# BLADE-PDE: BLENDED-ACTIVATION DENOISING AND LLM-AIDED DISCOVERY OF PARTIAL DIFFERENTIAL EQUATIONS FROM NOISY DATA

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Partial differential equations govern physical systems, and complex systems often require data-driven recovery of their governing equations from observations. Practical observations carry measurement noise, which is challenging because differentiation amplifies the noise and the corrupted derivatives hurt discovery. Implicit network fitting can denoise the observations before discovery, yet different kinds of PDEs prefer different activation representations in such implicit denoising; this motivates a blending activation, which learns a mixture of standard activations adapted to each field. Our denoising method fits the noisy field with an implicit network whose blending activation applies an affine transform to each activation basis, and the affine transform makes a SIREN-like initialization of the sine basis well-defined. LLM-based discovery brings broad prior knowledge and generates free-form candidates without a hand-crafted library, yet the LLM is not specialized in PDEs, and the raw data alone convey limited information about the system. We therefore run an LLM-aided discovery stage on the denoised derivatives that supplies the LLM with two kinds of information; equation skeletons retrieved from a math handbook provide PDE-specific structural knowledge, and mathematical tools extract essential properties of the system that guide the LLM inference and refinement. The LLM receives a property card of invariance and structure probes computed from the data, equation skeletons retrieved from a math handbook, and correlations of a derivative library, and proposes a sparse combination of library terms; mathematical tools such as weak-form, divergence-form, and scaling-invariance filters then score the proposal and return their notes to the LLM for another round of refinement. On the MDBench benchmark the discovery stage recovers 7 of 14 systems from clean data, more than any tested estimator; the full pipeline recovers all 6 systems of the noisy subset at 20 dB where no baseline recovers any; and under a strict leakage-control protocol the method discovers a synthesized equation outside well-known families with coefficients within a few percent.

## 1 INTRODUCTION

Partial differential equations (PDEs) are the fundamental language of continuum physics: they encode the conservation laws and constitutive relations that govern fluids, materials, climate, and many other dynamical systems. In most complex settings the governing PDE cannot be derived from first principles, so recovering it from observations is a central scientific task. Data-driven PDE discovery yields an explicit equation rather than a black-box predictor, enabling mechanistic interpretation, analysis, and extrapolation (Rudy et al., 2017).

Methods for this task have developed along several lines. Sparse regression, including SINDy and PDE-FIND, selects terms from a hand-crafted library (Brunton et al., 2016; Rudy et al., 2017). Physics-informed neural networks (PINNs) embed a parametric library residual in the training loss (Chen et al., 2021). Genetic algorithms search open-form expression trees (Chen et al., 2022), and reinforcement learning treats operator or term choice as a sequential decision problem (Du et al., 2024b). Across this lineage, observational noise remains a shared obstacle: numerical differentiation



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.

amplifies high-frequency error, and the difficulty is especially severe for high-order PDEs, where higher spatial derivatives magnify noise (Thanasutives et al., 2023).

Implicit coordinate networks and PINN-style denoisers, as well as weak-form estimators, were introduced to mitigate noise (Both et al., 2021; Chen et al., 2021; Messenger & Bortz, 2021; Stephany & Earls, 2024b). They still degrade on high-order fields: automatic differentiation of a poorly represented implicit fit, and weak-form testing of high-order operators, both remain sensitive to residual roughness. Moreover, weak-form and PINN discovery are library-based, so they cannot recover free-form structure absent from the candidate set. Some PDEs, including those whose nonlinearities do not admit a convenient weak-form expression, fall outside that toolkit.

Adaptive activations have been studied outside PDE discovery. Adaptive Blending Units (ABUs) learn a mixture of standard nonlinearities (Sütfeld et al., 2020); Jagtap-style adaptive activations introduce trainable slope parameters in PINNs for forward and inverse PDE solves (Jagtap et al., 2020); and SIREN uses periodic activations with a dedicated initialization for implicit signal representation (Sitzmann et al., 2020). None of these mechanisms has been used to denoise observations for subsequent PDE discovery.

Large language models (LLMs) have recently been applied to equation discovery. An LLM carries broad prior knowledge of mathematical and physical structures, proposes free-form equations without a hand-crafted library, and refines its proposals from feedback across rounds (Shojaee et al., 2025; Du et al., 2024a). Two weaknesses limit this trend for PDE discovery. A general LLM is not specialized in PDEs; training a generative model on PDEs collected from a math handbook supplies that specialization, but such a small dedicated model gives up the broad knowledge and reasoning of the LLM (Xu et al., 2025). Moreover, the observed data enter the LLM only as raw numeric inputs, which convey limited information about the underlying system, so the LLM must guess structure that a mathematical analysis could establish. We target both weaknesses; equation skeletons retrieved from a math handbook give the LLM PDE-specific structural knowledge without retraining, and mathematical tools extract essential properties of the system that inform the LLM proposal and refinement.

We propose BLADE-PDE, a method that couples blended-activation denoising with LLM-aided discovery. Denoising for discovery is not the same as fitting or solving a clean PDE; the network must recover a field whose derivatives remain faithful, without overfitting measurement noise. The denoiser therefore uses an ABU activation so that the implicit network can represent the diverse local shapes of PDE solutions, and an affine transform on each basis so that a SIREN-like initialization remains well-defined. The discovery stage runs on the derivatives of the denoised field; the LLM receives the handbook skeletons and the tool-extracted properties together with a derivative library summary, proposes a sparse combination of library terms, and refines the proposal across rounds using the notes returned by the mathematical tools.

**Contributions.**

- We propose an implicit denoising network with a blending activation, addressing the mismatch between a fixed activation and the diverse shapes of PDE solutions; the learned mixture adapts the representation to each noisy field so that its derivatives stay faithful for discovery.
- We introduce an affine transform on each activation basis, addressing the fact that a SIREN-like initialization is not defined for a sine component inside a mixture; the transform carries the initialization scale, so the periodic basis starts well-conditioned.
- We develop an LLM-aided discovery stage that closes two information gaps; equation skeletons retrieved from a math handbook supply the PDE-specific structural knowledge a general LLM lacks, mathematical tools extract system properties that raw numeric data alone do not convey, and the same tools score the LLM proposals and drive their refinement.

## 2 Related Work

Sparse regression remains a primary route to interpretable governing equations. SINDy and PDE-FIND assemble a library of candidate terms and select a sparse subset by sequential thresholding or

regularized least squares (Brunton et al., 2016; Rudy et al., 2017). Weak-form sparse variants such as WSINDy and ensemble WSINDy reduce explicit differentiation by integrating against test functions (Messenger & Bortz, 2021; Fasel et al., 2022), while Bayesian sparse formulations quantify coefficient uncertainty (More et al., 2023). Observational noise is still the shared obstacle: numerical derivatives amplify high-frequency error, and the difficulty is especially severe for high-order spatial operators (Thanasutives et al., 2023). In contrast, we do not discover from noisy numerical derivatives — we denoise the observed field first so that later identification sees a smoother state.

A complementary literature treats denoising as part of the estimator. PINN-style and DeepMoD-like models fit an implicit coordinate network while penalizing a library residual (Chen et al., 2021; Both et al., 2021), earlier neural networks learn differential operators from data (Berg & Nyström, 2019; Long et al., 2018; 2019), and weak-form estimators such as PDE-Learn project onto test functions (Stephany & Earls, 2022; 2024b;a). These approaches remain library-based and still degrade on high-order fields. Our implicit denoiser is instead a preprocessor that can feed any subsequent identifier.

Open-form search removes the candidate library. Genetic algorithms evolve expression trees (Chen et al., 2022); reinforcement learning treats operator or term choice as a sequential decision (Du et al., 2024b; Zhang et al., 2024); and symbolic-regression engines, including PySR, Operon, and unified deep symbolic regression, grow algebraic formulae from primitives (Cranmer, 2023; Burlacu et al., 2020; Landajuela et al., 2022). Transformer and neural sequence models similarly emit expressions without a fixed PDE library (Kamienny et al., 2022; Biggio et al., 2021). These methods inherit whatever derivatives or residual features they are given: when those inputs are computed from noisy observations, high-order terms remain unreliable. Open-form discovery still needs clean derivatives, which we supply by blended-activation implicit denoising before search.

LLM-based equation discovery is a recent branch of this open-form line. LLM-SR and LLM4ED let a general LLM propose equations and refine them against data (Shojaee et al., 2025; Du et al., 2024a), relying on knowledge that is not specialized in PDEs, while EqGPT trains a dedicated generative model on a small corpus of handbook PDE structures (Xu et al., 2025). Our discovery stage keeps the general LLM and supplies the missing information explicitly, retrieving handbook skeletons and providing system properties extracted by mathematical tools that also drive refinement.

Adaptive activations and implicit representations have been developed outside discovery. Adaptive Blending Units (ABUs) learn a mixture of standard nonlinearities (Sütfeld et al., 2020), and ABU-PINN tailors such blends to PINN solves by keeping only candidates with usable higher-order derivatives and adding adaptive slopes (Wang et al., 2023). Jagtap-style adaptive activations introduce trainable slopes that accelerate PINN forward and inverse solves (Jagtap et al., 2020; Chen et al., 2021), and SIREN uses periodic activations with a dedicated initialization for implicit signal representation (Sitzmann et al., 2020). Implicit coordinate networks have also appeared as denoisers inside library-based discovery pipelines (Both et al., 2021). None of ABU, ABU-PINN, Jagtap adaptive activations, or SIREN has been used for discovery-denoising. We apply them to denoising-for-discovery, an ABU implicit network with affine, SIREN-like initialization, rather than a clean-data PINN solve.

# 3 BLADE-PDE

BLADE-PDE consists of two stages, summarized in Figure 1. In the denoising stage, an implicit network with a blending activation fits the noisy observations, and automatic differentiation of the fitted network supplies the denoised field and its derivatives. In the discovery stage, an LLM proposes the governing equation from those derivatives; the LLM receives a property card of system probes computed from the data, equation skeletons retrieved from a math handbook, and a summary of a derivative library, and mathematical tools score each proposal and return notes that drive further rounds of refinement.

## 3.1 Implicit denoising with ABU activations

An implicit coordinate network $u_\theta(x, t)$ is fit to the noisy observations by mean squared error, with no physics residual in the objective, and automatic differentiation of the fitted network supplies the denoised field and its derivatives. Every hidden layer uses an Adaptive Blending Unit (ABU)

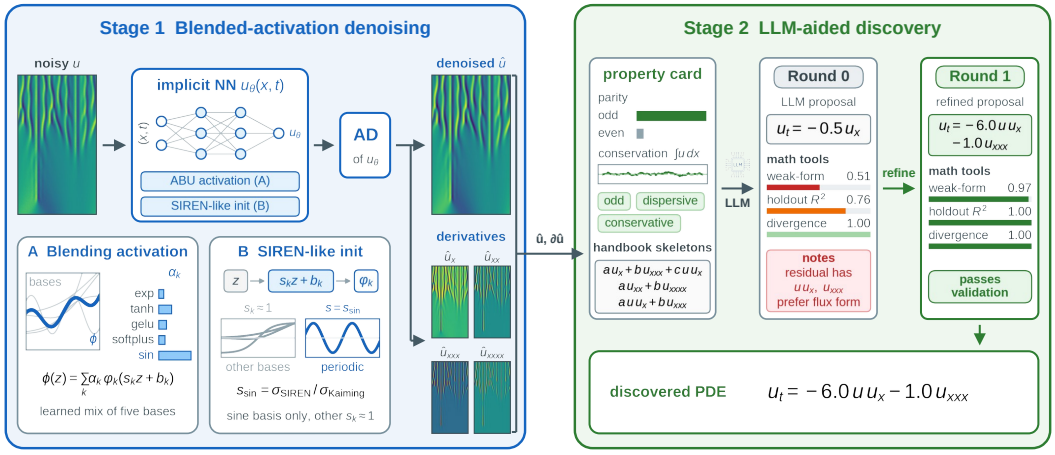



Figure 1: Overview of BLADE-PDE. In the denoising stage, an implicit coordinate network with a blending activation and a SIREN-like initialization on the sine channel fits the noisy field, and automatic differentiation yields the denoised field and its derivatives. In the discovery stage, the LLM receives a property card computed from the data and equation skeletons retrieved from a math handbook, proposes a sparse equation, and refines it across rounds using the scores and notes returned by the mathematical tools.

(Sütfeld et al., 2020), a weighted sum of $K{=}5$ standard nonlinearities after an affine map on each basis applied to the hidden pre-activation $z$,

$$\phi(z) \;=\; \sum_{k=1}^{K} \alpha_k \, \varphi_k(s_k z + b_k), \qquad \varphi_k \in \{e^z - 1,\ \tanh,\ \mathrm{GELU},\ \mathrm{softplus},\ \sin\}, \tag{1}$$

with mixture weights initialized at $\alpha_k = 1/K$ and trained together with $(s_k, b_k)$ by the same reconstruction loss. The growing bases represent fronts and ramps, $\tanh$ represents smooth transitions, and the sine represents oscillatory and high-frequency content, so the data select the activation of each layer. The per-basis affine map lets each basis choose its own operating range on top of the shared linear layers. Figure 2 shows the layer, and Appendix A discusses the bases, the affine map, and the fit.

### 3.2 SIREN-LIKE INITIALIZATION

A sine basis needs the input scale prescribed by SIREN (Sitzmann et al., 2020), but the linear layers of an ABU are shared by all five bases. Weights drawn at the SIREN scale would leave the non-periodic bases with inputs far too small, and weights drawn at a standard scale would overdrive the sine. We therefore keep a standard Kaiming-style initialization on the shared layers and initialize the affine scale of the sine channel to the compensation factor $s_{\sin} = \sqrt{\sigma^2_{\text{SIREN}}/\sigma^2_{\text{Kaiming}}}$, where $\sigma^2_{\text{SIREN}}$ and $\sigma^2_{\text{Kaiming}}$ are the weight variances of the two recipes, which gives $s_{\sin} = 1/\omega_0$ at hidden layers. The sine channel then behaves as if its layer had been SIREN-initialized, while the other four bases start at unit scale and zero shift and are unaffected. Appendix B gives the variances and discusses SIREN initialization of the shared layers as an alternative.

### 3.3 LLM-AIDED DISCOVERY

The discovery stage regresses the temporal derivative $\hat{u}_t$ on a candidate library $\{\theta_m\}$ of the state, its spatial derivatives, and their products, such as $\hat{u}$, $\hat{u}_x$ to $\hat{u}_{xxxx}$, $\hat{u}\hat{u}_x$, and $\hat{u}^2$, so a proposal is a sparse support with coefficients. An LLM brings broad prior knowledge to this choice but is not specialized in PDEs, so we supply the missing information from two sides. A property card of neutral probes computed from the data reports the parity, conserved densities, dominant scaling balance, and candidate linear orders of the system, and its tags retrieve nameless equation skeletons from a math

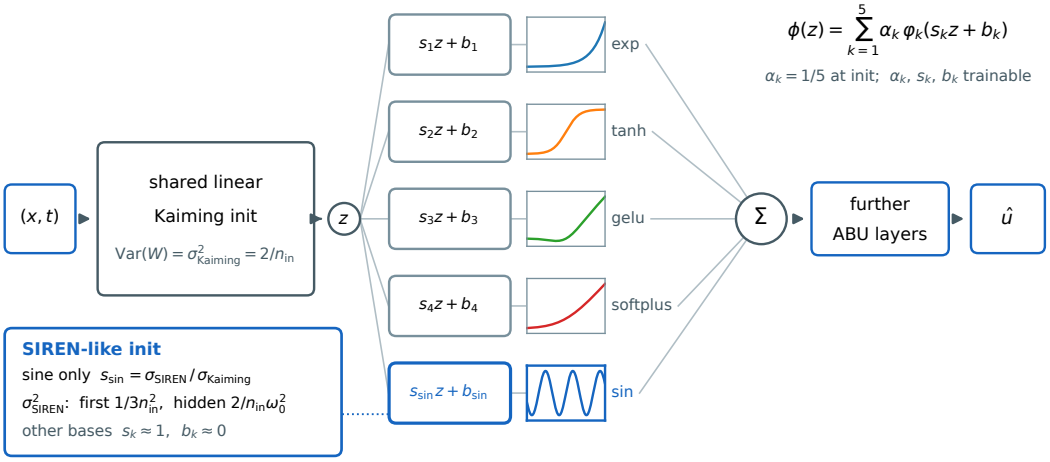



Figure 2: ABU layer with SIREN-like initialization. The coordinate input $(x, t)$ passes through shared linear layers with Kaiming initialization; the pre-activation $z$ fans out through the per-basis affine maps $s_k z + b_k$ into the five bases, which are blended with weights $\alpha_k$, initialized at $1/5$, before the later ABU layers produce $\hat{u}$. The four non-periodic bases start at $s_k \approx 1$ and $b_k \approx 0$, while the sine branch carries the compensation scale $s_{\sin}$ that maps the Kaiming variance onto the SIREN target.

handbook as structural priors. The LLM receives the card, the skeletons, the library correlations, and a few numeric samples under an opaque case identifier and proposes a sparse equation. Mathematical tools then verify the proposal with dependency, weak-form, scaling, conservation-form, variational, and dispersion filters, and return their scores together with the residual correlations and a joint least-squares refit as notes for the next round. The stage runs two rounds, and the reported equation is the final joint refit, so the LLM selects the support and least squares on the data sets the coefficients. Appendices C to F define the probes, the retrieval, the filters, and the refinement protocol, and Figure 6 in Appendix G walks through one run.

# 4 Experiments

## 4.1 Clean PDE discovery on MDBench

We first evaluate the discovery stage alone on clean data. MDBench is a benchmark of fourteen field datasets for data-driven model discovery (Ziaei Bideh et al., 2026). It pairs the canonical equations of the PDE-discovery literature, advection, Burgers, Korteweg–de Vries (KdV), Kuramoto–Sivashinsky (KS), and two-dimensional advection–diffusion (AD), with challenging systems drawn from real research problems, heat conduction in one, two, and three spatial dimensions under solar forcing together with a laser-heating case, a coupled nonlinear Schrödinger (NLS) system, reaction–diffusion (RD), and Navier–Stokes and reaction–diffusion flows in a channel or past a cylinder. The benchmark tests eight published estimators on the official grids, PDE-FIND, Bayesian sparse regression, WSINDy, EWSINDy, DeepMoD, and MDBench versions of PySR, Operon, and uDSR, whose outcomes we cite from the MDBench study (Ziaei Bideh et al., 2026). We additionally run the handbook-trained generative model EqGPT and the LLM-based method LLM4ED on the same fields (Xu et al., 2025; Du et al., 2024a), with Grok 4.6 as the language model of LLM4ED. A discovery is counted as correct when the equation contains exactly the terms of the governing form and every coefficient is within $5\%$ of the true value.

Clean fields require no denoising, so this experiment isolates the LLM-aided discovery stage. The candidate library and the temporal derivative are computed by finite differences on the official grids over the first $80$ percent of the time axis. The property card, the handbook skeletons, and the library summary are assembled as in Section 3.3 and passed to a general-purpose LLM under the hard-control protocol; each field appears only as an opaque case identifier, tool use is disabled,

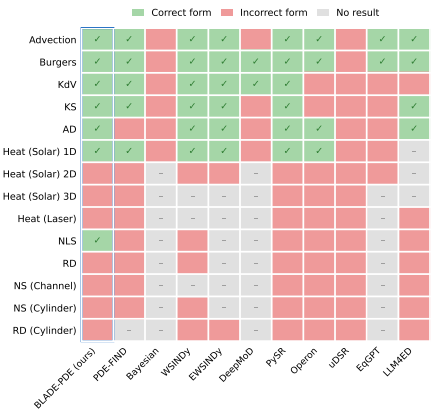



Figure 3: Clean-data discovery on MDBench. Dataset-by-method outcome matrix distinguishing correct form, incorrect form, and no result, comparing BLADE-PDE with the eight estimators tested by the MDBench study and with EqGPT and LLM4ED; the BLADE-PDE column is outlined.

Table 1: Clean MDBench systems recovered by BLADE-PDE. Mean relative coefficient error is averaged over the ground-truth terms of each equation.

| Dataset | Ground truth | Discovered | Coeff. err. |
|---|---|---|---|
| Advection | $u_t = -0.1\,u_x$ | $u_t = -0.1000\,u_x$ | $< 0.1\%$ |
| Burgers | $u_t = -u\,u_x + 0.1\,u_{xx}$ | $u_t = -0.9997\,u\,u_x + 0.1012\,u_{xx}$ | $0.6\%$ |
| KdV | $u_t = -6\,u\,u_x - u_{xxx}$ | $u_t = -6.041\,u\,u_x - 1.023\,u_{xxx}$ | $1.5\%$ |
| KS | $u_t = -u\,u_x - u_{xx} - u_{xxxx}$ | $u_t = -0.9926\,u\,u_x - 1.012\,u_{xx} - 1.016\,u_{xxxx}$ | $1.2\%$ |
| Heat (Solar) 1D | $u_t = 9.58 \times 10^{-7}\,u_{xx}$ | $u_t = 9.582 \times 10^{-7}\,u_{xx}$ | $< 0.1\%$ |
| NLS | $u_t = 0.5\,v_{xx} + u^2 v + v^3$ | $u_t = 0.5140\,v_{xx} + 1.005\,u^2 v + 1.001\,v^3$ | $1.2\%$ |
| | $v_t = -0.5\,u_{xx} - uv^2 - u^3$ | $v_t = -0.5124\,u_{xx} - 1.000\,uv^2 - 1.004\,u^3$ | $1.0\%$ |
| AD | $u_t = 0.25\,u_x + 0.5\,u_y + 0.5\,u_{xx} + 0.5\,u_{yy}$ | $u_t = 0.2507\,u_x + 0.5010\,u_y + 0.5034\,u_{xx} + 0.5024\,u_{yy}$ | $0.4\%$ |

and the proposal is refined for at most a small budget of rounds using the scores and notes of the mathematical tools.

Figure 3 reports the per-dataset outcomes. BLADE-PDE recovers the correct governing form on 7 of the 14 datasets, advection, Burgers, KdV, KS, AD, Heat (Solar) 1D, and the coupled NLS system, exceeding the strongest MDBench estimators, WSINDy, EWSINDy, and PySR, which each recover 6. NLS is recovered by no tested MDBench method. On the remaining 7 datasets the stage returns an incorrect form rather than no result; the pipeline is applicable to every dataset and always outputs an equation. Table 1 lists the recovered equations against the ground truth. Every recovered equation matches the governing form term for term, and the mean relative coefficient error stays at or below $1.5\%$ on every recovered system. Figure 7 in Appendix H visualizes 6 of the recovered systems, and the $u_t$ field predicted by the discovered right-hand side is visually indistinguishable from the ground-truth temporal derivative.

### 4.2 Noisy PDE discovery on MDBench

Measured fields are never clean, and noise is harder for PDE discovery than for ordinary regression because the regression target and most candidate terms are derivatives of the observed field, which numerical differentiation amplifies. MDBench provides a noisy protocol for this setting (Ziaei Bideh et al., 2026); each field is corrupted with Gaussian noise at signal-to-noise ratios of 40, 30, 20, and 10 dB, and noise robustness is evaluated on a subset of six datasets, advection, Burgers, KdV, KS, Heat (Soil) 1D, and two-dimensional advection–diffusion. We follow this subset and report the 20 dB level. Here the full BLADE-PDE pipeline runs; the implicit network with the blending activation and SIREN-like initialization is trained on the noisy observations, automatic differentiation supplies the derivatives, and the same hard-control LLM stage of Section 4.1 discovers the equation from them. As baselines we rerun the MDBench estimators that complete on the noisy fields, PDE-FIND, WSINDy, PySR, Operon, and uDSR (Rudy et al., 2017; Messenger & Bortz, 2021; Cranmer, 2023; Burlacu et al., 2020; Landajuela et al., 2022), and we additionally run the generative model

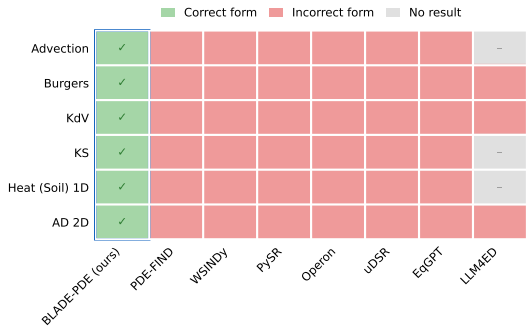



Figure 4: Noisy-data discovery on the MDBench six-dataset noise subset at $20\,\mathrm{dB}$. Dataset-by-method outcome matrix distinguishing correct form, incorrect form, and no result, comparing BLADE-PDE with the five MDBench estimators that complete on the noisy fields and with EqGPT and LLM4ED; the BLADE-PDE column is outlined.

EqGPT and the LLM-based method LLM4ED on the same fields (Xu et al., 2025; Du et al., 2024a). We keep the correct-form criterion of the clean experiment with the coefficient tolerance extended to $15\%$.

Figure 4 reports the outcomes at $20\,\mathrm{dB}$. BLADE-PDE recovers the correct governing form on all 6 datasets, and no baseline recovers a single one. The failures follow the derivative amplification described above; library methods keep the dominant transport term but lose the weak terms carried by higher derivatives, search methods collapse to near-constant or single-term expressions or wrap the dynamics into composite surrogates, and the handbook-trained EqGPT returns structurally inflated forms. LLM4ED, which differentiates the noisy field numerically, either exhausts its call budget without an equation or returns incomplete term sets. The margin comes from the two stages together; automatic differentiation of the blended-activation fit hands the LLM stage nearly clean derivatives, and the handbook skeletons with the mathematical tools keep the proposal on the governing structure rather than on the residual noise.

### 4.3 Discovering a new PDE under leakage control

An LLM evaluation must rule out the possibility that the model recognizes a benchmark rather than discovers an equation, so every LLM discovery run in this paper follows a hard-control protocol. Each field is staged under an opaque case identifier, and the model-visible data contain only the raw arrays $(t, x, u, u_t)$; the private mapping from identifiers to source systems and ground-truth equations is used for grading only and never enters a prompt. Prompts are assembled exclusively from the neutral summaries of Section 3.3, the property card, the nameless handbook skeletons, the library correlations, and a few numeric samples, and an automatic audit scans every prompt against a banned list of benchmark names, system names, noise levels, and repository terms before any call is made. The LLM runs with all tools disabled, so it cannot browse, search, or read files. After every round the full reasoning transcript is scanned against the same banned list, and a run that names a known equation family anywhere in its private reasoning is flagged as a violation.

These controls keep the answer out of the prompt, but a canonical benchmark equation could still coincide with structure the LLM has memorized. To show that the method discovers new expressions rather than recalls known ones, we synthesize a PDE that belongs to no well-known family by tweaking KdV; the quadratic flux and the dispersion carry noncanonical coefficients and an additional linear transport term is added,

$$u_t \;=\; -4\,uu_x \;-\; 0.25\,u_{xxx} \;-\; 1.5\,u_x. \tag{2}$$

The field is simulated fresh from a new initial condition, so the data exist nowhere outside this experiment, and it is staged as a fresh opaque case for the unchanged discovery stage. The stage recovers the generating support $\{u_x,\ uu_x,\ u_{xxx}\}$ in two rounds, and the least-squares refit gives $u_t = -1.52\,u_x - 4.03\,uu_x - 0.26\,u_{xxx}$, every coefficient within $5\%$ of the generating value with a

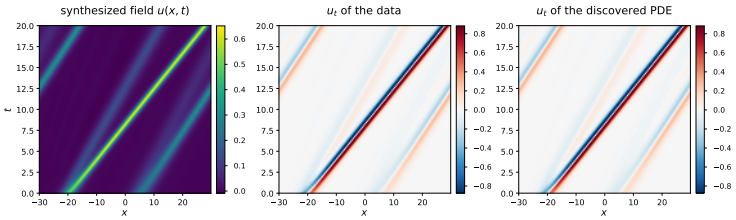



Figure 5: Discovery of the tweaked PDE outside well-known families. Left, the synthesized field. Middle, the temporal derivative of the data. Right, the temporal derivative evaluated from the discovered equation on the same field, which reproduces $u_t$ to a relative $\ell_2$ error of $1\%$.

Table 2: Component ablation on the MDBench noise subset at $20\,\mathrm{dB}$. Each variant changes one component and keeps the LLM discovery stage fixed. A check mark denotes a recovered governing form and a cross denotes a miss.

| Variant | Adv. | Burgers | KdV | KS | Heat 1D | AD | Total |
|---|---|---|---|---|---|---|---|
| no denoiser (FD) + LLM stage | × | × | × | × | × | × | 0/6 |
| fixed-tanh denoiser + LLM stage | × | ✓ | × | × | × | ✓ | 2/6 |
| ABU w/o SIREN init + LLM stage | ✓ | ✓ | × | × | ✓ | ✓ | 4/6 |
| BLADE-PDE (full) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 6/6 |

mean relative error of $2.1\%$. No banned name appears in the reasoning transcripts of either round. The discovered equation reproduces the temporal derivative of the data to a relative $\ell_2$ error of $1\%$, as shown in Figure 5. Appendix I gives the solver, the quality checks of the field, and the round-by-round trace.

### 4.4 Ablations

The ablations trace the noisy-data margin of Section 4.2 to its sources, the denoiser, the two design choices inside it, and the two information sources of the discovery stage. All variants run on the six-dataset MDBench noise subset at $20\,\mathrm{dB}$ with the protocol of Section 4.2 unchanged, and we report the correct-form outcome per dataset under the same $15\%$ coefficient tolerance.

The first ablation removes one component of the pipeline at a time while the LLM discovery stage stays fixed. Removing the denoiser entirely passes finite-difference derivatives of the noisy field to the discovery stage, which isolates the contribution of the implicit fit as a whole. Replacing the blending activation with a fixed tanh keeps the denoising protocol, the architecture size, and the training budget but removes the representation freedom, which isolates the ABU blend. Keeping the ABU blend but dropping the SIREN-like compensation on the sine channel, so that the sine basis starts at the same unit scale as the other bases, isolates the initialization. The full method completes the grid. Table 2 reports the outcomes.

Each component is needed. Without the denoiser, the discovery stage recovers none of the six systems, because finite differences of the noisy field bury the terms carried by higher derivatives. The fixed-tanh denoiser recovers only Burgers and AD, so a single fixed activation does not represent the range of field shapes in the subset. The ABU blend without the SIREN-like compensation recovers four systems but misses KdV and KS, the two systems with third- and fourth-order terms, which is where the high-frequency content carried by the sine channel matters most. The full method recovers all six systems.

The second ablation holds the full denoiser fixed and removes the information sources of the discovery stage. Removing the handbook drops the retrieved skeletons and their instructions from the prompt. Removing the mathematical tools drops the property card, the filter scores and notes, and the residual and refit notes, so the second round sees only the previous proposal. Every arm uses the denoised fields of the full method, the same LLM, and the same budget of two rounds with an early stop on success. Table 3 reports the outcomes and the round in which each success occurs.

Table 3: LLM mechanism ablation on the MDBench noise subset at $20\,\mathrm{dB}$. Every arm uses the full BLADE-PDE denoiser and changes only the information given to the LLM. A check mark denotes a recovered governing form, a superscript 2 marks a success reached only in the second round, and a cross denotes a miss.

| Discovery stage | Adv. | Burgers | KdV | KS | Heat 1D | AD | Total |
|---|---|---|---|---|---|---|---|
| w/o handbook, w/o math tools | ✓ | × | ✓ | $✓^2$ | × | × | 3/6 |
| w/o handbook | ✓ | ✓ | $✓^2$ | $✓^2$ | ✓ | ✓ | 6/6 |
| w/o math tools | ✓ | × | ✓ | ✓ | × | × | 3/6 |
| BLADE-PDE (full) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 6/6 |

The mathematical tools are essential for discovery. Without them, the LLM recovers only advection, KdV, and KS, with or without the handbook, and misses Burgers, Heat 1D, and AD in both rounds. On these systems the proposal omits a governing term, such as the diffusion term of Burgers or the $x$-direction terms of AD, or misses a coefficient by more than the tolerance, and without the residual notes the second round has no evidence to correct it. The handbook raises efficiency. With the tools but without the handbook, the LLM still recovers all six systems, but KdV and KS need a second round, whereas the full method recovers both in the first round. Among the arms without tools, the handbook likewise moves the KS success from the second round to the first. The skeletons therefore shorten the search for high-order structure, while the tools supply the evidence that makes discovery succeed.

## 5 Discussion

Each contribution addresses a specific failure mode, and the experiments trace the gains back to them. The blending activation addresses the mismatch between a fixed activation and the diverse local shapes of PDE fields. Its effect appears where discovery consumes derivatives of the implicit fit; at $20\,\mathrm{dB}$ every baseline loses the weak terms carried by higher derivatives while the full pipeline recovers all 6 governing forms (Figure 4), and the component ablation of Table 2 isolates the blend against a fixed-tanh denoiser under an identical protocol.

The affine transform on each basis addresses the initialization of the periodic component. It carries the SIREN-like compensation, so the sine channel starts in its useful regime while the shared layers keep a standard initialization (Figure 2). The compensation matters most for the high-frequency content behind high-order operators, the fourth-order KS terms and the third-order dispersion of KdV, and dropping it while keeping the blend is isolated as a separate row of the same ablation.

The LLM-aided discovery stage addresses two information gaps, PDE-specific structural knowledge supplied by the handbook skeletons and system properties supplied by the mathematical tools. On clean data the stage alone recovers 7 of 14 MDBench systems, exceeding the strongest tested estimators (Figure 3), and the walkthrough of Figure 6 in Appendix G shows the mechanism at work, the tool notes turn a wrong sparse guess into the governing form within one round. The mechanism ablation of Table 3 shows that the tools are essential for discovery and that the skeletons raise its efficiency, and the leakage-controlled experiment of Section 4.3 shows that the stage discovers a new expression rather than recalling a benchmark.

## 6 Conclusion

We presented BLADE-PDE, a two-stage method that couples blended-activation denoising with LLM-aided discovery. An implicit network with a blending activation, made SIREN-initializable by a per-basis affine transform, recovers noisy fields whose automatic-differentiation derivatives remain faithful, and an LLM stage guided by handbook skeletons and mathematical tools identifies the governing equation from those derivatives. On MDBench the discovery stage recovers 7 of 14 clean systems, the full pipeline recovers all 6 noisy-subset systems at $20\,\mathrm{dB}$ where no baseline recovers any, and under strict leakage control the method discovers a synthesized equation outside well-known families with coefficients within a few percent.

### AI USE STATEMENT

We used generative AI tools to assist with implementing the proposed method and with revising the writing of the paper. We did not use generative AI tools to come up with the ideas behind the contributions of this work. The authors checked all AI-assisted code and text, and we take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

Anonymized code will be provided to reviewers via OpenReview.

## REFERENCES

Jens Berg and Kaj Nyström. Data-driven discovery of pdes in complex datasets. *Journal of Computational Physics*, 384:239–252, 2019.

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

G.-J. Both, S. Choudhury, P. Sens, and R. Kusters. DeepMoD: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428:109985, 2021.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Bogdan Burlacu, Gabriel Kronberger, and Michael Kommenda. Operon C++: An efficient genetic programming framework for symbolic regression. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 1562–1570. Association for Computing Machinery, 2020. doi: 10.1145/3377929.3398099. URL https://doi.org/10.1145/3377929.3398099.

Y. Chen, Y. Luo, Q. Liu, H. Xu, and D. Zhang. Symbolic genetic algorithm for discovering open-form partial differential equations (sga-pde). *Physical Review Research*, 4(2):023174, 2022. doi: 10.1103/PhysRevResearch.4.023174.

Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, 12(1):6136, 2021. doi: 10.1038/s41467-021-26434-1.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*, 2023.

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: 097121, 2024a. doi: 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, 2024b. doi: 10.1103/PhysRevResearch.6.013182.

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

Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10269–10281, 2022.

Mikel Landajuela, Chak Shing Lee, Jiachen Yang, Ruben Glatt, Claudio P. Santiago, Ignacio Aravena, Terrell Mundhenk, Garrett Mulcahy, and Brenden K. Petersen. A unified framework for deep symbolic regression. In *Advances in Neural Information Processing Systems*, volume 35, pp. 33985–33998, 2022.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International conference on machine learning*, pp. 3208–3216. PMLR, 2018.

Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

D. A. Messenger and D. M. Bortz. Weak SINDy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021.

K. S. More, T. Tripura, R. Nayek, and S. Chakraborty. A Bayesian framework for learning governing partial differential equation from data. *Physica D: Nonlinear Phenomena*, 456:133927, 2023.

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.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=m2nmp8P5in.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Robert Stephany and Christopher Earls. Pde-read: Human-readable partial differential equation discovery using deep learning. *Neural Networks*, 154:360–382, 2022.

Robert Stephany and Christopher Earls. PDE-LEARN: Using deep learning to discover partial differential equations from noisy, limited data. *Neural Networks*, 174:106242, 2024a. doi: 10.1016/j.neunet.2024.106242.

Robert Stephany and Christopher Earls. Weak-PDE-LEARN: A weak form based approach to discovering PDEs from noisy, limited data. *Journal of Computational Physics*, 506:112950, 2024b. doi: 10.1016/j.jcp.2024.112950.

Leon René Sütfeld, Flemming Brieger, Holger Finger, Sonja Füllhase, and Gordon Pipa. Adaptive blending units: Trainable activation functions for deep neural networks. In *Intelligent Computing – Proceedings of the 2020 Computing Conference, Volume 3*, volume 1230 of *Advances in Intelligent Systems and Computing*, pp. 37–50. Springer, 2020. doi: 10.1007/978-3-030-52243-8_4.

Pongpisit Thanasutives, Takashi Morita, Masayuki Numao, and Ken-ichi Fukui. Noise-aware physics-informed machine learning for robust PDE discovery. *Machine Learning: Science and Technology*, 4:015009, 2023. doi: 10.1088/2632-2153/acb1f0.

Honghui Wang, Lu Lu, Shiji Song, and Gao Huang. Learning specialized activation functions for physics-informed neural networks. *Communications in Computational Physics*, 34:869–906, 2023. doi: 10.4208/cicp.OA-2023-0058.

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(1):10255, 2025. doi: 10.1038/s41467-025-65114-2. URL https://www.nature.com/articles/s41467-025-65114-2.

Xiaoxia Zhang, Junsheng Guan, Yanjun Liu, and Guoyin Wang. MORL4PDEs: Data-driven discovery of PDEs based on multi-objective optimization and reinforcement learning. *Chaos, Solitons & Fractals*, 180:114536, 2024. doi: 10.1016/j.chaos.2024.114536.

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.

## A Details of the ABU denoiser

The bases of the ABU layer fall into three families. The one-sided growing maps, the shifted exponential $e^z - 1$, GELU, and softplus, represent fronts, ramps, and localized growth. The saturating $\tanh$ represents smooth monotone transitions, and the periodic sine represents oscillatory and high-frequency content. Every hidden layer carries its own mixture and affine parameters, so depth does not force a single compromise. One layer can lean on the sine to track fine structure while another leans on the non-periodic bases that assemble the large-scale field.

The affine wrap is what makes the blend meaningful. All five bases share the same linear layers, so without $(s_k, b_k)$ one weight matrix would impose a single input scale on every basis. The small scale that suits the sine leaves the non-periodic bases in their near-linear regime, and a standard scale overdrives the sine, whose argument is further multiplied by $\omega_0$. The per-basis scale $s_k$ lets each basis choose its own operating range on top of the shared pre-activation, and the shift $b_k$ gives each basis a threshold or phase freedom. At initialization the four non-periodic bases start at unit scale with small jitter and near-zero shifts, while the sine scale carries the compensation of Section 3.2. The mixture weights $\alpha_k$ are trained by the same reconstruction loss as the network weights, with no auxiliary objective, so the data itself selects the activation. Different PDE families settle to different mixtures, which is precisely the representation freedom that motivates blending. Slope-adaptive activations in the sense of Jagtap et al. remain available as a related mechanism (Jagtap et al., 2020), but here the trainable degrees of freedom are the ABU mixture and the per-basis affine map.

The implicit denoiser is a coordinate MLP that uses this activation in every hidden layer. No physics residual enters the objective, so the stage requires no hypothesis about the equation. The derivatives read off by automatic differentiation are exact derivatives of the reconstructed surface rather than finite differences of noisy samples. Differentiation amplifies whatever roughness the representation retains, so the quality of a blend is judged by the derivatives it yields rather than by the reconstruction error alone.

## B Details of the SIREN-like initialization

SIREN feeds $\sin(\omega_0 z)$ with hidden weights uniform in $\pm\sqrt{6/n_{\text{in}}}/\omega_0$ and first-layer weights uniform in $\pm 1/n_{\text{in}}$ (Sitzmann et al., 2020). With these weights the pre-activations keep a variance for which the activations stay well spread over the period and gradients neither vanish nor explode, and the input span is mapped into the same regime. The corresponding weight variances are $\sigma^2_{\text{SIREN}} = 1/3n^2_{\text{in}}$ at the first layer and $\sigma^2_{\text{SIREN}} = 2/n_{\text{in}}\omega_0^2$ at hidden layers, while the shared Kaiming-style initialization has $\sigma^2_{\text{Kaiming}} = 2/n_{\text{in}}$. Both variances act on the same layer input, so their ratio is also the ratio of the pre-activation scales, and the compensation factor of Section 3.2 equals $1/\omega_0$ at hidden layers. The sine channel computes $\sin(\omega_0(s_{\sin} z + b_{\sin}))$, with the shift $b_{\sin}$ playing the role of a phase offset.

The linear maps themselves can alternatively be initialized with the full SIREN recipe, in which case the compensation reduces toward unity. We keep the two mechanisms separate so that the affine compensation is well-defined on its own, which is why the ABU layer carries $(s_k, b_k)$ rather than applying $\sin$ directly to $z$. The sine channel can always be brought to its correct operating scale regardless of how the shared weights were drawn.

## C PROPERTY CARD PROBES

The discovery stage regresses the temporal derivative on the candidate library,

$$\hat{u}_t \;\approx\; \sum_m c_m\,\theta_m, \qquad \theta_m \in \{1,\ \hat{u},\ \hat{u}_x,\ \ldots,\ \hat{u}_{xxxx},\ \hat{u}\hat{u}_x,\ \hat{u}^2,\ \hat{u}^3,\ \ldots\}, \tag{3}$$

and everything the LLM sees is computed offline from $(\hat{u}, \hat{u}_t)$ and carries no equation name. The property card is a set of neutral probes that summarize the system without naming any equation, and its outcomes are mapped to retrieval tags.

A parity probe correlates $\hat{u}_t$ with odd operators, $\hat{u}_x$, $\hat{u}_{xxx}$, and $\hat{u}\hat{u}_x$, and with even operators, $\hat{u}$, $\hat{u}_{xx}$, $\hat{u}_{xxxx}$, and $\hat{u}^2$, and reports the preferred parity. A conservation probe tracks the relative drift of $\int \rho\,dx$ for trial densities $\rho \in \{u, u^2\}$ against interior activity and flags a conservation-like system. A scaling probe estimates characteristic scales $(U, L, T)$ and tests which dominant balance, transport with $T \sim L/U$, diffusive with $T \sim L^2$, dispersive with $T \sim L^3$, or hyperviscous with $T \sim L^4$, best matches the measured magnitude of $\hat{u}_t$. A dispersion probe estimates the linear spectral symbol from the Fourier transform in $x$ and fits it by parity,

$$\hat{\sigma}(k) = \Big\langle\, \widehat{u_t}(k,t) \,/\, \widehat{u}(k,t) \,\Big\rangle_t, \qquad \mathrm{Re}\,\hat{\sigma}(k) \approx -a_2k^2 + a_4k^4, \quad \mathrm{Im}\,\hat{\sigma}(k) \approx a_1k - a_3k^3, \tag{4}$$

where the real part carries the dissipative even orders, the imaginary part carries the transport and dispersive odd orders, and the fitted magnitudes rank the candidate linear orders. Temporal homogeneity and shift-equivariance probes complete the card.

## D HANDBOOK RETRIEVAL AND PROMPT

Equation skeletons are retrieved from a math handbook collection of right-hand-side structures. The skeletons carry abstract tokens and tags only, no coefficients and no names. They are ranked by tag overlap with the property card and by match with the library correlations, and the top candidates are supplied to the LLM as soft priors. The LLM receives the property card, the retrieved skeletons, the library terms with their correlations against the target, and a small set of numeric samples, all under an opaque case identifier. It proposes a sparse support with coefficients, reasoned from the supplied evidence rather than from a recognized equation.

## E VERIFICATION FILTERS

Per-term scores from the filters below are aggregated by averaging, and terms whose aggregate falls below a threshold are dropped unless they are the only term.

The dependency filter removes linearly dependent library columns by QR factorization with column pivoting and notes product-rule identities such as $(u^2)_x = 2\hat{u}\hat{u}_x$. The weak-form filter projects both sides of the proposal onto smooth compact-support test functions $\varphi$ and scores the relative residual,

$$r_{\mathrm{weak}} \;=\; \mathrm{rms}_\varphi\Big(\langle \hat{u}_t, \varphi\rangle - \sum_m c_m \langle \theta_m, \varphi\rangle\Big) \Big/\; \mathrm{std}_\varphi\,\langle \hat{u}_t, \varphi\rangle, \tag{5}$$

averaged over test locations and times. The projection integrates against $\varphi$ instead of differentiating the data again, so the score stays robust to residual differentiation noise. The scaling filter assigns each monomial term its weight under the scaling estimated by the property card, with $x \to \lambda x$ and $u \to \lambda^{w_u} u$,

$$w\big(\hat{u}^{\,p}\,\partial_x^q \hat{u}\big) \;=\; (p+1)\,w_u - q, \tag{6}$$

and down-weights terms whose weight sits far from the median weight of the proposal, rejecting strong outliers. A Lie-point variant scores equivariance of each term under the same estimated scaling generator. The divergence-form filter rewrites terms as fluxes, $\hat{u}\hat{u}_x = \partial_x(\hat{u}^2/2)$ and $\hat{u}_{xxx} = \partial_x(\hat{u}_{xx})$, and prefers the conservation form when the card flags a conserved density. The variational filter applies a discrete self-adjointness proxy to tag gradient-flow-like versus skew or Hamiltonian-like structure, and the dispersion filter compares the proposed linear orders with the measured symbol of Equation 4, suggesting the dominant order when it is absent from the proposal.

## F REFINEMENT PROTOCOL

After each round, the residual of the proposal, $\hat{u}_t - \sum_m c_m \theta_m$, is correlated with every library column. A joint least-squares fit is then computed on the proposed support together with up to eight library terms whose residual correlation exceeds $0.15$ in magnitude. The notes returned to the LLM contain the filter notes, including suggestions such as a missing linear order or a preferred flux form, and numeric summaries of the previous proposal, its $R^2$, the residual correlations, and the joint fit with its $R^2$. The notes also ask for a sparse support and direct the LLM to take coefficient signs from the joint fit rather than from single-term correlations, because correlated library columns confound those signs. When the $R^2$ falls below $0.95$, the notes also list the candidate terms with high residual correlation.

## G WALKTHROUGH OF THE DISCOVERY STAGE

Figure 6 follows the discovery stage of Section 3.3 through two rounds on a clean MDBench field.

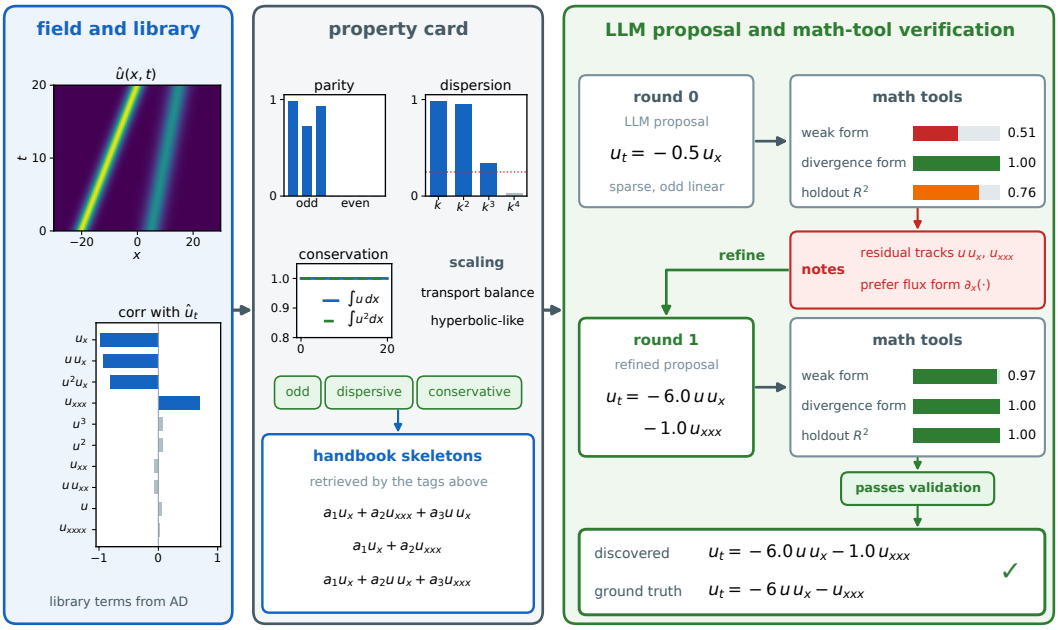



Figure 6: LLM-aided discovery on a clean MDBench field. Left, the field $\hat{u}(x,t)$ and the correlations of the library terms with $\hat{u}_t$. Middle, the property card computed from the data, an odd parity preference, dispersion order scores that favor orders one to three, conserved $\int u\,dx$ and $\int u^2 dx$, a hyperbolic-like regime with a transport balance, and the handbook skeletons retrieved by the resulting tags. Right, the LLM proposes $u_t = -0.5\,u_x$ in round zero; the math-tool scores, a weak-form score of $0.51$ and a holdout $R^2$ of $0.76$, and the notes that the residual correlates with $u\,u_x$ and $u_{xxx}$ and that a conservation form is preferred are returned, and the round-one proposal $u_t = -6.0\,u\,u_x - 1.0\,u_{xxx}$ passes validation and matches the governing equation.

## H RECOVERED CLEAN SYSTEMS

Figure 7 compares the ground-truth and predicted temporal derivatives for 6 of the 7 clean MDBench systems recovered in Section 4.1.

## I DETAILS OF THE TWEAKED PDE EXPERIMENT

The field of Equation 2 is integrated from a new initial condition with an integrating-factor RK4 spectral solver with $2/3$ dealiasing on a $512 \times 201$ grid over $T = 20$. Quality checks confirm that

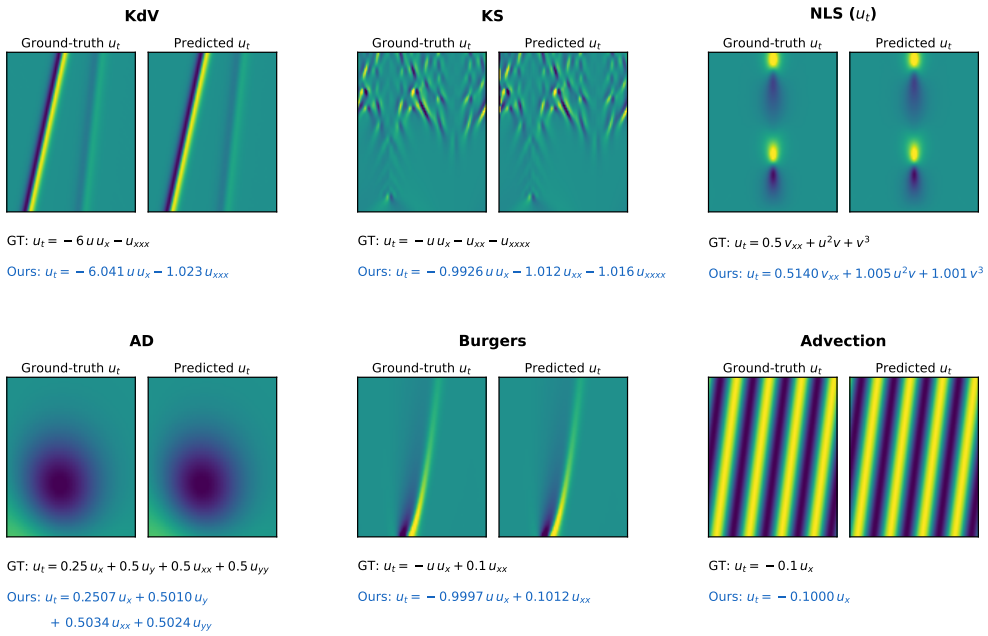



Figure 7: Ground-truth versus predicted $u_t$ for 6 of the 7 recovered clean MDBench systems, with the ground-truth and discovered equations. Predicted $u_t$ is the discovered right-hand side evaluated with finite-difference derivatives of the clean field. One-dimensional PDEs are shown as space–time fields with $x$ horizontal and $t$ vertical, and AD is an $x$–$y$ field at the midpoint of the training time window.

the field realizes the generating equation. The locked-coefficient residual RMSE is $2 \times 10^{-6}$, and a least-squares fit on the generating support returns the coefficients with $R^2 \approx 1$.

Round zero proposes the single dispersive term $u_t = -2.63\,u_{xxx}$. The weak-form score of $0.48$ and the returned notes, a preferred conservation form and residual correlations with $u_x$ and $u\,u_x$, redirect the proposal. Round one proposes the support $\{u_x,\ uu_x,\ u_{xxx}\}$, which matches the generating support exactly, and the least-squares refit on this support gives the equation reported in Section 4.3.