Soft Computing Fusion with Applications

www.scfa.reapress.com

Soft. Comput. Fusion. Appl. Vol. 3, No. 2 (2026) 139–152.

Paper Type: Original Article

Smoothing Screening and Reweighted Sparse Discovery of Noisy Governing Equations

Nadia Ghasemabadi[1,*], Setayesh Zahiri[2], Fatemeh Rasoulpour[1]

[1] Department of Computer Engineering and Information Technology, La.C., Islamic Azad University, Lahijan, Iran; ghasemabadinadia@gmail.com; rasouliisetare@gmail.com.

[2] Morvarid Intelligent Industrial Systems Research Group, Iran; setayeshzahiri70@gmail.com.

Citation:

Received: 25 August 2025 *Revised: 13 November 2025* *Accepted: 18 January 2026*	Ghasemabadi, N., Zahiri, S., & Rasoulpour, F. (2026). Smoothing screening and reweighted sparse discovery of noisy governing equations. *Soft Computing Fusion with Applications, 3*(2), 139-152.

Abstract

Discovering governing equations from noisy trajectories depends on derivative estimation, library conditioning, and model selection. This manuscript proposes HERMES, a broader slow-fast and symbolic architecture, and evaluates a regression-side realization combining per-state Savitzky–Golay filtering, window selection, correlation screening, and reweighted thresholded least squares. The reported campaign uses Lotka–Volterra, unforced Duffing, and a three-species multiscale decay chain, with four noise levels and twenty realizations per system and level. At 5% noise the pooled reported exact-recovery rate is 0.667 versus 0.000 for the tested raw-finite-difference STLSQ and ensemble-lite baselines; coefficient relative error is 0.0551±0.0647 versus 0.4615±0.5710 for STLSQ. The chain has no reported exact recovery at any level. At 10% noise Duffing's 0.900 rate corresponds to two failures among twenty realizations. The campaign does not test hidden-state recovery, neural slow-fast decomposition, evolutionary refinement, or a Pareto-front output. Aggregation units and statistical implementation require original records, and unequal derivative preprocessing limits attribution of the observed advantage to sparse-discovery methodology.

Keywords: Equation discovery, Multiscale systems, Symbolic regression, Sparse identification, Slow-fast dynamics, Noise robustness.

1 | Introduction

More than a century ago, biochemists reduced stiff networks of elementary reactions to two-line rate laws by exploiting the separation between fast binding and slow turnover [1]. That manoeuvre, singular perturbation in modern language, remains the template for how science extracts simple laws from multiscale reality: the observable dynamics live on a slow manifold, and the fast modes matter mainly through the constraints they impose on it. Today the extraction is increasingly attempted by machines. Automated reverse engineering inferred nonlinear models from measured time series from scratch [2], genetic programming distilled free-

Corresponding Author: ghasemabadinadia@gmail.com

https://doi.org/10.22105/scfa.v3i2.106

Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0).

form natural laws from experimental data [3], sparse identification recovers governing equations from trajectory libraries [4], and neural ordinary differential equations learn vector fields without any library at all [5]. The promise is explicit, interpretable, transferable models. The obstacle, largely unaddressed, is that real multiscale data are noisy and contaminated by exactly the fast transients that the target laws do not describe.

Geometric singular perturbation theory gives persistence of suitable compact normally hyperbolic critical manifolds under small perturbations [6]. Attraction additionally requires stable fast transverse directions; normal hyperbolicity alone also permits repelling and saddle-type manifolds. Classical reductions describe boundary layers and approximation errors under their stated conditions [7]. Delay-coordinate results provide embeddings under generic observation and dynamical assumptions [8], not a unique recovery of physical slow coordinates from every partially observed noisy record.

Sparse discovery [4], noise-aware formulations [9], symbolic regression [10], [11], relaxed sparse regression [12], and coordinate discovery [13] offer complementary approaches. Their strengths and limitations depend on coordinates, derivative estimators, libraries, and optimization choices. The broader HERMES design combines slow-fast representation learning with sparse and symbolic stages. The present experiments test only its smoothing–screening–reweighting core, rather than establishing an exhaustive novelty claim for the full composition.

The reported campaign uses full-state observations of three low-dimensional benchmark ODEs with known polynomial laws. It permits a controlled noise sweep, but does not evaluate partial observation or reduced hidden-manifold equations. The formal architecture and the executed regression realization must therefore be read separately.

The proposed architecture separates representation learning, sparse library regression, and seeded symbolic refinement (Sections 3 and 4). The executed realization combines per-state smoothing and differentiation with correlation screening and reweighted thresholding. A complexity–accuracy Pareto output is a proposed deliverable; the executed realization returns one model per record. The reported empirical campaign compares four implementations across three ODE systems and four noise levels, with twenty realizations per system and level. The scope distinguishes the tested regression core from unexecuted neural decomposition, hidden-state recovery, and symbolic refinement.

The remainder of the manuscript is organized as follows. Section 2 synthesizes related work across sparse and symbolic discovery, neural continuous-time models, and multiscale theory, and distils the gap. Sections 3 and 4 develop the research questions, the formal problem, and the proposed pipeline, and Section 5 reports the executed experimental campaign and its analysis. Section 6 discusses limitations and counterarguments, and Section 7 concludes.

2 | Related Work

2.1 | Sparse and Symbolic Equation Discovery

Equation discovery began as search over model structures. Automated reverse engineering evolved both model forms and parameters to reproduce measured time series [2], and symbolic regression by genetic programming famously distilled conserved quantities and free-form laws from physical data [3]. Sparse identification of nonlinear dynamics inverted the paradigm: with a fixed library of candidate terms, the governing equation becomes a sparse coefficient vector recoverable by thresholded least squares [4]. The software ecosystem around this idea is mature [14], extensions reach partial differential equations and noisy derivative estimation [9], and relaxed regularizers such as SR3 unify robust and structural regression with improved convergence behaviour [12]. Coordinate discovery adds a further layer, learning observability transformations under which the sparse representation becomes compact [13], and low-data-limit variants target control applications [15]. AI Feynman demonstrated that physics-inspired constraints, such as symmetries and separability, can make the symbolic search space tractable [10], while modern evolutionary engines such as PySR scale the search with typed operators and parsimony pressure [11].

Library discovery depends on a coordinate system and derivative estimates in which the target law can be represented. Coordinate-learning and noise-aware variants address some of these issues [9], [13]. A decomposition stage is a design option for slow-fast data, but full-state multiscale ODEs need not be nonsparse by construction. The chain used here already has a sparse full-state polynomial law.

2.2 | Neural Continuous-Time Models and Physical Structure

Neural models of dynamics learn flows rather than equations. Neural ordinary differential equations parameterize the vector field and integrate it with differentiable solvers [5]. Universal differential equations embed neural components inside known mechanistic structure, enabling hybrid models whose unknown parts are learned [16]. Physics-informed networks constrain the learner with residual losses from known partial differential equations [17], a programme whose scope and open problems are charted in [18]. Energy-based structure offers another constraint: Hamiltonian networks conserve a learned energy by construction [19], and Lagrangian networks extend this to general coordinates [20]. Linear-operator alternatives such as dynamic mode decomposition trade expressivity for spectral interpretability [21].

Neural models can absorb fast transients into latent states and predict observations without identifying a unique symbolic slow law. Hybrid models recover some structure [16], but symbolic postprocessing is still a separate proposed step. The HERMES architecture assigns representation learning, decomposition, and refinement to such possible stages; the experiment here tests only a smoothing–screening–reweighting regression core on observed full-state trajectories.

2.3 | Multiscale Structure, Noise, and Model Selection

Attracting slow manifolds require more than scale separation: normally hyperbolic transverse directions must be stable, with suitable compactness and regularity conditions [6], [7]. Regularized noisy-data differentiation [22], robust losses [23], stability selection [24], and information criteria [25], [26] address different statistical issues. Their guarantees depend on assumptions and tuning; citing them does not establish that the proposed composition inherits all of them.

Multiscale model reduction has its own data-driven tradition, from equation-free computation to manifold learning, but the outputs are numerically defined objects rather than closed-form equations. The present design accepts that limitation for its neural stages and refuses it at the output: whatever the intermediate representations, the deliverable is a symbolic equation whose complexity can be counted and compared. This stance forces the model-selection question to the front. A discovery method that cannot say how complex its answer is, or how that complexity trades against predictive fidelity, is not yet a scientific instrument, and the trade-off is therefore formalized as a first-class object in Section 4.

2.4 | Synthesis and the Research Gap

Table 2 summarizes selected architectures. HERMES's row describes proposed stages, not all executed components. Sparse regression can use noise-aware derivatives, coordinate discovery can learn representations, and symbolic models can incorporate physical constraints. The present question is whether the specific regression implementation improves the displayed outcomes under the stated preprocessing and budget.

Table 2. Selected equation-discovery architectures.

Family	Representative Works	Noise Handling	Multiscale Handling	Output Form
Sparse library regression	[4], [12], [14]	Robust variants, regularized derivatives [9]	None; assumes given coordinates	Sparse linear model
Evolutionary symbolic regression	[3], [10], [11]	Limited; sensitive to noise	None natively	Symbolic expression
Neural continuous-time models	[5], [16], [17]	Implicit, in training	Implicit, in hidden state	Opaque vector field
Energy-structure learning	[19], [20]	Implicit	Conservative systems	Conserved quantity + flow
Coordinate discovery	[13]	Partial	Learned observability	Transform + sparse model
Proposed HERMES (this work)	-	Robust thresholds, Huber losses, stability selection	Explicit neural slow-fast decomposition	Symbolic law on Pareto front

Note: The HERMES row describes the proposed full design; only a regression-side realization is evaluated.

3 | Problem Formulation and Research Questions

3.1 | Research Questions and Hypotheses

The design is organized around three research questions. They concern, respectively, whether a neural decomposition can prepare slow coordinates well enough for identification, whether hybridizing library regression with seeded evolutionary search improves the discovery front, and how robustness degrades under noise and weak separation. Each question is paired with a falsifiable hypothesis.

RQ1 asks whether a learned slow-fast decomposition supports identification of reduced terms under noisy partial observations.

RQ2 asks whether seeded symbolic refinement improves the complexity–accuracy front relative to its parent methods.

RQ3 asks how recovery changes with noise, separation, and record length.

These are design-level questions. The current campaign sweeps observation noise for three fixed full-state systems; it does not execute neural decomposition, evolutionary refinement, or systematic separation/record-length sweeps.

H1 predicts useful term recovery after a specified learned decomposition. It remains untested without that stage and independently verified coordinates.

H2 predicts a benefit of seeded refinement at matched search vocabulary and budget. It remains untested.

H3 predicts a recovery boundary under noise and separation sweeps. The current chain failure is a limitation of one implementation, not an estimated general boundary.

3.2 | System Model and Notation

We consider slow-fast systems in the standard form of *Eq. (1)*, where x collects the slow variables, z the fast variables, eps the ratio of timescales, and mu a parameter. Only a subset of the state is observed, through the measurement model of *Eq. (2)*, with eta_k denoting additive noise whose level is summarized by a signal-to-

noise ratio. The observed record therefore mixes slow trend, fast transients, and measurement noise in unknown proportions. The unknown objects of interest are the slow manifold Z and the reduced drift F, and *Table 1* collects the notation used throughout.

$$\dot{x} = f(x, z; \mu), \quad \varepsilon \dot{z} = g(x, z; \mu), \quad 0 < \varepsilon \ll 1. \tag{1}$$

$$y_k = h\big(x(t_k), z(t_k)\big) + \eta_k, \quad k = 1, \dots, K. \tag{2}$$

Table 1. Principal symbols used throughout the manuscript.

Symbol	Meaning	Role
x, z	Slow and fast state variables	Latent decomposition
eps	Timescale-separation ratio	System parameter
mu	System parameter	Sweep axis
y_k	Observation at time t_k	Data
h	Observation operator	Partial observability
SNR	Signal-to-noise ratio of the record	Stress factor
U	Slow coordinates recovered in Stage A	Regression input
Z, Z_theta	True and learned slow manifold	Invariance object
F	Reduced drift on the manifold	Discovery target
Theta(U)	Library of candidate terms	Stage B basis
xi	Sparse coefficient vector	Stage B output
c(F)	Symbolic node count of F	Complexity measure
H, L_roll	Rollout horizon and loss	Fidelity measure

3.3 | Formal Problem Statement

For a smooth slow-fast system, the critical manifold at ε=0 satisfies the first relation of *Eq. (3)*. Under appropriate compact normal hyperbolicity, a nearby locally invariant manifold can persist for small positive ε [6], [7]; attraction requires stable transverse fast dynamics. The finite-ε invariance relation contains DZεf, so setting g=0 alone is a quasi-steady approximation, not exact invariance. *Eq. (4)* defines the leading-order reduced drift. Recovering this object from noisy partial observations is the proposed discovery problem; the current experiments instead identify full-state ODEs.

$$\begin{aligned} g(x, Z_0(x; \mu); \mu) &= 0, \\ g(x, Z_\varepsilon(x; \mu); \mu) &= \varepsilon D_x Z_\varepsilon(x; \mu) f(x, Z_\varepsilon(x; \mu); \mu). \end{aligned} \tag{3}$$

$$\dot{x} = F_0(x; \mu) = f(x, Z_0(x; \mu); \mu). \tag{4}$$

A future implementation could report candidate laws and their validation-error/complexity front. Structural agreement with a known law belongs to evaluation, not training selection. Support error and coefficient error are separate: an error metric restricted to true support can ignore additional false terms, which must be assessed through support precision and exact matching.

4 | Proposed Methodology: The HERMES Pipeline

4.1 | Stage A: Neural Slow-Fast Decomposition

Stage A is proposed to learn slow and fast coordinates from delayed observations, reconstruct measured channels, and predict trajectories. A learned manifold head Zθ can be regularized by a finite-ε invariance residual, rather than requiring fast drift to vanish exactly. *Eq. (5)* states that residual for compatible learned drift models. These coordinates, networks, and regularization weights must be specified and validated before any manifold-recovery claim. Robust loss options [23] do not resolve identifiability by themselves.

$$P_{man} = \frac{1}{N}\sum_{i=1}^{N} \left\| g_\theta\left(u_i, Z_\theta(u_i)\right) - \varepsilon D Z_\theta(u_i) f_\theta\left(u_i, Z_\theta(u_i)\right) \right\|^2. \quad (5)$$

Proposed slow/fast heads, gating, and ensemble disagreement are modeling choices. Delay embedding [8] can supply an alternative state representation under its assumptions, but not uniquely named physical coordinates or sparse reduced laws. The neural output and its uncertainty remain unexecuted design components in this study.

4.2 | Stage B: Noise-Robust Sparse Library Regression

The proposed sparse stage uses a declared library and derivative estimates. For a Huber loss ρ_δ, *Eq. (6)* sums losses over residual components, with optional nonnegative weights; it does not square the Huber loss again. Sequential thresholding is an algorithmic approximation to sparse selection and is not guaranteed to solve the stated nonconvex objective globally. Stability selection [24] and BIC-style scoring [26] require their own assumptions, which may fail for autocorrelated trajectories. The implemented reweighted STLSQ core is a different realization; Huber optimization and stability-selection guarantees are not assigned to it.

$$\min_{\xi} \sum_{i,j} w_{ij}\, \rho_\delta\left(\dot{U}_{ij} - [\Theta(U)\xi]_{ij}\right) + \lambda \parallel \xi \parallel_0, \quad w_{ij} \geq 0. \quad (6)$$

Noise may affect support and coefficient estimates through derivative error and library conditioning. The proposed uncertainty-weighted regression requires validated weights; it is not tested here. The executed core uses correlation screening and reweighted thresholding on smoothed full-state data.

4.3 | Stage C: Seeded Symbolic Refinement

Stage C refines the shortlist by evolutionary symbolic search in the style of PySR [11]. Each candidate support seeds the initial population, so the search begins from a structurally informed region rather than from random expressions, while standard mutation and crossover operators explore replacements, simplifications, and eliminations. Complexity is counted as the number of symbolic nodes, which makes parsimony explicit and comparable across candidates. The search optimizes the objective of *Eq. (7)*, a rollout loss plus a complexity penalty, evaluated by integrating each candidate equation and comparing it with withheld slow-trajectory segments. Multi-objective bookkeeping retains the full Pareto set rather than a single winner.

$$\min_{F} \mathcal{L}_{roll}(F) + \lambda_c c(F). \quad (7)$$

Seeding symbolic search may improve efficiency, but neither a better Pareto front nor domination of unseeded search follows from initialization alone. A matched-budget ablation must test it. The current component ablations concern regression only.

4.4 | Model Selection and the Complexity-Accuracy Trade-off

A future Pareto output should retain candidates nondominated in validation rollout error and complexity. Scalarization by *Eq. (7)* samples only part of a potentially nonconvex front. A knee selection needs an explicit normalization and distance/tie rule rather than an informal corner description. No such front is produced in the executed study.

This framing has two consequences. It forbids the common practice of reporting the single best-recovered equation while the method actually produces many, and it renders the method honest about degeneracy: multiple simple laws may explain finite data, and their coexistence on the front is a finding about the data rather than a defect of the algorithm. The executed campaign accordingly scores the single delivered model per realization against the known true support and coefficients. In this respect the pipeline is closer to a measurement instrument for structure than to a point estimator.

4.5 | Algorithm, Complexity, Assumptions, and Failure Conditions

Algorithm 1 describes the proposed full architecture. Exact networks, losses, parameter ranges, supports, budgets, integrators, and selection rules remain necessary for reimplementation. Section 5 specifies a separate regression-only realization.

Algorithm 1. HERMES - hybrid equation recovery for noisy multiscale data.

```
Input: observations {(t_k, y_k)}, library Theta, operator set O,
       thresholds schedule T, generations G, complexity weight lam_c
1: Stage A: train slow-fast networks with manifold penalty of Eq. (5);
   ensemble across seeds to obtain U, Z_theta, uncertainty weights
2: estimate dU/dt on smoothed slow trajectories (regularized method)
3: Stage B: for each threshold in schedule T do
4:     solve thresholded robust regression of Eq. (6) on subsamples
5:     retain terms by stability selection; score supports by BIC
6: end for; output ranked candidate supports S_1, S_2, ...
7: Stage C: seed evolutionary search with each support
8: for generation = 1 to G do
9:     mutate / crossover symbolic candidates over operator set O
10:    evaluate L_roll on withheld segments; update Pareto front
11: end for
12: apply an explicitly specified normalized selection rule; report the full front
Output: Pareto front of symbolic laws with the knee selection
```

Computational cost includes neural training and integration, library construction, repeated least-squares solves, screening, bagging or stability subsamples, and candidate rollouts. For K rows and p library terms, a dense least-squares solve commonly requires $O(Kp^2+p^3)$ work; thresholds and iterations multiply this cost. Evolutionary cost depends on population size, expression complexity, integration horizon, and solver steps. Memory must include trajectories, the K×p library, models, and candidate populations. The small regression runtime does not measure full-architecture cost.

The following are assumptions of the proposed architecture, not conditions all checked by the full-state campaign:

I. A1 requires a suitable compact normally hyperbolic critical manifold; attraction additionally requires stable fast transverse directions [6].

II. A2 requires adequate observations for a valid embedding or identifiable coordinate model; an embedding alone does not recover unique slow physical coordinates [8].

III. A3 requires the target law to be representable in the library or reachable operator vocabulary.

IV. A4 specifies an approximately additive stationary noise model for the intended record.

V. A5 requires data covering the time constants and excitation needed to identify the target terms.

Complementarily, the pipeline is expected to fail in identifiable situations, declared here as scope rather than observed. Anticipating these failures disciplines the claims made on the pipeline's behalf. The stress grid treats each condition as a cell to be probed deliberately.

Without useful separation, slow-fast reduction may be inappropriate; that does not imply the full dynamics lack an identifiable model. An omitted functional form cannot be recovered by the fixed library. Derivative error and ill-conditioning may prevent support recovery under some noise levels. Short or insufficiently excited records can leave multiple candidate laws indistinguishable.

5 | Experimental Results and Analysis

5.1 | Setup: Systems, Baselines, Protocol, and Scope

The campaign contains models labeled Lotka–Volterra [27] and Duffing [28] with two states each, and a three-state linear chain. These historical works identify the model families; they do not validate the specific modern equations or the present simulation. The stated libraries, sample counts, and initial conditions are two-degree/501 samples at $\Delta t=0.02$ for Lotka–Volterra, cubic/626 samples at $\Delta t=0.02$ for Duffing, and degree-two/851 samples at $\Delta t=0.1$ for the chain. Chain rates are 2.0, 0.2, and 0.02, corresponding to time constants 0.5, 5, and 50. Gaussian noise levels are 0,1,5,10% of per-state standard deviation. Twenty realizations per system and level give 240 system-realization records overall and 60 at each level. The underlying state-equation count is $(2+2+3)\times 20=140$ per level, not 60. The reported table aggregation must therefore be clarified from the original records before modelwise and equationwise interpretations are assigned.

The methods share observation records but differ in preprocessing. Ridge regression [29] uses the full library; STLSQ [4] uses fixed thresholds 0.2,0.02,0.05 for the respective systems. The ensemble-lite variant bags 25 fits on 80% of time rows with a 60% support-retention rule; it is not the full Ensemble-SINDy method [30]. HERMES uses Savitzky–Golay filtering/differentiation [31], a window bank {5,9,15,25,41,61}, correlation screening, and reweighted STLSQ with adaptive thresholds and $\varkappa=0.5$. Its described generalized-cross-validation selection adapts a criterion developed for linear smoothing/regularization [32]. Baselines use raw second-order finite differences. Matched-smoothed baselines are needed to isolate benefits of sparse selection from derivative preprocessing.

The reported metrics include support precision/recall, norm-based coefficient relative error on the true support, and a 10% maximum per-term coefficient tolerance for exact recovery. False extra coefficients are excluded from the true-support coefficient metric and must be assessed separately. Whether exactness requires all equations in a model or is averaged equationwise remains unresolved by the source summaries. The 5% level is the designated endpoint. The stated master seed 20261011, 36.3 s runtime, and software configuration, including SciPy [33] and matplotlib [34], require code and logs. Neural decomposition, hidden-state recovery, evolutionary refinement, and Pareto output are not executed; no release identifier or dated preregistration is established.

5.2 | Main Comparison (RQ1)

At 5% noise *Table 3* reports pooled exact rate 0.667 for HERMES and 0.000 for all three baselines. The source interpretation is 40 successes among 60 realization records, requiring confirmation of modelwise aggregation. Coefficient errors are 0.0551 ± 0.0647 for HERMES, 0.4615 ± 0.5710 for STLSQ, 0.4601 ± 0.5696 for ensemble-lite, and 0.8926 ± 0.1073 for ridge. These summaries demonstrate a difference between the tested implementations, but their pooling units and raw records are needed for precise inference. Reported p-values are retained descriptively pending verification of test method and assumptions.

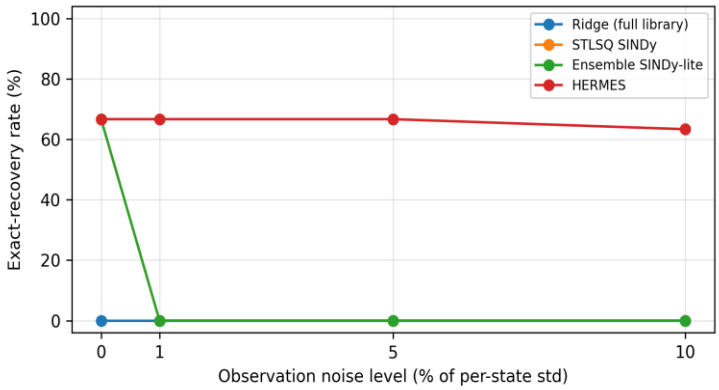


Fig. 1. Original reported pooled exact-recovery curves across observation-noise levels (twenty realizations per system).

At zero noise the plotted STLSQ curve is 0%, whereas the source prose claims a pooled STLSQ rate of 0.667 and success on both single-scale systems. The contradiction and aggregation unit require the original per-system indicators and plotting code; there are sixty system-realization records and 140 underlying state equations per level (*Fig. 1*).

Table 3. Reported comparison at 5% noise.

Method	Exact Rate	Coeff. Rel. Error	Precision	Recall	LV Exact	Duffing Exact	Chain Exact
Ridge (full library)	0.000	0.8926 ± 0.1073	0.233 ± 0.072	1.000 ± 0.000	0.000	0.000	0.000
STLSQ SINDy [4]	0.000	0.4615 ± 0.5710	0.246 ± 0.065	0.997 ± 0.021	0.000	0.000	0.000
Ensemble SINDy	0.000	0.4601 ± 0.5696	0.234 ± 0.072	1.000 ± 0.000	0.000	0.000	0.000
HERMES (proposed)	0.667	0.0551 ± 0.0647	1.000 ± 0.000	0.944 ± 0.079	1.000	1.000	0.000

Note: Pooled summaries and per-system exact rates are preserved; modelwise versus equationwise aggregation requires original records.

The per-system summaries show exact rate 1.000 for HERMES on Lotka–Volterra and Duffing at 5% noise, and 0.000 on the chain. At 10% Duffing's rate 0.900 means 18 of twenty successes and two failures if the unit is a realization, not one failure. The chain's reported precision 1.000 and recall 0.833 are consistent with omitting a weak true term. The claimed third-equation omission of coefficient −0.02 and selected threshold 2.0 require the delivered support records; aggregate precision/recall alone do not identify them.

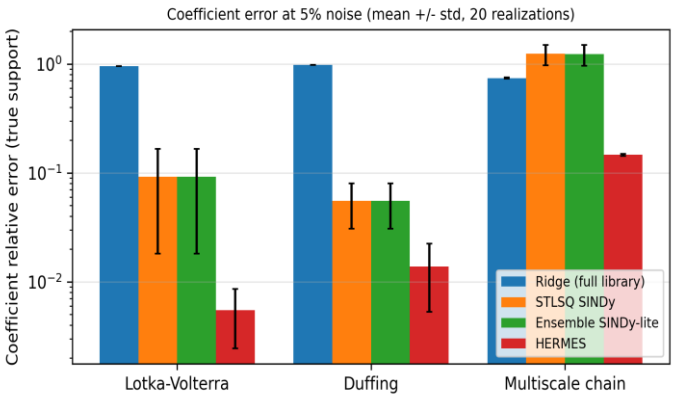


Fig. 2. Reported coefficient relative error on the true support at 5% noise (mean ± SD over twenty realizations per system; logarithmic vertical axis).

HERMES has the lowest displayed error on Lotka–Volterra and Duffing. Its error rises on the multiscale chain, where the displayed STLSQ and ensemble-lite errors are approximately order one. The plotted comparisons are descriptive; the underlying records are needed to resolve the pooling unit in *Table 3*.

5.3 | Ablation of the Pipeline Components (RQ2)

Table 4 compares the implemented core with three removals. Without smoothing, the 5% exact rate is 0.633 versus 0.667, and the 10% rate is 0.350 versus 0.633. Removing screening gives 0.000 at 5%; removing reweighting gives 0.033, with precision 0.677 and recall 0.944. The lower precision directly indicates false extra terms, so the loss of exactness cannot be attributed solely to coefficient accuracy. These are scoped component contrasts, not proof that each stage is universally necessary or that neural/evolutionary hybridization works.

Table 4. Reported ablations of the regression-side realization.

Variant	σ	Exact Rate	Coeff. Rel. Error	Precision	Recall
Full pipeline	1%	0.667	0.0481 ± 0.0648	1.000	0.944
w/o SG smoothing	1%	0.667	0.0484 ± 0.0638	1.000	0.944
w/o library pruning	1%	0.000	0.4022 ± 1.1173	0.441	0.861
w/o reweighting	1%	0.000	0.0512 ± 0.0628	0.735	0.944
Full pipeline	5%	0.667	0.0551 ± 0.0647	1.000	0.944
w/o SG smoothing	5%	0.633	0.0639 ± 0.0586	0.992	0.944
w/o library pruning	5%	0.000	0.5139 ± 1.1354	0.356	0.808
w/o reweighting	5%	0.033	0.0730 ± 0.0597	0.677	0.944
Full pipeline	10%	0.633	0.0672 ± 0.0625	0.996	0.944
w/o SG smoothing	10%	0.350	0.3426 ± 0.4182	0.808	0.842
w/o library pruning	10%	0.000	0.4967 ± 0.7430	0.330	0.781
w/o reweighting	10%	0.000	0.1024 ± 0.0595	0.599	0.944

Note: Pooling unit requires confirmation from the sixty system-realization records at each level.

5.4 | Robustness across Noise Levels and Timescales (RQ3)

At zero noise, the source prose claims STLSQ and ensemble-lite each recovered Lotka–Volterra and Duffing exactly, corresponding to pooled 0.667 if the unit is a system realization. *Fig. 1* instead plots STLSQ at 0% and ensemble-lite near 66.7%. STLSQ noiseless exact success is therefore unestablished until per-system exact indicators, the aggregation rule, and plotting code reconcile text and figure. The source reports no exact chain recovery. HERMES's Duffing rate of 0.900 at 10% implies two failures among twenty realizations if that is the unit. Reported chain smoothing windows include 15 samples for the fast state and 61 for the slow state; the bank also contains 5 and 9, so 15 is not its shortest window. Recorded windows and validation errors are needed to assess their selection.

The noiseless single-scale coefficient errors favor STLSQ: 0.0005 versus 0.0018 on Lotka–Volterra, and 0.0006 versus 0.0142 on Duffing. This is consistent with a smoothing-related trade-off for these records. Chain failure at zero noise does not prove that denoising, alternative thresholds, longer excitation, or another estimator cannot recover it. Conditioning and support-selection effects should be examined before declaring a structural impossibility.

5.5 | Statistical Analysis

The source reports Wilcoxon comparisons [35], Holm adjustments [36], and Friedman summaries [37]. The claim that all p-values are exact is inconsistent with their scale: forty nonzero same-sign binary differences give a two-sided sign-randomization value $2/2^{40} \approx 1.819e{-}12$, rather than the reported 2.540e−10. With sixty nonzero same-sign continuous differences, W=0 has exact value $2/2^{60} \approx 1.735e{-}18$, rather than 1.630e−11. The reported values resemble normal approximations, but the actual zero/tie handling and software method require code. Tables retain source values as descriptive, unverified summaries. Shared system records, equations, trajectories, and noise realizations must be handled at the appropriate independent unit before confirmatory inference.

5.6 | Findings, Losses, and Scope

The tested smoothing–screening–reweighting combination has the highest reported pooled exact rate at 5% noise, alongside lower true-support coefficient error. The baselines have different derivative preprocessing, which limits method-level attribution. Removal contrasts show differences among these variants. The multiscale chain remains unrecovered, and reported omitted-term details require raw support traces.

The core has larger noiseless single-scale coefficient errors than STLSQ and lower chain recall at 5% (0.833 versus 0.992), while selecting fewer false terms. H1 about learned slow-coordinate recovery and H2 about seeded refinement remain untested. Three fixed systems and a noise sweep cannot estimate a general separation or record-length boundary for H3.

The empirical scope is full-state polynomial ODE identification for three fixed systems with twenty realizations per cell. The hidden-state, neural-decomposition, symbolic-refinement, and Pareto claims remain proposed architecture components. Chain failure informs limitations of the tested realization without falsifying an unexecuted decomposition.

The reported proof-of-concept outcomes require the original trajectory generation, noise realizations, delivered supports, coefficient vectors, and analysis code for verification.

6 | Discussion: Limitations and Counterarguments

6.1 | Limitations

Slow-manifold approximation errors depend on separation, stability, regularity, domain, and observation horizon [6], [7]. Partial observations need not identify physical coordinates. Several candidate laws can fit finite records, and library or operator restrictions define what can be recovered. The proposed Pareto representation may expose ambiguity but does not eliminate it. None of these full-architecture issues is resolved by the present full-state regression experiment.

The proposed decomposition cannot remove closure error merely by penalizing residuals: unresolved fast modes may influence a slow approximation, and proposed robust losses or uncertainty weights have not been executed or validated here. Partial observability adds a further layer, since delay embedding under its assumptions can recover a representation of attractor dynamics, not unique named physical coordinates or an identified sparse law [8]. Real records also contain drift, unmodeled inputs, and instrument artifacts; the current full-state simulated campaign does not establish transfer to them.

6.2 | Alternative Viewpoints and Responses

Stage outputs can make failures easier to inspect only if they are actually generated and evaluated. The current ablations do not test neural-stage attribution or symbolic-refinement fronts. Matched preprocessing and independent rollout validation would make the regression comparison more informative.

Longer or cleaner records may help when remeasurement is feasible. For archived records, the proposed hybrid pipeline would still need an explicit approximation horizon and an independently held-out rollout test before it could be treated as a reduced model. Fast variables need not disappear after a transient, and a finite-horizon validation target cannot guarantee global asymptotic fidelity.

7 | Conclusion and Future Directions

The reported campaign shows a higher pooled exact-recovery rate for the specific smoothing–screening–reweighting core than for the tested raw-finite-difference baselines at 5% noise, while the multiscale chain remains unrecovered. The 10% Duffing rate corresponds to two failures among twenty realizations. Unequal differentiation, unresolved aggregation units, and unverified inferential implementation limit the strength of the comparison. The full hidden slow-fast and symbolic architecture remains unexecuted.

Next steps are matched-smoothed baseline comparisons, verification of supports and test calculations, and independent rollout validation. Hidden-state and reduced-manifold discovery require explicit decomposition tests; evolutionary refinement requires matched-budget seeded/unseeded comparisons and a defined Pareto selection rule. Original data and executable records must be archived before reproducibility is claimed.

Author Contributions

All authors read and approved the final manuscript.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The study uses generated ODE trajectories. Noise realizations, initial conditions, sample times, supports, coefficients, and raw outputs are required for reproduction; a repository identifier is not specified.

Code Availability

A public repository and archived version for the reported implementation are not specified.

Conflicts of Interest

The authors declare no conflicts of interest.

Generative AI Disclosure

AI-assisted drafting tools were used in preparing the manuscript text under the direction and scientific responsibility of the authors; all methodological choices, claims, and final content were determined and verified by the authors.

References

[1] Michaelis, L., & Menten, M. L. (1913). Die kinetik der invertinwirkung. *Biochem. z*, *49*(333–369), 352. https://www.google.com/goto?url=CAESkwEBxCJF1wK-GOVfJ3ezRkPV2vcfbNziyq4F-1yzdwHUvDMVf9YvEA1zynTMVR4yzQnT4A71alHCWQyaEdC7xsKynoP9_c4h-ufe4iDkDNdDXVlV-BpbTB06qFdC_CO_igeVLB-ytytxObddbn_jrzXxdNab27g-dFkL3a5DOrDuOQNOXDu4VjyVcqATnZjgUBVkIdA

[2] Bongard, J., & Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, *104*(24), 9943–9948. https://doi.org/10.1073/pnas.0609476104

[3] Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, *324*(5923), 81–85. https://doi.org/10.1126/science.1165893

[4] Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, *113*(15), 3932–3937. https://doi.org/10.1073/pnas.1517384113

[5] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, *31*. https://proceedings.neurips.cc/paper_files/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html

[6] Fenichel, N. (1979). Geometric singular perturbation theory for ordinary differential equations. *Journal of differential equations*, *31*(1), 53–98. https://doi.org/10.1016/0022-0396(79)90152-9

[7] Verhulst, F. (2005). *Methods and applications of singular perturbations: boundary layers and multiple timescale dynamics*. Springer. https://www.amazon.com/dp/1441919929?lv=shuf&channelId=500&plpRedirect=mhFallback

[8] Takens, F. (2006). Detecting strange attractors in turbulence. *Dynamical systems and turbulence, warwick 1980: proceedings of a symposium held at the university of warwick 1979/80* (pp. 366–381). Springer. https://doi.org/10.1007/BFb0091924

[9] Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Science advances*, *3*(4), e1602614. https://doi.org/10.1126/sciadv.1602614

[10] Udrescu, S.-M., & Tegmark, M. (2020). AI Feynman: A physics-inspired method for symbolic regression. *Science advances*, *6*(16), eaay2631. https://doi.org/10.1126/sciadv.aay2631

[11] Cranmer, M. (2023). *Interpretable machine learning for science with PySR and symbolicregression*. https://doi.org/10.48550/arXiv.2305.01582

[12] Zheng, P., Askham, T., Brunton, S. L., Kutz, J. N., & Aravkin, A. Y. (2018). A unified framework for sparse relaxed regularized regression: SR3. *IEEe access*, *7*, 1404–1423. https://doi.org/10.1109/ACCESS.2018.2886528

[13] Champion, K., Lusch, B., Kutz, J. N., & Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the national academy of sciences*, *116*(45), 22445–22451. https://doi.org/10.1073/pnas.1906995116

[14] De Silva, B. M., Champion, K., Quade, M., Loiseau, J. C., Kutz, J. N., & Brunton, S. L. (2020). *Pysindy: A python package for the sparse identification of nonlinear dynamics from data*. https://doi.org/10.48550/arXiv.2004.08424

[15] Kaiser, E., Kutz, J. N., & Brunton, S. L. (2018). Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings mathematical, physical, and engineering sciences*, *474*(2219), 20180335. https://doi.org/10.1098/rspa.2018.0335

[16] Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., … & Edelman, A. (2020). *Universal differential equations for scientific machine learning*. https://doi.org/10.48550/arXiv.2001.04385

[17] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of computational physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[18] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature reviews physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5%0A%0A

[19] Greydanus, S., Dzamba, M., & Yosinski, J. (2019). Hamiltonian neural networks. *Advances in neural information processing systems* (Vol. 32). NeurIPS Proceedings. https://proceedings.neurips.cc/paper_files/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html

[20] Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., & Ho, S. (2020). *Lagrangian neural networks*. https://doi.org/10.48550/arXiv.2003.04630

[21] Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, *656*, 5–28. https://doi.org/10.1017/S0022112010001217

[22] Chartrand, R. (2011). Numerical differentiation of noisy, nonsmooth data. *International scholarly research notices*, *2011*(1), 164564. https://doi.org/10.5402/2011/164564

[23] Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics: methodology and distribution* (pp. 492–518). Springer. https://doi.org/10.1007/978-1-4612-4380-9_35%0A%0A

[24] Meinshausen, N., & Bühlmann, P. (2010). Stability selection. *Journal of the royal statistical society: Series b (statistical methodology)*, *72*(4), 417–473. https://doi.org/10.1111/j.1467-9868.2010.00740.x

[25] Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, *19*(6), 716–723. https://doi.org/10.1109/TAC.1974.1100705

[26] Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 461–464. https://www.jstor.org/stable/2958889

[27] Lotka, A. J. (1998). *Analytical theory of biological populations*. Springer Science & Business Media. https://doi.org/10.1007/978-1-4757-9176-1

[28] Duffing, G. (1918). *Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung*. Vieweg. https://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp106773

[29] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

[30] Fasel, U., Kutz, J. N., Brunton, B. W., & Brunton, S. L. (2022). Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings mathematical, physical, and engineering sciences*, *478*(2260), 20210904. https://doi.org/10.1098/rspa.2021.0904

[31] Savitzky, A., & Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, *36*(8), 1627–1639. https://doi.org/10.1021/ac60214a047

[32] Golub, G. H., Heath, M., & Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, *21*(2), 215–223. https://doi.org/10.1080/00401706.1979.10489751

[33] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., & Cournapeau, D. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[34] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

[35] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, *1*, 80. https://doi.org/10.2307/3001968

[36] Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, *6*(2), 65–70. https://www.jstor.org/stable/4615733

[37] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, *32*, 675–701. https://doi.org/10.1080/01621459.1937.10503522