families
Global/local intervals	32/8 observed latent candidates
Distribution	Diagonal Gaussian
Scale range	$[0.18, 2]$
Elite fraction/minimum	20% / 3 valid candidates
Recent evidence window	128 candidate observations
Mean-target/log-scale step limits	0.75σ / 0.35
Smoothing/prior shrinkage	$0.45/0.04$
Surrogate minimum observations	16
Surrogate drafts/acquisition coefficient κ	$48/0.45$
Final shortlist width k_{rerank}	4 existing candidates
Worst-subset/dispersion weights	$0.30/0.15$
Progress weight	0.5
Seed	42

C.3 LATENT DIMENSION BREAKDOWN

Table 6 breaks down the 40-dimensional latent policy referenced in §3.2.

The global block encodes variable relevance, operator preferences, interaction, and complexity; the local block encodes residual families and edit actions. Unavailable variables are masked during compilation. The compiler converts sampled coordinates into a small set of preferred labels. These

labels guide generation without restricting programs to the latent control vocabulary; the numerical backend separately enforces valid operations and interfaces.

Table 6: The 40-dimensional latent representation (dimensions are not update intervals).

Block	Coordinates	Size	Meaning and compiled choice
Global	Variable relevance	10	Up to three available variable names
	Operator preference	12	Four preferred operator labels
	Interaction	4	Additive, multiplicative, ratio, composed
	Complexity	4	Short, medium, deep, constant-rich
Local	Residual family	6	Two preferred residual categories
	Edit action	4	Explore, refine, simplify, stabilize

C.4 Distribution Update Rule

Rank-weighted elite moments define a Gaussian target with a variance floor. For local residual coordinates, measured residual-family evidence supplies the mean target; other coordinates retain their elite-based targets. After bounding the target displacement and scale change, an eligible block follows Eq. (5), applied to its mean and standard deviation. Here ρ controls smoothing and λ_0 controls shrinkage toward the initial distribution. Ineligible blocks remain unchanged; final bounds keep the proposal numerically stable. Appendix D analyzes the Gaussian fit and the bounded update.

C.5 Parent Compatibility

Parent selection combines program quality with agreement between latent preferences and parsed program features. For m binary features, let $A(p, b) = m^{-1} \sum_{j=1}^{m} [p_j b_j + (1 - p_j)(1 - b_j)]$. Compatibility is a weighted sum across variable, operator, interaction, complexity, and residual groups:

$$M(z, e) = \sum_g \alpha_g A_g(z, e), \qquad \alpha_g \geq 0, \quad \sum_g \alpha_g = 1. \tag{8}$$

The complexity group uses the probability assigned to the program's class. The first parent maximizes compatibility within a quality-filtered shortlist; a distinct second parent follows the archive's quality ranking. Thus the latent steers retrieval among already useful programs.

C.6 Surrogate Model

The surrogate regresses utility normalized by its median and median absolute deviation (Rousseeuw & Croux, 1993) on compiled latent features. Let X_ϕ contain the observed feature vectors with an intercept, Λ be a positive diagonal regularizer, and $A = X_\phi^\top X_\phi + \Lambda$. Ridge fitting gives $\widehat{\beta} = A^{-1} X_\phi^\top \widetilde{u}$, with prediction

$$\widehat{u}(z) = x^\top \widehat{\beta}, \qquad \widehat{s}^2(z) = \widehat{\nu}\big(1 + x^\top A^{-1} x\big), \qquad x = [\phi(z); 1]. \tag{9}$$

Here $\widehat{\nu}$ is a stabilized residual-variance estimate. After 16 latent observations, acquisition selects the latent draft maximizing $\widehat{u}(z) + \kappa \widehat{s}(z)$. This uncertainty score guides exploration rather than providing a calibrated confidence bound; only the selected draft invokes the decoder.

C.7 Search Utility and Ranking

For an internal evaluation subset V_k and fitting subset $A_k = \mathcal{D} \setminus V_k$,

$$E_k(e) = \frac{\mathrm{MSE}_{V_k}(e, \widehat{c}_{A_k})}{\max\{\mathrm{Var}_{V_k}(y),\, \delta_{\mathrm{rel}} \mathrm{Var}_{\mathcal{D}}(y),\, \delta_{\mathrm{abs}}\}}, \tag{10}$$

where the positive denominator floors stabilize near-constant targets. The separate utility floor $\varepsilon = 10^{-15}$ is applied before logarithms and leaves reported prediction errors unchanged. Invalid or non-finite evaluations are excluded from the valid elite set. Define E_{med} and $E_{\max}$ as the median and

maximum subset errors, with a positive floor applied before logarithms, and let S_E denote log-error dispersion. These quantities define the utility in Eq. (3). The score rewards typical accuracy and consistency across subsets. Elite ranking additionally rewards log-error improvement relative to the best error available before the candidate was proposed; the surrogate learns utility without this progress bonus.

C.8 Constant Optimization

When a program separates as $f(x; a, b) = F_b(x)a + g_b(x)$, the backend solves its affine parameters a by regularized least squares within an outer fit over nonlinear parameters b. The outer fit uses SciPy's bounded `least_squares` with multiple starts (Virtanen et al., 2020). Candidate quality is computed from the fitted predictions rather than clipped optimizer residuals.

Table 7: Three fitting stages. Limits are per nonlinear-optimizer restart, not numbers of generated formulas.

Stage	Restarts	`max_nfev`	Internal subsets
Search candidate	4	150	Random plus two projected tails
Final shortlist reranking	8	300	Random plus two tails per variable
Selected structure refit	16	600	All task-training points

Restarts combine available warm starts with generic and Sobol space-filling initializations (Sobol', 1967). Search scoring, shortlist reranking, and final parameter estimation use progressively stronger fitting budgets, as summarized above. A global-optimization fallback is available only for the final low-dimensional nonlinear fit.

C.9 Training Resampling and Final Selection

Search evaluation combines a random subset with both tails of a standardized principal-component ordering (Jolliffe & Cadima, 2016). Strong reranking additionally tests tails along individual variables. Subset fits may be initialized from the full task-training fit, so the score is warm-started training resampling. Hierarchical selection first retains candidates within $\max(10^{-10}, 0.01E^*_{\max})$ of the best worst-subset NMSE $E^*_{\max}$, then within $\max(0.01, 0.05S^*_E)$ of the best remaining log-error dispersion S^*_E, and finally minimizes median error, fitting discrepancy, and complexity lexicographically. If strong reranking yields no valid result, the search-score selection is retained. The chosen structure is then refitted on all task-training observations; external data are used only for assessment.

D Mathematical Interpretation and Proofs

This appendix supports §3.4 with the statistical meaning of the distribution fit, the role of regularization, and proofs of the estimation and stability properties used there. The analysis uses established Gaussian estimation and cross-entropy foundations (de Boer et al., 2005; Rubinstein & Kroese, 2004); the contribution lies in their integration with residual-informed global–local adaptation and latent-guided generation.

D.1 From Numerical Utility to an Inference Target

At a fixed search history, $V_t(z)$ is the expected utility of programs generated under latent z as defined in §3.2; an evaluated candidate supplies one noisy observation. A quality-weighted inference target is

$$\pi_t(z) = \frac{p_0(z)\exp\{\beta V_t(z)\}}{Z_t}, \qquad Z_t = \int p_0(v)\exp\{\beta V_t(v)\}\,dv. \tag{11}$$

Assume p_0 is a normalized prior density, V_t is measurable, finite p_0-almost everywhere, and bounded above by M, and $0 < \beta < \infty$. The integrand is positive p_0-almost everywhere and $Z_t \le \exp(\beta M)$,

establishing $0 < Z_t < \infty$. Thus Eq. (11) defines a probability density that favors useful latent regions while retaining prior support. The utility floor and nonnegative penalties in Eq. (3) provide an upper bound for valid candidate utilities; invalid or non-finite evaluations instead receive the fixed failure utility -20 and are excluded from the valid elite set.

This density supplies an inference interpretation. The algorithm constructs a tractable search distribution through finite-sample elite ranking and Gaussian fitting, followed by residual adjustment and regularization; it does not explicitly compute Z_t or an exact projection onto π_t. The Gaussian denotes the proposal before coordinate clipping, and β is conceptual rather than an additional implementation parameter.

D.2 Why Weighted Latent Moments?

All latent observations share a coordinate system, so weighted averaging estimates a location in a common representational space shared by syntactically different programs. With exponential rank weights, a favorable observation has more influence while several retained observations still contribute; the weighted variance complements the mean, with agreement among elites suggesting concentration and disagreement preserving a broader region for subsequent sampling.

Proposition 1 (Weighted Gaussian projection)**.** *Let $z_1, \dots, z_m \in \mathbb{R}^d$ have positive weights with $\sum_i w_i = 1$. Among diagonal Gaussian densities with variances $v_j \geq v_{\min} > 0$, the maximizer of $\sum_i w_i \log q(z_i)$ has*

$$\mu_j^\star = \sum_i w_i z_{i,j}, \qquad v_j^\star = \max\Big\{ \sum_i w_i (z_{i,j} - \mu_j^\star)^2,\, v_{\min} \Big\}. \tag{12}$$

Proof. Up to an additive constant, the weighted log density is $L(\mu, v) = -\frac{1}{2}\sum_j [\log v_j + \sum_i w_i (z_{i,j} - \mu_j)^2 / v_j]$. Writing $\bar{z}_j = \sum_i w_i z_{i,j}$ and expanding the square, $\sum_i w_i (z_{i,j} - \mu_j)^2 = \sum_i w_i (z_{i,j} - \bar{z}_j)^2 + (\mu_j - \bar{z}_j)^2$ (the cross term vanishes since $\sum_i w_i (z_{i,j} - \bar{z}_j) = 0$), so for any $v_j > 0$ the unique maximizing mean is $\bar{z}_j$. With $s_j^2 = \sum_i w_i (z_{i,j} - \bar{z}_j)^2$, the remaining objective $-\frac{1}{2}(\log v_j + s_j^2 / v_j)$ has derivative $(s_j^2 - v_j)/(2v_j^2)$, increasing until s_j^2 and decreasing thereafter; restricting $v_j \geq v_{\min}$ yields $v_j^\star = \max(s_j^2, v_{\min})$, including the boundary case $s_j^2 = 0$. □

Proposition 1 characterizes only the raw Gaussian fitting target before residual adjustment and update regularization. Residual adjustment introduces additional training-data feedback about unexplained response patterns; displacement limits, smoothing, and prior shrinkage then control the movement toward that target. Compilation maps the resulting latent to discrete preferences, so the argument does not require a smooth latent-to-program mapping.

D.3 Controlled Adaptation on Two Timescales

The mean update (Eq. (5)) can be written as

$$\mu^+ = (1 - \lambda_0)(1 - \rho)\mu + (1 - \lambda_0)\rho\, \mu^{\mathrm{tar}} + \lambda_0 \mu_0, \tag{13}$$

a convex combination (for $\rho, \lambda_0 \in [0, 1]$) of the current state, the limited target, and the initial reference, with nonnegative coefficients summing to one. Smoothing limits how strongly one favorable batch redirects the distribution given that each utility belongs to a stochastic decode with a finite numerical fit; prior shrinkage retains some influence from the initial representation even after many updates.

Proposition 2 (Bounded mean displacement)**.** *Suppose $\rho, \lambda_0 \in [0, 1]$, $\mu_j \in I$ for a closed interval I, and $|\mu_j^{\mathrm{tar}} - \mu_j| \leq d_j$. The regularized update, including final clipping, satisfies*

$$|\mu_j^+ - \mu_j| \leq (1 - \lambda_0)\rho\, d_j + \lambda_0 |\mu_{0,j} - \mu_j|. \tag{14}$$

Proof. Subtracting μ_j from Eq. (13) gives $\mu_j^+ - \mu_j = (1 - \lambda_0)\rho(\mu_j^{\mathrm{tar}} - \mu_j) + \lambda_0(\mu_{0,j} - \mu_j)$; the triangle inequality and the target bound give Eq. (14). Projection onto I is nonexpansive relative to any point already inside the interval, including μ_j, so the bound also holds after clipping. □

Analogously, a bounded log-scale step limits relative changes in proposal spread, and final projection preserves the allowed scale range. Interpolation updates the parameters of one Gaussian; it does not form a Gaussian mixture. These bounds control update size, not convergence or improvement in decoded equation quality.

D.4 Properties of Numerical Evaluation

Proposition 3 (Conditional scale invariance of a subset score)**.** *Fix predictions and a nonzero scalar a. Transforming all responses by $y \mapsto ay + b$ and predictions by $\widehat{y} \mapsto a\widehat{y} + b$ leaves Eq.* (10) *unchanged whenever its absolute floor is inactive before and after the transform.*

Proof. Both MSE and the variances in the denominator are multiplied by a^2; the common factor cancels. □

For fixed nonlinear constants, the scaled ridge matrix $\widetilde{F}^\top \widetilde{F} + \eta I$ ($\eta > 0$) satisfies $v^\top(\widetilde{F}^\top \widetilde{F} + \eta I)v = \|\widetilde{F}v\|_2^2 + \eta\|v\|_2^2 > 0$ for any $v \neq 0$, so the unconstrained ridge solution is unique even when design columns are linearly dependent (subsequent coefficient clipping enforces the implementation's numerical bounds).

D.5 Qualitative Interpretation

Consistent accuracy. Candidates with uniformly low subset error receive higher utility than candidates with comparable median error but a difficult tail. The dispersion term further distinguishes stable fits from uneven ones.

Robust calibration. Median-based centering and scale estimation reduce the influence of extreme utility values on the surrogate. Bounding normalized targets prevents one exceptional observation from dominating acquisition.

Gradual adaptation. Elite means shift the proposal toward successful preferences, while elite dispersion controls its spread. Step limits, smoothing, and prior shrinkage moderate this movement, allowing local corrections without abruptly replacing a global search direction.

E Additional Case Studies

E.1 Full Structural Recovery Derivations

This subsection derives the dominant forms discussed in Section 4.3 from the full-precision discovered expressions. It also provides two further recovery examples and parameter comparisons.

CRK16 (reaction kinetics). The reference rate is $\dot{A} = k_f A^{1/3} - kA$, $k = 0.2458124901$. The saved expression, after combining multiplicative constants, is $\widehat{\dot{A}} = 0.9709294316\, e^{-1.82\times10^{-10}t} A^{0.3333333304} - 0.2458124828A + 2.84 \times 10^{-12}t^2 - 1.90 \times 10^{-9}\sin(-1.0048883t)$, whose compact dominant form $0.9709294316\, A^{1/3} - 0.2458124828A$ differs from the saved expression by at most 2.43×10^{-8} on OOD points. Fitting the symbolic k_f to training data gives $k_f \simeq 0.9709294496$, consistent with the recovered amplitude.

PO18 (oscillator dynamics). The reference is $\dot{v} = -\beta\sin(x)v - \omega_0^2 x$. The generated program $\widehat{\dot{v}} = (a_0 + a_1 t)\sin(a_2 x)v + a_3 e^{-a_4 t}(a_5 x + a_6 v)$ has $a_2 = -1.0000000372$ and near-zero a_1, a_4, a_6, so using the oddness of sine and the near-unit exponential factor yields dominant form $\widehat{\dot{v}}_{\mathrm{dom}} = -0.5836560167\sin(x)v - 1.8126475323x$, matching training-only estimates $\beta \simeq 0.5836560400$, $\omega_0^2 \simeq 1.8126475358$ (max. difference from the full expression 2.42×10^{-8} on OOD).

MatSci22 (material response). The reference is $\sigma = 72.3310900778\,\epsilon^2 + 7.7465460881\,\epsilon\sin(T - 287.7739091906)$. The two principal recovered components are $\widehat{\sigma}_{\mathrm{dom}} = 72.3310952760\,\epsilon^2 + 7.7465455321\,\epsilon\cos(T - 0.3181820646)$; using $\sin u = \cos(u - \pi/2)$ and reducing phase modulo 2π, the reference phase and recovered phase agree to within 6.77×10^{-7} radians and the fitted frequency is 1.00000000165. Replacing the full saved program by this dominant form changes OOD predictions by at most 6.27×10^{-6}; an additional independent-input check (4,096 combinations of strain and temperature drawn within training marginal ranges) gives relative ℓ_2 error 4.10×10^{-7} between the unchanged program and the reference.

II.34.29b_3_0 (multivariable physics). Substituting the recovered coefficients gives the following expression, with displayed parameters rounded from the saved full-precision values:

$$\widehat{\mu} = \frac{0.159154943366\,E_n^{0.999999998650}\,h^{1.000000000688}}{g^{0.999999995413}\,J_z^{0.999999996778}\,B^{0.999999997633}}.$$

There are no additional terms. The recorded ID NMSE of 6.23×10^{-15} is computed from the saved full-precision program, not its rounded dominant form in Table 2; the task supplies no OOD split. Table 8 compares the individual parameters with the reference values.

Table 8: Reference and recovered parameters for $\widehat{\mu} = aE_n^{p_E} g^{p_g} h^{p_h} J_z^{p_J} B^{p_B}$ in II.34.29b_3_0. Denominator factors have negative exponents.

Parameter	Reference	Recovered
a	$1/(2\pi) \simeq 0.159154943092$	0.159154943366
p_E	1	0.999999998650
p_g	-1	-0.999999995413
p_h	1	1.000000000688
p_J	-1	-0.999999996778
p_B	-1	-0.999999997633

Two supplementary recoveries. BPG10 has reference rate $\dot{P} = 0.101008100332(P + P^{1/3})$; the dominant recovered dependence $0.101008100646P + 0.101008104544P^{1/3}$ agrees closely, with fitted exponent 0.333333319170 (ID NMSE 2.13×10^{-15}, OOD NMSE 7.20×10^{-10}). PO25 has reference $\dot{v} = F_0 \sin t - \beta \sin v$; recovered dominant terms $0.938430446668 \sin t - 0.866973741523 \sin v$ match training-only symbolic-coefficient estimates $F_0 = 0.938430539171$, $\beta = 0.866974142259$ (ID NMSE 5.55×10^{-13}, OOD NMSE 2.61×10^{-12}). Both pass Acc@0.01 on ID and OOD.

Structure and parameter recovery, side by side. Table 9 compares individual constants for CRK16, PO18, and MatSci22, complementing Table 8 for II.34.29b_3_0. Relative parameter errors range from approximately 10^{-9} to 2.1×10^{-6}; the largest is the reduced phase offset for MatSci22. Reference values marked † are left symbolic in the benchmark metadata (CRK16's rate constant k_f and both PO18 coefficients) and are estimated from training observations for comparison only; the other reference values come from benchmark metadata.

Table 9: Parameter recovery for CRK16, PO18, and MatSci22. Table 8 covers II.34.29b_3_0. Relative error is $|\widehat{c} - c_{\text{ref}}|/|c_{\text{ref}}|$, computed before rounding. Entries refer to the dominant components; the MatSci22 phase uses $\cos(\omega(T-\phi))$. A dagger marks a training-data estimate of a symbolic reference coefficient.

Task	Parameter	Reference	Recovered	Relative error
CRK16	Rate constant k_f	$0.9709294496^{\dagger}$	0.9709294316	1.85×10^{-8}
	Loss coefficient k	0.2458124901	0.2458124828	2.95×10^{-8}
	Power of A	0.3333333333	0.3333333304	8.83×10^{-9}
PO18	Damping β	$0.5836560400^{\dagger}$	0.5836560167	3.99×10^{-8}
	Frequency^2 ω_0^2	$1.8126475358^{\dagger}$	1.8126475323	1.93×10^{-9}
MatSci22	Quadratic-strain coeff.	72.3310900778	72.3310952760	7.19×10^{-8}
	Interaction coeff.	7.7465460881	7.7465455321	7.18×10^{-8}
	Phase offset ϕ, rad	0.3181813872	0.3181820646	2.13×10^{-6}
	Frequency	1.0000000000	1.0000000017	1.65×10^{-9}

Reference coefficients. When benchmark metadata leave a coefficient symbolic, its reference value is estimated from training observations for the parameter comparison. This applies to CRK16's production coefficient and the symbolic coefficients in PO18 and PO25. These estimates are used only for analysis and are not substituted into the generated equations.

F Stress–Strain Application: Full Derivation

This appendix expands §4.3. The source is the aluminum 6061-T651 dataset of Aakash et al. (2019): specimens from nine commercial material lots at six temperatures (20–300 °C), with engineering stress–strain curves from uniaxial and plane-strain tension tests, strain measured by digital image correlation and stress from load-cell readings and cross-sectional areas. The application uses the uniaxial-tension task released with LLM-SR (Shojaee et al., 2025a), with inputs engineering strain ϵ and temperature T and target engineering stress σ:

$$\widehat{\sigma} = f_e(\epsilon, T; \widehat{c}), \qquad \widehat{c} \approx \arg\min_{c} \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{i \in \mathcal{D}_{\text{train}}} \big[f_e(\epsilon_i, T_i; c) - \sigma_i\big]^2, \tag{15}$$

with the 200 °C condition assigned to the OOD split, testing transfer to a held-out temperature within the measured range. BiTempo uses Llama-3.1-8B-Instruct and the configuration in Appendix C. Table 3 reports predictive performance.

Discovered expression. Let s, t denote strain and temperature as represented in the task files and $\widehat{\sigma}_{\text{data}}$ the predicted stress in the task's numerical scale. Substituting fitted constants,

$$\begin{aligned}\widehat{\sigma}_{\text{data}}(s,t) = \; & \sin(-0.663582s - 1.735583t) \cdot \big[0.111170e^{1.323106|s|} - 0.469549\big] \\ & + \cos(1.312904t) \cdot \big[-0.093616e^{2.432090t} + 0.896190\tanh(60.466063s)\big].\end{aligned} \tag{16}$$

Displayed coefficients are rounded; predictive errors use the saved full-precision constants. This can be organized as $\widehat{\sigma}_{\text{data}}(s,t) = A(s)\sin(as + bt) + B(t) + C(t)\tanh(ks)$, where $a = -0.663582$, $b = -1.735583$, $k = 60.466063$, $A(s) = 0.111170e^{1.323106|s|} - 0.469549$, $B(t) = -0.093616e^{2.432090t}\cos(1.312904t)$, $C(t) = 0.896190\cos(1.312904t)$. Table 10 summarizes the resulting three components.

Table 10: Mathematical roles of the components in the discovered Stress–Strain expression.

Component	Mathematical behavior	Material-response interpretation
$A(s)\sin(as + bt)$	Both inputs affect the argument; amplitude also varies with strain	A nonlinear interaction that can alter response shape across temperature
$B(t)$	Depends on temperature alone	An additional temperature-dependent response offset
$C(t)\tanh(ks)$	Locally linear in s near zero, bounded at large $\lvert ks\rvert$; amplitude depends on t	A smooth transition between a locally steep and a more gradual response

Interpretation. The $\tanh$ term provides a useful shape basis: $\tanh(ks) \approx ks$ for $|ks| \ll 1$ and approaches a constant as ks grows, with strain derivative $kC(t)[1 - \tanh^2(ks)]$ decreasing in magnitude away from zero at fixed temperature — a component that can express a change in slope, with the size of that change set by temperature. Measured tensile curves have an initially near-linear region followed by nonlinear deformation whose shape varies with temperature (Aakash et al., 2019; Shojaee et al., 2025a), which motivates reading the saturating and coupled terms as empirical shape representations rather than as evidence of a specific physical mechanism: the trigonometric components act as empirical corrections, their presence does not by itself establish a periodic material mechanism, and the local linearity of one component does not establish an elastic law for the complete expression. The explicit form nonetheless permits direct quantitative sensitivity analysis (derivatives with respect to s and t for each component), which the discovered symbolic form makes tractable in a way a black-box predictor would not.

G Notation Reference

Table 11: Principal symbols. Scientific constants c and latent representations z are different objects.

Symbol	Meaning
$\mathcal{D}$	Observations supplied for equation inference and fitting
A_k, V_k	Internal fitting/evaluation subsets of $\mathcal{D}$
s	Task specification used in the generator context
e, c	Program structure and its fitted numerical constants
$\mathcal{E}, \mathcal{C}_e$	Admissible equation programs; constant space of structure e
z_g, z_l	Global and local generation-preference coordinates
q_t, μ_t, σ_t	Adapted proposal and its Gaussian parameters
p_0, μ_0, σ_0	Initial reference distribution and parameters
$C(z), P_t(z)$	Compiled semantic guidance and ordered parent programs
θ	Pretrained language-model parameters
$n_{\text{obs}}, X_\phi, \widehat{\beta}$	Surrogate observation count, latent-feature matrix, regression coefficients
$\phi(z)$	Discrete compiler features for the surrogate
E_k, u, Δ	Internal normalized error, robust utility, and relative progress
$V_t(z)$	Expected utility of programs generated under latent z at step t
$\mathcal{E}_t, w_i$	Elite observations and normalized rank weights
N_t, N_g, N_l	Observed latent-candidate count and last block-update counts
K_g, K_l	Global and local update intervals, 32 and 8
ρ, λ_0	Update smoothing and prior-shrinkage coefficients
$\eta_b, \Pi_{\mathcal{B}}$	Block distribution parameters and projection onto their allowed ranges
$\widehat{u}_t, \widehat{s}_t, \kappa$	Surrogate utility prediction, uncertainty score, and exploration weight
$\varepsilon, \alpha_w, \alpha_s$	Internal utility floor and subset-consistency penalty weights
B	Candidate budget per run, as specified in the result tables
NMSE, Acc @0.01	Variance-normalized squared error and relative-error coverage success

G.1 Result Verification

Verification recomputes split-level errors and the pointwise Acc@0.01 coverage criterion from saved programs and fitted constants, then aggregates task scores by domain. Structural case studies additionally compare the discovered expression with the reference relation.