SYMBOLIC BY DESIGN: ZERO-PRIOR NEURAL ODE DISCOVERY VIA DECOUPLED AFFINE KANS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural scientific discovery seeks governing laws that can be inspected and reused. It should not return only predictors that interpolate observed trajectories. For ordinary differential equations (ODEs), flexible neural models often fit noisy dynamics but hide the law in non-symbolic functions. Symbolic regression can return formulas. However, it often requires a suitable library, operator set, differential order, or search budget. We propose DA-KAN, a decoupled affine Kolmogorov-Arnold network for affine-coefficient ODE discovery in a restricted zero-prior setting. The learner is not given the true differential order or the final symbolic coefficients. DA-KAN represents each ODE coefficient with a global affine path and an annealed local RBF residual path. The affine path carries the recoverable law. The residual path absorbs optimization mismatch and noise during training. The pipeline combines this cell with Wronskian multi-trajectory excitation, cascade order validation, and a fixed affine extractor based on projection and residual-checked snapping. Under a strict five-seed protocol on a 20-system affine-coefficient ODE benchmark, DA-KAN recovered 100/100 equations exactly through 2% derivative-channel noise without access to the true order. It recovered 94/100, 89/100, and 88/100 systems at 3%, 4%, and 5% noise, respectively. In the main fixed-budget comparison, it matched the best external methods at clean and 1% noise and achieved the highest external-baseline recovery from 2% to 5% noise. These results support the claim that a complete affine discovery protocol with explicit coefficient structure, order validation, and residual-checked extraction improves exact equation recovery in scoped affine-coefficient ODE discovery.

1 INTRODUCTION

Scientific discovery from data requires models that recover mechanisms, not only accurate predictions. For dynamical systems, this goal requires compact ordinary differential equations (ODEs). Their terms should be inspectable, simulatable, and reusable beyond the observed trajectories. Current discovery methods face a persistent tension. Symbolic regression methods search directly for readable formulas. Early examples included SINDy and PDE-FIND (Brunton et al., 2016; Rudy et al., 2017). Physics-inspired systems such as AI Feynman provided another route (Udrescu & Tegmark, 2020). Neural and search-augmented systems later expanded this class (Petersen et al., 2021; Biggio et al., 2021; Cranmer, 2023; Scholl et al., 2025; Yu et al., 2025). These methods returned closed-form equations. However, their success often depended on choices made before search. These choices included the candidate library, operator set, differential order, pruning rule, and search budget. Figure 1 illustrates this search-based route.

Flexible neural models took the opposite route. Neural ODEs and physics-informed neural networks fitted continuous-time dynamics and imposed differential residuals during training (Chen et al., 2018; Raissi et al., 2019). Kolmogorov-Arnold Networks (KANs) exposed learnable univariate functions on network edges and became attractive for interpretable scientific modeling (Liu et al., 2025). Recent KAN-based symbolic regression and extraction methods showed that learned edge functions could guide formula search (Bühler & Guillén-Gosálbez, 2025; Faroughi et al., 2026; Howard et al., 2026). Some methods also regularized edge functions toward symbolic primitives. Others coupled KAN components with sparse identification. In-context symbolic regression for KANs highlighted a failure mode in this setting. Post-hoc per-edge conversions could be unstable under small training changes.

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.

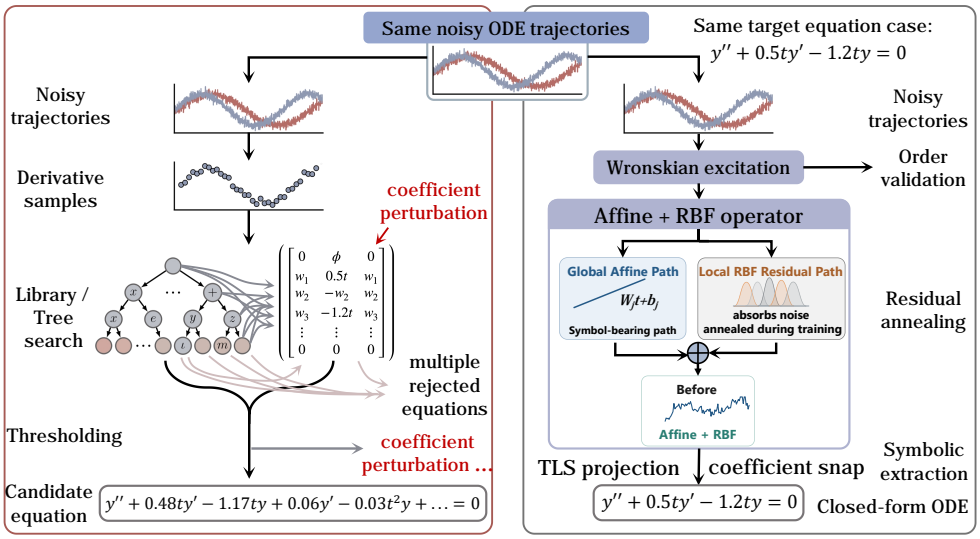


Figure 1: **Structural contrast between conventional symbolic regression and DA-KAN.** Conventional symbolic regression obtains interpretability after derivative estimation, library construction, and sparse or expression search. Recovery is therefore sensitive to choices of operators, libraries, orders, and search budgets. DA-KAN instead embeds each coefficient in a decoupled neural cell. The global affine path carries the recoverable symbolic law. The local RBF residual path temporarily absorbs optimization mismatch and noise before annealing, affine projection, and residual-checked snapping.

They could also propagate errors when local operator choices were not scored in the end-to-end context (Sovrano et al., 2026). For ODE discovery, the relevant context is the differential residual shared across derivative channels and trajectories. A small residual on one trajectory does not identify either the differential order or the coefficient law. Flexible local functions can compensate for a missing derivative channel or a biased coefficient. Thus, fitting a smooth coefficient curve is not the same as recovering a law. A spline or radial-basis expansion may reduce residual error. It may still fail to collapse to a constant, an affine coefficient, or another compact expression that satisfies the global ODE residual. This gap motivates our central question: can symbolic recoverability be made an architectural property of neural ODE discovery, rather than a post-hoc outcome of expression search?

We study this question in restricted zero-prior affine-coefficient ODE discovery. Here, we use "zero-prior" in a restricted non-oracle sense. The learner does not receive the true differential order or the final symbolic coefficients. It does not mean that the model has no family-level inductive bias. We target scalar ODEs of the form

$$y^{(N)}(t) + \sum_{j=0}^{N-1} \Phi_j(t) y^{(j)}(t) = 0, \tag{1}$$

where each coefficient function $\Phi_j(t)$ is constant or affine in the independent variable. This family is narrower than general symbolic regression. It captures interpretable linear ODEs with time-varying coefficients. It also gives a controlled setting for exact symbolic recovery under noisy observations and non-oracle order selection.

We propose DA-KAN, a decoupled affine Kolmogorov-Arnold network that makes *symbolifiability* explicit in the model design. For each ODE coefficient, DA-KAN separates the representation into a global affine path and a local radial-basis-function (RBF) residual path. The affine path carries the recoverable symbolic law. The RBF path absorbs optimization mismatch and noise during training. Before extraction, the residual path is penalized and annealed. An accepted equation must therefore be explained by the affine scaffold rather than opaque local basis functions. Within the affine-coefficient

ODE family, this design reduces symbolic discovery from a large expression-tree search to recovery of a low-dimensional coefficient operator.

The DA-KAN pipeline couples this operator with three mechanisms for the restricted zero-prior protocol. Wronskian multi-trajectory excitation improves the identifiability of the coefficient functions. Cascade order validation tests whether a lower-order fit remains valid after its local correction path is removed. A fixed adaptive affine extractor compares direct affine readout with affine projection. It then applies residual-checked coefficient snapping with a small complexity penalty. Together, these components preserve a flexible training path and enforce a compact symbolic route at extraction time.

The experiments supported three conclusions. First, the benchmark contained 20 second- and third-order affine-coefficient ODEs. Under five base seeds, DA-KAN recovered 100/100 equations exactly through 2% derivative-channel noise without oracle access to the true order. It recovered 94/100, 89/100, and 88/100 systems at 3%, 4%, and 5% noise, respectively. A direct-readout ablation bypassed adaptive affine extraction. It recovered 100/100, 97/100, 79/100, and 57/100 systems at 0%, 1%, 2%, and 3% noise, respectively. These results showed the value of the extraction stage within the unified DA-KAN pipeline. Second, DA-KAN tied the strongest external baselines at clean and 1% noise and outperformed the SINDy variants, PySR, and a non-decoupled pykan coefficient learner from 2% to 5% noise. Separate PySR order and budget analyses showed that expression-search performance remained competitive but depended on the search protocol. Third, the scope analysis clarified that the claim concerned scalar affine-coefficient ODEs. Small compatible MDBench cases provided an external sanity check. Nonlinear ODEs, biological network dynamics, and PDE settings required extensions beyond the present architecture.

The contributions are:

- We identify symbolifiability as a distinct objective in neural ODE discovery. Accurate residual fitting is insufficient unless the learned representation exposes an extractable closed-form law.
- We introduce DA-KAN, a decoupled affine KAN coefficient operator. It separates a symbol-bearing affine path from an annealed local residual path, making affine symbolic recovery an architectural property.
- We develop a restricted zero-prior discovery protocol. It combines Wronskian multi-trajectory excitation, a rank-based identifiability condition, cascade order validation, affine projection, and residual-checked snapping. The learner does not receive the true differential order.
- We provide a scoped exact-recovery evaluation with explicit reporting conventions. The evaluation shows robust recovery on affine-coefficient ODEs. It delimits nonlinear ODE, biological ODE, and PDE settings as extension targets.

2 Related Work

Symbolic law and governing-equation discovery. Early symbolic law-discovery systems showed that nonlinear dynamical equations and compact natural laws could be inferred from data through automated model search and symbolic regression (Bongard & Lipson, 2007; Schmidt & Lipson, 2009). SINDy made this idea practical for nonlinear dynamical systems by combining a candidate function library with sparse regression (Brunton et al., 2016). PDE-FIND extended the same sparse-discovery principle to partial differential equations (Rudy et al., 2017). Subsequent variants improved robustness to noisy or limited data. They included weak-form identification and ensemble sparse discovery (Messenger & Bortz, 2021; Fasel et al., 2022). PDE-Net used learnable differential filters for PDE discovery (Long et al., 2018). They returned explicit governing equations rather than only predictors. They assumed that the discovery procedure had access to relevant operators, candidate terms, or a sparse model class. DA-KAN has a narrower target family but a different mechanism. Instead of searching a broad dictionary, it represents the admissible affine law with an explicit global path. It then validates whether that path explains the trained residual.

Neural symbolic regression and differentiable dynamics. AI Feynman reduced symbolic search by exploiting physics-inspired regularities such as separability and symmetry (Udrescu & Tegmark, 2020). Deep Symbolic Regression trained a neural policy to emit expression trees (Petersen et al.,

2021). Neural symbolic regression at scale used large synthetic pretraining to map point sets to candidate formulas (Biggio et al., 2021). Symbolic model extraction used inductive biases to distill analytic relations from deep representations (Cranmer et al., 2020). Recent symbolic-regression systems extended this comparison class. ParFam converted parts of discrete expression search into continuous optimization over parametric families. MDLformer-guided search replaced pure prediction-error ranking with a minimum-description-length objective (Scholl et al., 2025; Yu et al., 2025). These works had a broader scope than DA-KAN. Their generality required larger operator spaces, expression-tree choices, pretraining corpora, or search budgets. Neural ODEs learned continuous-time vector fields through differentiable solvers (Chen et al., 2018). Physics-informed neural networks imposed differential residuals as learning constraints for forward and inverse problems (Raissi et al., 2019). These models demonstrated the value of differentiable dynamics, but their learned functions did not automatically yield compact equations. For the problem studied here, a model can fit derivative residuals accurately while hiding the recoverable coefficient law in flexible basis functions. This fitability–symbolifiability mismatch motivated the decoupled DA-KAN operator and its fixed extraction protocol.

KAN-based symbolic discovery. Kolmogorov-Arnold Networks replaced fixed node activations with learnable univariate edge functions. They were proposed as interpretable alternatives to multi-layer perceptrons in scientific tasks (Liu et al., 2025). KAN-SR and related KAN-based extraction methods used KAN edge functions to guide formula search or replace learned curves with analytic primitives (Bühler & Guillén-Gosálbez, 2025; Faroughi et al., 2026). SINDy-KANs combined KAN function composition with sparse-identification ideas for nonlinear dynamics. They therefore provided a relevant 2026 reference point for interpretable dynamical discovery (Howard et al., 2026). These studies showed that KANs were a promising substrate for symbolic extraction. They also exposed a practical difficulty. Flexible edge functions could fit data well without yielding a globally valid symbolic law. In-context symbolic regression made this issue explicit. It showed that isolated per-edge KAN-to-symbol fitting could be unstable and could ignore the effect of local substitutions on the full model objective (Sovrano et al., 2026). DA-KAN addresses a different, ODE-specific level of structure. It does not treat each KAN edge as an independent symbolic candidate or search over a general operator library. Instead, it represents each ODE coefficient as a decoupled affine-plus-residual cell. It accepts a formula only after order validation, affine projection, and residual-checked snapping against the differential residual.

Dynamics benchmarks and scope. Formal benchmarks for model discovery, biological dynamical systems, and PDE learning clarified the boundary of the present study (Villaverde et al., 2015; Takamoto et al., 2022; Bideh et al., 2026). MDBench provided a broad benchmark for model discovery across ODEs and PDEs. It included equation-level evaluation under noise (Bideh et al., 2026). BioPreDyn-Bench targeted dynamic modeling in systems biology (Villaverde et al., 2015). PDEBench focused on time-dependent PDE simulation tasks for scientific machine learning (Takamoto et al., 2022). Search-based dynamics discovery, such as the Symbolic Physics Learner, also targeted nonlinear governing equations through expression-tree exploration (Sun et al., 2023). These settings are intentionally broader than the current model class. We used them to position the contribution. DA-KAN's claims are limited to scalar affine-coefficient ODE discovery under restricted zero-prior order and coefficient recovery. Nonlinear ODEs, biological network dynamics, and PDE discovery remain extension targets.

3 METHOD

DA-KAN was a discovery pipeline for scalar affine-coefficient ODEs under the restricted zero-prior protocol defined above. During optimization, the model used local flexibility. The final equation had to use explicit low-dimensional affine coefficients. The pipeline separated derivative excitation, decoupled coefficient learning, cascade order validation, and fixed adaptive affine extraction. Figure 2 summarizes the recovery procedure. The pipeline had four stages. First, we stacked derivative observations from multiple trajectories. The same coefficient functions then had to explain different solution curves. Second, for a candidate order N, we represented every ODE coefficient with a decoupled affine KAN cell. Third, a cascade validation rule accepted an order only when the learned dynamics remained accurate after the local residual path was removed. Finally, a fixed extraction rule converted the accepted affine scaffold into a symbolic ODE. The rule used direct readout, total-least-

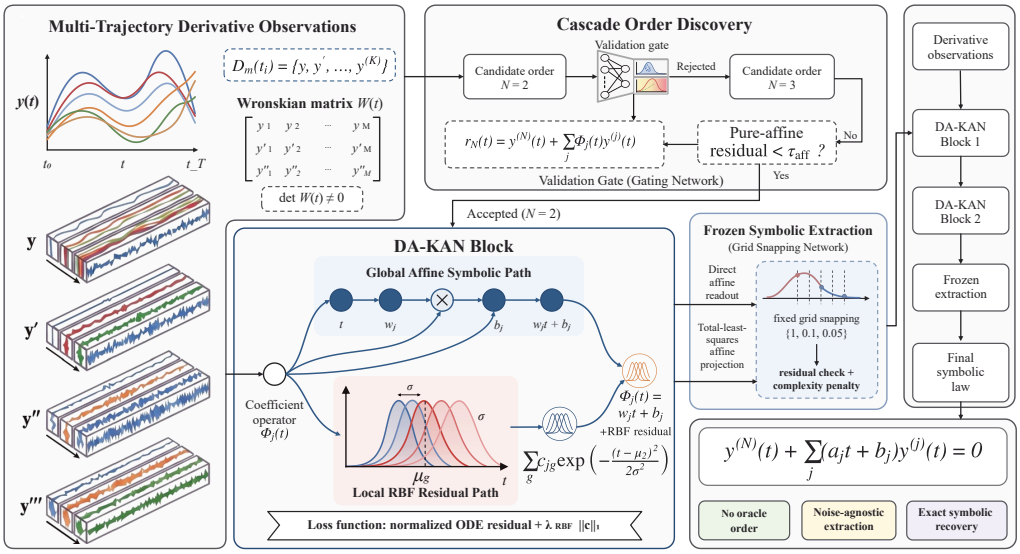


Figure 2: **DA-KAN pipeline for restricted zero-prior affine-coefficient ODE discovery.** Multi-trajectory derivative observations provided excitation for coefficient identification. For each candidate order, DA-KAN represented every coefficient with a global affine path and a local RBF residual path. It trained the normalized differential residual and validated whether the pure affine path explained the candidate order. The accepted order was passed to a fixed adaptive affine extractor. The extractor compared direct affine readout with affine projection. It applied coefficient snapping. It selected the symbolic equation by residual and complexity.

squares affine projection, residual-checked snapping, and a small complexity penalty. We reported direct affine readout only as an ablation. The main DA-KAN result included the fixed adaptive extraction stage.

3.1 Problem Formulation and Scope

We observed M trajectories of a scalar state $y(t)$ on a shared time grid $\{t_i\}_{i=1}^{T}$. For trajectory m, the available derivative observation at t_i was

$$D_m(t_i) = \{y_m(t_i), y'_m(t_i), \ldots, y_m^{(K)}(t_i)\}, \tag{2}$$

where K was the largest available derivative order. We wrote $\mathcal{D} = \{D_m(t_i) : m = 1, \ldots, M;\ i = 1, \ldots, T\}$ for the full derivative dataset. The task was to recover both the differential order N and a compact monic ODE

$$y^{(N)}(t) + \sum_{j=0}^{N-1} \Phi_j(t) y^{(j)}(t) = 0, \tag{3}$$

without giving the learner the true order or the final symbolic coefficients. We focused on the affine-coefficient family

$$\Phi_j(t) = a_j t + b_j, \tag{4}$$

where $\{a_j, b_j\}_{j=0}^{N-1}$ were the symbolic quantities to be recovered. Thus, "zero-prior" referred to a restricted non-oracle setting, not to the absence of a family-level inductive bias. In the implementation, the candidate order set was $\mathcal{N} = \{2, 3\}$. It matched the second- and third-order affine-coefficient systems used in the experiments.

3.2 Multi-Trajectory Excitation

For a candidate order N, the residual equations were shared across trajectories:

$$r_{m,N}(t_i) = y_m^{(N)}(t_i) + \sum_{j=0}^{N-1} \Phi_j(t_i) y_m^{(j)}(t_i). \tag{5}$$

At each time point, the lower-order derivative channels formed a Wronskian-style excitation matrix

$$\mathbf{W}_N(t_i) = \begin{bmatrix} y_1^{(0)}(t_i) & \cdots & y_1^{(N-1)}(t_i) \\ \vdots & & \vdots \\ y_M^{(0)}(t_i) & \cdots & y_M^{(N-1)}(t_i) \end{bmatrix}. \tag{6}$$

If a single trajectory made columns nearly dependent, different coefficient combinations could yield similar residuals. For $M = N$, $\mathbf{W}_N(t_i)^\top$ was the usual Wronskian matrix of the N trajectories. For $M \geq N$, the corresponding requirement was full column rank.

Rank condition. Full column rank of $\mathbf{W}_N(t_i)$ makes the coefficient values pointwise identifiable in the noiseless affine family. With at least two distinct time points, the affine restriction in Equation 4 then identifies each pair (a_j, b_j). Using distinct initial conditions improved rank and conditioning over the grid; the benchmark used two trajectories for second-order systems and three for third-order systems. The full rank argument and noise-conditioning discussion are provided in the appendix.

3.3 Decoupled Affine Cell and Training Objective

The core architectural choice was to make the symbolic part of each coefficient explicit. For each $j < N$, DA-KAN parameterized

$$\Phi_j(t) = \underbrace{a_j t + b_j}_{\text{global affine path}} + \underbrace{\rho_j(t)}_{\text{local residual path}}. \tag{7}$$

with

$$\rho_j(t) = \sum_{g=1}^{G} c_{jg} \exp\left(-\frac{(t-\mu_g)^2}{2\sigma^2}\right). \tag{8}$$

The centers $\{\mu_g\}_{g=1}^G$ were fixed on the problem domain. The implementation set the width from the grid spacing as

$$\sigma = \frac{t_{\max} - t_{\min}}{0.4G}. \tag{9}$$

The learnable parameters were the affine coefficients (a_j, b_j) and the RBF weights $\{c_{jg}\}_{g=1}^G$. Figure 3 illustrates how the two branches entered the ODE residual. The affine path was the symbol-bearing route used in the final equation. The RBF residual path was excluded from the final symbolic law. It gave the optimizer a local correction channel for noise and mismatch. The training procedure penalized and annealed this path. This decoupling differed from a generic KAN coefficient learner, where the entire learned univariate function had to be approximated or interpreted after training.

For candidate order N, DA-KAN minimized a normalized differential residual over all trajectories and grid points. We defined

$$\rho_{N,m}(t_i) = 1 + \sum_{k=0}^{N} |y_m^{(k)}(t_i)|. \tag{10}$$

The training objective was

$$\mathcal{L}_N = \frac{1}{MT} \sum_{m=1}^{M} \sum_{i=1}^{T} \left(\frac{r_{m,N}(t_i)}{\rho_{N,m}(t_i)}\right)^2 + \lambda_{\text{rbf}} \sum_{j=0}^{N-1} \sum_{g=1}^{G} |c_{jg}|. \tag{11}$$

The normalization prevented high-magnitude derivative channels from dominating the fit. The ℓ_1 penalty acted only on the RBF weights. It preserved the affine path as the low-dimensional route for symbolic recovery. After a warm-up period, we soft-thresholded the RBF weights:

$$c_{jg} \leftarrow \text{sign}(c_{jg}) \max(|c_{jg}| - \eta, 0), \tag{12}$$

This procedure gradually removed local corrections that were not needed to explain the residual. The experimental implementation used Adam (Kingma & Ba, 2015) with learning rate 0.04, $\lambda_{\text{rbf}} = 0.01$, and $\eta = 0.01$. For $N = 2$, it trained for 800 steps and began annealing after step 600. For $N = 3$, it trained for 1200 steps and began annealing after step 800. These budgets were fixed across systems rather than tuned for each equation.

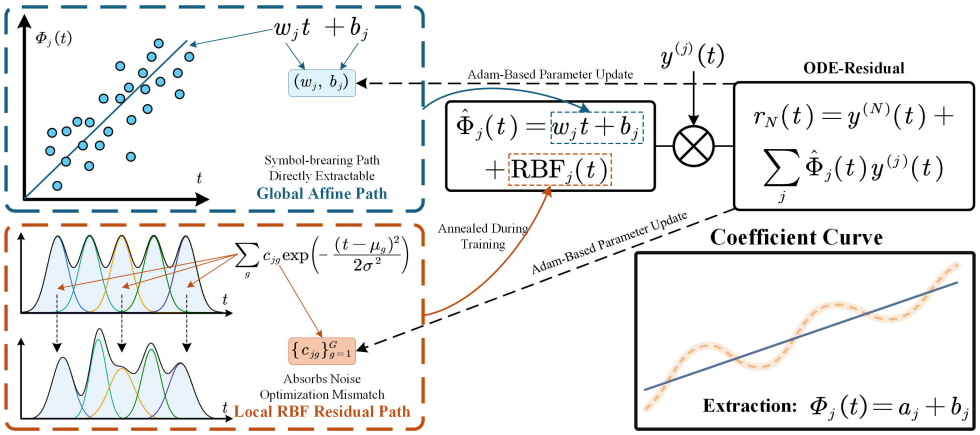


Figure 3: **Decoupled affine KAN coefficient cell.** Each ODE coefficient was represented by a global affine path and a local RBF residual path. The affine path carried the coefficient used in the final recovered law. The RBF path provided local flexibility during optimization. The training procedure penalized and annealed it before extraction. The resulting coefficient multiplied the corresponding derivative channel inside the ODE residual. We extracted the final symbolic equation from the affine scaffold after residual checking.

3.4 Cascade Order Validation

We handled order discovery with an increasing-order cascade. After training order N, DA-KAN evaluated the residual obtained by deleting the local RBF paths:

$$E_{\mathrm{aff}}(N) = \frac{1}{MT} \sum_{m=1}^{M} \sum_{i=1}^{T} \left(\frac{\delta_{m,N}^{\mathrm{aff}}(t_i)}{\rho_{N,m}(t_i)} \right)^2 .$$
$$\delta_{m,N}^{\mathrm{aff}}(t_i) = y_m^{(N)}(t_i) + \sum_{j=0}^{N-1} (a_j t_i + b_j) y_m^{(j)}(t_i). \tag{13}$$

The candidate was accepted if $E_{\mathrm{aff}}(N) \le \tau_{\mathrm{aff}}$. Otherwise, the cascade moved to the next order. This criterion was stricter than checking the full neural residual. It asked whether the symbolic path alone explained the data. It prevented a flexible residual path from masking a missing derivative channel at a lower candidate order. In our experiments, the cascade trained $N = 2$ first. It accepted this candidate when $\tau_{\mathrm{aff}} = 0.02$ was satisfied. Otherwise, it trained the $N = 3$ model and proceeded to extraction at the largest candidate order. If a residual path remained large at extraction time, we treated the resulting expression as unresolved and counted it as a failed exact recovery.

3.5 Fixed Adaptive Extraction and Equation Output

Once an order was selected, DA-KAN extracted coefficients with a fixed rule rather than a per-system search.

The extractor details are given separately. We collected the affine coefficients as

$$\alpha = (b_0, a_0, b_1, a_1, \ldots, b_{N-1}, a_{N-1})^\top \tag{14}$$

The direct candidate α^{dir} was read from the affine paths of the trained cells. We also formed an affine projection candidate from the stacked linear system

$$Z_N \alpha + d_N \approx 0, \tag{15}$$

where each row of Z_N contained

$$[y_m(t_i),\ t_i y_m(t_i), \ldots,\quad y_m^{(N-1)}(t_i),\ t_i y_m^{(N-1)}(t_i)] \tag{16}$$

and d_N contained $y_m^{(N)}(t_i)$. We computed the projection by total least squares (Golub & Van Loan, 1980) on the scaled augmented matrix $[Z_N, d_N]$. This matched the extraction script used in the experiments.

We snapped both the direct and projected candidates to simple rational grids with denominators in $\{1, 2, 5, 10, 20\}$. We included a zero candidate when a coefficient was sufficiently small. DA-KAN then evaluated the normalized residual

$$R_N(\alpha) = \frac{1}{MT} \sum_{m=1}^{M} \sum_{i=1}^{T} \left(\frac{\delta_{m,N}^{\alpha}(t_i)}{\rho_{N,m}(t_i)} \right)^2$$
$$\delta_{m,N}^{\alpha}(t_i) = y_m^{(N)}(t_i) + \sum_{j=0}^{N-1} (a_j t_i + b_j) y_m^{(j)}(t_i). \tag{17}$$

and selected

$$\widehat{\alpha} = \arg\min_{\alpha \in \mathcal{A}} R_N(\alpha) + \lambda_C C(\alpha). \tag{18}$$

Here $\mathcal{A}$ contained the snapped direct and snapped projection candidates. The complexity term $C(\alpha)$ ignored zeros and mildly favored small rational denominators from $\{1, 2, 5, 10, 20\}$. The implementation used $\lambda_C = 3 \times 10^{-4}$ in the main DA-KAN extraction. This stage was part of DA-KAN. Direct affine readout served as an ablation that isolated the trained coefficient cell. The complete method could correct small training bias when residual-checked affine projection yielded a simpler and more consistent equation. The full discovery and extraction procedure is given in the appendix.

The final output was the monic equation

$$y^{(\widehat{N})}(t) + \sum_{j=0}^{\widehat{N}-1} (\widehat{a}_j t + \widehat{b}_j) y^{(j)}(t) = 0. \tag{19}$$

This output form was deliberately narrower than general symbolic regression. It made exact recovery auditable. Every coefficient in the final law was visible and low-dimensional. We checked each coefficient against the normalized differential residual. We therefore evaluated exact symbolic recovery at the equation level. We separately reported trajectory integration error and ablation results. These results showed when fitting accuracy and symbolifiability diverged.

4 Experiments

The experiments tested whether explicit coefficient structure, non-oracle order validation, and residual-checked extraction improve exact law recovery under noisy derivative observations. We asked whether DA-KAN could recover exact equations under noise, whether adaptive extraction helped, which architectural components mattered, and where the affine-coefficient design applies.

4.1 Experimental Setup and Evaluation Protocol

Benchmark. The main benchmark contained 20 scalar ODEs with affine or constant coefficients. It consisted of 10 second-order and 10 third-order systems. The equations included constant-coefficient oscillators, damped systems, Airy-type equations, and mixed time-varying coefficient systems. We simulated each system on its own time domain with 200 grid points. Second-order systems used two independent initial conditions, and third-order systems used three. These counts matched the number of independent state directions available for each order. The learner was not given the true differential order. It selected from the candidate set $\mathcal{N} = \{2, 3\}$. The strict headline protocol repeated the full 20-system sweep with five base seeds, $\{42, 43, 44, 45, 46\}$. Thus, each noise level contained 100 recovery trials.

Noise model. We evaluated derivative-channel Gaussian noise at levels $\epsilon \in \{0, 0.01, \ldots, 0.05\}$. For each available channel $y^{(k)}$, the noisy observation was

$$\widetilde{y}^{(k)}(t_i) = y^{(k)}(t_i) + \epsilon \, \mathrm{std}(y^{(k)}) \, \xi_i, \xi_i \sim \mathcal{N}(0, 1). \tag{20}$$

This setting isolated symbolic recovery from numerical differentiation of raw observations. For each system, seed, and noise level, the noisy derivative tensors were generated once and reused in memory by every method.

Table 1: **Exact symbolic recovery on the 20-system affine-coefficient ODE benchmark.** Each entry aggregates five base seeds, so each noise level contains 100 recovery trials. All methods used the same noisy derivative tensors and non-oracle order selection when applicable. PySR numbers follow the main fixed-budget comparison used for the headline benchmark. Internal DA-KAN diagnostics, including direct readout and TLS-only projection, are reported separately in the appendix. Bold marks the best recovery count in each noise column.

Method	0%	1%	2%	3%	4%	5%
DA-KAN	**100/100**	**100/100**	**100/100**	**94/100**	**89/100**	**88/100**
PySR	68/100	66/100	68/100	72/100	70/100	64/100
E-SINDy	**100/100**	**100/100**	99/100	91/100	79/100	70/100
WSINDy	**100/100**	54/100	41/100	35/100	31/100	28/100
SINDy	**100/100**	0/100	1/100	0/100	0/100	0/100
KAN (pykan)	5/100	0/100	0/100	0/100	0/100	0/100

Metrics. We used *exact symbolic recovery* as the primary metric. After order selection and coefficient extraction, we compared the recovered monic ODE with the ground-truth symbolic equation. An equation counted as exact only under the extractor's coefficient-snapping convention. We also reported order correctness, normalized residuals, trajectory mean-squared error (MSE), and wall-clock seconds per system. We treated MSE as a diagnostic rather than the main metric. A model could fit trajectories while failing to return the correct symbolic law.

Baselines and Protocol. We compared DA-KAN with SINDy (Brunton et al., 2016), WSINDy (Messenger & Bortz, 2021), E-SINDy (Fasel et al., 2022), PySR (Cranmer, 2023), and the non-decoupled KAN baseline (Liu et al., 2025) under the same derivative-observation setting. Every method selected its order from $\{2, 3\}$ under the same normalized residual threshold. The appendix gives the full extractor settings, internal diagnostics, and PySR budget sweep.

4.2 Exact Symbolic Recovery under Noise

Table 1 reports the headline result. DA-KAN recovered 100/100 equations exactly through 2% derivative-channel noise and 94/100, 89/100, and 88/100 equations at 3%, 4%, and 5% noise. The appendix reports the adaptive-extraction and TLS-only diagnostics.

The sparse baselines showed the opposite trade-off: SINDy was exact in the clean setting but collapsed under noise, WSINDy and E-SINDy were more robust, and PySR remained the strongest expression-search baseline while still being budget-sensitive.

A detailed discussion of fitability versus symbolifiability, and of order selection versus coefficient recovery, is given in the appendix.

4.3 Mechanism Ablations

Ablations in the appendix show that the complete protocol is necessary: direct readout, single-trajectory input, no affine decoupling, no order cascade, and no symbolic snapping all reduce exact recovery.

5 Conclusion

This paper argues that exact scientific recovery improves when symbolic structure is built into the learner rather than left to post-hoc search. DA-KAN combined multi-trajectory excitation, cascade order validation, and residual-checked affine extraction to recover 100/100 benchmark equations through 2% derivative-channel noise and 94/100 at 3%. The scope is intentionally limited to derivative-channel observations and affine-coefficient ODEs.

AI USE STATEMENT

Generative AI assistance was used to migrate the manuscript's LaTeX source from the AAAI template to the ICLR 2027 template and to check compilation and layout.

REFERENCES

Amirmohammad Ziaei Bideh, Aleksandra Georgievska, and Jonathan Gryak. MDBench: Benchmarking data-driven methods for model discovery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19746–19754, 2026. doi: 10.1609/aaai.v40i24.39056. URL https://doi.org/10.1609/aaai.v40i24.39056.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, et al. 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. PMLR, 2021. URL https://proceedings.mlr.press/v139/biggio21a.html.

Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007. doi: 10.1073/pnas.0609476104. URL https://doi.org/10.1073/pnas.0609476104.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.

Marco Andrea Bühler and Gonzalo Guillén-Gosálbez. KAN-SR: A kolmogorov-arnold network guided symbolic regression framework. *arXiv preprint arXiv:2509.10089*, 2025. doi: 10.48550/arXiv.2509.10089. URL https://arxiv.org/abs/2509.10089.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, et al. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper/7892-neural-ordinary-differential-equations.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*, 2023. doi: 10.48550/arXiv.2305.01582. URL https://arxiv.org/abs/2305.01582.

Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, et al. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, 2020.

Salah A. Faroughi, Farinaz Mostajeran, Amirhossein Arzani, et al. Symbolic–KAN: Kolmogorov-arnold networks with discrete symbolic structure for interpretable learning. *arXiv preprint arXiv:2603.23854*, 2026. doi: 10.48550/arXiv.2603.23854. URL https://arxiv.org/abs/2603.23854.

Urban Fasel, J. Nathan Kutz, Bingni W. Brunton, et al. Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2260):20210904, 2022. doi: 10.1098/rspa.2021.0904. URL https://doi.org/10.1098/rspa.2021.0904.

Gene H. Golub and Charles F. Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, 1980. doi: 10.1137/0717073. URL https://doi.org/10.1137/0717073.

Amanda A. Howard, Nicholas Zolman, Bruno Jacob, et al. SINDy-KANs: Sparse identification of non-linear dynamics through kolmogorov-arnold networks. *arXiv preprint arXiv:2603.18548*, 2026. URL https://arxiv.org/abs/2603.18548.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Ziming Liu, Yixuan Wang, Sachin Vaidya, et al. KAN: Kolmogorov–arnold networks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ozo7qJ5vZi.

Zichao Long, Yiping Lu, Xianzhong Ma, et al. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, 2018.

Daniel A. Messenger and David M. Bortz. Weak SINDy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021. doi: 10.1016/j.jcp.2021.110525. URL https://doi.org/10.1016/j.jcp.2021.110525.

Brenden K. Petersen, Mikel Landajuela, T. Nathan Mundhenk, et al. 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.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. 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, 2019. doi: 10.1016/j.jcp.2018.10.045.

Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, et al. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017. doi: 10.1126/sciadv.1602614.

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

Philipp Scholl, Katharina Bieker, Hillary Hauger, et al. ParFam – (neural guided) symbolic regression via continuous global optimization. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8y5Uf6oEiB.

Francesco Sovrano, Lidia Losavio, Giulia Vilone, et al. In-context symbolic regression for robustness-improved kolmogorov-arnold networks. *arXiv preprint arXiv:2603.15250*, 2026. doi: 10.48550/arXiv.2603.15250. URL https://arxiv.org/abs/2603.15250.

Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press, 2 edition, 2015.

Fangzheng Sun, Yang Liu, Jian-Xun Wang, et al. Symbolic physics learner: Discovering governing equations via monte carlo tree search. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ZTK3SefE8_Z.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, et al. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0a9747136d411fb83f0cf81820d44afb-Abstract-Datasets_and_Benchmarks.html.

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.

Alejandro F. Villaverde, David Henriques, Kieran Smallbone, et al. BioPreDyn-bench: a suite of benchmark problems for dynamic modelling in systems biology. *BMC Systems Biology*, 9(1), 2015. doi: 10.1186/s12918-015-0144-4.

Zihan Yu, Jingtao Ding, Yong Li, et al. Symbolic regression via MDLformer-guided search: From minimizing prediction error to minimizing description length. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ljAS7cPAU0.

A Multi-Trajectory Identifiability

This appendix makes explicit the rank argument used by the multi-trajectory stage. It concerns coefficient identifiability in the affine ODE family, not convergence of the neural optimizer or exact symbolic recovery under noise.

A.1 Pointwise Coefficient Uniqueness

Fix a candidate order N and a time point t_i. Let $\mathbf{\Phi}_i = [\Phi_0(t_i), \ldots, \Phi_{N-1}(t_i)]^\mathsf{T}$ and let $\mathbf{d}_{N,i} = [y_1^{(N)}(t_i), \ldots, y_M^{(N)}(t_i)]^\mathsf{T}$. Stacking the shared residual equation across M trajectories gives

$$\mathbf{W}_N(t_i)\mathbf{\Phi}_i = -\mathbf{d}_{N,i}, \tag{21}$$

where $\mathbf{W}_N(t_i)$ is defined in Equation 6. If $\operatorname{rank}(\mathbf{W}_N(t_i)) = N$, then $\mathbf{W}_N(t_i)$ has no nonzero null vector. Thus, if two vectors $\mathbf{\Phi}_i$ and $\widetilde{\mathbf{\Phi}}_i$ both satisfy Equation 21,

$$\mathbf{W}_N(t_i)(\mathbf{\Phi}_i - \widetilde{\mathbf{\Phi}}_i) = \mathbf{0}, \tag{22}$$

which implies $\mathbf{\Phi}_i = \widetilde{\mathbf{\Phi}}_i$. For $M = N$, this reduces to the Wronskian inverse

$$\mathbf{\Phi}_i = -\mathbf{W}_N(t_i)^{-1}\mathbf{d}_{N,i}. \tag{23}$$

For $M > N$, the unique least-squares solution is $-\mathbf{W}_N(t_i)^\dagger \mathbf{d}_{N,i}$, where $\dagger$ denotes the Moore–Penrose pseudoinverse.

A.2 Identification of Affine Parameters

The previous argument identifies coefficient values at individual times. The model class in Equation 4 further ties these values across time. For coefficient j, stack the pointwise values at L distinct grid locations:

$$\begin{bmatrix} t_1 & 1 \\ \vdots & \vdots \\ t_L & 1 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} = \begin{bmatrix} \Phi_j(t_1) \\ \vdots \\ \Phi_j(t_L) \end{bmatrix}. \tag{24}$$

At least two distinct times make the left-hand design matrix full column rank. Consequently, the pair (a_j, b_j) is unique whenever the pointwise coefficient values are unique. This separation clarifies the purpose of the two data dimensions in DA-KAN: independent trajectories identify the coefficient value at a time, while distinct time points identify the global affine law.

A.3 Noise and Conditioning

Full rank is necessary but not sufficient for stable recovery. For a perturbed target in Equation 21, while holding the design matrix fixed, least-squares sensitivity satisfies

$$\|\widehat{\mathbf{\Phi}}_i - \mathbf{\Phi}_i\|_2 \le \|\mathbf{W}_N(t_i)^\dagger\|_2 \, \|\mathbf{e}_i\|_2, \tag{25}$$

where $\mathbf{e}_i$ is the perturbation of the highest-derivative target. Derivative-channel noise also perturbs $\mathbf{W}_N(t_i)$, so Equation 25 is only a local diagnostic. A nearly singular excitation matrix can amplify either source of error. The multi-trajectory design was therefore used to improve rank and conditioning, not as a blanket guarantee under noisy or misspecified dynamics.

B Experimental Protocol and Implementation Details

B.1 Benchmark Construction and Noise

The benchmark contained 20 scalar affine-coefficient ODEs: 10 second-order systems and 10 third-order systems. Each equation used its own time domain and a uniform grid of 200 points per trajectory. Second-order systems used two initial conditions, while third-order systems used three. The candidate-order set was $\{2, 3\}$, and the true order was withheld from the learner. The headline protocol used five base seeds, $\{42, 43, 44, 45, 46\}$, giving 100 recovery trials at each noise level.

Table 2: **Fixed DA-KAN settings and extractor audit.** The upper panel lists the shared experimental settings. The lower panel compares direct affine readout, the non-neural TLS-only diagnostic, and the complete fixed extractor on the strict headline benchmark. Direct readout and complete DA-KAN decode the same trained DA-KAN cells. TLS-only uses the same order cascade and canonicalization grid as the TLS candidate inside DA-KAN. The noise levels refer to the derivative-channel protocol. "Post-acceptance update" means that extraction did not further optimize the neural coefficient cells.

Setting	Value
Benchmark systems	20 (10 order-2, 10 order-3)
Grid points	200 per trajectory
Trajectories	2 for order 2; 3 for order 3
Candidate orders	$\{2, 3\}$
Base seeds	$\{42, 43, 44, 45, 46\}$
Trials per noise	100
Noise levels	$0\%, 1\%, 2\%, 3\%, 4\%, 5\%$
Noise type	Shared channelwise Gaussian
RBF bases	$G = 60$ fixed Gaussian centers
RBF width	$(t_{\max} - t_{\min})/(0.4G)$
Optimizer	Adam, learning rate 0.04
RBF ℓ_1 weight	$\lambda_{\text{rbf}} = 0.01$
Order-2 budget	800 steps; anneal after 600
Order-3 budget	1200 steps; anneal after 800
Soft-threshold step	$\eta = 0.01$
Affine acceptance	$\tau_{\text{aff}} = 0.02$
Extraction sources	Affine-path and TLS candidates
Rational denominators	$\{1, 2, 5, 10, 20\}$
Snap tolerance	0.08
Zero tolerance	0.05
Complexity weight	$\lambda_C = 3 \times 10^{-4}$
Post-acceptance update	None

Method	0%	1%	2%	3%	4%	5%
Direct affine readout	100/100	97/100	79/100	57/100	37/100	19/100
TLS-only diagnostic	**100/100**	**100/100**	**100/100**	**94/100**	**89/100**	**88/100**
Complete DA-KAN	**100/100**	**100/100**	**100/100**	**94/100**	**89/100**	**88/100**

All headline results used the derivative-channel protocol in Equation 20. We perturbed every available derivative channel independently with Gaussian noise scaled by that channel's standard deviation. For each system, seed, and noise level, the noisy derivative tensors were generated once and reused by all methods. The pipeline did not estimate derivatives from noisy raw trajectories and did not use spline differentiation, boundary truncation, or an RUV acceptance test. This distinction is important when comparing the present results with alternative noisy-observation protocols.

B.2 Fixed Training and Extraction Settings

Table 2 lists the settings shared across the reported DA-KAN runs. The optimization budget depended only on the candidate order, not on the individual equation. After cascade acceptance, the neural cells were frozen. The extractor then compared a snapped affine-path candidate with a snapped total-least-squares (TLS) projection. It selected between these two fixed candidates by Equation 18.

The affine-path candidate was initialized from the learned (a_j, b_j) values and snapped with residual-guided local updates. The TLS candidate was computed from the affine design system in Equation 15 and then snapped independently. The reported DA-KAN equation was the lower residual-plus-complexity candidate, rather than the result of an unrestricted expression search. This implementation mapping explains the workflow in Figure 2. Multi-trajectory data constrained the shared residual.

Algorithm 1 DA-KAN Discovery

Require: Derivative data $\mathcal{D}$, orders $\mathcal{N}$, thresholds
Ensure: Symbolic affine-coefficient ODE or unresolved output
for orders N in increasing order in $\mathcal{N}$ **do**
  Build N cells using Eq. (7).
  Train with loss $\mathcal{L}_N$ in Eq. (11).
  Anneal the RBF weights by soft-thresholding the coefficients c_{jg}.
  Compute the pure affine validation score $E_{\mathrm{aff}}(N)$.
  if $E_{\mathrm{aff}}(N) \le \tau_{\mathrm{aff}}$ or $N = \max \mathcal{N}$ **then**
    Set $\widehat{N} \leftarrow N$ and keep the affine paths.
    Stop the cascade.
  end if
end for
Read direct affine coefficients α^{dir} from the accepted cells.
Form $Z_{\widehat{N}}\alpha + d_{\widehat{N}} \approx 0$.
Compute TLS affine projection α^{tls}.
Snap both candidate sets to rational grids.
Use denominators $\{1, 2, 5, 10, 20\}$.
Select $\widehat{\alpha}$ by Eq. (18).
return recovered ODE if the residual check passes.
Otherwise return unresolved.

Table 3: **Internal ablations at 3% noise.** The first block uses the strict five-seed protocol and isolates extraction diagnostics. TLS-only uses the same order cascade and canonicalization grid as the TLS candidate inside DA-KAN, but it does not use learned neural coefficient cells. The second block uses an independent 20-system 3% batch to isolate architectural and validation components. Runtime is reported only where the variant had a comparable discovery pipeline. Bold marks the best exact recovery count within each block.

Variant	Source	Exact	Sec.
Complete DA-KAN	strict protocol	**94/100**	2.83
TLS-only projection	strict diagnostic	**94/100**	5.4×10^{-4}
Direct readout	strict diagnostic	57/100	–
Complete DA-KAN	indep. ablation	**17/20**	4.88
Single trajectory	indep. ablation	7/20	2.13
No affine decoupling	indep. ablation	0/20	–
No order cascade	indep. ablation	8/20	1.24
No symbolic snapping	indep. ablation	0/20	–

The RBF path was annealed during optimization. Pure-affine validation selected an order. Frozen affine candidates were converted to the final equation.

C Detailed Recovery Audit

C.1 Fixed Extractor Across Noise Levels

The lower panel of Table 2 expands the adaptive-extraction ablation in Table 3 over the full headline noise sweep. The DA-KAN rows used the same trained affine cells across the same five-seed, 20-system protocol. The direct-readout row used the affine branch without the adaptive TLS comparison. The complete row used the fixed two-candidate extractor described above. The TLS-only diagnostic used the same cascade threshold and canonicalization grid without the neural coefficient cells.

C.2 Mechanism Ablations and Runtime

Table 3 and Figure 6(a) identify the mechanisms behind the main result. Without adaptive affine extraction, direct readout from the trained affine branch dropped from 94/100 to 57/100 at 3% noise under the strict five-seed protocol. Across the full noise sweep, direct readout recovered 100/100, 97/100, 79/100, 57/100, 37/100, and 19/100 systems. This gap showed that the extractor was not cosmetic post-processing. Residual-checked affine projection corrected small training bias. Without correction, that bias appeared as incorrect coefficients or extra terms. The TLS-only diagnostic matched complete DA-KAN under the strict protocol. We therefore interpreted the headline result as evidence for the complete restricted affine discovery protocol rather than for direct neural readout alone. Using only a single trajectory reduced exact recovery from 17/20 to 7/20 on the independent 3% batch. This result confirmed the identifiability role of multi-trajectory excitation. This result was consistent with the rank-based argument in Equation 23. Insufficient excitation left several coefficient combinations compatible with a similar residual. Removing affine decoupling eliminated the explicit symbol-bearing path and yielded no exact recoveries in this ablation. Removing cascade validation caused wrong-order acceptances and reduced recovery to 8/20. Removing symbolic snapping also yielded no exact recoveries. This result showed that final coefficient discretization and residual checking were essential for converting a trained affine scaffold into a symbolic ODE.

Runtime and PySR Search-Budget Sensitivity. Figure 6(b) reports wall-clock runtime per system at 3% noise. The complete DA-KAN pipeline took about 2.83 seconds per system in the strict 3% run. PySR and pykan took about 29.3 and 39.5 seconds per system, respectively, at 3% noise. E-SINDy, WSINDy, SINDy, and TLS-only were faster. They took about 4.05×10^{-2}, 2.64×10^{-2}, 1.13×10^{-3}, and 5.37×10^{-4} seconds per system, respectively. Their exact recovery, except for the internal TLS-only diagnostic, was lower on the noisy affine-coefficient benchmark. The PySR budget sweep in Figure 5(d) further shows that search performance was not monotone in timeout on this benchmark. At 3% noise, exact recovery ranged from 15/20 to 18/20 across the tested budgets. This result did not weaken PySR as a baseline. It clarified that DA-KAN and PySR obtained interpretability through different mechanisms: affine coefficient recovery with fixed extraction and expression-tree search, respectively.

C.3 Fitability versus Symbolifiability

Low trajectory error did not imply exact symbolic recovery. At 1% noise, SINDy produced a mean trajectory MSE of 6.83×10^{-5} but recovered 0/100 exact symbolic equations. Small coefficient deviations or spurious terms invalidated the recovered law. For exact DA-KAN recoveries, the accepted equations had near-machine-precision integration error because the symbolic equations themselves were recovered. At 3–5% noise, we counted the remaining DA-KAN failures strictly, even when their numerical errors were moderate. We applied this strict metric intentionally. A low trajectory error over the evaluated horizon was diagnostic evidence, not proof that the recovered equation had the correct symbolic structure.

The evaluation also separated order selection from coefficient recovery. DA-KAN selected the correct order for all 600 strict-protocol trials across all tested noise levels. This result supported cascade order validation in the non-oracle setting. In this panel, "order" denotes whether the selected differential order matched the ground truth. "Exact" additionally required the snapped affine coefficients to match the ground-truth monic ODE. Thus, the DA-KAN failures at 3–5% noise were coefficient-extraction failures rather than wrong-order acceptances. PySR also had high order accuracy, but exact recovery additionally depended on the expression selected by the search procedure. The contrast was consistent with the design difference. DA-KAN reduced the final symbolic search to a low-dimensional affine coefficient problem, whereas PySR searched over a broader expression space.

C.4 Per-System Audit at 3% Noise

Table 4 gives the complete 3% audit for base seed 42, where all 20 recovered equations matched their respective ground-truth equations after fixed extraction. "Aff." denotes the residual-checked snapped candidate initialized from the learned affine path; "TLS" denotes the independently projected candidate. Across the full five-seed 3% protocol, DA-KAN recovered 94/100 equations exactly. The six failures all had the correct selected order, but their snapped coefficients contained small spurious

Table 4: **Per-system recovery audit at 3% noise.** Each displayed equation was both the ground truth and the final DA-KAN output for base seed 42 in the strict protocol.

Second-order systems				
ID	System	N	Source	$F^\star = \widehat{F}$
1	Harmonic	2	TLS	$y'' + 9y = 0$
2	Damped	2	Aff.	$y'' + 0.5y' + 4y = 0$
3	Airy	2	TLS	$y'' - ty = 0$
4	Hermite	2	TLS	$y'' + 2ty' + 2y = 0$
5	Var-Stiff	2	TLS	$y'' + 0.8y' + (2t + 5)y = 0$
6	Decaying	2	Aff.	$y'' + 1.5ty' + 3y = 0$
7	Aero-Drag	2	Aff.	$y'' + (1.2t + 0.5)y' + 2.5y = 0$
8	Pure Dissip.	2	TLS	$y'' + 2.5y' = 0$
9	Dual Fields	2	Aff.	$y'' + 0.5ty' - 1.2ty = 0$
10	Coupled Decay	2	Aff.	$y'' + 0.5y' + 2.5y = 0$
Third-order systems				
ID	System	N	Source	$F^\star = \widehat{F}$
11	Oscil. Jerk	3	Aff.	$y''' + 4y' = 0$
12	Damped Oscil.	3	Aff.	$y''' + 0.5y'' + 5y' = 0$
13	Airy Jerk	3	TLS	$y''' - 1.5ty = 0$
14	Var-Damp	3	TLS	$y''' + 0.5ty'' + 2y = 0$
15	Affine Stiff	3	Aff.	$y''' + 1.2y' + (2t + 1)y = 0$
16	Affine Oscil.	3	Aff.	$y''' + (1.5t + 5)y' = 0$
17	Stable Mixed	3	TLS	$y''' + 0.8y'' + 4y' + 2y = 0$
18	Shifted Var.	3	Aff.	$y''' + 0.5y'' + (t + 3)y' = 0$
19	Dual Fields	3	TLS	$y''' + 0.5ty'' - 1.2ty' + 1.5y = 0$
20	Coupled Third	3	Aff.	$y''' + 0.4y'' + 6y' + 1.5y = 0$

terms or coefficient shifts. The failures occurred twice for Pure Dissip. and once each for Airy, Hermite, Airy Jerk, and Var-Damp.

D Scope and Boundary Discussion

DA-KAN was designed for scalar ODEs whose coefficients were constant or affine functions of the independent variable. The validation and extraction gates were intended to return an unresolved residual in three cases. The cases were degenerate derivative excitation, a true order outside the candidate order set, and a governing equation with non-affine coefficients or nonlinear state interactions. Nonlinear ODE systems, biological network dynamics, and PDE settings therefore tested the boundary of the present architecture rather than the same in-family recovery problem.

We used related public benchmarks to identify compatibility rather than to claim broad ODE/PDE discovery. MDBench (Bideh et al., 2026) contained a small compatible affine-ODE subset. It provided an external compatibility check but was too small to support a broad benchmark claim. Strogatz nonlinear ODEs (Strogatz, 2015) required nonlinear state functions such as $\sin(y)$ and y^2y'. BioPreDyn-Bench systems (Villaverde et al., 2015) required multivariate biological dynamics. PDEBench-style tasks (Takamoto et al., 2022) required spatial derivatives and PDE-specific residuals. These settings were natural extension targets, but they were outside the present hypothesis class. The supported conclusion was therefore deliberately scoped. Within affine-coefficient ODE discovery, exact recovery improved when an explicit affine path carried the symbolic law and the ODE residual validated it.

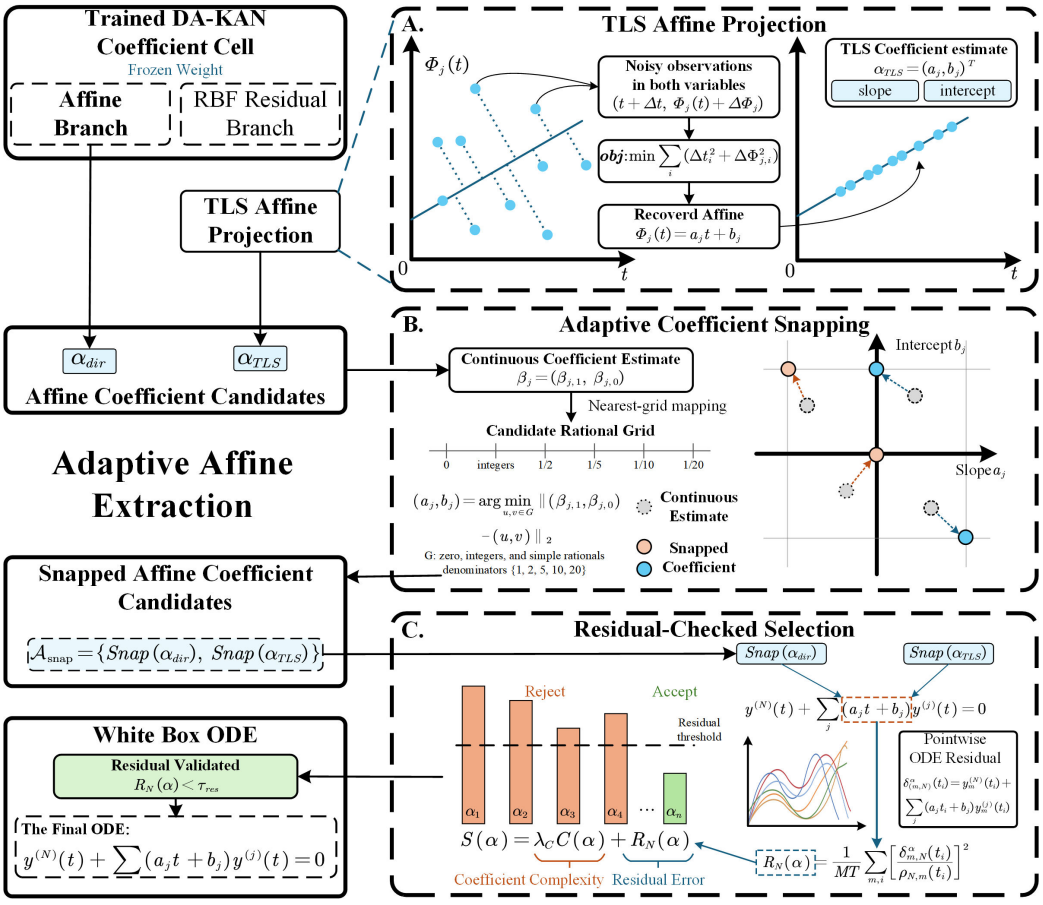


Figure 4: **Fixed adaptive affine extraction.** After cascade order validation, DA-KAN froze the trained coefficient cells and extracted the final symbolic law without further neural adaptation. The extractor compared direct affine readout with a total-least-squares affine projection. It snapped candidate coefficients to a rational grid. It selected the final ODE by a residual-plus-complexity score.

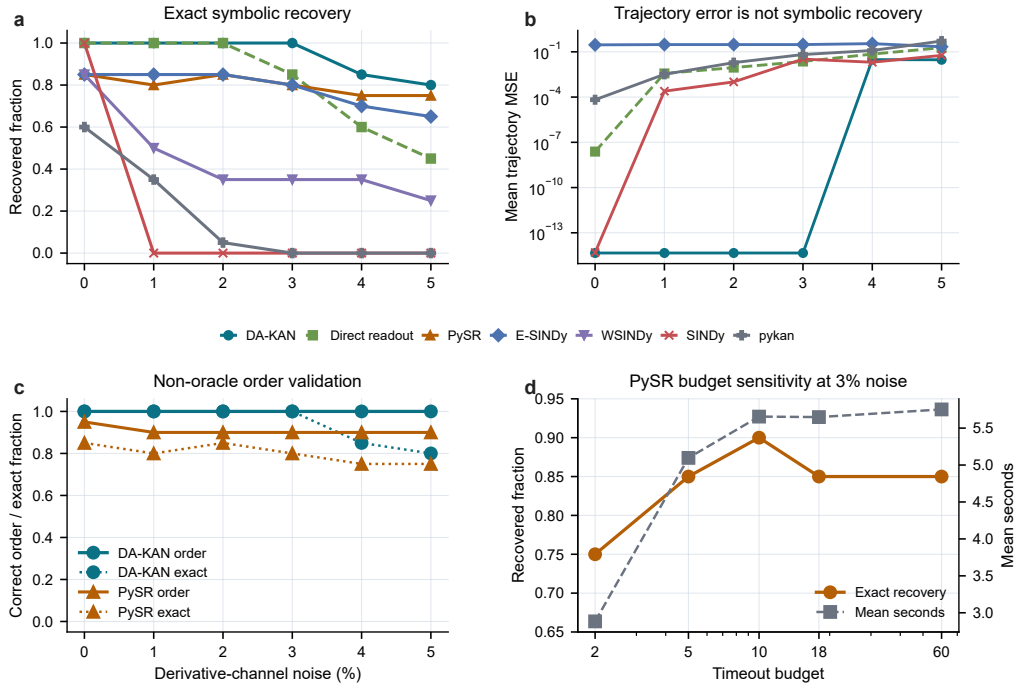


Figure 5: **Main experimental behavior under derivative-channel noise.** (a) Exact symbolic recovery shows that DA-KAN preserved full recovery through 2% noise and retained high recovery at 3–5% noise under the five-seed strict protocol. (b) Trajectory MSE alone is not a sufficient discovery metric. Several models obtained low prediction error while failing exact equation recovery. (c) The cascade validated the correct differential order for all DA-KAN trials. Here, "order" denotes correct order selection. "Exact" additionally requires coefficient recovery after snapping. PySR order and expression recovery varied with noise in the main fixed-budget run. (d) PySR recovery at 3% noise was sensitive to the expression-search budget on a separate 20-system sweep. This motivated a separate budget report rather than a single search setting.

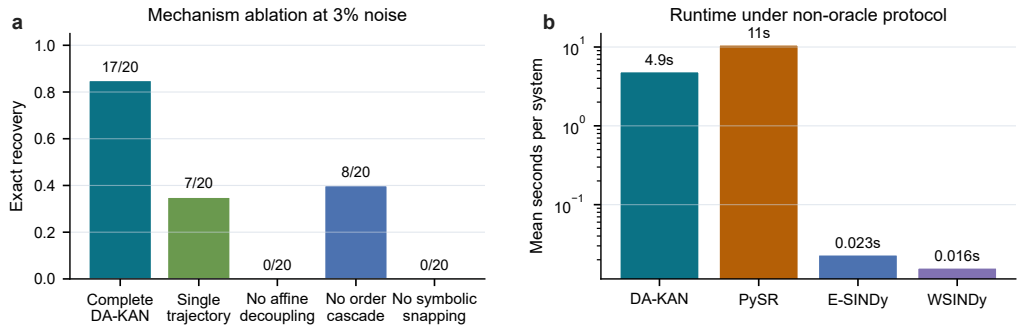


Figure 6: **Ablation and runtime evidence.** (a) Removing multi-trajectory excitation, affine decoupling, cascade validation, or symbolic snapping reduced exact recovery on the independent 3% ablation batch. (b) DA-KAN had moderate per-system runtime under the strict 3% protocol. Sparse baselines and TLS projection were faster but less reliable for exact recovery, except for the internal TLS-only diagnostic. PySR and pykan were slower in this run.