PDEDISCOVERYBENCH: EVALUATING DIFFERENTIAL EQUATION DISCOVERY FROM SPATIOTEMPORAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Discovering a partial differential equation (PDE) from observed fields requires choosing which terms may enter the law. Existing methods address this task, but published results rarely report how much prior information a method received, and to our knowledge no benchmark controls that information across method types. We introduce PDEDiscoveryBench, 337 ODE and PDE systems in one machine-readable format: 332 with reference equations and five unlabeled blood-flow fields simulated with computational fluid dynamics (CFD). Each labeled system is posed at three prior levels, from generic assumptions (L1) to equation-informed guidance (L3), written as natural-language prior cards that each method receives in its native form, and at four noise tiers. We evaluate seven sparse-regression, symbolic-search and large language model (LLM)-based methods by exact structural recovery and held-out normalized mean squared error (NMSE). Exact recovery is higher at L3 than at L1 for every method at every noise tier. Four case studies examine the prior from different sides: the data did not override a wrong prior card; LLM-PDESR recovered a KdV equation exactly only under its published, more specific specification; telling two LLM agents the noise model raised their exact recovery; and LLM agents given no physical description returned relations containing the Stokes balance for all five CFD fields. PDEDiscoveryBench makes prior information an explicit, controlled variable, so that new classical and LLM-based methods can be compared under the same priors and noise; we release its data, prior cards and scoring code. Our results suggest that discovery claims should state the prior behind them.

1 INTRODUCTION

Partial differential equations (PDEs) are central to modeling physical systems that vary over space and time, describing interactions among transport, diffusion, reactions, and other processes (Karniadakis et al., 2021). Their applications range from numerical weather prediction (Bauer et al., 2015) to cardiovascular flow modeling from medical measurements (Kissas et al., 2020). *PDE discovery* is the task of identifying the differential operators and functional interactions that govern observed data, together with their coefficients (Rudy et al., 2017; Kacprzyk et al., 2023). It inverts the *forward problem*, which computes a solution field from a given PDE. PDE discovery has played a pivotal role in the history of science, yielding explicit equations whose mechanisms can be inspected, compared with physical principles, and built upon for further scientific discovery.

Existing PDE-discovery methods span sparse regression (Brunton et al., 2016; Rudy et al., 2017), weak and variational formulations (Messenger & Bortz, 2021; Kacprzyk et al., 2023), evolutionary and reinforcement-learning search (Chen et al., 2022; Du et al., 2024c), and scientific foundation models (Xu et al., 2025). More recently, large language models (LLMs) have entered equation discovery (Du et al., 2024b; Shojaee et al., 2025a; Xia et al., 2026), including PDE-specific methods such as KeplerAgent, LLM4PD, HER-PDE and LLM-PDESR (Yang et al., 2026; Luo et al., 2025; Huang et al., 2026; Du et al., 2026).

Several methods (Shojaee et al., 2025a; Xia et al., 2026; Yang et al., 2026) reduce differential equation discovery to symbolic regression, which searches for an expression relating precomputed input derivatives to a precomputed target derivative. Such derivatives are often clean, and choosing which ones to supply already reveals which terms may enter the equation. PDE discovery from observed

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.

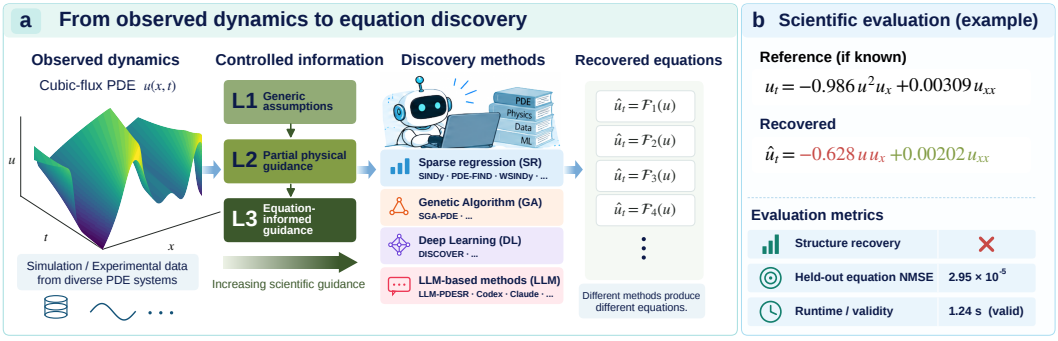


Figure 1: Overview of PDEDiscoveryBench. (a) A discovery method receives observed fields under one of three prior levels (L1–L3) and returns a candidate equation. (b) For a labeled system, the candidate is compared with the reference equation by structural recovery and held-out NMSE, and its runtime and validity are recorded (Appendix C.2); the recovered equation and scores shown are illustrative.

fields is harder. An end-to-end method must first decide which spatial and temporal derivatives may enter the equation, then estimate them from noisy, discretely sampled fields or reformulate the problem to avoid pointwise differentiation.

Public benchmark datasets enable systematic evaluation of several PDE-related modeling tasks. For the forward problem, PDEArena, PDEBench, and The Well provide infrastructure for learning from simulated physical fields, with an emphasis on surrogate modeling and prediction (Gupta & Brandstetter, 2023; Takamoto et al., 2022; Ohana et al., 2024). On the other hand, PDE discovery has less benchmark infrastructure. MDBench evaluates ODE and PDE discovery across several algorithmic families through an interface built on trajectories, coordinates and time derivatives (Bideh et al., 2026). PhysPDE studies the selection of physical hypotheses within supplied theories (Feng et al., 2025). PDE-discovery benchmarks have been especially slow to accommodate LLM-based methods, and benchmarks built for LLMs do not pose PDE discovery from raw fields. LLM-SRBench, for example, frames LLM-based discovery as symbolic regression over supplied inputs and targets (Shojaee et al., 2025b).

Evaluating PDE discovery is itself difficult, because no single metric suffices. Exact structural recovery compares the discovered terms with a reference equation, so it cannot score systems that lack one. Prediction error, such as the normalized mean squared error (NMSE) between the predicted right-hand side and the held-out temporal derivative, needs no reference but can reward overcomplete models, because spurious terms absorb residual error. A benchmark should therefore report both metrics and show where they disagree.

Discovery methods also depend on prior information, and published evaluations rarely report or control how much of it a method receives. A sparse regression-based method requires a library of candidate terms that is fixed in advance (Brunton et al., 2016; Rudy et al., 2017). Expression-tree methods (Chen et al., 2022; Du et al., 2024c) instead fix the operators from which terms are built, and that choice affects the search as well as what it can represent. LLM-based methods can also receive prior information directly as text, in whatever amount a prompt supplies. A recovered equation is then hard to read as evidence of capability unless the disclosed prior is reported with it. PhysGym shows this dependence in interactive physics discovery: removing problem context, variable descriptions and meaningful names lowers aggregate success (Chen et al., 2025). For LLM-based methods, a benchmark should assess how numerical evidence and disclosed scientific context jointly support equation recovery. To our knowledge, none of the existing benchmarks compares classical and LLM-based methods across controlled levels of disclosed prior information for field-input PDE discovery.

To address these gaps, we introduce **PDEDiscoveryBench** (Figure 1), a dataset and evaluation protocol for field-input PDE discovery with classical and LLM-based methods.

Our contributions are as follows.

- **An LLM-native dataset for PDE discovery.** PDEDiscoveryBench stores 337 ODE and PDE systems in one format, each with a machine-readable observation schema that bridges LLM agents and classical methods. Each labeled system is posed at three prior levels: L1 (generic assumptions), L2 (partial physical) and L3 (equation-informed). At L2 and L3, a natural-language *prior card* states what the governing law may contain; the two coagulation systems have no L2 card. The five unlabeled systems, blood-flow fields simulated with computational fluid dynamics (CFD), are posed at L1 only, so they test discovery with minimal prior information.

- **A protocol that controls the scientific prior across noise levels.** We cross the three prior levels with four noise tiers, translating each level into the native input of seven methods: SINDy, WSINDy, SGA-PDE, DISCOVER, LLM-PDESR (clean data only), GPT-5.6-sol and Claude Opus 5. We score each outcome by exact structural recovery and held-out NMSE.

- **A controlled evaluation of seven classical and LLM-based methods.** Exact recovery is higher at L3 than at L1 at every noise tier for all six methods run under noise, so a reported recovery rate should be read together with the prior that produced it.

2 Related work

Symbolic regression. Symbolic regression searches for a closed-form expression that relates input variables to a target. The classical approach is evolutionary search over expression trees, and a representative implementation is PySR (Cranmer, 2023). Recent advances include neural-guided search (Petersen et al., 2021) and LLM-guided approaches, in which a language model proposes equation structures and revises them using numerical feedback (Shojaee et al., 2025a; Xia et al., 2026). Applied to PDE discovery, symbolic regression takes precomputed derivatives as inputs and so skips the derivative selection and estimation that end-to-end discovery requires (Section 1). In PDEDiscoveryBench, no method receives precomputed derivatives; each must compute or avoid them within its own pipeline.

PDE discovery methods. We group PDE-discovery methods by computational framework, as in Section 4.1. **Sparse regression** selects a sparse combination of terms from a fixed candidate library, as in SINDy, PDE-FIND and the weak-form WSINDy (Brunton et al., 2016; Rudy et al., 2017; Messenger & Bortz, 2021), all implemented in PySINDy (Brian et al., 2023). **Genetic algorithm** methods such as SGA-PDE evolve populations of expressions (Chen et al., 2022). **Deep-learning** methods include DISCOVER and R-DISCOVER, which sample expression trees from a reinforcement-learning policy (Du et al., 2024c;a), and EqGPT, a generative model trained on known PDEs (Xu et al., 2025). **LLM-based** methods use a language model to propose or select terms, configure tools or interpret field diagnostics, as in LLM4ED (Du et al., 2024b), LLM4PD (Luo et al., 2025), KeplerAgent (Yang et al., 2026) and LLM-PDESR (Du et al., 2026), among others (Ivanchik et al., 2025; Qu et al., 2025; Huang et al., 2026; Pekhale et al., 2026; Yang & Thomson, 2026).

Discovery benchmarks and prior information. SRBench and LLM-SRBench evaluate symbolic regression over supplied inputs, including precomputed derivatives (La Cava et al., 2021b; Shojaee et al., 2025b), and LLM-SRBench also gives LLMs the physical context. MDBench evaluates ODE and PDE discovery across algorithmic families through an interface of trajectories, coordinates and time derivatives (Bideh et al., 2026), and PhysPDE asks methods to select hypotheses and discover PDEs within supplied physical theories (Feng et al., 2025). In these benchmarks, the prior information a method receives is fixed by the task. PhysGym instead varies it, progressively removing problem context, variable descriptions and meaningful names in interactive physics discovery; aggregate success falls, although individual tasks do not always follow this trend (Chen et al., 2025). PDEDiscoveryBench treats the prior as an experimental factor for field-input PDE discovery, translating each level into every method's native input, from candidate libraries to prompt text (Section 3.3), so that a reported recovery rate reflects both the method and its prior.

Table 1: PDEDiscoveryBench systems by family. Systems 0001–0332 (blue) have reference equations: 12 ODEs and 320 time-dependent PDEs. The five stationary CFD blood-flow fields 0333–0337 (green) have no documented reference equation and serve as case studies. Count treats a multi-target system as one system. Data shape is the stored observation-array shape, with space before time and channels last; the five CFD grid sizes are given in Appendix A.1. The last column lists target channels before the semicolon and supplied auxiliaries after it; auxiliaries are inputs, not equations to recover.

Family	System IDs	Count	Source	Data shape	Targets; auxiliaries
Reference equation available					
8D Lorenz–96 ODE	0001	1	Authors	2000×8	8 states; none
60D Lorenz–96 ODE	0002	1	Authors	12000×60	60 states; none
Heat	0003–0027	25	PROSE-PDE	$256 \times 256 \times 1$	u; none
Viscous conservation laws	0028–0102	75	PROSE-PDE	$256 \times 256 \times 1$	u; none
Inviscid conservation laws	0103–0137	35	PROSE-PDE	$256 \times 256 \times 1$	u; none
1D reaction–diffusion	0138–0162, 0183–0232, 0242–0266	100	PROSE-PDE	$256 \times 256 \times 1$	u; none
Fokker–Planck	0163–0182	20	PROSE-PDE	$512 \times 512 \times 1$	u; none
Cahn–Hilliard	0233–0241	9	PROSE-PDE	$256 \times 256 \times 1$	u; none
Wave (u_{tt})	0267–0291	25	PROSE-PDE	$256 \times 256 \times 1$	u; none
Porous medium	0292–0316	25	PROSE-PDE	$256 \times 256 \times 1$	u; none
Advection	0317	1	MDBench	$1024 \times 201 \times 1$	u; none
Viscous Burgers	0318	1	MDBench	$256 \times 101 \times 1$	u; none
KdV	0319	1	MDBench	$512 \times 201 \times 1$	u; none
KS	0320	1	MDBench	$1024 \times 251 \times 1$	u; none
2D Navier–Stokes	0321	1	MDBench	$250 \times 50 \times 500 \times 3$	(u, v); p
2D reaction–diffusion	0322	1	MDBench	$250 \times 50 \times 500 \times 9$	(c_0, c_1, c_2); (v_x, v_y, p), (f_0, f_1, f_2)
Glider ODE	0323–0326	4	ODE-Strogatz	100×2	(v, θ); none
Predator–prey ODE	0327–0330	4	ODE-Strogatz	100×2	(x, y); none
8D blood coagulation ODE	0331	1	Authors	1869×12	8 targets; 4 intermediates
25D blood coagulation ODE	0332	1	Authors	5000×41	25 targets; 16 intermediates
No documented reference equation					
Stationary CFD blood-flow fields	0333–0337	5	Authors	$N_x \times N_y \times 3$	(u, v); p
Total		337			

3 PDEDISCOVERYBENCH

PDEDiscoveryBench comprises 337 systems: 332 with reference equations and five without (Table 1). It spans ODEs, time-dependent PDEs and stationary spatial fields. A multi-field system counts once, with a separate equation sought for each target field. The five unlabeled CFD fields serve as exploratory case studies rather than reference-based scoring tasks.

3.1 DATASET DESCRIPTION

The labeled collection combines trajectories generated with PROSE-PDE's data generation functionality (Sun et al., 2025), PDE cases adapted from MDBench (Bideh et al., 2026), and ODE systems from Lorenz–96 (Lorenz, 1996), ODE-Strogatz (La Cava et al., 2021a) and coagulation models (Zarnitsina et al., 1996; Anand et al., 2008). These sources mix generated trajectories with adapted benchmark observations; systems within a family may share a symbolic form while differing in coefficients and initial states. Table 1 gives the coverage, and Appendix A explains selection and preparation.

Table 2: Prior levels in detail, with system 0078 as a concrete example; its hidden reference is $u_t = -0.986u^2u_x + 0.00309u_{xx}$ (Figure 1). Cards are abridged; Table 4 gives the full texts for all systems.

Level	Disclosed information	Card for system 0078
L1: Generic assumptions	Observed fields, coordinates, auxiliary roles and data layout; a method may add a finite generic search envelope.	No scientific disclosure beyond the anonymous observed fields, coordinates, auxiliary-channel roles and sampling layout.
L2: Partial physical guidance	L1 plus broad structural facts: whether the dynamics are a source-free balance law, whether local sources are permitted and whether coefficients may depend on position.	The field obeys a local source-free balance law with unknown flux. Constitutive coefficients have no explicit position dependence.
L3: Equation-informed guidance	L2 plus a narrower constitutive family, chosen with knowledge of the reference equation, that retains several alternatives.	Homogeneous viscous transport with an unknown polynomial flux of degree at most four and constant diffusivity. Active powers, signs and coefficients are unknown.

A discovery task supplies sampled fields and coordinates together with target and auxiliary roles and, for some geometries, a spatial mask. An auxiliary may enter a target law but is not a separate equation to recover. The 25 wave systems require u_{tt} rather than u_t. Appendix A specifies the stored shapes, channel roles and mask semantics.

We call the collection LLM-native in the operational sense that an agent can read observed arrays, their schema and a prior card directly, without a method-specific feature pipeline. No task supplies precomputed derivatives or a reference equation. For scored temporal tasks, fitting uses the first 70% of time samples and the final segment is held out. Other methods use the same observations through their native adapters; differences in prior access are detailed in Section 3.3.

3.2 Noise modeling

We report clean, 1%, 5% and 10% observation tiers. For every system with a clean baseline, the nominal k-percent perturbation is applied independently at each point and channel, including supplied auxiliaries, as $u^{(k)} = u + ((k/100)|u| + 10^{-10})\xi$, where u is the clean value and $\xi \sim \mathcal{N}(0, 1)$. The noisy CFD variants were provided with the fields and match these levels (residual standard deviations of 1.00, 5.00 and 10.01% of channel magnitude). Of the 332 labeled systems, 330 have all four tiers, as do the five CFD systems; coagulation system 0331 has only 5% observations, and 0332 has clean and 5%. Appendix A details tier availability, the treatment of missing observations in scoring and the temporal split.

3.3 Controlled prior information

Each labeled system is posed at three cumulative prior levels, L1–L3 (Table 2). A level is a natural-language prior card shared by all methods; each method translates the card into its own search space, or *hypothesis space*, such as a candidate library for SINDy or prompt text for an LLM agent (Appendix B.1). At every level, the active terms and all coefficients remain to be inferred from the observations. The physical roles of channels also count as guidance: L1 uses anonymous channel names, and a higher-level card may disclose them. For example, the cylinder-flow L2 card identifies momentum components and a pressure-like auxiliary without specifying their equations (Table 4).

For a fixed task the admissible hypothesis spaces are nested, $\mathcal{H}_3 \subseteq \mathcal{H}_2 \subseteq \mathcal{H}_1$. The levels measure disclosed information, not numerical quality: L1 is a bounded search space rather than an absence of assumptions, and L3 names a family but never its active terms. For system 0078, the L3 card admits flux powers one through four, whereas the reference uses only the cubic flux (Table 2). A conservation card constrains a complete flux, so a differentiated product keeps a single coefficient even when its expansion contains several terms. The curators wrote the cards with knowledge of the

catalog, so comparisons across levels are exploratory rather than blind. The five unlabeled systems receive L1 only.

3.4 Evaluation metrics

We score every labeled system with two metrics that answer different questions: whether the discovered equation reproduces held-out dynamics, and whether it has the reference structure.

Clean-reference NMSE. For each target field, let y_i be the reference temporal derivative (u_t, or u_{tt} for second-order targets), computed numerically from the clean observations at the m eligible points of the held-out final segment, and let $\hat{y}_i$ be the discovered right-hand side at the same points. Fitting may use noisy data, but scoring always uses the clean reference. After discarding the $\lfloor 0.03m \rfloor$ largest squared residuals, we compute

$$\mathrm{NMSE} = \frac{\sum_{i \in I} (y_i - \hat{y}_i)^2}{\sum_{i \in I} y_i^2 + 10^{-10}}, \tag{1}$$

where I indexes the remaining points. A multi-target system receives the median of its target scores. All methods share one population per noise tier: 331, 330, 331 and 330 systems at clean, 1%, 5% and 10% noise, since system 0331 has no clean reference and 0332 has only clean and 5% observations. An unsupported system, a failed run, a wrong temporal order or an unscoreable equation receives NMSE 1, the score of a zero prediction; finite values above 1 are not clipped. The target, populations and failure rule are common to all methods, but differentiation stencils and feature evaluation remain method-specific (Appendix B). Because y is itself a numerical estimate, even the reference equation need not reach zero NMSE.

Exact structural recovery. A labeled system counts as recovered when, for every target, the discovered equation has the reference temporal order and the same nonzero terms as the reference after both are expanded symbolically into canonical atomic terms; coefficients are not scored. The recovery rate divides by all 332 labeled systems at every noise tier: unsupported systems, failed runs and missing tiers count as non-recoveries, so declining a case never raises a rate. SGA-PDE and LLM-PDESR support only a first-order temporal target for one field in one spatial dimension, which excludes the 12 ODE systems, the two 2D systems and the 25 wave systems; interface coverage alone thus caps their recovery at $293/332 = 88.3\%$. For SGA-PDE, which runs three seeds per system, NMSE is the median and recovery the mean over seeds.

4 Experiments and results

4.1 Discovery methods

We benchmark seven representative methods from the four groups of Section 2: SINDy (Brunton et al., 2016; Rudy et al., 2017) and WSINDy (Messenger & Bortz, 2021) (sparse regression), SGA-PDE (Chen et al., 2022) (genetic algorithm), DISCOVER (Du et al., 2024c) (deep learning), and LLM-PDESR (Du et al., 2026), GPT-5.6-sol and Claude Opus 5 (LLM-based; we run the latter two as tool-using *agents*). Each selected method (i) supports PDE discovery, (ii) requires no precomputed derivatives and (iii) requires no pretraining on additional data. Methods such as ODEFormer (d'Ascoli et al., 2024), LLM-SR (Shojaee et al., 2025a) and SR-Scientist (Xia et al., 2026) do not meet these criteria without substantial adaptation. Appendix B describes each method's implementation and settings.

4.2 Experimental setup

All methods fit the same training prefix and are scored on the same held-out segment (Section 3.4); held-out data never select a policy, a candidate or a final answer. Each prior card becomes a method's native input: a candidate library for SINDy and WSINDy, an expression grammar for SGA-PDE and DISCOVER, and prompt text for LLM-PDESR and the two agents (Appendix B.1). Three asymmetries matter for reading the results. First, SINDy's and WSINDy's hyperparameters are tuned by grid search on the training prefix, per family, noise tier and prior level, with noisy-tier policies validated on clean inner-validation data (Appendices B.3 and B.5), and both methods receive the reference term count K_{GT} as a cap on the fitted support, so they run as oracle-assisted variants; the

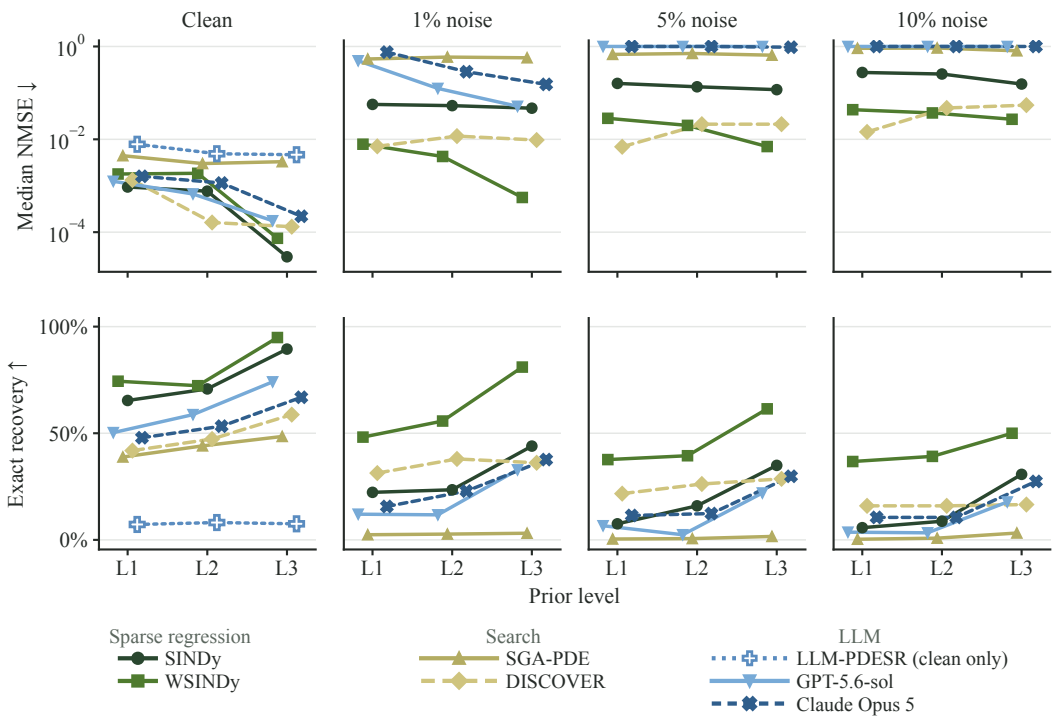


Figure 2: Main results by noise tier. Each column is one noise tier, and the horizontal axis is the prior level. **Top:** median held-out clean-reference NMSE (lower is better; failures score 1). **Bottom:** exact structural recovery over all 332 labeled systems (higher is better). Metrics and populations follow Section 3.4; LLM-PDESR has clean-data results only. Appendix C shows the same values with the noise tier on the horizontal axis.

other methods use fixed configurations, and the agent prompts state no noise level. The same tuning procedure is applied independently at every level and never uses held-out data or the identity of the reference terms, so differences in the selected hyperparameters across levels come from running the same validation on different hypothesis spaces; differences across methods are not controlled prior effects. Second, the agents write their own derivative estimators and infer the temporal order, whereas the other methods use the estimators built into their pipelines. Third, each agent reports its first accepted answer under retries, so its rates are cumulative over attempts with agent-specific budgets (Appendices B.6 and B.7); SGA-PDE averages three seeds, and DISCOVER and LLM-PDESR run once per case. Budgets are method-native, not equalized; Appendix B gives the full protocol and Appendix C.2 the recorded compute costs.

4.3 Main results

Figure 2 summarizes the seven methods across the four noise tiers and three prior levels; Figure 5 (Appendix C) plots the same values with the noise tier on the horizontal axis, which shows the effect of noise at each prior level. Each method computes or avoids derivatives within its own pipeline: SINDy and WSINDy receive the declared temporal target order, while the agents must infer it.

The top row reports the clean-reference NMSE of each final equation (Section 3.4). Within the common populations, scores of one assigned to unsupported, failed or unscoreable outcomes affect the medians; a median of one does not mean that most runs failed, because finite scores can also lie near or above one. For example, Claude Opus 5 at L3 and 10% noise has finite scores for 299 of 330 systems but a median NMSE of one. The bottom row reports exact structural recovery over all 332 labeled systems (Section 3.4); seeds, retries and the K_{GT} cap follow Section 4.2.

Prior level. Exact recovery rises with the prior level almost everywhere: L3 exceeds L1 for every method at every noise tier, also with the 19 families weighted equally (Table 9), and L1 $\leq$ L2 $\leq$ L3 holds in 19 of 25 method–tier cells. Most of this gain comes from the equation-informed card: under noise, L1 to L2 changes exact recovery by at most 8.5 points for any method, whereas L2 to L3 adds 10.8–25.3 points for SINDy, WSINDy and both agents. Median NMSE usually follows, falling from L1 to L3 in 18 of 25 cells; the exceptions are DISCOVER under noise, SGA-PDE at 1% noise, and agent medians that saturate at one. The prior is not the only factor: at each clean level the six methods still differ by at least 28.2 points in exact recovery, more than the largest clean L1-to-L3 gain (SINDy's 24.1 points), because they also differ in toolboxes and in priors the levels do not vary, such as K_{GT} and noise-conditioned tuning.

Noise and derivative estimation. Noise costs every method more than the prior adds: from clean data to 10% noise, exact recovery falls by 26–62 points, more than each method's largest L1-to-L3 gain. The two methods that avoid pointwise numerical differentiation hold up best. WSINDy, whose weak form integrates the data against test functions, has the highest exact recovery in every cell of the figure, and DISCOVER, which obtains derivatives by automatic differentiation of a PINN fit, is second at L1 and L2; together they hold the two lowest median NMSEs in every noisy cell. Methods that differentiate noisy samples directly fall furthest: SGA-PDE recovers at most 3.2% of systems at any noisy tier, and SINDy drops from 65.4–89.5% on clean data to 5.7–30.7% at 10% noise despite noise-tuned smoothing. The prior still helps when the derivative estimate is usable: at 10% noise, L3 adds 25.0 points over L1 for SINDy and 13.3–16.9 for WSINDy and the two agents, but 2.9 for SGA-PDE.

Agents, fit and structure. On clean data, GPT-5.6-sol ranks third in exact recovery at every level, behind the two sparse methods that receive K_{GT}; under noise, both agents lose most of their L1 accuracy at 1% noise (GPT-5.6-sol from 50.3% to 12.0%, Claude Opus 5 from 47.9% to 15.7%). At 5% and 10% noise the agents' median NMSE is at or near one at every level, although Claude Opus 5 still recovers 27.4–29.8% of systems exactly at L3. Fit and structure also rank methods differently: on clean data, DISCOVER has the lowest L2 median NMSE but recovers exact structure less often than SINDy, WSINDy, GPT-5.6-sol or Claude Opus 5, whereas WSINDy has the highest clean L1 exact recovery but only the fifth-lowest L1 median NMSE. A clean-data ablation of the K_{GT} cap on the 318 scalar PDEs shows how NMSE can mislead (Table 10): retuned without the cap, SINDy and WSINDy reach a lower median held-out NMSE at every level, yet exact recovery falls from 68–97% to 6–43%, and 97–171 systems per method and level that the capped fit recovers exactly get a lower NMSE with the wrong terms. Structure recovery thus needs an explicit sparsity constraint, and the sparse methods' clean lead depends on this oracle: without it, both fall below the agents, DISCOVER and SGA-PDE on these systems at every level.

4.4 Additional findings and case studies

Four findings in Figure 2 raise questions that we test with targeted case studies; Appendix C.1 gives the full records.

Wrong priors. Most of the prior's gain comes from the L3 card, so we asked whether the data can correct a wrong one. On systems 0079 and 0139, we replaced the L3 card with one for a different family that excludes true terms. No outcome recovered an excluded term, and none was exact: 0 of 12 agent outcomes and 0 of 4 SINDy and WSINDy fits, against 4 of 4 correct-card controls in each group (Table 11). For sparse regression this is expected, since a library without the true term cannot fit it; the agents, however, could write any equation and still followed the card. In this case study, a wrong prompt constrained the agents almost as firmly as a wrong library constrained sparse regression.

Published priors. LLM-PDESR stays at 7.2–8.1% exact recovery at every level, although the share of its equations that contain every reference term rises from 35% to 80% of the systems it supports. Its proposals also stall: the median search stops after 65 of 500 allowed samples, and the median run finds its best candidate at the third sample. Yet its original evaluation recovers a KdV equation with the same structure as our system 0319 (Du et al., 2026). That evaluation's specification is more concrete than our L3 card: its seed program already contains the dispersion term. Run on our 0319 data with our settings, it returned the exact structure in all three repeats, each at the first generated candidate, whereas our L3 card yielded all reference terms plus three extra ones (Table 12). The

specification also changes the parameter count, so this does not isolate the wording, but it shows that a published success can rest on a prior more specific than the one a benchmark discloses. LLM-based methods take the prior as free text, whose specificity no library bounds; in Figure 2 the agents' L1-to-L3 gains are comparable to those of the library-based sparse methods.

Noise as prior information. The agents lose the most under noise, and their prompts, unlike the sparse methods' tuning, state no noise level. We therefore told them. On the 20 Fokker–Planck systems at L3 and 5% noise, where SINDy recovers none and WSINDy all 20, a card stating only the noise model raised exact recovery from 6 to 11 of 20 for GPT-5.6-sol and from 2 to 10 for Claude Opus 5 (16 paired gains against 3 losses, $p = 0.004$), and both agents reported weak-form or integral estimators more often (Table 13). Noise information thus helps an agent through the estimator it chooses. The comparison uses historical controls, one family, one tier and one level, and GPT-5.6-sol's gain alone is not significant ($p = 0.18$).

Discovery with a minimal prior. These case studies show that the prior shapes much of what a method recovers. The five unlabeled CFD blood-flow fields, which have no documented governing equation, test the other end. Outside the benchmark protocol, Claude Code subagents running Claude Opus 5.5 at high effort received anonymized, rescaled channels with no physical description, told only that the fields are steady and two-dimensional, given a term budget and asked to consider nonlinear terms. For all five fields they returned relations containing the steady Stokes balance, and for the two with the highest reported Reynolds numbers, the steady Navier–Stokes momentum equations (Table 14). The convective term is small, 0.07–0.24% of the pressure gradient and largest in a Stokes field, so the data do not separate the two forms, and no continuity equation was returned (Appendix A.1). The case study suggests that an agent can recover physically consistent structure without a disclosed equation.

5 Conclusions

We introduced PDEDiscoveryBench, 337 ODE and PDE systems with natural-language prior cards, and a protocol that crosses three prior levels with four noise tiers for seven classical and LLM-based methods, scored by exact structural recovery and clean-reference NMSE. Results depend strongly on the scientific prior at every noise level: for all six methods run under noise, exact recovery is higher at L3 than at L1 at every tier, and most of the gain comes from the equation-informed card. Case studies sharpen this picture: the data did not override a wrong card, a published success rested on a specification more specific than our L3 card, and telling the agents the noise model raised their recovery through the estimators they chose. The prior is not the only factor: moving from clean data to 10% noise costs every method more than the prior adds, and the methods that avoid pointwise differentiation hold up best. Without a disclosed equation, agents still returned relations containing the Stokes balance for all five unlabeled CFD fields, which suggests a path toward discovery beyond reference equations. Evaluations of PDE discovery should report the prior each method receives and pair NMSE, which rewards extra terms, with structural recovery.

Limitations. All LLM-based methods use proprietary models; compute limits precluded an open-weight model. Pooled rates are weighted toward 1D diffusion–reaction and viscous conservation-law systems (175 of 332); with the 19 families weighted equally, L3 still exceeds L1 in all 25 method–tier cells, but family-bootstrap 95% intervals of the gain exclude zero in only 17 of 25 pooled cells (Table 9). The curators wrote the cards knowing the catalog, so comparisons across levels are exploratory. Methods receive unequal side information: SINDy and WSINDy get the reference term count and noise-conditioned tuning, and the agents' rates are cumulative over retries with unequal budgets. The case studies cover one to twenty systems each and show mechanisms, not rates; the CFD study ran outside the benchmark protocol on simulated fields. Many catalog equations are textbook PDEs, so anonymization cannot separate an LLM's recognition of an equation from inference. Exact recovery needs a reference equation and cannot score unlabeled systems.

AI USE STATEMENT

In this work, we used generative AI tools, the coding agents Codex and Claude Code, to implement the scoring, aggregation and analysis pipelines, to help design experiments and give feedback on the evaluation methodology, and to help interpret the results. We have not used them to generate synthetic data, clean or reformat the benchmark data, propose or refine hypotheses, or develop theoretical models or conceptual frameworks; mathematical claims, proofs, translation and qualitative or thematic data analysis are not applicable to this work. We also used them to create and edit code and figures, draft and edit parts of the manuscript, suggest its structure and format references. We have reviewed all AI-assisted work: pipeline outputs were checked against saved per-system outcomes, every plotted value traces to a hashed source table, AI-generated code was reviewed and tested by five authors, and every AI-assisted interpretation and text was checked by the authors against the underlying data. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics and involves no human subjects or personal data. The benchmark consists of simulated data: trajectories we generated with the PROSE-PDE generator, public MDBench and ODE-Strogatz cases, and ODE models we simulated; the five CFD fields were simulated on vessel geometries from a published mouse study (Meng et al., 2022), and we performed no animal experiments. Third-party data are used under their licenses, which the release lists per subset, and third-party method code is not redistributed. A discovered equation is a candidate model, not a verified law; because a low NMSE can accompany a wrong structure, we report exact structural recovery alongside it. Appendix C.2 reports the recorded compute and API usage.

REPRODUCIBILITY STATEMENT

The anonymous dataset is at `https://anonymous-hf.com/a/vex2ala7am9i/` and the evaluation code at `https://anonymous.4open.science/r/PDEDiscoveryBench`. The code checks the data against SHA-256 hashes, scores equations with the metrics of Section 3.4, reruns each method, and rebuilds the reported figures and tables from saved per-system outcomes, including the saved outputs of the proprietary LLMs. Appendices A–C give the dataset construction, method settings and prompts, and the plotted values.

REFERENCES

M. Anand, K. Rajagopal, and K.R. Rajagopal. A model for the formation, growth, and lysis of clots in quiescent plasma. a comparison between the effects of antithrombin iii deficiency and protein c deficiency. *Journal of Theoretical Biology*, 253(4):725–738, 2008. ISSN 0022-5193. doi: https://doi.org/10.1016/j.jtbi.2008.04.015. URL https://www.sciencedirect.com/science/article/pii/S002251930800180X.

Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525:47–55, 2015. doi: 10.1038/nature14956. URL https://www.nature.com/articles/nature14956.

Amirmohammad Ziaei Bideh, Aleksandra Georgievska, and Jonathan Gryak. MDBench: Benchmarking data-driven methods for model discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19746–19754, 2026. doi: 10.1609/aaai.v40i24.39056. URL https://ojs.aaai.org/index.php/AAAI/article/view/39056.

Brian, Markus Quade, Jacob Stevens-Haas, Zachary Nicolaou, Kathleen Champion, Alan Kaptanoglu, OliviaZ0826, Jared Callaham, Marat Kopytjuk, TarenGorman, Bill Tubbs, Thomasillo, Matteo Manzi, Shih-Chi Liao, Thomas O'Connor, and Tobias Uelwer. dynamicslab/pysindy: Small fixes to docs and scs version, April 2023. URL https://doi.org/10.5281/zenodo.7808834.

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 Na-*

tional Academy of Sciences, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113. URL https://doi.org/10.1073/pnas.1517384113.

Yimeng Chen, Piotr Piękos, Mateusz Ostaszewski, Firas Laakom, and Jürgen Schmidhuber. PhysGym: Benchmarking LLMs in interactive physics discovery with controlled priors. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1556. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/42abcf9dffa48ca44fea1c497539a914-Abstract-Datasets_and_Benchmarks_Track.html.

Yuntian Chen, Yingtao Luo, Qiang Liu, Hao Xu, and Dongxiao Zhang. Symbolic genetic algorithm for discovering open-form partial differential equations (sga-pde). *Phys. Rev. Res.*, 4:023174, Jun 2022. doi: 10.1103/PhysRevResearch.4.023174. URL https://link.aps.org/doi/10.1103/PhysRevResearch.4.023174.

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

Stéphane d'Ascoli, Sören Becker, Alexander Mathis, Philippe Schwaller, and Niki Kilbertus. ODEFormer: Symbolic regression of dynamical systems with transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TzoHLiGVMo.

Jinyang Du, Hao Ma, Xiaohu Shi, Bo Yang, Yanchun Liang, Heow Pueh Lee, and Chunguo Wu. LLM-PDESR: Robust PDE discovery via subdomain weighted residuals and LLM-guided symbolic hypothesis generation. *arXiv preprint arXiv:2607.10546*, 2026. URL https://arxiv.org/abs/2607.10546.

Mengge Du, Yuntian Chen, Longfeng Nie, Siyu Lou, and Dongxiao Zhang. Physics-constrained robust learning of open-form partial differential equations from limited and noisy data. *Physics of Fluids*, 36(5):057123, 2024a. doi: 10.1063/5.0204187. URL https://doi.org/10.1063/5.0204187.

Mengge Du, Yuntian Chen, Zhongzheng Wang, Longfeng Nie, and Dongxiao Zhang. Large language models for automatic equation discovery of nonlinear dynamics. *Physics of Fluids*, 36(9):097121, 2024b. doi: 10.1063/5.0224297. URL https://doi.org/10.1063/5.0224297.

Mengge Du, Yuntian Chen, and Dongxiao Zhang. DISCOVER: Deep identification of symbolically concise open-form partial differential equations via enhanced reinforcement learning. *Physical Review Research*, 6(1):013182, 2024c. doi: 10.1103/PhysRevResearch.6.013182. URL https://doi.org/10.1103/PhysRevResearch.6.013182.

Mingquan Feng, Yixin Huang, Yizhou Liu, Bofang Jiang, and Junchi Yan. PhysPDE: Rethinking PDE discovery and a physical HYpothesis selection benchmark. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/05d42b7bd130ecbc57fdf02cbdcd370e-Abstract-Conference.html.

Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized PDE modeling. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=dPSTDbGtBY.

YuJie Huang, WenWu He, ZhuoEr Lin, Congcong Liu, Dong Liang, and Zhuo-Xu Cui. Hypothesize, evaluate, refine: A scientific agent for PDE discovery with unknown spatial coefficient fields. *arXiv preprint arXiv:2608.27475*, 2026. URL https://arxiv.org/abs/2608.27475.

Elizaveta Ivanchik, Timur Bavshin, and Alexander Hvatov. Does LLM dream of differential equation discovery? In *NeurIPS Workshop on AI for Science: The Reach and Limits of AI for Scientific Discovery*, 2025. URL https://neurips.cc/virtual/2025/125859.

Krzysztof Kacprzyk, Zhaozhi Qian, and Mihaela van der Schaar. D-CIPHER: Discovery of closed-form partial differential equations. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1201. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/57c30b677add9aa78e1745f0643104d0-Abstract-Conference.html.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021. doi: 10.1038/s42254-021-00314-5. URL https://www.nature.com/articles/s42254-021-00314-5.

Georgios Kissas, Yibo Yang, Eileen Hwuang, Walter R. Witschey, John A. Detre, and Paris Perdikaris. Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 358:112623, 2020. doi: 10.1016/j.cma.2019.112623. URL https://doi.org/10.1016/j.cma.2019.112623.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabricio de Franca, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason Moore. Contemporary symbolic regression methods and their relative performance. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021a. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/c0c7c76d30bd3dcaefc96f40275bdc0a-Paper-round1.pdf.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/c0c7c76d30bd3dcaefc96f40275bdc0a-Abstract-round1.html.

Edward N. Lorenz. Predictability: A problem partly solved. In *Proceedings of the Seminar on Predictability*, volume 1, pp. 1–18, Reading, UK, 1996. European Centre for Medium-Range Weather Forecasts.

Xiao Luo, Changhu Wang, Yizhou Sun, and Wei Wang. How do large language models perform on PDE discovery: A coarse-to-fine perspective. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2684–2697. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.145. URL https://aclanthology.org/2025.findings-emnlp.145/.

Guanghan Meng, Jian Zhong, Qinrong Zhang, Justin S. J. Wong, Jianglai Wu, Kevin K. Tsia, and Na Ji. Ultrafast two-photon fluorescence imaging of cerebral blood circulation in the mouse brain in vivo. *Proceedings of the National Academy of Sciences*, 119(23):e2117346119, 2022. doi: 10.1073/pnas.2117346119. URL https://www.pnas.org/doi/abs/10.1073/pnas.2117346119.

Daniel A. Messenger and David M. Bortz. Weak sindy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110525. URL https://www.sciencedirect.com/science/article/pii/S0021999121004204.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1430. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/4f9a5acd91ac76569f2fe291b1f4772b-Abstract-Datasets_and_Benchmarks_Track.html.

Sarang Manoj Pekhale, Amartya Roy, Rajat Sarkar, and Souvik Chakraborty. See, hypothesize, validate: Multimodal agentic framework for discovering governing PDEs. *arXiv preprint arXiv:2608.27869*, 2026. URL https://arxiv.org/abs/2608.27869.

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

Junqi Qu, Yan Zhang, Shangqian Gao, and Shibo Li. Dynamic bayesian optimization framework for instruction tuning in partial differential equation discovery. *arXiv preprint arXiv:2601.00088*, 2025. URL https://arxiv.org/abs/2601.00088.

Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017. doi: 10.1126/sciadv.1602614. URL https://doi.org/10.1126/sciadv.1602614.

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

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

Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equations: Multioperator learning and extrapolation. *Phys. Rev. E*, 111:035304, Mar 2025. doi: 10.1103/PhysRevE.111.035304. URL https://link.aps.org/doi/10.1103/PhysRevE.111.035304.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0117. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0a9747136d411fb83f0cf81820d44afb-Abstract-Datasets_and_Benchmarks.html.

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

Hao Xu, Yuntian Chen, Rui Cao, Tianning Tang, Mengge Du, Jian Li, Adrian H. Callaghan, and Dongxiao Zhang. Generative discovery of partial differential equations by learning from math handbooks. *Nature Communications*, 16:10255, 2025. doi: 10.1038/s41467-025-65114-2. URL https://www.nature.com/articles/s41467-025-65114-2.

Fan Yang and Matt Thomson. What should a large language model see? physical invariants as a data representation for PDE discovery. *arXiv preprint arXiv:2608.25189*, 2026. URL https://arxiv.org/abs/2608.25189.

Jianke Yang, Ohm Venkatachalam, Mohammad Kianezhad, Sharvaree Vadgama, and Rose Yu. Think like a scientist: Physics-guided LLM agent for equation discovery. *arXiv preprint arXiv:2602.12259*, 2026. URL https://arxiv.org/abs/2602.12259.

V.I. Zarnitsina, A.V. Pokhilko, and F.I. Ataullakhanov. A mathematical model for the spatio-temporal dynamics of intrinsic pathway of blood coagulation. i. the model description. *Thrombosis Research*, 84(4):225–236, 1996. ISSN 0049-3848. doi: https://doi.org/10.1016/S0049-3848(96)00182-X. URL https://www.sciencedirect.com/science/article/pii/S004938489600182X.

A DATASET CONSTRUCTION AND CONTENTS

Observation unit and schema. A catalog entry is one system or trajectory, even when it has several target equations. Each entry provides sampled observations at its available noise tiers, coordinates, field-role and axis metadata, and a spatial mask or temporal split where applicable. With d target channels and a supplied auxiliaries, ODE arrays have shape $(N_t, d + a)$, time-dependent PDE arrays have shape $(N_{x_1}, \ldots, N_{x_n}, N_t, d + a)$ for $n \leq 2$, and stationary 2D arrays have shape $(N_x, N_y, d + a)$. Spatial axes precede time and target channels precede auxiliaries; scalar fields retain the final channel axis. Exact reference expressions, numerical coefficients and available parameter or trajectory indices are kept separately from discovery inputs. No task supplies precomputed derivatives. All observations in the current release are simulated.

PROSE-PDE-generated trajectories. We used the PROSE-PDE generator (Sun et al., 2025) to simulate 25 trajectories for each of 16 one-dimensional equation families, combining five coefficient settings with five initial conditions, on a 256×256 grid over $x, t \in [0, 2]$. The generator also emits each trajectory's governing equation in implicit residual form, e.g. $u_t - 0.00297u_{xx} = 0$. We solved each residual symbolically for u_t, or for u_{tt} in second-order-in-time families, to obtain the explicit reference equation, and checked that every family had one label per trajectory. For Fokker–Planck, we corrected the sign of the $\sin(10x)u_x$ term in the generator, regenerated the trajectories on a 512×512 grid and verified the coefficients and signs of the resulting labels. We then curated the collection with a family allow-list and a trajectory-level exclusion list from manual numerical and label-quality review, converted the retained trajectories to the benchmark schema and renumbered them consecutively with their labels unchanged. The final collection contains 314 trajectories from 14 families.

MDBench. The four one-dimensional cases (0317–0320) are the clean fields of the public MDBench release (Bideh et al., 2026), used without cropping or resampling; we did not use its SNR-based noisy versions. The released cylinder cases cover only $t \in [1, 2]$ with 51 frames on a 100×30 grid, so for 0321–0322 we reran the MDBench FEniCS scripts on $t \in [0, 5]$ and linearly interpolated the solutions onto a 250×50 grid with its structuring scripts. The Navier–Stokes case was integrated with $\Delta t = 10^{-3}$ and saved every tenth step; the reaction–diffusion case was integrated with $\Delta t = 10^{-2}$ and saved at every step. Each case has 500 frames.

ODE-Strogatz. Each glider and predator–prey file of ODE-Strogatz (La Cava et al., 2021a) concatenates four 100-sample trajectories from different initial conditions, sampled at a step of about 0.1. We split each file into four systems (0323–0330) and used only the state columns. Integrating the reference equations from the first sample of each trajectory reproduces it to within about 1–3% for glider and 5–9% for predator–prey, so the source trajectories themselves carry some integration error.

Lorenz–96. The two Lorenz–96 systems (Lorenz, 1996) (0001–0002) use $(N, F) = (8, 10)$ and $(60, 8)$. They were integrated with LSODA (SciPy `odeint`, default tolerances) at $\Delta t = 0.01$ from $x = F\mathbf{1}$ with x_1 perturbed by 0.01.

Coagulation. The 8-dimensional system (0331) follows the coagulation model of Zarnitsina et al. (1996) with states IXa, Xa, IIa, II, VIIIa, Va, APC and Ia and four intermediates as auxiliaries. Its time grid is nonuniform, with steps from 8×10^{-6} to 0.9. The 25-dimensional system (0332) simulates the coagulation model of Anand et al. (2008). Each initial concentration was drawn as its nominal value times $U(0.5, 1.5)$. The system was integrated with LSODA at $\Delta t = 0.002$, and every tenth of the first 50,000 samples was kept, giving 5000 samples at $\Delta t = 0.02$. Its 16 auxiliary fields hold rational rate terms and the forcing $\exp(-972.58t)$.

Coupled tasks and reference labels. The two MDBench cylinder cases (0321-0322) couple several target fields to observed pressure, velocity or forcing fields. Their masks exclude the solid cylinder when estimating derivatives. The coagulation systems likewise supply biochemical intermediates as inputs rather than additional targets. Reference equations and coefficients are kept separate from task observations. Table 3 shows the equation families; the reference catalog con-

tains the system-specific expressions used for scoring. For Cahn–Hilliard, coefficients of u^2u_{xx} and $u(u_x)^2$ are rounded separately, so the stored expanded expression is authoritative.

Table 3: Family-wise reference equations. Each nonblank row gives one governing equation or constraint. For 0322, $\mathbf{v} = (v_x, v_y)$. PROSE-PDE coefficients vary by system; exact numerical expressions are in the separate reference catalog.

Family	System IDs	Reference equation
Lorenz–96	0001	$\dot{x}_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + 10$ (cyclic i).
	0002	$\dot{x}_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + 8$ (cyclic i).
Heat	0003–0027	$u_t = \kappa u_{xx}$.
Viscous conservation	0028–0052	$u_t = -au_x + \nu u_{xx}$.
	0053–0077	$u_t = -a\cos(u)u_x + \nu u_{xx}$.
	0078–0102	$u_t = -au^2u_x + \nu u_{xx}$.
Inviscid conservation	0103–0127	$u_t = -a\cos(u)u_x$.
	0128–0137	$u_t = -au^2u_x$.
1D reaction–diffusion	0138–0162	$u_t = \nu u_{xx} + ru(1-u)(u-\alpha)$.
	0183–0207	$u_t = \nu u_{xx} + ru$.
	0208–0232	$u_t = \nu u_{xx} + ru^2(1-u)^2$.
	0242–0266	$u_t = \nu u_{xx} + ru(1-u)$.
Fokker–Planck	0163–0182	$u_t = Du_{xx} - \partial_x[b\sin(10x)u]$.
Cahn–Hilliard	0233–0241	$u_t = Au^2u_{xx} + Bu(u_x)^2 - Cu_{xx} - Eu_{xxxx}$.
Wave	0267–0291	$u_{tt} = c^2u_{xx}$.
Porous medium	0292–0296, 0307–0311	$u_t = \frac{1}{20}\partial_{xx}(u^4)$.
	0297–0301	$u_t = \frac{1}{20}\partial_{xx}(u^3)$.
	0302–0306, 0312–0316	$u_t = \frac{1}{20}\partial_{xx}(u^2)$.
Advection	0317	$u_t = -0.1u_x$.
Viscous Burgers	0318	$u_t = -uu_x + 0.1u_{xx}$.
KdV	0319	$u_t = -6uu_x - u_{xxx}$.
KS	0320	$u_t = -uu_x - u_{xx} - u_{xxxx}$.
2D Navier–Stokes	0321	$u_t = -uu_x - vu_y - p_x + 0.001\Delta u$.
	0321	$v_t = -uv_x - vv_y - p_y + 0.001\Delta v$.
	0321	$u_x + v_y = 0$.
2D reaction–diffusion	0322	$\partial_t c_0 = -\mathbf{v}\cdot\nabla c_0 + 0.01\Delta c_0 + f_0 - 10c_0c_1$.
	0322	$\partial_t c_1 = -\mathbf{v}\cdot\nabla c_1 + 0.01\Delta c_1 + f_1 - 10c_0c_1$.
	0322	$\partial_t c_2 = -\mathbf{v}\cdot\nabla c_2 + 0.01\Delta c_2 + f_2 + 10c_0c_1 - 10c_2$.
Glider	0323–0326	$\dot{v} = -0.05v^2 - \sin\theta$.
	0323–0326	$\dot{\theta} = v - \cos\theta/v$.
Predator–prey	0327–0330	$\dot{x} = x(4 - x - y/(1+x))$.
	0327–0330	$\dot{y} = y(x/(1+x) - 0.075y)$.
Coagulation	0331 (8D)	See Zarnitsina et al. (1996)
	0332 (25D)	See Anand et al. (2008)

Noise tiers and scoring populations. Noisy tiers for all sources use the pointwise perturbation of Section 3, applied to targets and supplied auxiliaries alike. The release contains clean, 1%, 5% and 10% observations only. The 330 labeled systems other than 0331 and 0332, together with all five stationary CFD systems, have all four tiers. System 0331 has only 5% observations; 0332 has clean and 5%. Thus the labeled observation populations are 331, 330, 332 and 330 in tier order. Clean-reference NMSE excludes 0331 even at 5% because no clean observation is available for it, giving scoring populations of 331, 330, 331 and 330. Exact structural recovery retains the

fixed denominator of 332 labeled systems at every tier, counting an unavailable tier or unsupported interface as a non-recovery; the five unlabeled CFD fields enter neither reference-based metric. The CFD noisy variants were generated with multiplicative perturbations at the same nominal tiers.

Temporal evaluation split. For each temporal system, the first $\lfloor 0.7N_t \rfloor$ time points form the fitting prefix and the rest form a held-out future segment. This tests later times of the same trajectory, not transfer to unseen initial conditions, coefficients or geometries. The stationary CFD fields have no temporal holdout.

A.1 Unlabeled CFD blood-flow fields

The five author-provided CFD systems (0333–0337) are individual stationary two-dimensional simulated snapshots. They were obtained by running CFD simulations on irregular regions shaped from mouse blood vessels (Meng et al., 2022). No governing equation was documented for these snapshots, and neither a momentum balance nor two-dimensional incompressibility is assigned as a ground-truth label. The irregular regions are padded to rectangular grids, with recorded masks indicating the original regions. Each system has two velocity targets and a supplied pressure field on the associated padded rectangular grid. In ID order, their (N_x, N_y) grid sizes are $(432, 159)$, $(274, 403)$, $(319, 448)$, $(241, 491)$ and $(269, 411)$.

Figures 3 and 4 visualize the clean fields and post hoc diagnostics for the separate strict-blind case study. Derivatives in the displayed diagnostics use fourth-order centered differences with mean grid spacing (relative spacing variation below 5.4×10^{-5}) and only fully fluid interior points with a 7×7 valid neighborhood. The fitted coefficients used to visualize momentum-term magnitudes, including $\mu \approx 3$, come from an independent least-squares calculation, not from the discovery agents.

Panel (b) rescales pressure to $[0, 1]$ within each field. Panel (c) plots $|\rho(\mathbf{u}\cdot\nabla)\mathbf{u}|/|\mu\nabla^2\mathbf{u}|$ on a shared logarithmic 10^{-4}–10^{-1} scale, saturated outside that range; this relative-magnitude diagnostic is not a detection threshold. Panel (d) divides the RMS magnitude of inertia by that of the pressure gradient. One strict-blind Claude Opus 5.5 subagent at high effort analyzed each field with permuted, rescaled anonymous channels and no disclosed physical roles. They proposed Stokes forms for 0333, 0335 and 0336, and Navier–Stokes momentum for 0334 and 0337. These are hypotheses, not exact recoveries against a documented governing equation.

B Methods and hyperparameter settings

We treat SINDy and PDE-FIND as one method because PDE-FIND extends SINDy to spatiotemporal data, and WSINDy as a separate method because its weak formulation avoids pointwise derivative estimates; PySINDy implements all three (Brian et al., 2023). LLM-PDESR uses a language model to propose equation forms inside an evolutionary search, whereas GPT-5.6-sol and Claude Opus 5, both at high reasoning effort, act as tool-using agents that choose the numerical analysis as well as the equation. The subsections below give each method's translation of the prior cards, its tuning and its attempt protocol. Throughout the appendices, Levels 1–3 denote the prior levels L1–L3 of Section 3.3.

B.1 Cross-method setup details and Prior cards

Temporal order and differentiation. SINDy and WSINDy receive the declared temporal target (u_t or u_{tt}), whereas the agents infer temporal order themselves. The benchmark supplies no precomputed derivatives. SINDy uses finite differences with optional, tuned Savitzky–Golay smoothing; WSINDy needs no pointwise temporal derivative; SGA-PDE and DISCOVER use finite differences, except in DISCOVER's noisy runs, which use automatic differentiation through a PINN; LLM-PDESR's evaluator supplies the spatial derivative operator; and each agent writes its own estimator. Information about noise can only choose among a method's estimators, so noisy-tier differences combine the estimator toolbox, noise information and search.

Tuning and noise information. Only SINDy and WSINDy are tuned per group: one numerical policy per physical family, noise condition and prior level, with coefficients and support fitted per

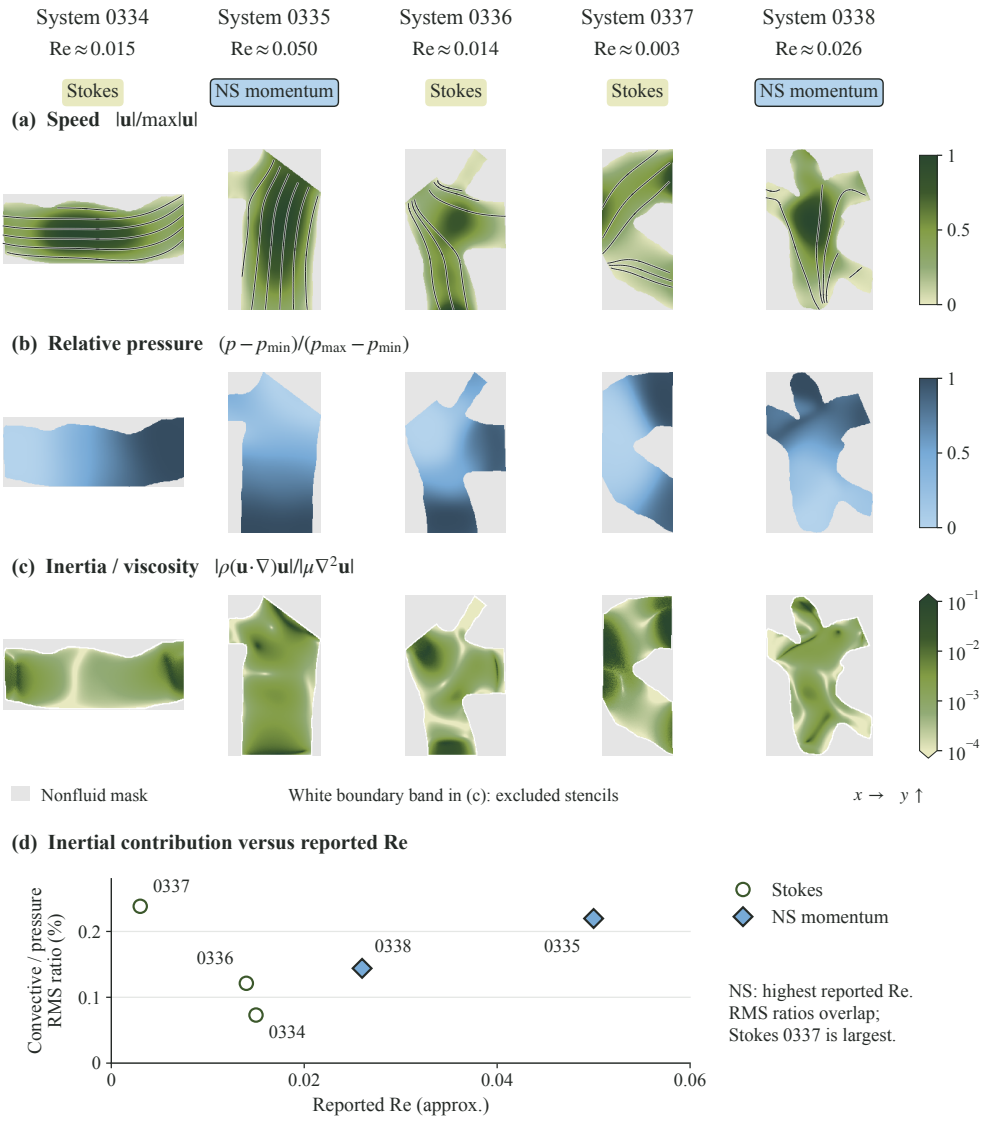


Figure 3: Five clean stationary CFD fields (0333–0337): (a) normalized speed and streamlines, (b) rescaled pressure, (c) local inertia/viscosity ratio, and (d) RMS inertia/pressure-gradient ratio versus reported Reynolds number. Gray marks outside-domain padding; white bands mark excluded stencils.

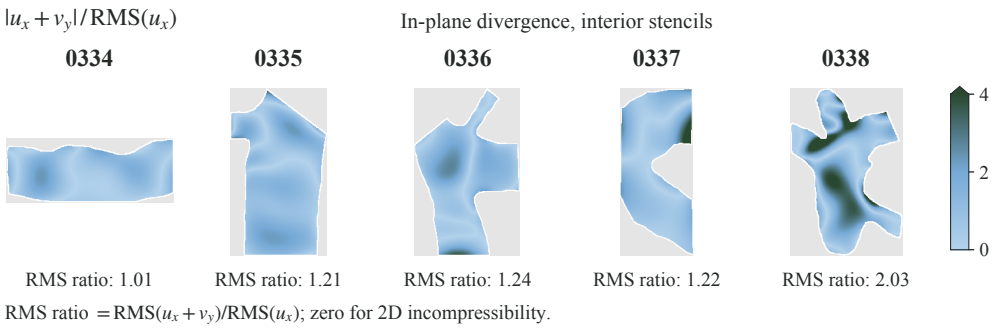


Figure 4: In-plane divergence of the same five clean CFD fields. Color shows $|u_x + v_y|/\mathrm{RMS}(u_x)$ on the valid interior, with the scale saturated above 4. Labels give $\mathrm{RMS}(u_x + v_y)/\mathrm{RMS}(u_x)$, which ranges from 1.01 to 2.03. Gray denotes nonfluid padding and white bands excluded derivative stencils. The nonzero in-plane divergence does not identify its cause or establish a violation of three-dimensional mass conservation.

system (Appendices B.3 and B.5). Each policy fixes the differentiation, smoothing or weak-form settings for a known noise condition and family, so both methods effectively know the noise level. The other methods run fixed configurations: SGA-PDE applies one policy selected on clean data to all tiers; DISCOVER uses frozen settings; LLM-PDESR runs upstream commit `0b491b1` with a changed model backend; and the agents make their numerical choices inside each attempt's single analysis call, which has no short time limit. The agent prompts disclose no noise level, and their fixed text states no periodicity or boundary conditions (Appendices B.6 and B.7), so under noise SINDy and WSINDy know the noise level and DISCOVER, through its PINN switch, knows that the data are noisy, whereas the agents must infer both from the arrays.

Method applicability. Applicability is recorded separately from discovery success. The native SGA-PDE capability audit and the LLM-PDESR clean-case manifest agree on a 293-system supported subset of the 332 labeled systems: 318 scalar one-dimensional PDEs minus 25 wave systems (0267–0291) whose reference target is u_{tt}. Both implementations fit u_t and have no source-native u_{tt} target. They also exclude twelve labeled ODE systems (0001–0002 and 0323–0332) and two 2D PDE systems (0321–0322) because their interfaces assume one scalar field and one spatial coordinate. These 39 labeled systems are identified as unsupported, not as attempted but failed fits. They count as non-recoveries in the 332-system denominator of Section 3.4; a 293-system conditional rate is supplementary. The SGA-PDE audit further classifies only 223 of its 293 input-compatible references as representable by its fixed search grammar, with 70 not representable. This additional search-space limit must accompany any SGA-PDE result. The five unlabeled systems are outside all exact-recovery denominators.

Prior cards. One key innovation of our benchmark is to assign system-wise cards of different levels which are later translated into method-specific hypothesis spaces. The 332 labeled systems receive cards at all three levels. The five unlabeled systems (0333–0337) receive Level 1 only, because they have no reference equation from which to derive a Level 3 card or an exact-support score.

Section 3.3 defines the three levels. The same prior card can require different method-specific implementations. Numerical differentiation, sparsification and validation choices belong to a method's tuning protocol rather than to the disclosed prior. A lower error with more prior information, or with less noise, is a testable outcome rather than a selection rule.

For the 318 scalar one-dimensional PDE systems (0003–0320), the Level 2 cards are source-free homogeneous conservation (173 systems), homogeneous spatial coupling with permitted local sources (125), and source-free conservation with permitted coordinate-dependent flux (20). Their Level 3 cards specify twelve constitutive families, with alternatives enumerated in Table 5. The ODE extension preserves the same informational progression: Lorenz–96 adds radius-two locality and ring-

shared coefficients at Level 2; glider dynamics add angular periodicity; predator–prey dynamics add extinction-boundary structure. The two biochemical systems have no independently established Level 2 restriction beyond their schema, so $\mathcal{H}_2 = \mathcal{H}_1$ there. The two 2D PDEs add source-free homogeneous momentum or homogeneous transport with sources at Level 2.

Table 4 gives the exact card names and there disclosure texts. The scalar families use three Level 2 and twelve Level 3 cards. The ODE and 2D extensions use six family pairs; the biochemical pair has no extra Level 2 text. These textual disclosures do not prescribe a derivative estimator, basis, support count, or regression algorithm to either agent.

The disclosure texts of the prior cards are directly imported by GPT-5.6-sol and Claude Opus 5. Their Level 1 hypothesis spaces contain only the numerical schema and the fixed method prompt. Level 2 appends the relevant broad structural card; Level 3 appends that card, when nonempty, and a constitutive-family card with alternatives.

Table 4: Scientific prior cards used by PDEDiscoveryBench. The disclosure texts are directly used in the GPT-5.6-sol and Claude Opus 5 prompts. Level 3 includes its Level 2 card when that card is nonempty.

Card	Disclosure text
Scalar PDE Level 2	
Conservative, uniform	The field obeys a local source-free balance law with unknown flux. Constitutive coefficients have no explicit position dependence. The flux, derivative content, nonlinearities and coefficients are unknown.
Sources, uniform	The field admits spatial coupling and local source or restoring terms. Constitutive coefficients have no explicit position dependence. Source laws, spatial operators, nonlinearities and coefficients are unknown.
Conservative, spatial	The field obeys a local source-free balance law with unknown flux. The flux may depend explicitly on position. Its coordinate profile, nonlinearities, derivatives and coefficients are unknown.
Scalar PDE Level 3	
H	Linear homogeneous transport and diffusion; either process can be absent. The drift and diffusivity are unknown.
PV	Homogeneous viscous transport with an unknown polynomial flux of degree at most four and constant diffusivity. Active powers, signs and coefficients are unknown.
PI	Homogeneous inviscid transport with an unknown polynomial flux of degree at most four. Active powers and coefficients are unknown.
TV	Homogeneous viscous transport with an unknown combination of linear, sine and cosine fluxes and constant diffusion. Phase, relative weights and coefficients are unknown.
TI	Homogeneous inviscid transport with an unknown combination of linear, sine and cosine fluxes. Phase, relative weights and coefficients are unknown.
R	Homogeneous reaction-diffusion with constant diffusivity and an unknown polynomial local reaction of degree at most four, including a possible constant source. Degree, active powers and all coefficients are unknown.
FP	A linear conservative drift-diffusion law with constant diffusivity and an unknown position-dependent drift. The drift may be constant, polynomial through degree two, or a sine/cosine combination at the declared frequencies 1 through 10. Active profiles and all coefficients are unknown.
CH	A conservative homogeneous chemical-potential law with an unknown polynomial state potential of degree at most four and a possible linear fourth-order gradient regularizer. Active powers, regularization and coefficients are unknown.
W	A homogeneous linear second-time-order field law with possible uniform forcing, linear restoring force, damping, spatial drift and second spatial derivative. Which contributions are active and their coefficients are unknown.
PM	Homogeneous conservative nonlinear diffusion with an unknown polynomial constitutive potential spanning powers one through four. The exponent, combinations of powers and coefficients are unknown; transport and local reaction are excluded.
KD	Homogeneous conservative dispersive transport with an unknown polynomial flux through degree four and a possible linear third spatial derivative. Active powers and coefficients are unknown.

Table 4 continued

Card	Disclosure text
KS	Homogeneous conservative polynomial transport through degree four with possible second- and fourth-order linear spatial terms. Active powers, signs and coefficients are unknown.
ODE/2D extension	
lorenz96 L2	The observed target channels form a periodic ring in their listed order. Each channel's evolution depends on a radius-two neighborhood, and the same local law is shared across ring sites. Coefficients need not match between independent trajectories.
lorenz96 L3	Consider autonomous local quadratic transport and linear relaxation, with a possible constant drive. The relevant two-site products, squares, and central linear term remain alternatives; their active identities and coefficients are unknown.
cylinder flow L2	Targets u0 and u1 are the two components of a source-free, spatially homogeneous, approximately incompressible momentum field on the x0,x1 plane. Auxiliary a0 is a measured pressure-like field. The stored mask marks valid locations. Incompressibility is a physical structural prior, not an assertion that the discrete divergence vanishes exactly.
cylinder flow L3	Consider competing quadratic momentum-flux couplings, pressure-gradient directions, linear drift, and anisotropic second-order viscosity. Do not assume which coupling or direction is active or that coefficients are tied across components.
transport reaction L2	Targets u0,u1,u2 are concentrations on the x0,x1 plane. Auxiliaries a0,a1 are measured carrier-velocity components, a2 is pressure-like, and a3,a4,a5 are supplied forcing channels. Evolution is spatially homogeneous in its constitutive coefficients but can include local reactions and supplied forcing. The carrier is approximately incompressible; verify any discrete divergence assumption.
transport reaction L3	Consider competing measured-flow transport, linear mixing/diffusion, linear or quadratic interactions among concentrations, and supplied forcing alternatives. Active reaction pairs, forcing assignments, directions, and coefficients remain unknown.
glider L2	The two target channels are speed u0 and orientation angle u1. Orientation is periodic, so nonperiodic powers of u1 are structurally inappropriate. Evolution is autonomous and local in the observed states.
glider L3	Consider competing speed drag laws through cubic order and first angular harmonics, including speed-weighted or reciprocal-speed angular effects where data support their domain. No harmonic, power, or coefficient is identified as active.
predator prey L2	The two target channels are interacting nonnegative populations u0 and u1. Each population's own time derivative vanishes at its own zero boundary. Retained denominator alternatives must remain finite at that boundary.
predator prey L3	Consider growth, self-limitation, cross-population interaction, and alternative saturation by either population at several positive scales. Which interaction and saturation are active, and their coefficients, remain unknown.
biochemical L2	No additional Level 2 disclosure.
biochemical L3	All supplied target and auxiliary channels may enter autonomous local kinetics. Consider constant, linear, and quadratic combinations, including interactions among auxiliary channels. No reaction graph, target-specific assignment, or active term is supplied.

B.2 SINDy TRANSLATION: MATHEMATICAL CANDIDATE LIBRARIES

This subsection specifies the admissible columns, before sparse fitting. Let D_x denote a native PySINDy spatial derivative applied to the entire expression in parentheses. $D_x^r u$ is its order-r derivative. All libraries are assembled from native PySINDy feature-library constructors; numerical differentiation uses native differentiator objects. The symbols below define candidate mathematics, not a handwritten stencil or a new fitting API. A common backend makes every Level 3 column an identical column of Level 2, and every Level 2 column an identical column of Level 1. Separately tuned backends can change the numerical values of those columns.

Scalar 1D PDEs (0003–0320). The common Level 1 has five blocks:

$$\begin{aligned}
F &= \{D_x u, D_x(u^2), D_x(u^3), D_x(u^4), D_x \sin u, D_x \cos u,\\
&\quad D_x(uD_x u), D_x(u^2 D_x u), D_x(u^3 D_x u), D_x^2 u, D_x^3 u, D_x^4 u\},\\
S &= \{1, u, u^2, u^3, u^4, \sin u, \cos u\},\\
G &= \{(D_x u)^2, u(D_x u)^2, u^2(D_x u)^2, u^3(D_x u)^2\},\\
X_F &= \{D_x(xu), D_x(x^2 u)\} \cup \{D_x(\sin(kx)u), D_x(\cos(kx)u) : 1 \le k \le 10\},\\
X_S &= \{xu, x^2 u\} \cup \{\sin(kx)u, \cos(kx)u : 1 \le k \le 10\}.
\end{aligned}$$

Thus Level 1 has $12+7+4+22+22 = 67$ columns. The 25 wave systems with declared target u_{tt} additionally receive the measured u_t input at every level, giving 68 columns. Level 2 retains F for homogeneous source-free conservation, $F \cup S \cup G$ for homogeneous equations that permit sources (plus u_t for wave), or $F \cup X_F$ for coordinate-dependent source-free conservation. These contain 12, 23 (24 for wave), or 34 columns respectively. The state degree-four cap, derivative-order-four cap and coordinate-frequency range are finite envelope choices, not claims about the active equation.

Table 5: Complete Level 3 candidate menus for the 318 scalar 1D PDE systems. Every entry is a whole native library column and all coefficients are unknown.

L3 card	Systems	Retained candidate columns
H	0003–0052, 0317	$D_x u, D_x^2 u$
PV	0078–0102, 0318	$D_x(u^k), k = 1,2,3,4$; $D_x^2 u$
PI	0128–0137	$D_x(u^k), k = 1,2,3,4$
TV	0053–0077	$D_x u, D_x \sin u, D_x \cos u, D_x^2 u$
TI	0103–0127	$D_x u, D_x \sin u, D_x \cos u$
R	0138–0162, 0183–0232, 0242–0266	$1, u, u^2, u^3, u^4, D_x^2 u$
FP	0163–0182	$D_x u, D_x^2 u$ and every column of X_F
CH	0233–0241	$D_x^2 u, D_x(uD_x u), D_x(u^2 D_x u), D_x(u^3 D_x u), D_x^4 u$
W	0267–0291	$1, u, u_t, D_x u, D_x^2 u$
PM	0292–0316	$D_x^2 u, D_x(uD_x u), D_x(u^2 D_x u), D_x(u^3 D_x u)$
KD	0319	$D_x(u^k), k = 1,2,3,4$; $D_x^3 u$
KS	0320	$D_x(u^k), k = 1,2,3,4$; $D_x^2 u, D_x^4 u$

Table 5 is the complete scalar Level 3 menu; no target is given just its true term. In particular, PM includes the quadratic conservative flux derivative $D_x(uD_x u)$ for every porous-medium case. The complete flux derivatives retain their coefficient ties through sparse fitting.

ODE systems (0001–0002, 0323–0332). For observed and supplied channels $z_1, \ldots, z_n$, the generic local-source block S has 1, every z_i^p for $p = 1,2,3,4$, every pair $z_i z_j$ for $i < j$, and $\sin z_i, \cos z_i$. When $n \le 4$, Level 1 also contains $z_i^p \sin z_j, z_i^p \cos z_j$ for $i \ne j$, $p = 1,2$; $1/z_i$, $\sin z_j/z_i$, $\cos z_j/z_i$ when the denominator channel has a predeclared positive domain; and $z_j/(K+z_i)$ plus $z_i z_j/(K+z_i)$ for the admitted indices and $K \in \{0.1, 1, 10\}$. For autonomous ODEs the spatial blocks F, G, X_F, X_S are inactive. These rational scales are fixed dictionary elements, not continuously inferred coefficients.

Lorenz–96 Level 1 uses this generic bank over all eight or sixty state channels (77 or 2,131 columns). Level 2 uses the same functions on the five radius-two sites and shares one local law around the ring (41 columns). Level 3 keeps a constant, the central state, the five local squares and all ten distinct local pair products (17 columns), leaving competing transport products available. Glider Level 1 has 34 or 46 columns depending on fit-prefix positive-domain admission. Level 2 removes nonperiodic occurrences of the orientation θ: it keeps 1, v through v^4, $\sin v, \cos v, 1/v$, $v/(K+v)$, $\sin\theta, \cos\theta$, $v^p \sin\theta, v^p \cos\theta$ for $p = 1,2$, and $\sin\theta/v, \cos\theta/v$ (19 columns). Level 3 keeps $v, v^2, v^3, \sin\theta, \cos\theta, v\sin\theta, v\cos\theta, \sin\theta/v, \cos\theta/v$ (nine).

For predator–prey target q and other state r, Level 1 has 46 common columns. Level 2 keeps only terms vanishing at $q = 0$ with finite denominators there: $q, q^2, q^3, q^4, \sin q, qr, q^p \sin r, q^p \cos r$ for $p = 1,2$, and $q/(K+q), q/(K+r), qr/(K+q), qr/(K+r)$ for all three K (22 columns). Level

3 retains q, q^2, qr and $qr/(K+x), qr/(K+y)$ for every K (nine columns). For biochemical systems 0331/0332, supplied auxiliary fields enter the same generic source grammar as observed states. Their Level 2 library equals Level 1 (139/1,067 columns), because variable labels alone do not establish a reaction graph. Level 3 retains every constant, linear and quadratic combination of all supplied channels (91/903 columns); it still does not reveal target-specific active interactions.

Two-dimensional PDEs (0321–0322). For each physical input channel q and direction $a \in \{x, y\}$, Level 1 adds $D_a(q^k)$ for $k = 1, \ldots, 4$, $D_a \sin q$, $D_a \cos q$, $D_a(q^k D_a q)$ for $k = 1, 2, 3$, $D_a^r q$ for $r = 2, 3, 4$, and $D_x D_y q$ to F; each distinct channel pair q, r adds $D_x(qr), D_y(qr)$. The S block has a constant, all q^k, $\sin q, \cos q$, and pair products. The G block has $q^k (D_a q)^2$ for $k = 0, \ldots, 3$. The coordinate blocks contain $D_a(h(a)q)$ or $h(a)q$, with $h(a) \in \{a, a^2, \sin(ka), \cos(ka) : 1 \le k \le 10\}$. These produce 391 Level 1 columns for cylinder momentum and 1,252 for transport–reaction.

Cylinder Level 2 keeps source-free homogeneous F (81 columns). Its target-specific Level 3 retains six fluxes $D_x(u^2), D_y(u^2), D_x(v^2), D_y(v^2), D_x(uv), D_y(uv)$, both pressure gradients, both first derivatives of the target, and D_{xx}, D_{yy}, D_{xy} of the target (13 columns). Transport–reaction Level 2 keeps $F \cup S \cup G$ (460 columns). For target c_i, Level 3 keeps $D_x(c_i v_x), D_y(c_i v_x), D_x(c_i v_y), D_y(c_i v_y)$; $D_x c_i, D_y c_i, D_{xx} c_i, D_{yy} c_i, D_{xy} c_i$; a constant, all c_j, c_j^2, all distinct $c_j c_k$, and each supplied forcing field f_0, f_1, f_2 (22 columns). Provided velocity, pressure and source fields are available at every level; their roles narrow the library only when the card discloses them. Discrete product rules and measured incompressibility need not hold exactly, so structural scoring uses the declared strict canonical representation and records those approximation limits.

B.3 Pointwise SINDy tuning and evaluation contract

The numerical selection unit is a named physical family × noise condition × prior level. One selected differentiation/optimizer policy is shared by all trajectories in that group; each trajectory estimates its own coefficients and support. The scalar experiment has 18 families, four retained noise conditions (clean, 1%, 5%, 10%), and three levels, yielding 216 selected policies. The labeled ODE/2D extension follows the same sharing rule within its applicable families; 0331 is a separate 5%-only matching-noise group. Table 6 groups all 285 selected policies into 95 family–noise rows, with the three levels shown side by side. It combines two frozen experiment strata and must not be read as one joint grid search.

The original HDF5 time split places the fitting prefix before a held-out final suffix. The fitting prefix is divided 60/40 for inner fit and validation; differentiation and smoothing are performed inside each split. For the scalar families, 73 of 318 trajectories calibrate shared policies and 245 do not; singleton families use temporal validation only. For the expanded systems, calibration uses up to four fixed targets per trajectory except biochemical systems, for which all eight or 25 targets participate. Validation minimizes the family mean of $\log_{10}(\max(\mathrm{NMSE}, 10^{-12}))$, with failures penalized and deterministic complexity/configuration tie-breaking. No final-suffix score or recovered support selects a policy. The noisy paired arms have access to clean inner-validation observations and temporal references; that is a disclosed calibration privilege, not a noise-blind benchmark. System 0331 lacks clean observations and uses matching-noise validation.

Every group searches native finite differences or native Savitzky–Golay-smoothed finite differences, ridge α, nominal sparsity threshold τ, column normalization and STLSQ unbiasing. Scalar first-order clean targets search finite difference and seven-point smoothing; noisy or second-order targets search seven- and fifteen-point smoothing, all with cubic local polynomial and finite-difference accuracy two. The scalar coarse grid contains 400 settings per group followed by one local refinement round around a family-wide winner. The ODE/2D extension uses finite difference and seven-point cubic smoothing at clean, 1%, 5%, and 10% noise, with 192 coarse settings per group and one refinement round. The irregularly sampled 0331 uses finite difference alone. Thus the two strata have different derivative menus and search budgets, which limits a causal interpretation of their pooled comparison. Selected α and τ values in Table 6 are the saved nominal values, rounded to three significant digits for display; `methods/sindy/selected_family_policies.csv` retains full precision. The normalized-threshold implementation rescales the effective threshold according to the number of fitting rows.

Native STLSQ first proposes a sparse support. If the union of canonical atoms in its active whole columns exceeds the evaluator-supplied true term count K_{GT}, native SSR drops whole columns along a fixed backward path until the first support meeting that count, followed by native restricted STLSQ refitting at zero threshold. This uses the true *count*, not term identities, and is applied at all three levels. The 2D transport–reaction extension uses $K_{\mathrm{GT}} + 1$ for its declared flux-column accounting; cylinder-flow accounting uses the explicitly declared incompressibility normal form. This known-count procedure is an oracle-assisted variant of SINDy. It can yield a literal-zero model or a failed fit, and neither is removed from the structural denominator. Maximum STLSQ iterations are 50 and fitting uses at most 4,096 deterministic sampled rows in both strata. The ODE/2D extension also has a recorded resource guard when an SSR proposal has more than 128 active columns. Weak-form SINDy is a separate method and is excluded here.

Table 6: Selected SINDy policies, one row per family and noise condition, with the three prior levels side by side. Each level cell shows D/α above τ/NU: $D = F$ is native finite difference, $S7$ or $S15$ is native smoothing with the stated window, and $N, U \in \{Y, N\}$ mark column normalization and STLSQ unbiasing. K is the true-atom cap; $*$ means target-specific. Biochemical denotes 0332; the separate 0331 row uses matching-noise validation. Full precision is in `methods/sindy/selected_family_policies.csv`.

Family	Noise	K	Level 1	Level 2	Level 3
			F/0.0001	F/0	F/0
advection1d	clean	1	0/YY	0/NN	0/NN
			S15/10	S15/10	S15/0
advection1d	1	1	0/NY	0/NY	0/YY
			S15/0.1	S15/0	S15/0
advection1d	5	1	0/YY	0/NN	0/NN
			S15/0.000316	S15/1e-08	S15/0
advection1d	10	1	0/NY	0/YY	0/NY
			S7/0.1	S7/0.1	F/0
biochemical	clean	*	0.00333/YY	0.00333/YY	0.1/NN
			F/10	F/10	F/0.001
biochemical	5	*	0.1/NY	0.1/NY	9/NN
			F/1e-05	F/1e-05	F/0.0001
biochemical_0331	5	*	0.01/YY	0.01/YY	0/NY
			F/0.0001	F/0.0001	F/0
burgers	clean	2	0/NY	0/NY	0/YN
			S15/100	S15/1e-06	S15/0.01
burgers	1	2	0/NN	0/YY	0/NN
			S15/0.001	S15/0.001	S15/0
burgers	5	2	0/YY	0/YY	0/YN
			S15/1	S15/0.316	S7/0.1
burgers	10	2	0/YN	0/YN	0/YN
			F/0.01	F/0.0001	F/0
cahnhilliard_1D	clean	4	0/YY	0/YY	0/YN
			S15/0.316	S15/0.316	S15/0.316
cahnhilliard_1D	1	4	0.00949/YN	0/YN	0/YN
			S15/0.316	S15/0.0316	S15/0.0316
cahnhilliard_1D	5	4	0/YY	0/YN	0/YN
			S15/0.0316	S15/0.01	S15/0.0316
cahnhilliard_1D	10	4	0/YY	0.00949/YY	0/YY
			F/0.001	F/0.001	F/0.001
conservation_cubicflux	clean	2	0/YN	0/YN	0/YN
			S7/0.01	S7/0.00316	S7/0.001
conservation_cubicflux	1	2	0.1/YN	0/YN	0/YN
			S15/1	S15/0.1	S15/0.316
conservation_cubicflux	5	2	0/YY	0/NN	0.03/NN
			S15/0.1	S15/3.16	S15/0.001
conservation_cubicflux	10	2	0/YY	0/NY	0/YN
			F/1e-06	F/1e-06	F/0.316
conservation_linearflux	clean	2	0/YN	0.01/YN	0/NN
			S7/0.1	S15/3.16e-06	S15/0
conservation_linearflux	1	2	0.03/NY	0.03/YY	0/YN
			S15/0.1	S15/1e-06	S15/0
conservation_linearflux	5	2	0.03/NY	0.1/YN	0/NY
			S15/0.000316	S15/0.0001	S15/0
conservation_linearflux	10	2	0/YY	0.3/YY	0/NN

Table 6 continued

Family	Noise	K	Level 1	Level 2	Level 3
			F/3.16e-07	F/1e-05	F/1e-08
conservation_sinflux	clean	2	0.00949/YN	0.0316/YN	0/YN
			S7/0.001	S7/3.16e-06	S7/1e-05
conservation_sinflux	1	2	0.1/YN	0/YN	0/YN
			S7/0.001	S7/0.0001	S15/1e-05
conservation_sinflux	5	2	0/YY	0/YY	0/YN
			S15/0.0001	S15/0.316	S15/0
conservation_sinflux	10	2	0.1/YY	0.0316/NN	0.3/YN
			S7/10	S7/3.33e-05	S7/1e-06
cylinder_flow	clean	*	0.01/NY	0.0333/YY	0/YN
			S7/10	S7/10	S7/0
cylinder_flow	1	*	0.01/NY	0.1/NY	0/YY
			S7/0.1	S7/100	S7/0
cylinder_flow	5	*	0.01/YN	0/NN	0.1/NN
			F/0.1	S7/10	S7/0.1
cylinder_flow	10	*	0.01/YN	0/NY	0/YY
			S7/1e-05	S7/1e-10	S7/1e-08
diff_bistablereact_1D	clean	4	0/YY	0.316/YY	0/YN
			S15/0.1	S15/0.1	S15/1
diff_bistablereact_1D	1	4	0/YY	0/YY	0.0316/YY
			S15/0.1	S15/0.1	S15/31.6
diff_bistablereact_1D	5	4	0/YN	0/YN	0.1/NY
			S15/10	S15/10	S15/100
diff_bistablereact_1D	10	4	0/YY	0.03/NY	0.03/NY
			S7/3.16e-06	S7/0	S7/3.16e-06
diff_linearreact_1D	clean	2	0/NN	0/YN	0/NN
			S15/0.01	S15/0.01	S15/0.001
diff_linearreact_1D	1	2	0.0316/YY	0/YY	0.3/YY
			S15/0.1	S15/0.1	S15/0.1
diff_linearreact_1D	5	2	0/YY	0/YY	0.03/YY
			S15/0.001	S15/1e-08	S15/0.01
diff_linearreact_1D	10	2	0/YN	0/NN	0/YY
			S7/1e-07	S7/1e-06	S7/3.16e-06
diff_logisreact_1D	clean	3	0/YN	0/YN	0/YN
			S15/0.01	S15/0.000316	S15/0.000316
diff_logisreact_1D	1	3	0.0316/YN	0.3/YY	0/YN
			S15/0.1	S15/1e-05	S15/0.316
diff_logisreact_1D	5	3	0/YY	0.00949/YN	0/YY
			S15/0.1	S15/0.1	S15/0.001
diff_logisreact_1D	10	3	0/YY	0/YY	0/YY
			S7/3.16e-06	S7/1e-06	S7/3.16e-06
diff_squarelogisticreact_1D	clean	4	0/YY	0.3/YY	0/NN
			S15/0.1	S15/0.316	S15/0.000316
diff_squarelogisticreact_1D	1	4	0/YY	0/YY	0/YN
			S15/0.0316	S15/0.001	S15/1e-05
diff_squarelogisticreact_1D	5	4	0/YN	0.03/YN	0/NN
			S15/0.0316	S15/10	S15/0.1
diff_squarelogisticreact_1D	10	4	0.0949/NY	0/YY	0/YY
			S7/1e-05	S7/0.00316	S7/0.00316
fplanck	clean	3	0/YN	0/NN	0/NN
			S7/1e-06	S15/0.0001	S15/0.0001
fplanck	1	3	3/YN	3/YN	3.16/YN
			S7/1e-06	S15/0.00316	S15/0.0001
fplanck	5	3	0/NN	0.316/NN	1/YN
			S15/3.16e-07	S15/0.0001	S15/0.0001
fplanck	10	3	0/NN	1/YN	1/YN
			S7/0.0001	S7/0.001	S7/3.33e-05
glider	clean	*	0/NN	0.03/NN	0.1/NN
			S7/0.1	F/0.0333	S7/0.1
glider	1	*	0/YY	0/NN	0/NN

Table 6 continued

Family	Noise	K	Level 1	Level 2	Level 3
			S7/100	S7/10	S7/0.1
glider	5	*	0/YY	0/YY	0/NN
			S7/0.1	F/1	S7/3.33
glider	10	*	0.1/NN	0/NN	0/NN
			S7/100	S7/100	S7/100
heat	clean	1	0/NN	0/NN	0/NN
			S15/0.01	S15/0.001	S15/0
heat	1	1	0/YY	0/YY	0/NY
			S15/0.0316	S15/0.0316	S15/0
heat	5	1	0/NN	0/YY	0/YY
			S15/1	S7/3.16	S15/0
heat	10	1	0/YN	0/YN	0.03/YN
			F/0.1	F/1	F/10
inviscid_conservation_ cubicflux	clean	1	0.1/YN	0.1/NN	0/NN
			S7/100	S7/0.0316	S7/0.0316
inviscid_conservation_ cubicflux	1	1	0/YY	0.01/NN	0/YY
			S7/1e-06	S7/1	S7/10
inviscid_conservation_ cubicflux	5	1	0.3/YN	0.00316/NN	0/NN
			S7/0.1	S7/31.6	S7/0.000316
inviscid_conservation_ cubicflux	10	1	0/YY	0/NN	0/YN
			F/0.000316	F/0.000316	F/31.6
inviscid_conservation_ sinflux	clean	1	0.0316/YN	0/YN	0.0316/NN
			S7/0.000316	S7/0.000316	S7/0.001
inviscid_conservation_ sinflux	1	1	0/YY	0/YY	0.3/YY
			S7/0.001	S7/0.001	S7/3.16
inviscid_conservation_ sinflux	5	1	0/YY	0/YY	0/NN
			S7/0.000316	S7/0.000316	S7/1e-05
inviscid_conservation_ sinflux	10	1	0/YN	0/YN	0/YN
			S7/3.16e-06	S7/3.16e-06	S7/0
kdv	clean	2	0/YN	0/YN	0/NN
			S7/0.00316	S15/0.00316	S15/0.01
kdv	1	2	0/NY	0/NY	0/NY
			S15/0.001	S15/0.01	S15/0.01
kdv	5	2	0.3/NN	0/NY	0/YY
			S15/0	S15/0	S15/0.01
kdv	10	2	0/YN	0/YN	0/NY
			F/0.01	F/0.01	F/0.01
ks	clean	3	0/NN	0/NN	0/NN
			S15/1	S7/31.6	S7/3.16
ks	1	3	0/NN	0.0949/NY	0.0316/YY
			S15/31.6	S7/0.1	S7/0
ks	5	3	0.0949/NY	0.1/YY	0.1/NN
			S15/0.001	S15/10	S15/1
ks	10	3	0/YN	0.3/NY	0.0316/YY
			S7/0	S7/0	S7/0
lorenz96	clean	*	0.00333/NN	0/NN	0/NN
			S7/0	S7/0	S7/0
lorenz96	1	*	0.01/NN	0/NY	0/NY
			S7/0.01	S7/0	S7/0
lorenz96	5	*	0.0333/YY	0/YN	0/YN
			S7/0.0333	S7/0	S7/0
lorenz96	10	*	0.0333/YY	0/YN	0/NN
			F/10	F/10	F/10
porous_medium	clean	2	0/NN	0/NN	0/NN
			S15/0.316	S15/0.0001	S15/0
porous_medium	1	2	0.1/YY	0/YY	0/YY

Table 6 continued

Family	Noise	K	Level 1	Level 2	Level 3
			S15/1	S15/0.01	S15/0.01
porous_medium	5	2	0/YN	0/YY	0/YY
			S15/1	S15/0.0316	S15/0.0316
porous_medium	10	2	0.00949/YN	0/YY	0/YY
			S7/3.33e-05	F/3.33e-05	F/1e-05
predator_prey	clean	*	0.1/YN	0/YY	0/YY
			S7/100	S7/0.001	S7/0.0003
predator_prey	1	*	0/NN	0/YY	0/YY
			S7/10	S7/3.33	F/3.33
predator_prey	5	*	0.01/NN	0/YN	0/YN
			F/0.0001	F/3.33e-05	S7/3.33
predator_prey	10	*	9/NN	3/YN	0/YN
			S7/0.001	S7/0.0001	S7/0.01
transport_reaction	clean	*	0.0333/YN	0.1/YN	0/YY
			S7/0.1	S7/0.001	S7/0.01
transport_reaction	1	*	0.01/YY	0.1/YN	0/YY
			S7/0.0001	S7/0.0001	S7/0.001
transport_reaction	5	*	0.1/YN	1/NY	0/YY
			S7/1	S7/0.001	S7/0.01
transport_reaction	10	*	0.03/YY	0.1/YN	0.3/YY
			S15/0.1	S15/0.1	S15/0.1
wave	clean	1	0/YN	0/YN	0/YN
			S15/0.00316	S15/0.0001	S7/3.16
wave	1	1	0.949/NY	9.49/NN	0/YN
			S15/0.0001	S15/0.0001	S7/3.16
wave	5	1	1/YN	9.49/NN	0/YN
			S15/0.0001	S15/0.0001	S7/3.16
wave	10	1	9.49/NN	9.49/NN	0/YN

B.4 WSINDy TRANSLATION: THE SAME CANDIDATES IN WEAK FORM

WSINDy is a separate discovery method under the same method-independent prior cards in Appendix B.1. Its admissible *mathematical* terms are exactly those specified for pointwise SINDy in Appendix B.2: the scalar F, S, G, X_F, X_S blocks, the twelve scalar Level 3 menus in Table 5, and the ODE and two-dimensional menus there. Thus the scalar Level 1 has 67 candidate terms (68 for wave), and the three Level 2 sizes are 12, 23 (24 for wave), or 34, before a constitutive Level 3 subset is selected. The ODE and two-dimensional candidate counts likewise match their SINDy cards. Biochemical Level 2 equals Level 1 because the disclosed information does not support an independent reaction-graph restriction. These are fixed prior choices, not terms selected by inspecting held-out results.

The numerical translation changes the regression equations. Native PySINDy `WeakPDELibrary` integrates scalar pure derivatives, whole conservative flux derivatives, sources, coordinate-dependent fluxes and the temporal target against test functions on sampled space–time supports. A candidate such as $D_x(u^2 D_x u)$ remains one column with one coefficient, rather than a set of independently fitted product-rule terms. The identity $D_x(u^k D_x u) = D_x^2(u^{k+1})/(k+1)$ gives native pure-derivative integration for the nonlinear-diffusion columns. Coordinate fluxes such as $D_x(a(x)u)$ retain their coefficient tie. The independent gradient-square block $u^k(D_x u)^2$ uses a weak pure-derivative column together with a native interaction column whose inner gradient is evaluated by PySINDy's spatial finite differences. For the 25 wave systems, the target is a native weak second temporal derivative and the optional u_t damping term is a weak-integrated RHS column. First-order temporal targets use native `convert_u_dot_integral`. No pointwise temporal derivative is used to fit the weak coefficients.

For the ODE extension, native `CustomLibrary`/`GeneralizedLibrary` evaluates the same algebraic terms as SINDy; native `WeakPDELibrary` then integrates those fields and constructs the temporal target. Lorenz–96 Level 2 and Level 3 share a local law across cyclic sites. For the two-dimensional systems, native weak integration covers the outer flux and pure derivatives; nonlinear fluxes and independent gradient products require native pointwise spatial differences for their inner gradients. The cylinder's weak supports lie in a fixed valid rectangle, $x[40::4], y[::2], t$[phase start :

phase end : 2], so they avoid the obstacle mask. Held-out scoring uses the full valid grid. This training crop and the inner-gradient approximation can make WSINDy's numerics differ from pointwise SINDy's even though their candidate mathematics and disclosures coincide. At fixed data and weak geometry, the Level 3 weak columns are a literal subset of Level 2, itself a subset of Level 1; independently selected geometries need not yield identical numerical columns across levels.

B.5 WSINDy TUNING AND EVALUATION CONTRACT

The policy-selection unit remains physical family $\times$ noise condition $\times$ prior level. All trajectories in a group share one numerical policy, then estimate their own coefficients and supports. The 318 scalar systems produce 216 selected policies, and the remaining labeled systems produce 66, for 282 total. Table 7 groups these into 94 family–noise rows with all three levels side by side. `methods/wsindy/selected_family_policies.csv` retains full precision. As with SINDy, the scalar and expanded experiments are two frozen strata with different resource menus; their policy rows should not be interpreted as one uniform grid search.

WSINDy's native weak-support count K, support-width fraction h, and test-function exponent p replace SINDy's pointwise differentiation choice. Here K counts integration supports and is distinct from the true-term-count cap K_{GT}. The remaining tuned settings are ridge regularization α, nominal STLSQ threshold τ, column normalization and unbiasing. The scalar search uses $K = 100$, $p = 4$, two support widths, five ridge values, ten thresholds, and both binary flags: 400 coarse configurations per group followed by one local refinement. Its maximum STLSQ iterations are 20. The ODE extension uses $p = 2$ and a 72-configuration ordinary menu; the two-dimensional extension uses $p = 4$ and a 54-configuration menu. These vary weak geometry, α, τ, and normalization, with maximum iterations 50 and unbiasing off. The biochemical systems use a documented bounded threshold menu, calibrated with a nonzero-fit requirement, because a nearly stationary target otherwise made an all-zero family law appear favorable. The same biochemical menu and selection rule apply at every level. Selected K varies from 40 to 300 in the extension. The scalar and extension definitions of h refer to saved space–time and temporal support fractions, respectively.

The original HDF5 fitting prefix is split 60/40 for inner fit and validation; the held-out final suffix does not select a policy. Scalar families use their predeclared calibration trajectories and minimize the family mean log clean-reference validation NMSE with failures penalized. The extension uses fixed pointwise validation NMSE rather than a weak residual. Its biochemical family uses median log validation NMSE across predeclared channels and requires a nonzero calibration fit. As in the SINDy protocol, clean inner-validation observations are available even when fitting noisy paired observations; system 0331 has only 5%-noise data and uses a matching-noise reference. This calibration privilege is disclosed in Appendix B.3. All final coefficients are fit on the full outer prefix under the frozen policy.

Native STLSQ proposes support; native SSR prunes whole candidate columns only when the disclosed K_{GT} atomic-count cap binds, followed by native restricted STLSQ refitting. The cap supplies the true *number* of atomic terms, never their identities. The two-dimensional extension declares its incompressibility normal form for 0321 and a one-atom allowance for 0322; its 128-proposal-column guard records a resource failure. Literal-zero models and failed targets remain in the structural denominator. The primary reported score is the clean-reference NMSE of Eq. 1. Saved weak coefficients are expanded into the canonical strong-form RHS for this evaluation; the weak training residual is not reported as NMSE. This evaluation translation is shared across WSINDy levels but differs numerically from pointwise SINDy's selected RHS backend. Exact structure requires strict canonical support equality across every target of a system.

Table 7: Selected family-shared WSINDy policies, one row per family and noise, with L1–L3 side by side. Each level cell shows weak support count K and support fraction h, then ridge α and threshold τ, then normalization/unbias flags (Y or N). The exponent p is shared across levels within a row. K_{GT} is a distinct true-atom support cap; $*$ denotes target-specific caps. Scalar h is the saved H_{xt} fraction; extension h is the saved temporal support fraction. Values below one omit a leading zero; full precision is in `methods/wsindy/selected_family_policies.csv`.

Family	Noise	p	K_{GT}	Level 1	Level 2	Level 3
advection1d	clean	4	1	100/.05 1e-6/0 YN	100/.05 1e-6/0 YN	100/.05 1e-6/0 YN
advection1d	1	4	1	100/.1 1e-6/0 NN	100/.1 1e-6/0 NN	100/.1 .01/0 YN
advection1d	5	4	1	100/.05 1e-6/0 NY	100/.05 1e-8/0 YY	100/.05 0/0 NY
advection1d	10	4	1	100/.1 .01/0 YY	100/.1 0/0 NY	100/.1 0/0 YY
biochemical	clean	2	*	100/.08 0/1e-12 NN	100/.08 0/1e-12 NN	100/.08 1e-6/5 YN
biochemical	5	2	*	300/.08 .001/1 YN	300/.08 .001/1 YN	100/.08 .001/1 YN
burgers	clean	4	2	100/.1 0/0 NY	100/.1 0/0 NY	100/.1 0/0 NY
burgers	1	4	2	100/.1 1e-6/.00316 YY	100/.1 1e-6/0 YY	100/.1 0/0 YN
burgers	5	4	2	100/.1 .0316/0 NN	100/.05 .1/.01 YN	100/.1 .01/0 YN
burgers	10	4	2	100/.1 .001/0 YY	100/.1 .1/0 NN	100/.1 0/.03 YN
cahnhilliard_1D	clean	4	4	100/.05 1e-5/0 YN	100/.1 1e-6/0 YN	100/.1 1e-6/0 YN
cahnhilliard_1D	1	4	4	100/.05 3.16e-6/.001 NY	100/.05 1/0 YN	100/.05 1/0 YN
cahnhilliard_1D	5	4	4	100/.1 1e-6/.0316 YN	100/.1 3.16/0 NN	100/.1 3.16/0 NN
cahnhilliard_1D	10	4	4	100/.1 3.16/0 NN	100/.1 1/1e-5 NN	100/.1 .316/.0001 NN
conservation_cubicflux	clean	4	2	100/.05 3.16e-7/.001 YN	100/.1 1e-5/0 YN	100/.1 1e-5/0 YN
conservation_cubicflux	1	4	2	100/.1 1e-5/.000316 YN	100/.1 1e-5/.000316 YN	100/.1 0/0 YY
conservation_cubicflux	5	4	2	100/.1 .001/0 YN	100/.1 3.16e-6/0 YY	100/.1 0/0 YN
conservation_cubicflux	10	4	2	100/.05 .1/0 YN	100/.05 .1/0 YN	100/.1 0/0 YY
conservation_linearflux	clean	4	2	100/.05 3.16e-7/.001 YY	100/.1 1e-7/.00316 YY	100/.05 0/0 YY
conservation_linearflux	1	4	2	100/.1 1e-6/0 NN	100/.1 3.16e-7/0 YN	100/.1 1e-8/0 NN

Table 7 continued

Family	Noise	p	K_{GT}	Level 1	Level 2	Level 3
				100/.1 1e-7/.0316	100/.1 3.16e-6/0	100/.1 0/0
conservation_linearflux	5	4	2	NN	NN	YN
				100/.1 1e-5/0	100/.05 .001/0	100/.1 0/0
conservation_linearflux	10	4	2	NN	YN	NY
				100/.05 3.16e-7/0	100/.05 3.16e-7/0	100/.05 1e-5/0
conservation_sinflux	clean	4	2	YN	YN	YN
				100/.05 .01/0	100/.05 1e-5/0	100/.1 1e-5/0
conservation_sinflux	1	4	2	YN	YN	YN
				100/.05 1e-5/0	100/.1 1e-6/0	100/.05 1e-5/0
conservation_sinflux	5	4	2	NN	YN	YN
				100/.05 .01/0	100/.05 3.16e-6/.0316	100/.05 10/0
conservation_sinflux	10	4	2	YY	NN	NY
				100/.08 .0001/.01	100/.08 .0001/0	100/.08 .0001/0
cylinder_flow	clean	4	*	NN	YN	YN
				100/.08 .0001/.01	40/.08 .01/0	100/.08 .01/0
cylinder_flow	1	4	*	NN	YN	YN
				100/.08 .0001/.01	40/.08 .0001/0	100/.08 .0001/.1
cylinder_flow	5	4	*	NN	YN	NN
				40/.08 0/.01	40/.08 .01/0	100/.08 .0001/.1
cylinder_flow	10	4	*	NN	YN	NN
				100/.1 1e-6/0	100/.05 .001/0	100/.1 1e-8/0
diff_bistablereact_1D	clean	4	4	YY	YY	YN
				100/.1 .000316/0	100/.1 .001/0	100/.1 1e-8/0
diff_bistablereact_1D	1	4	4	YN	YN	YN
				100/.1 .1/0	100/.1 .1/.001	100/.1 .1/0
diff_bistablereact_1D	5	4	4	YY	YY	YY
				100/.1 .0316/0	100/.1 0/0	100/.1 .1/.001
diff_bistablereact_1D	10	4	4	YN	NY	YY
				100/.1 1e-6/0	100/.1 1e-6/0	100/.1 0/0
diff_linearreact_1D	clean	4	2	YY	YY	NN
				100/.1 .01/0	100/.1 .000316/0	100/.1 3.16e-6/0
diff_linearreact_1D	1	4	2	YN	YY	YN
				100/.1 .1/0	100/.1 .01/0	100/.1 .01/0
diff_linearreact_1D	5	4	2	YY	YY	YY
				100/.1 1/0	100/.1 .1/0	100/.1 100/0
diff_linearreact_1D	10	4	2	YY	YN	YY
				100/.05 3.16e-7/.000316	100/.05 3.16e-7/.000316	100/.05 3.16e-6/0
diff_logisreact_1D	clean	4	3	YN	YN	YN
				100/.05 .000316/.001	100/.1 3.16e-7/0	100/.1 3.16e-6/0
diff_logisreact_1D	1	4	3	YY	YN	YN
				100/.05 .316/0	100/.1 .0001/0	100/.1 .0001/0
diff_logisreact_1D	5	4	3	YY	NY	YN
				100/.1 .1/0	100/.05 1/0	100/.1 10/0
diff_logisreact_1D	10	4	3	YY	YY	YY

Table 7 continued

Family	Noise	p	K_{GT}	Level 1	Level 2	Level 3
				100/.05 3.16e-7/0	100/.05 1e-7/0	100/.05 1e-8/0
diff_squarelogisticreact_1D	clean	4	4	YY	YY	YN
				100/.1 .01/0	100/.1 .001/0	100/.1 0/0
diff_squarelogisticreact_1D	1	4	4	YY	YY	YY
				100/.1 .1/0	100/.1 .316/0	100/.1 1/0
diff_squarelogisticreact_1D	5	4	4	YY	YY	NY
				100/.1 .1/0	100/.1 3.16/.000316	100/.1 .0316/0
diff_squarelogisticreact_1D	10	4	4	YN	YY	YN
				100/.05 .001/0	100/.05 1e-5/0	100/.05 1e-5/0
fplanck	clean	4	3	YY	YN	YN
				100/.05 .001/0	100/.1 1e-5/0	100/.1 1e-5/0
fplanck	1	4	3	YY	YN	YN
				100/.1 .0001/0	100/.1 1e-5/0	100/.1 1e-5/0
fplanck	5	4	3	YN	YN	YN
				100/.1 .0001/.000316	100/.1 1e-5/0	100/.1 1e-5/0
fplanck	10	4	3	YN	YN	YN
				40/.05 .0001/0	40/.05 .0001/.1	100/.05 0/0
glider	clean	2	*	NN	NN	YN
				100/.05 .01/0	40/.05 .01/.1	100/.05 .01/0
glider	1	2	*	YN	NN	YN
				40/.05 0/0	40/.05 .01/.01	100/.1 .01/0
glider	5	2	*	YN	NN	YN
				40/.1 0/0	100/.1 0/0	40/.05 .01/0
glider	10	2	*	YN	YN	NN
				100/.05 1e-5/0	100/.05 1e-5/0	100/.05 1e-5/0
heat	clean	4	1	NN	NN	NN
				100/.1 .01/0	100/.1 .01/0	100/.1 .01/0
heat	1	4	1	YN	YN	YN
				100/.1 .0316/0	100/.1 .0316/0	100/.1 .0316/0
heat	5	4	1	YN	YN	YN
				100/.1 .316/0	100/.1 .316/0	100/.1 .316/0
heat	10	4	1	NN	NN	NN
				100/.1 .0001/0	100/.1 .0001/0	100/.1 .0001/0
inviscid_conservation_cubicflux	clean	4	1	NN	NN	NN
				100/.05 .00316/0	100/.05 .00316/0	100/.05 .00316/0
inviscid_conservation_cubicflux	1	4	1	YN	YN	YN
				100/.1 3.16e-6/.1	100/.1 3.16e-6/0	100/.1 3.16e-6/0
inviscid_conservation_cubicflux	5	4	1	NN	NN	NN

Table 7 continued

Family	Noise	p	K_{GT}	Level 1	Level 2	Level 3
inviscid_conservation_cubicflux	10	4	1	100/.1 1e-5/.00316 NN	100/.1 1e-6/0 NN	100/.1 1e-5/.00949 NN
inviscid_conservation_sinflux	clean	4	1	100/.1 .001/0 NY	100/.1 .001/0 NY	100/.1 .001/0 NY
inviscid_conservation_sinflux	1	4	1	100/.1 .001/0 NY	100/.1 .001/0 NY	100/.1 .001/0 NY
inviscid_conservation_sinflux	5	4	1	100/.1 1e-5/0 NN	100/.05 3.16e-6/.00316 YN	100/.1 1e-5/0 NN
inviscid_conservation_sinflux	10	4	1	100/.1 .01/0 YN	100/.1 .01/0 YN	100/.1 3.16e-7/.0316 NN
kdv	clean	4	2	100/.1 1e-6/0 NN	100/.1 1e-6/0 NN	100/.1 .001/0 YN
kdv	1	4	2	100/.05 1e-6/0 NN	100/.05 1e-6/0 NN	100/.05 .001/0 YN
kdv	5	4	2	100/.05 1e-5/0 YN	100/.1 1e-6/0 NY	100/.1 3.16e-6/0 YN
kdv	10	4	2	100/.05 .01/0 YN	100/.05 .1/0 YN	100/.1 .001/0 NN
ks	clean	4	3	100/.05 .0001/0 YN	100/.05 .0001/0 YN	100/.05 .0001/0 YN
ks	1	4	3	100/.05 .01/0 NN	100/.05 .01/0 NN	100/.05 .01/0 NN
ks	5	4	3	100/.1 1e-6/0 YY	100/.1 0/0 YN	100/.1 0/0 YN
ks	10	4	3	100/.05 .0001/0 YY	100/.05 .0001/0 YY	100/.1 .00316/0 YN
lorenz96	clean	2	*	100/.05 .01/.01 NN	40/.1 .01/0 NN	40/.1 .01/0 NN
lorenz96	1	2	*	100/.1 0/.01 NN	40/.05 .01/0 NN	40/.05 .01/0 NN
lorenz96	5	2	*	100/.05 .01/.1 NN	40/.05 .01/0 NN	40/.05 .01/0 NN
lorenz96	10	2	*	100/.05 .01/.1 NN	40/.05 .01/0 NN	40/.05 .01/0 NN
porous_medium	clean	4	2	100/.05 .00316/0 YN	100/.1 .00316/0 YN	100/.1 .00316/0 YN
porous_medium	1	4	2	100/.1 .001/0 YY	100/.1 .0001/0 NY	100/.1 0/0 NN
porous_medium	5	4	2	100/.1 .001/0 NY	100/.1 .0001/0 NY	100/.1 0/0 NN

Table 7 continued

Family	Noise	p	K_{GT}	Level 1	Level 2	Level 3
porous_medium	10	4	2	100/.1 .0316/0 YY	100/.1 .01/0 YY	100/.1 .01/0 YY
predator_prey	clean	2	*	100/.05 .01/0 NN	40/.1 .0001/0 YN	40/.05 .0001/0 NN
predator_prey	1	2	*	40/.1 .0001/.1 YN	40/.1 .0001/0 YN	40/.05 .0001/0 YN
predator_prey	5	2	*	40/.05 .01/.1 NN	40/.1 .0001/0 YN	100/.1 .0001/0 YN
predator_prey	10	2	*	40/.05 .01/.01 NN	100/.05 .01/.1 NN	40/.1 .01/.1 NN
transport_reaction	clean	4	*	100/.08 0/.01 NN	100/.08 0/.1 NN	100/.08 .01/0 YN
transport_reaction	1	4	*	100/.08 0/.01 NN	100/.08 0/.1 NN	100/.08 .01/0 YN
transport_reaction	5	4	*	100/.08 0/.1 NN	100/.08 0/.1 NN	100/.12 .01/0 NN
transport_reaction	10	4	*	100/.08 0/.1 NN	100/.08 0/.1 NN	100/.12 .01/0 NN
wave	clean	4	1	100/.05 .1/0 YN	100/.05 .1/0 YN	100/.05 .1/0 YN
wave	1	4	1	100/.1 .01/0 NY	100/.1 .01/0 NY	100/.1 .1/0 YN
wave	5	4	1	100/.1 .1/0 NN	100/.1 .0316/0 NN	100/.1 1/0 NY
wave	10	4	1	100/.1 1/0 NN	100/.1 1/0 NN	100/.1 1/0 NN

B.6 GPT-5.6-SOL PROMPT AND ATTEMPT PROTOCOL

The discovery agent was GPT-5.6-sol at high reasoning effort. It was given anonymous training observations, an observation schema, and the prior card specified in Appendix B.1. Unlike the SINDy implementations, the prompt did not specify a derivative estimator, candidate library, support count, or even the temporal order of the law. The agent could use one offline `analyze` tool call per attempt, submitting a shell script that could internally fit several candidates. The script could read only `/data/arrays.npz` and `/data/schema.json`; source labels and held-out observations were unavailable. After that one result, it returned either explicit equations and round-trip decimal coefficients in the prescribed JSON schema or `no_equation`. A failed analysis still consumed the call.

The following is the frozen Level 3 scalar prompt structure with the two case-card paragraphs replaced by placeholders. At Level 2, the Level 3 block is omitted; at Level 1, both blocks are omitted and the opening sentence says that no scientific description is supplied. The ODE/2D extension used the same data/tool/output contract with a small opening-paragraph wording change and headings "Additional structural disclosure" and "Additional constitutive disclosure." Thus this rendering is a prompt *template*, not a claim that every worker saw byte-identical introductory text. The template is released as `methods/agents/prompt_template.txt`; the prompts actually sent are in `methods/agents/prompts.json.gz`, and `methods/agents/prepare_all.py` rebuilds them from the dataset.

```
Infer the governing differential equation or system of equations for anonymous
numerical fields. The additional scientific disclosure for this case appears below.
Beyond the scientific disclosures below, no equation family, variable meanings,
known active terms, or reference coefficients are provided.

You have exactly ONE call to the `analyze` tool. Submit a complete shell script
that performs your chosen numerical analysis and prints the evidence you need
to propose the equation. You may use any available computational method or
develop your own. Internal loops, optimization, fitting several candidates,
subsampling, and numerical model selection inside this one execution are
allowed. Beyond the supplied scientific disclosures, no particular hypothesis
space, derivative estimator, polynomial degree, temporal derivative order, or
fitting algorithm is prescribed.

The tool runs your script offline in a fresh environment with Python, NumPy,
SciPy, SymPy and other installed system tools. It exposes `/data/arrays.npz`
and `/data/schema.json`, plus writable `/work` and `/tmp`. The NPZ contains
training observations only: `u` has spatial grid axes, then time, then anonymous
target fields; `t` is the time coordinate; `x0`, `x1`, ... are spatial coordinate
vectors. Optional `a` contains additional anonymous observed channels on the
same grid. The schema gives only layout, shapes, dtypes and generic channel
names. An optional boolean `mask` on the space-time grid marks valid sample
locations with true. Load arrays without pickle. Source metadata and evaluation data are
unavailable. Read the schema and numerical data within your one analysis script.

The script should print concise numerical findings, candidate equations with
their fitted coefficients, or other evidence you choose. Its combined output is
limited to 64,000 characters. Choose output informative for your final equation.
Do not request raw source files, reference answers, scientific type labels,
external web access or another case.

After receiving that ONE tool result, directly return your final equation(s)
using the required JSON schema. Do not make another analysis call, repair or
rerun code based on the output, or execute a separate final fit. An unsuccessful
analysis still consumes the call. There is no 30-second cutoff: think and compute
as needed within the environment's generous emergency safeguards.

Use target names `u0`, `u1`, ... and auxiliary names `a0`, `a1`, ... . Write
equations explicitly enough to evaluate, defining any derivative/custom notation
inside the equation strings. Include all fitted constants and serialize their
values as round-trip decimal strings. Do not assume the governing relation is
first-order in time. Return `no_equation` if the one result does not support a
defensible equation. Save only the equations, coefficients, status and a few
method keywords; omit derivations, reasoning transcripts and procedural prose.

Additional disclosure (Level 2)
{level2_prior}

Additional disclosure (Level 3)
{level3_prior}
```

Each clean scalar system–level slot had up to six recorded attempts; the 1% and 5% scalar slots had up to six, and the 10% scalar slots up to five. The 14 labeled extensions had one terminal attempt per available noise–level slot. Retries targeted cases without an accepted equation, that is, a well-formed answer with status `ok`, one equation per target and finite decimal coefficients in the prescribed JSON schema; the first accepted submission in chronological attempt order was retained even when a later numerical evaluation was poor. This is a failure-conditioned cumulative protocol, not a fresh single-attempt success estimate, and the unequal attempt budgets limit method-to-method comparisons. Slots used 2.3 attempts on average (1.5 on clean data and 2.4, 2.7 and 2.7 at 1%, 5% and 10% noise). No agent-specific hyperparameter sweep or family-shared tuning was conducted. Prompt text and card assignment were frozen across retries; coefficients and numerical procedure were chosen by each agent within its single tool call.

B.7 Claude Opus 5 prompt and attempt protocol

The Claude discovery arm used `claude-opus-5` at high effort in fresh, memory-free one-shot sessions. It received the same frozen per-case prompt, prior card and sanitized numerical bundle as the GPT-5.6-sol arm. The prompt structure and the level-specific placeholders are therefore the ones displayed in Appendix B.6; we do not repeat them here. Both agents received the prompts released in `methods/agents/prompts.json.gz`; `scripts/run_method.py --prepare-only` with method `claude` rebuilds each bundle from the dataset. Each attempt permitted one `analyze` call and required the same equation/coefficient JSON response contract.

The model and retry history differ. Each clean scalar and ODE/2D extension slot had one terminal Claude attempt, so Claude's clean rates are single-attempt rates. The scalar 1%, 5% and 10% slots had up to three rounds, retrying only slots without an accepted equation and retaining the first accepted response. No equation was chosen using held-out accuracy. Thus the two agents share the disclosed task inputs, but their aggregate outcomes are not a controlled single-attempt comparison: the GPT arm used up to six scalar attempts at clean, 1% and 5% noise and up to five at 10% noise. The Claude arm had no external family-shared hyperparameter search; any numerical choices were made within its single permitted analysis call.

B.8 LLM-PDESR TRANSLATION AND CLEAN-RUN PROTOCOL

We applied the upstream LLM-PDESR search to the 293 compatible labeled scalar one-dimensional PDEs at clean noise only. Its executable specification keeps the method's fixed evaluator, neutral seed, and evolution instructions at every level. A hash audit of all 879 specifications found one identical executable boilerplate and 1, 3 and 11 distinct opening descriptions at Levels 1–3, respectively; the wave-family Level 3 description is absent because wave is unsupported. The level-dependent text is the scientific disclosure inserted into that description: Level 1 has the observed $u(x, t)$ schema and no extra card; Level 2 inserts the broad structural card; Level 3 inserts that card plus the narrower constitutive-family card. A per-case text audit matched all 879 disclosure descriptions to the cards in Table 4. The cards are translated into this method's source-native specification rather than into a new library or operator. The per-case specifications are not stored separately: the released runner rebuilds each one from the upstream specification text and the benchmark card at run time, and the selected equations are in `results/equations/llmpdesr.jsonl.gz`.

The source-native candidate is Python code for $u_t = F(x, u, D_x u, \ldots)$, using an already supplied spatial derivative operator D_x (orders at most four) and at most six fitted parameters. The executable objective uses ten spatial test windows and a subdomain-weighted residual; it fits candidate coefficients by BFGS starting from ones and ranks candidate error against complexity. For the single reported outcome per case, the saved scorer selects the valid evaluated candidate with the lowest training weighted-residual MSE, before consulting its reference equation or held-out data. Our adapter changes the model backend to `gpt-6-luna` with `xhigh` reasoning and supplies the benchmark case and disclosure; it otherwise uses upstream commit `0b491b1`. All 879 saved run metadata records specify four candidates per prompt, a 30-second candidate-evaluation timeout, a 500-sample cap, and early stopping after 50 consecutive identical candidates. One Level 3 Fokker–Planck run (system 0164) was stopped by the operator at sample 377; its saved best candidate remains the reported outcome. These are method-specific compute choices, not additional information in any prior level.

The upstream target is u_t for one field in one spatial dimension, so the method supports the same 293-system subset as SGA-PDE (Appendix B.1). Our held-out numerical scorer refits each saved selected candidate with the method's own BFGS routine, translates the fitted RHS into benchmark terms, and applies the same clean-reference, 3%-trimmed NMSE as the other arms. It makes no new model calls and does not select a candidate using held-out data.

B.9 SGA-PDE TRANSLATION AND FROZEN SEARCH PROTOCOL

We translate the common prior cards into restrictions on SGA-PDE's native symbolic-tree proposals. Level 1 receives the scalar field and coordinate schema and a broad public differential-expression envelope, without a source or homogeneity fact. Level 2 adds one of the three structural cards in Table 4: source-free and homogeneous, source-allowing and homogeneous, or source-free and coordinate-dependent. Level 3 additionally restricts the tree grammar to the disclosed constitutive family, with alternative fluxes, derivative orders, nonlinear powers, or source terms still available. The family card does not give the active equation or coefficients. The frozen source specification is released as `methods/sga/protocol/cumulative_priors.json`. The implemented grammar is the intersection of these public operator families and SGA-PDE's executable native polynomial tree language; this intersection is narrower than the conceptual envelope. An independent reference-expression audit finds only 223 of the 293 interface-compatible systems representable at Level 1 and 217 at Levels 2 and 3. These limitations are reported with the results rather than treated as search failures.

The search uses upstream SGA-PDE commit `28893ad1`, with population 20, maximum tree depth 4, at most five generated trees plus its fixed u regression term, native STRidge fitting, and fitness $2k + 2\log(\mathrm{MSE}_{\mathrm{train}})$, where k is the number of nonzero fitted coefficients. The frozen `masks300_pure` policy admits up to 300 distinct raw candidate evaluations, ten effective generations, and a 120-second soft search wall; the enclosing job uses a 300-second search timeout and 60-second evaluation timeout. Three fixed random seeds (0, 1 and 2) are run for every compatible system, level and noise condition. The four observation conditions are clean, 1%, 5% and 10% noise. The same clean-selected search policy is transferred to noisy data without noise-specific retuning. Source finite differences are computed after slicing the 70% training prefix, and the remaining 30% is held out. Only the benchmark's shared clean-reference evaluator and structural scorer supply the reported metrics; neither held-out values nor the reference equation choose a tree.

The SGA-PDE interface supports scalar one-dimensional u_t equations, so it runs on the 293-system subset of Appendix B.1; the other 39 labeled systems count as non-recoveries. The five unlabeled systems are outside the structural benchmark. The frozen search options are in `configs/sga_search_options.json`, and one record per production run, with its status and saved scores, is in `results/equations/sga.jsonl.gz`.

B.10 DISCOVER TRANSLATION AND SEARCH SETTINGS

DISCOVER builds an expression tree for one temporal target. We use one seed (0) per saved target and noise condition. The saved Level 1 sweep uses a generic native token set, not the frozen finite-stem grammar used by the later Level 2 and Level 3 sweep. Its one-dimensional PDE tokens are `add, mul, div, diff, diff2, diff3, n2, n3`; the ODE set removes the three spatial differentiation tokens. At clean Level 1, the adapter runs one job per target of a multivariate ODE, but does not support two-dimensional PDEs. The clean Level 1 runs cover 308 systems (406 target records). The noisy Level 1 runs use the PINN path described below, which supports only scalar one-dimensional PDEs; they were attempted on 297 such systems per noise tier, of which 282, 291 and 288 completed at 1%, 5% and 10% noise; of the 30 failed runs, 22 reached the 1,800-second limit and 8 stopped with software or GPU-memory errors. Systems without a noisy Level 1 result, including the ODE and two-dimensional systems, count as failures. The Level 1 adapter also fits u_t for the 25 wave systems 0267–0291, whose reference target is u_{tt}, so Level 1 cannot recover them; the Level 2/3 adapters fit u_{tt} and offer u_t as a candidate stem.

The later Level 2/3 translation is defined for all 318 labeled scalar one-dimensional PDEs. Let $D = \partial_x$ and $P = \{x, x^2, \sin(kx), \cos(kx) : k = 1, \ldots, 10\}$. The frozen candidate stems are

Block	Candidate stems	Count
F	$D(u^j),\ j = 1{:}4;\ D(\sin u), D(\cos u);\ D(u^j Du),\ j = 1{:}3;\ D^2u, D^3u, D^4u$	12
S	$1, u, u^2, u^3, u^4, \sin u, \cos u$	7
G	$(Du)^2, u(Du)^2, u^2(Du)^2, u^3(Du)^2$	4
X_F	$D(pu),\ p \in P$	22
T	u_t (only for u_{tt} targets)	1

Level 2 supplies F for 173 homogeneous source-free cases; $F + S + G$ for 100 source-allowing cases; $F + S + G + T$ for the 25 wave cases; and $F + X_F$ for the 20 spatially inhomogeneous cases. The corresponding stem counts are 12, 23, 24 and 34. Level 3 narrows each system to the following constitutive card while retaining alternatives:

Card	Systems	Stems	Exact alternatives
H	51	2	Du, D^2u
PV	26	5	$D(u^j),\ j = 1{:}4;\ D^2u$
PI	10	4	$D(u^j),\ j = 1{:}4$
TV	25	4	$Du, D(\sin u), D(\cos u), D^2u$
TI	25	3	$Du, D(\sin u), D(\cos u)$
R	100	6	$1, u, u^2, u^3, u^4, D^2u$
FP	20	24	Du, D^2u and all X_F stems
CH	9	5	$D^2u, D(u^jDu),\ j = 1{:}3;\ D^4u$
W	25	5	$1, u, u_t, Du, D^2u$
PM	25	4	$D^2u, D(u^jDu),\ j = 1{:}3$
KD	1	5	$D(u^j),\ j = 1{:}4;\ D^3u$
KS	1	6	$D(u^j),\ j = 1{:}4;\ D^2u, D^4u$

These are complete native prefix traversals for candidate *stems*. The `CandidateGrammarPrior` masks next-token choices during recurrent tree generation. It allows the same seven composition operators `add`, `mul`, `n2`, `n3`, `n4`, `sin`, `cos` and at most five composition nodes at both levels. A discovered expression may therefore multiply or transform entire stems; a source-free card does not enforce a globally conservative final expression. Level 3 stems are a subset of their Level 2 stems, but Level 1 is not a superset of this later grammar: its published token list lacks fourth derivatives and trigonometric tokens. The case-to-card assignment and the candidate grammar are released in `methods/discover/tier_candidates.py` and `methods/discover/tier_grammar.py`. Their cards were curator-authored with benchmark-family knowledge; no active-term mask or coefficient is passed to DISCOVER at fit time.

For clean observations, all three levels use DISCOVER Mode 1 and its native finite-difference derivatives. The frozen controller has learning rate 0.0025, entropy weight 0.03, entropy decay 0.7, attention and priority-queue training enabled, queue size 10, and no GP meld. The task uses `pde_reward` with metric parameter 0.01, threshold 5×10^{-4}, no variance normalization and no added reward noise. Level 1 uses its generic length/repetition priors; Level 2/3 replace those with the candidate grammar and a soft length prior of location 10 and scale 5. Mode 1 samples 50,000 expressions in batches of 500, with exploration fraction 0.02, one core per batch and early stopping disabled. The clean Level 2/3 run directories show all 636 scheduled cases completed with a final expression.

Noisy observations use automatic differentiation through a PINN at every level. At Level 1, the runner's own PINN switch keeps the generic token list in its PyTorch form and uses a tanh network with eight layers and width 32, 25% sampled measurements, 50,000 collocation points, four PINN/discovery rounds, 1,000 PINN epochs, learning rate 0.001, and 30,000 sampled expressions in the first search and 10,000 in each of the four later searches, with batch size 1,000 and exploration fraction 0.01. The noisy Level 2/3 campaign uses DISCOVER Mode 2. Its frozen configuration uses a tanh network with eight layers and width 20, 4% sampled measurements, 50,000 collocation points, three PINN/discovery rounds, 1,000 PINN epochs, learning rate 0.001, and 20,000 initial sampled expressions with batch size 1,000 and exploration fraction 0.01; later rounds follow the native 10,000-sample rule. It retains the same controller and task reward settings. The noisy adapter sets the known u_{tt} temporal target for the 25 wave systems and uses u_t as a measured auxiliary stem. No clean field, held-out suffix, or reference expression is supplied to discovery. There is no documented family- or case-level hyperparameter selection for DISCOVER. All 1,908 scheduled noisy Level 2/3 slots have result records, but 269 ended without a final expression.

Setting	L1 clean	L1 noisy	L2/3 clean	L2/3 noisy
Differentiation	finite difference	PINN autodiff	finite difference	PINN autodiff
Initial samples	50,000	30,000	50,000	20,000
Batch size / exploration	$500/0.02$	$1{,}000/0.01$	$500/0.02$	$1{,}000/0.01$
Candidate prior	generic tokens	generic tokens	stem grammar	stem grammar
Neural field model	none	8×32 tanh	none	8×20 tanh
Seed / reward threshold	$0/5 \times 10^{-4}$	$0/5 \times 10^{-4}$	$0/5 \times 10^{-4}$	$0/5 \times 10^{-4}$

Because the Level 1 grammar, its noisy PINN settings and its coverage differ from those of Levels 2/3, a change between Level 1 and Level 2 cannot be attributed to prior information alone. The clean Level 2-to-Level 3 contrast uses the same numerical protocol, composition grammar and candidate-stem subset rule. The noisy Level 2-to-Level 3 contrast also holds its numerical protocol fixed. The frozen settings are in `methods/discover/configs.py`, and the saved expressions of all three levels are in `results/equations/discover.jsonl.gz`.

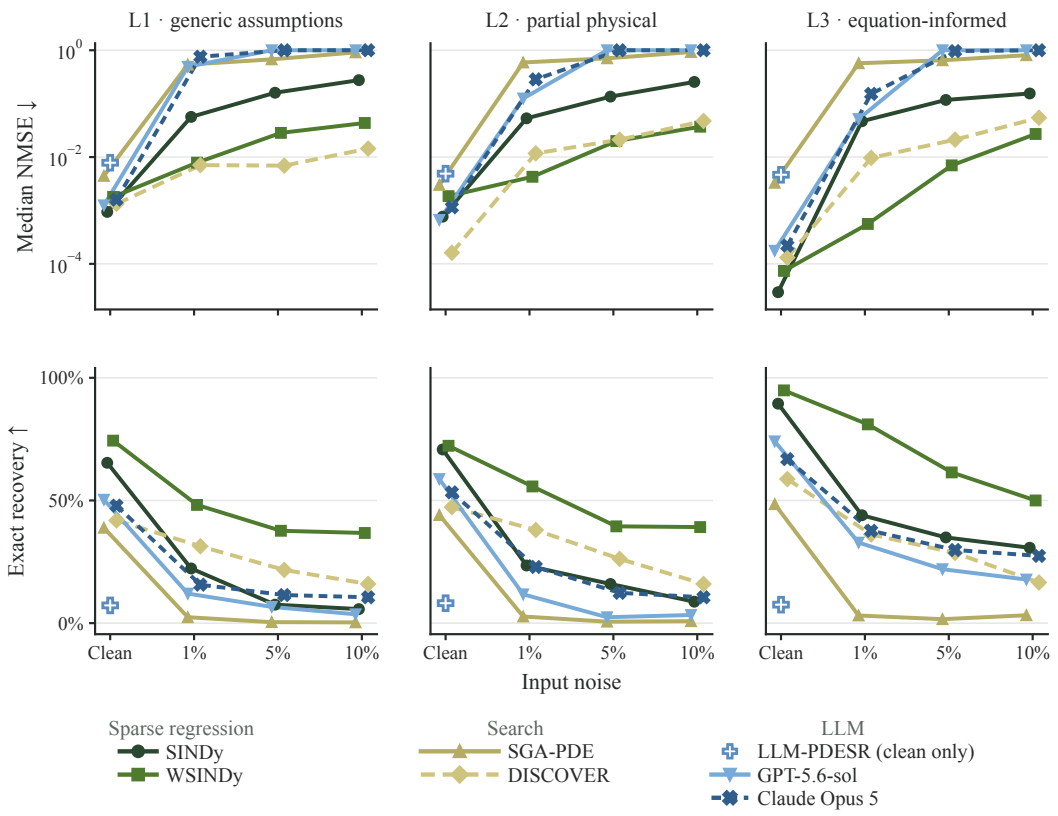


Figure 5: Field-input equation discovery at the measured noise tiers. Curves join categorical tiers, with small horizontal offsets to reveal ties. **Top:** Median held-out clean-reference NMSE (Section 3.4), after discarding the largest 3% of pointwise squared residuals within each target. Every method uses the same systems with observations and clean references: 331/330/331/330 at clean/1%/5%/10% noise. Unsupported, failed, or unscoreable outcomes receive NMSE one; finite evaluated values above one are retained. **Bottom:** Percentage of all 332 labeled systems whose complete equation structure is recovered exactly. The 39 systems unsupported by SGA-PDE and LLM-PDESR count as non-recoveries, capping their recovery at 293/332 (88.3%) from interface coverage alone. SGA-PDE aggregates three seeds per system by the median for NMSE and the mean for recovery. LLM-PDESR has clean-data results only. Among these runs, WSINDy retains the highest exact-recovery rate at 10% noise at every prior level.

C Additional Experimental Results

Figure 5 shows the main results with the noise tier on the horizontal axis, and Table 8 lists the plotted values.

C.1 Case-Study Records

The four tables below give the full records of the case studies in Section 4.4.

Table 8: Values plotted in Figures 2 and 5. NMSE is the median held-out clean-reference NMSE over 331/330/331/330 systems at clean/1%/5%/10% noise, with failures scored 1. Exact is exact structural recovery as a percentage of all 332 labeled systems; for SGA-PDE, of the 996 seed–system trials. Scored counts the systems with a finite evaluated NMSE; for SGA-PDE, it counts seed runs out of the 879 compatible runs. LLM-PDESR has clean-data results only; its selected equations contain every reference term for 103, 133 and 235 of its 293 supported systems at L1–L3. The per-system outcomes are in `results/outcomes.csv` of the released code, and these values in `results/paper_values.csv`.

		Median NMSE			Exact (%)			Scored		
Method	Noise	L1	L2	L3	L1	L2	L3	L1	L2	L3
SINDy	Clean	9.4×10^{-4}	7.6×10^{-4}	2.9×10^{-5}	65.4	70.8	89.5	330	330	331
	1%	5.6×10^{-2}	5.3×10^{-2}	4.7×10^{-2}	22.3	23.5	44.0	330	330	330
	5%	1.6×10^{-1}	1.4×10^{-1}	1.2×10^{-1}	7.5	16.0	34.9	331	331	331
	10%	2.8×10^{-1}	2.6×10^{-1}	1.6×10^{-1}	5.7	8.7	30.7	330	330	330
WSINDy	Clean	1.8×10^{-3}	1.9×10^{-3}	7.4×10^{-5}	74.4	72.3	94.9	331	331	330
	1%	7.9×10^{-3}	4.3×10^{-3}	5.6×10^{-4}	48.2	55.7	81.0	330	330	330
	5%	2.8×10^{-2}	2.0×10^{-2}	7.0×10^{-3}	37.7	39.5	61.4	330	330	330
	10%	4.3×10^{-2}	3.7×10^{-2}	2.7×10^{-2}	36.7	39.2	50.0	330	330	330
SGA-PDE	Clean	4.4×10^{-3}	3.0×10^{-3}	3.3×10^{-3}	38.9	44.1	48.5	876	879	879
	1%	5.4×10^{-1}	5.9×10^{-1}	5.7×10^{-1}	2.4	2.7	3.1	879	879	878
	5%	6.7×10^{-1}	7.1×10^{-1}	6.5×10^{-1}	0.4	0.6	1.6	866	875	871
	10%	9.1×10^{-1}	9.3×10^{-1}	8.1×10^{-1}	0.3	0.8	3.2	857	878	858
DISCOVER	Clean	1.3×10^{-3}	1.6×10^{-4}	1.3×10^{-4}	41.9	47.3	58.7	283	318	318
	1%	7.1×10^{-3}	1.2×10^{-2}	9.7×10^{-3}	31.3	38.0	36.1	260	287	259
	5%	6.9×10^{-3}	2.1×10^{-2}	2.1×10^{-2}	21.7	26.2	28.6	267	288	273
	10%	1.4×10^{-2}	4.7×10^{-2}	5.4×10^{-2}	16.0	16.0	16.6	265	275	257
LLM-PDESR	Clean	7.8×10^{-3}	4.9×10^{-3}	4.7×10^{-3}	7.2	8.1	7.5	293	293	293
GPT-5.6-sol	Clean	1.2×10^{-3}	6.6×10^{-4}	1.8×10^{-4}	50.3	58.7	74.1	295	306	318
	1%	4.9×10^{-1}	1.2×10^{-1}	5.1×10^{-2}	12.0	11.7	32.8	198	210	270
	5%	1	1	1	6.6	2.4	22.0	121	122	188
	10%	1	1	1	3.6	3.3	17.8	79	93	167
Claude Opus 5	Clean	1.6×10^{-3}	1.1×10^{-3}	2.2×10^{-4}	47.9	53.3	66.9	300	304	317
	1%	7.6×10^{-1}	2.9×10^{-1}	1.5×10^{-1}	15.7	22.9	37.7	260	272	317
	5%	1	1	9.6×10^{-1}	11.4	12.3	29.8	214	242	313
	10%	1	1	1	10.5	10.5	27.4	190	207	299

Table 9: Family-level robustness of exact structural recovery. Macro exact recovery is the unweighted mean, over 19 equation families, of each family's exact-recovery rate over all of its labeled systems; the families are the rows of Table 1, with the two Lorenz–96 systems forming one family, and 9 of them have only one or two systems. Unavailable tiers and unsupported, failed or unscoreable systems count as non-recoveries. Pooled L3−L1 is the difference of the 332-system rates in Figure 5. Brackets are 95% percentile intervals from 10,000 bootstrap resamples of families with replacement (seed 0); L1 and L3 are evaluated on the same resampled families, and a resampled pooled rate divides exact recoveries by the resampled system count. SGA-PDE averages the binary outcomes of its three seeds per system; LLM-PDESR has clean data only.

		Macro exact (%)			L3−L1 (pp), paired family-bootstrap 95% CI	
Method	Noise	L1	L2	L3	Pooled	Macro
SINDy	Clean	58.6	57.4	65.6	+24.1 [−3.5, +38.7]	+7.0 [−0.6, +15.5]
	1%	20.4	28.7	38.2	+21.7 [+3.9, +45.1]	+17.8 [+5.0, +33.3]
	5%	6.2	14.7	29.6	+27.4 [+8.1, +57.1]	+23.4 [+8.8, +40.5]
	10%	5.4	9.9	26.5	+25.0 [+8.1, +50.0]	+21.2 [+8.1, +36.6]
WSINDy	Clean	59.1	63.8	73.1	+20.5 [+1.8, +37.5]	+14.0 [+1.5, +29.8]
	1%	41.2	50.5	60.8	+32.8 [+1.0, +50.2]	+19.6 [+5.7, +35.8]
	5%	35.3	40.6	57.1	+23.8 [+6.4, +43.0]	+21.8 [+7.0, +39.8]
	10%	29.7	40.9	54.4	+13.3 [+1.5, +29.7]	+24.7 [+7.8, +44.3]
SGA-PDE	Clean	28.0	33.5	35.2	+9.6 [+1.3, +18.3]	+7.2 [+2.2, +13.2]
	1%	6.3	6.9	7.2	+0.7 [−2.3, +6.3]	+1.0 [−0.6, +3.3]
	5%	0.2	0.4	6.2	+1.2 [−0.2, +4.8]	+5.9 [−0.0, +17.2]
	10%	0.2	2.2	7.3	+2.9 [0.0, +9.6]	+7.1 [0.0, +18.9]
DISCOVER	Clean	24.6	33.5	36.0	+16.9 [+5.9, +29.3]	+11.4 [−6.6, +30.2]
	1%	16.4	16.6	20.8	+4.8 [−15.2, +23.9]	+4.4 [−1.4, +10.5]
	5%	11.6	10.9	16.2	+6.9 [−5.8, +21.0]	+4.6 [−0.2, +10.5]
	10%	4.2	7.0	7.2	+0.6 [−11.5, +18.0]	+3.0 [−1.7, +9.1]
LLM-PDESR	Clean	1.3	1.4	1.8	+0.3 [−16.1, +20.0]	+0.5 [−3.8, +5.3]
GPT-5.6-sol	Clean	41.3	47.2	57.5	+23.8 [+10.4, +39.5]	+16.2 [+5.5, +27.7]
	1%	13.5	23.5	29.3	+20.8 [+2.3, +40.9]	+15.7 [−4.4, +34.5]
	5%	16.9	6.2	32.6	+15.4 [+2.5, +29.3]	+15.7 [−3.3, +34.5]
	10%	12.6	19.4	23.0	+14.2 [+4.3, +24.7]	+10.4 [−6.8, +26.4]
Claude Opus 5	Clean	54.6	53.8	60.9	+19.0 [+10.9, +27.8]	+6.4 [−8.3, +17.8]
	1%	33.0	37.7	41.4	+22.0 [+10.8, +34.1]	+8.4 [−12.8, +27.6]
	5%	26.2	25.0	28.1	+18.4 [+8.7, +27.9]	+2.0 [−17.0, +18.4]
	10%	30.2	29.1	31.6	+16.9 [+8.5, +25.6]	+1.4 [−17.6, +17.7]

Table 10: Ablation of the true-term-count cap K_{GT} on clean data for the 318 scalar PDE systems. Cap: the frozen policies of the main results, with the cap. Frozen: the same policies without the cap. Retuned: policies reselected without the cap by the same grid search on the training prefix (same grid, inner 60/40 split and validation criterion; no held-out data), then refit and scored on the held-out final 30%. Exact recovery is over the 318 systems, and median NMSE is the clean-reference NMSE of Section 3.4. Terms count canonical atoms. The last column counts systems recovered exactly with the cap whose retuned uncapped fit has a lower NMSE but the wrong terms. With the cap on, the rerun reproduces the per-system outcomes and coefficients of the main results exactly. For example, without the cap, SINDy's frozen L2 fit to the conservation law 0028 grows from two to eight terms and loses its exact structure while its NMSE falls from 5.71×10^{-5} to 5.03×10^{-6}.

		Exact (%)			Median NMSE		Median terms		Lower NMSE,
Method	Level	Cap	Frozen	Retuned	Cap	Retuned	Cap	Retuned	wrong terms
SINDy	L1	67.9	0.0	13.5	8.13×10^{-4}	1.73×10^{-5}	2	39.5	125
	L2	73.3	6.6	17.0	5.79×10^{-4}	5.11×10^{-6}	2	12.5	152
	L3	92.8	8.2	42.5	2.58×10^{-5}	1.62×10^{-5}	2	4	97
WSINDy	L1	77.7	5.7	6.0	1.35×10^{-3}	2.33×10^{-4}	2	41	119
	L2	74.8	0.9	9.4	1.73×10^{-3}	1.85×10^{-5}	2	14	171
	L3	97.2	7.9	25.8	5.09×10^{-5}	4.02×10^{-5}	2	4	140

Table 11: Card-swap case study on clean data. Each system receives the prior cards of a donor from another constitutive family: 0079 (correct L3 card PV, viscous transport with polynomial flux) receives those of 0003 (H, linear transport and diffusion), and 0139 (correct L3 card R, reaction–diffusion with polynomial reaction) those of 0292 (PM, conservative nonlinear diffusion with reaction excluded). At L2, 0139 moves from S (spatial coupling with local sources) to C (source-free balance law). Excl. states whether the outcome contains a true term that its card excluded (n/a for correct cards). SINDy and WSINDy run once per card. The agents run three swapped repeats (r1–r3) at L3 only; their correct-card controls are the saved main-benchmark L3 answers. Terms are compared after expansion into atoms. $\mathcal{P}_4 = \{uu_{xx}, u_x^2, u^2u_{xx}, uu_x^2, u^3u_{xx}, u^2u_x^2\}$ are the atoms that $\partial_{xx}\Phi(u)$ adds to u_{xx} for a quartic potential Φ. NMSE is the clean held-out temporal-target NMSE. †Failed outcomes (no equation, no analysis call, or a wrong time order, which is not scored) receive NMSE 1 and are not exact. At L3, swapped cards give 0/4 exact outcomes for the sparse methods and 0/12 for the agents; correct cards give 4/4 for each. No outcome contains a true term that its card excluded. Two of the three Claude Opus 5 swaps on 0079 add uu_x, a nonlinear term that the H card excludes, but not the true u^2u_x. The L2 swap on 0079 is a null intervention because 0003 and 0079 share the same L2 card, so those runs are omitted. On 0139, the L2 swap gives 0/2 exact outcomes versus 1/2 with the correct card.

Method	Card	Exact	Excl.	Extra (+) and missing (−) terms	NMSE
0079, L3: $u_t = -0.986\,u^2u_x + 0.00309\,u_{xx}$; the swapped card H excludes u^2u_x.					
SINDy	PV	yes	n/a	none	3.39×10^{-4}
SINDy	H	no	no	$+u_x$; $-u^2u_x$	0.109
WSINDy	PV	yes	n/a	none	3.17×10^{-4}
WSINDy	H	no	no	$+u_x$; $-u^2u_x$	0.136
GPT-5.6-sol	PV	yes	n/a	none	5.06×10^{-4}
	H, r1	no	no	$+u_x$; $-u^2u_x$, $-u_{xx}$	0.171
	H, r2	no	no	no equation	$1^\dagger$
	H, r3	no	no	$+u_x$; $-u^2u_x$, $-u_{xx}$	0.180
Claude Opus 5	PV	yes	n/a	none	6.18×10^{-4}
	H, r1	no	no	$+uu_x$, $+u_x$; $-u^2u_x$	0.947
	H, r2	no	no	$+uu_x$, $+u_x$; $-u^2u_x$	6.58×10^{-3}
	H, r3	no	no	$+u_x$; $-u^2u_x$	0.0369
0139, L3: $u_t = -0.927\,u^3 + 1.71\,u^2 - 0.784\,u + 0.00318\,u_{xx}$; PM excludes u, u^2, u^3.					
SINDy	R	yes	n/a	none	8.41×10^{-8}
SINDy	PM	no	no	$+uu_{xx}$, $+u_x^2$; $-u$, $-u^2$, $-u^3$	0.852
WSINDy	R	yes	n/a	none	5.10×10^{-7}
WSINDy	PM	no	no	$+uu_{xx}$, $+u_x^2$, $+u^2u_{xx}$, $+uu_x^2$; $-u$, $-u^2$, $-u^3$, $-u_{xx}$	0.939
GPT-5.6-sol	R	yes	n/a	none	4.44×10^{-7}
	PM, r1	no	no	returned $u_{tt} = \partial_{xx}\Phi(u)$ (wrong time order)	$1^\dagger$
	PM, r2	no	no	$+\mathcal{P}_4$; $-u$, $-u^2$, $-u^3$	0.817
	PM, r3	no	no	no equation	$1^\dagger$
Claude Opus 5	R	yes	n/a	none	4.39×10^{-7}
	PM, r1	no	no	no analysis call, no equation	$1^\dagger$
	PM, r2	no	no	$+\mathcal{P}_4$; $-u$, $-u^2$, $-u^3$	0.835
	PM, r3	no	no	$+\mathcal{P}_4$; $-u$, $-u^2$, $-u^3$	0.820
0139, L2: the swapped card C excludes u, u^2, u^3.					
SINDy	S	yes	n/a	none	8.42×10^{-8}
SINDy	C	no	no	$+u_x$, $+uu_x$, $+u_x\cos u$, $+u_x\sin u$; all four reference terms missing	0.857
WSINDy	S	no	n/a	$+\sin u$, $+\cos u$; $-u$, $-u^3$	0.0286
WSINDy	C	no	no	$+uu_{xx}$, $+u_x^2$, $+u^3u_{xx}$, $+u^2u_x^2$; all four reference terms missing	0.954

Table 12: LLM-PDESR on KdV (0319) at L3 on clean data: our L3 specification versus F21, the unmodified upstream LLM-PDESR specification for this equation, on the same data with the same model and search settings (at most 500 samples, four per prompt, patience 50). Reference: $u_t = -6\,uu_x - u_{xxx}$; no run misses a reference term. Exact is exact structural recovery. First exact is the position of the first exact-structure candidate in the sample history (gen. = its rank among model-generated candidates). Stop is the sample at which the patience rule ended the search, and Bodies the number of distinct program bodies. Scored and failed candidates were 54/31 for L3 and 51/6, 51/18 and 52/21 for the three F21 repeats. Coefficients are the saved fitted values, rounded to three significant figures; for F21 they were reconstructed with the frozen scoring recipe because upstream logs only rounded coefficients. NMSE is the clean held-out temporal-target NMSE (main-figure convention); the reference equation itself scores 3.86×10^{-3} on the same mask. In the reference right-hand-side convention the values are 2.62×10^{-3} (L3) and 8.69×10^{-8} (F21). F21 differs from our L3 specification as a whole: its seed function already contains a dispersion term and it has five instead of six parameter slots. The contrast is therefore not an isolated wording change.

Run	Exact	First exact (sample)	Stop	Bodies	Recovered right-hand side of u_t	Extra terms	NMSE
L3 (ours)	no	–	84	11	$-4.22\,uu_x - 0.932\,u_{xxx}$ $-3.51\,u^2u_x - 0.660\,u^3u_x$ $-0.132\,u_x$	u_x, u^2u_x, u^3u_x	7.59×10^{-3}
F21, rep. 1	yes	2 (gen. 1)	57	11	$-6.00\,uu_x - 1.00\,u_{xxx}$	none	3.87×10^{-3}
F21, rep. 2	yes	2 (gen. 1)	69	11	$-6.00\,uu_x - 1.00\,u_{xxx}$	none	3.87×10^{-3}
F21, rep. 3	yes	2 (gen. 1)	71	16	$-6.00\,uu_x - 1.00\,u_{xxx}$	none	3.87×10^{-3}

Table 13: Noise-model disclosure on the 20 Fokker–Planck systems 0163–0182 at L3 with 5% noise. The control is the saved historical benchmark answer without the card. The card run uses the same prompt and numerical input, with one appended paragraph stating that the noise is independent pointwise Gaussian with standard deviation 5% of the local field magnitude and that the coordinates are noise-free. Each entry is control → card over 20 systems per agent. Gains and losses count systems that become or stop being exact. p is the two-sided exact McNemar test, a Binomial$(n, 0.5)$ sign test on the n discordant systems. The pooled test ignores the dependence between the two agents on the same system and is descriptive only. Transcription rejections are accepted answers whose literal equation the frozen transcription could not parse (Unicode derivative symbols, implicit multiplication, an undefined π or an unclosed bracket). They score NMSE 1 and are not exact; one of the two GPT-5.6-sol losses (0178) is such a rejection. Median NMSE is the clean held-out temporal-target NMSE, with failed answers at 1; finite values above 1 are not clipped. Keyword rows count answers whose reported method keywords, from the final answer and the analysis call, match fixed patterns: weak form or integral (for example weak form, integral, integration by parts); pointwise finite differences (finite difference, centered differences, fd) or spectral derivatives; and smoothing or filtering (for example Savitzky–Golay, smoothing, denoising, low-pass, low-rank). The keywords record what an agent said it used; the analysis code was not saved, so they cannot show which estimator changed a score. The controls are selected historical runs, not paired seeds with equal retry budgets.

Control → noise-model card	GPT-5.6-sol	Claude Opus 5	Pooled
Accepted answers	9 → 20	17 → 18	26 → 38
Exact structural recovery	6/20 → 11/20	2/20 → 10/20	8/40 → 21/40
Gains / losses (discordant systems)	7 / 2	9 / 1	16 / 3
Two-sided exact McNemar p	0.18	0.021	0.0044
Transcription rejections	0 → 7	0 → 1	0 → 8
Median NMSE	1 → 1	8.26 → 1.70	–
Reported estimator keywords (answers out of 20 per agent, 40 pooled; multi-label)			
Weak form or integral	5 → 19	6 → 14	11 → 33
Pointwise finite differences or spectral	7 → 0	20 → 8	27 → 8
Smoothing or filtering	16 → 12	0 → 7	16 → 19
No analysis call	1 → 0	0 → 2	1 → 2

Table 14: Strict-blind discovery on the five unlabeled stationary CFD fields 0333–0337. One independent Claude Opus 5.5 agent per field received the clean steady 2D field as three anonymous, permuted and rescaled channels on shifted and rescaled coordinates, with no physics, variable roles or units disclosed. The released data store neutral channels recorded as u, v and p. Each run returned two relations in its anonymous channels. We map them back to (u, v, p) through the recorded permutation and scaling, which the agents never saw, and normalize each so that the pressure-gradient coefficient is 1; no coefficient is refitted. $\Delta = \partial_{xx} + \partial_{yy}$. Re is the approximate Reynolds number reported in the study summary; it was not recomputed, and the velocity and length conventions differ between fields. Stokes: $\nabla p = \mu \Delta \mathbf{u}$ in both components with no convective term. Steady NS: the full steady Navier–Stokes momentum balance $\nabla p + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \mu \Delta \mathbf{u}$ in both components. Both NS forms contain the Stokes balance, and they occur on the two fields with the highest reported Re. Forward NMSE is the largest per-channel NMSE of each run's own boundary-value solve on its full valid domain, as recorded by the run. The runs' own normalized residuals, $\mathrm{RMS}(\text{residual})/\mathrm{RMS}(\text{largest term})$, are 7.1×10^{-4} to 3.5×10^{-3} on their primary derivative schemes; 0336 recorded no single primary scheme and ranges from 9.5×10^{-4} to 2.7×10^{-2} across its smoothing and weak-form variants. The FD check uses the same ratio with a common fourth-order centered finite-difference stencil on an interior mask eroded by 7×7 cells, without fitting. Convective share is $\mathrm{RMS}\,|\rho(\mathbf{u} \cdot \nabla)\mathbf{u}|/\mathrm{RMS}\,|\nabla p|$ with ρ from an independent joint least-squares fit of the steady NS momentum balance, not from the agents; it is largest on 0336, a Stokes field. No run returned a continuity relation, and the observed in-plane velocities violate $u_x + v_y = 0$: $\mathrm{RMS}(u_x + v_y)/\mathrm{RMS}(u_x)$ ranges from 1.01 to 2.03. These fields have no reference PDE, so the classes describe the returned momentum structure only.

Field	Re	Returned relations (x and y momentum)	Class	Forward NMSE	FD check (x / y)	Convective share (%)
0333	0.015	$p_x = 3.00\,\Delta u$ $p_y = 3.00\,\Delta v$	Stokes	6.5×10^{-7}	0.0074 / 0.026	0.0730
0334	0.05	$p_x + 0.964\,(uu_x + vu_y) = 3.00\,\Delta u$ $p_y + 0.964\,(uv_x + vv_y) = 3.00\,\Delta v$	Steady NS	2.2×10^{-7}	0.014 / 0.0082	0.220
0335	0.014	$p_x = 2.99\,\Delta u$ $p_y = 2.99\,\Delta v$	Stokes	7.9×10^{-7}	0.0098 / 0.011	0.121
0336	0.003	$p_x = 3.00\,\Delta u$ $p_y = 3.00\,\Delta v$	Stokes	8.6×10^{-8}	0.036 / 0.062	0.238
0337	0.026	$p_x + 0.950\,(uu_x + vu_y) = 3.00\,\Delta u$ $p_y + 0.950\,(uv_x + vv_y) = 3.00\,\Delta v$	Steady NS	4.5×10^{-7}	0.0059 / 0.0050	0.144

C.2 COMPUTE AND COST

Reserved GPU-hours measure allocated capacity, not GPU use, which no record preserves. All methods ran on a workstation with an AMD Ryzen Threadripper PRO 5955WX CPU (16 cores), 251 GiB of memory and two NVIDIA GeForce RTX 3090 GPUs. The LLM-based methods call hosted model APIs, so their costs are reported in session-hours and tokens.

Table 15: Recorded compute cost of each method. Units and coverage differ by row, so the values show scale rather than a ranking.

Method	Recorded cost (M: millions of tokens)	Coverage
SINDy	2.36 elapsed hours on four workers	Scalar-PDE tuning and fitting; selected-fit timing excludes cached feature construction
WSINDy	0.76 summed hours (library and fit); 3.10 summed hours (tuning groups)	Scalar PDEs
SGA-PDE	Clean: about 24 s median search and 35 s mean total time per run. Noisy: 75.91 summed wall-hours; 72.03 measured CPU-hours	Noisy CPU-hours from 880 scheduler jobs, 14 failed
DISCOVER	89.52 summed training hours (clean L1–L3); noisy L1: 345.5 summed training hours; noisy L2 and L3: 827.72 measured CPU-hours, 142.36 reserved GPU-hours	Training hours: clean L1–L3 runs. Noisy L1 excludes 30 failed runs (about 13.4 further hours). Noisy L2 and L3: 245 jobs, 87 failed; not separable by level
GPT-5.6-sol	527.73 session-hours; 367.15M input and 75.08M output tokens	120 attempts lack usage records
Claude Opus 5	361.32 session-hours; 103.64M input and 104.00M output tokens; $3,204.30	Dollar cost from the command-line client
LLM-PDESR	161.50 task-hours; 98,756 successful API responses; 138.07M input and 78.15M output tokens	All runs (clean data only)