LLM-ACES: Closed-Loop Discovery of Dynamical Systems with LLM-Guided Adaptive Search

Nikhil Abhyankar*, Sha Li*, Sanchit Kabra*, Naren Ramakrishnan, Yulia Gel, Chandan K. Reddy

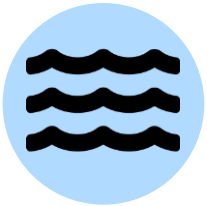
VIRGINIA TECH

Modeling Dynamical Systems with ODEs

Why do ODEs matter?

Ordinary differential equations provide compact, interpretable descriptions of continuous-time dynamics through vector fields the language scientists use to reason about how a system evolves.

Recovering these equations from experimental data is a fundamental driver of scientific progress, historically derived from first principles and, increasingly, from high-fidelity observational data.



Fluid Dynamics



Neuroscience



Epidemiology



Physics



Climate Systems

Current Methods for Dynamical Discovery

First-Principles



Derived analytically from physical law

Sparse Regression



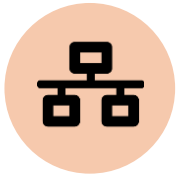
SINDy
E-SINDy
FFX

Evolutionary Search



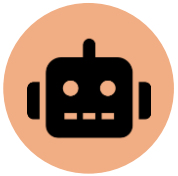
PySR AFP
Operon EPLEX

Transformer-Based



ODEFormer
End2End

LLM-Guided



LLM-SR
LLM-ODE

✓ **Problem:** Existing discovery methods treat discovery as a static problem over pre-collected, fixed datasets resulting in the **identifiability gap**.

Identifiability Gap: Challenge with Dynamical Discovery

A model can fit every observation and still be the wrong law.

- **Identifiability Gaps:** Structurally distinct equations match the observed states almost perfectly yet diverge outside the observed regime.
- **Data Limits**: Data from a limited range of observations results in identifiability issues, hindering the discovery of true underlying laws.
- ✓ **Goal**: Find the right observations to identify the true law that describes how a system moves and changes.

No amount of optimization on the same trajectory can reveal structure that is absent from the data.

Same early data, two different laws

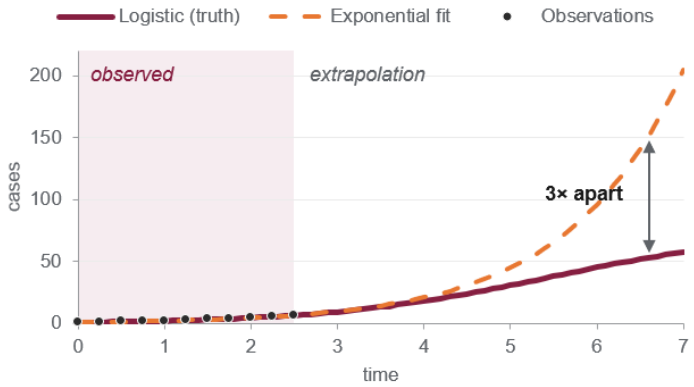


Figure: Illustrative example of Logistic growth from ODEBench dataset (dx/dt = 0.79x{1 − x/74.3}) and an exponential fit agree on the given data range to within ~1% but diverges later highlighting the identifiability gap.

Discovery is not just curve fitting; it is about *acquiring the right data to identify the governing law*

Closing the Loop Between Hypotheses and Data

- LLM-guided methods inject scientific priors into symbolic search but **still treat discovery as a passive regression task over a fixed dataset**.
- Active methods adaptively acquire new trajectories, but without using structured symbolic hypothesis spaces to guide experimentation.
- However, **existing active methods are data-intensive** requiring large amounts of data to identify the underlying governing laws.

Question: Can symbolic hypotheses themselves guide adaptive data acquisition toward more data-efficient, identifiable dynamical-system discovery?



We introduce LLM-ACES

A closed-loop framework that jointly performs symbolic hypothesis construction and adaptive trajectory acquisition, letting the hypothesis space and dataset co-evolve.

LLM-ACES | Framework

The hypothesis space and the acquired dataset are co-evolved resulting in data-efficient discovery

1 LLM-Induced Priors

The LLM proposes a set of operator subset based on the task metadata and prior feedback

2 Fit Hypotheses

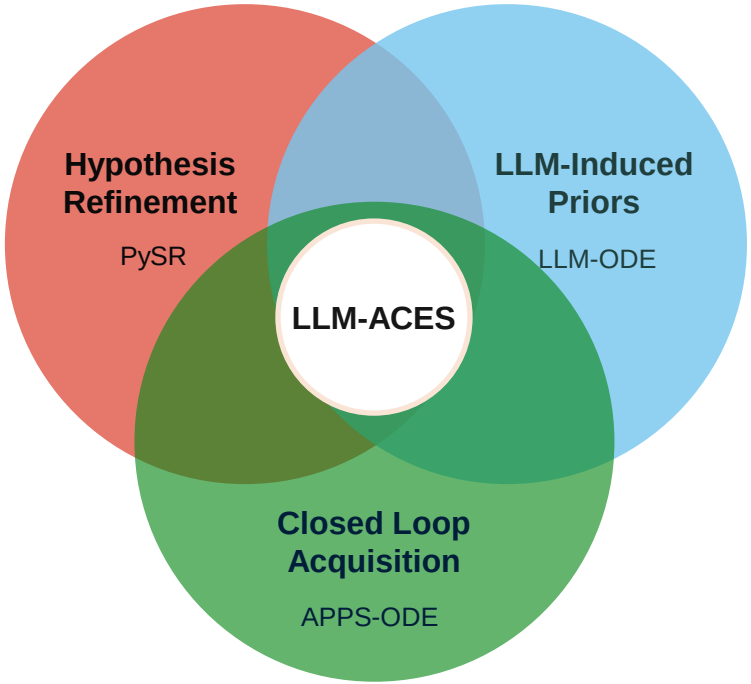
Symbolic Regression generates candidate equations within each operator sub-space

3 Data Acquisition

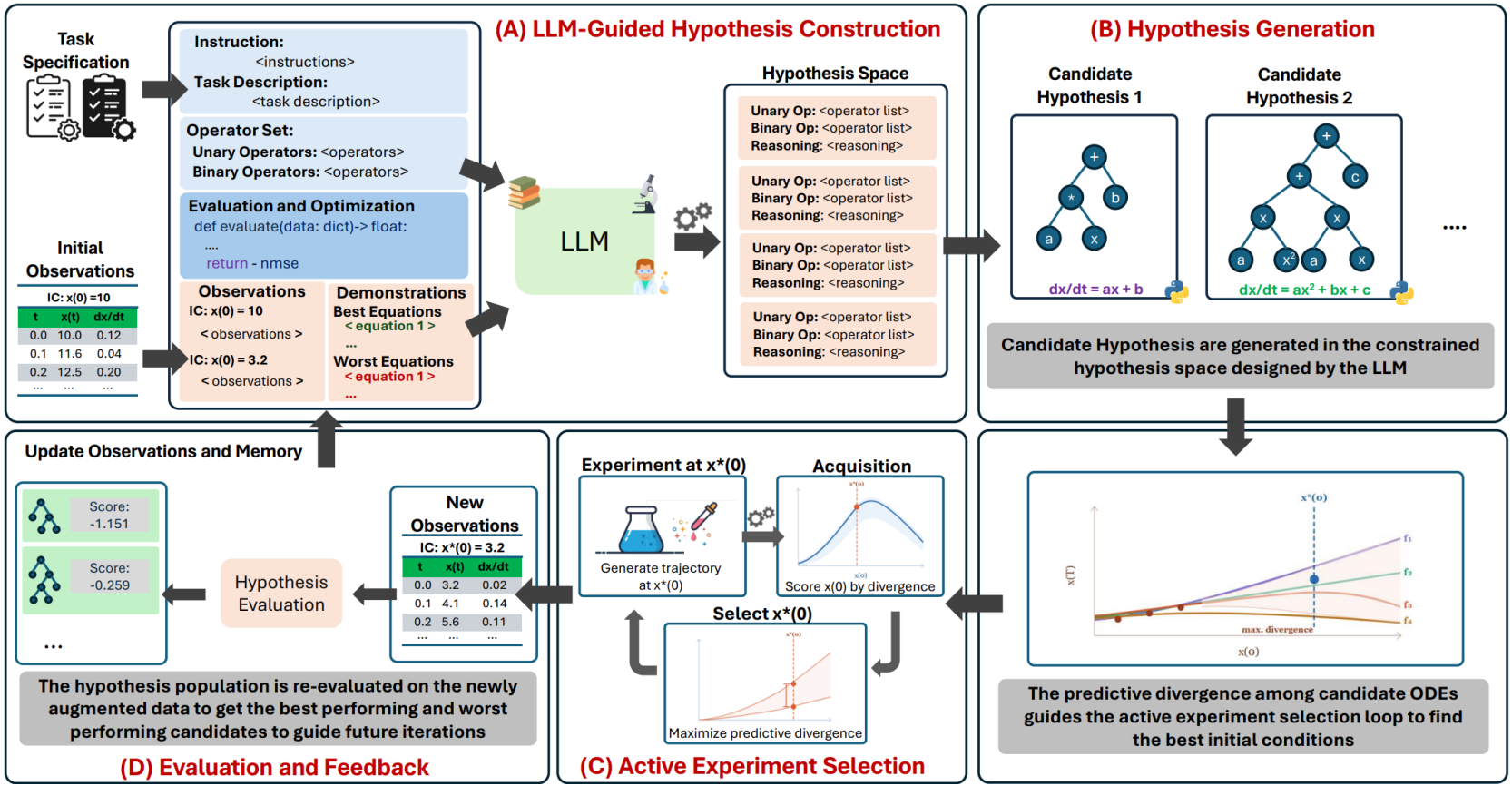
New trajectories are queried exactly where the candidate hypotheses disagree the most

4 Score and Update

The experience buffer closes the loop by storing the most effective and least effective candidates guiding future iterations

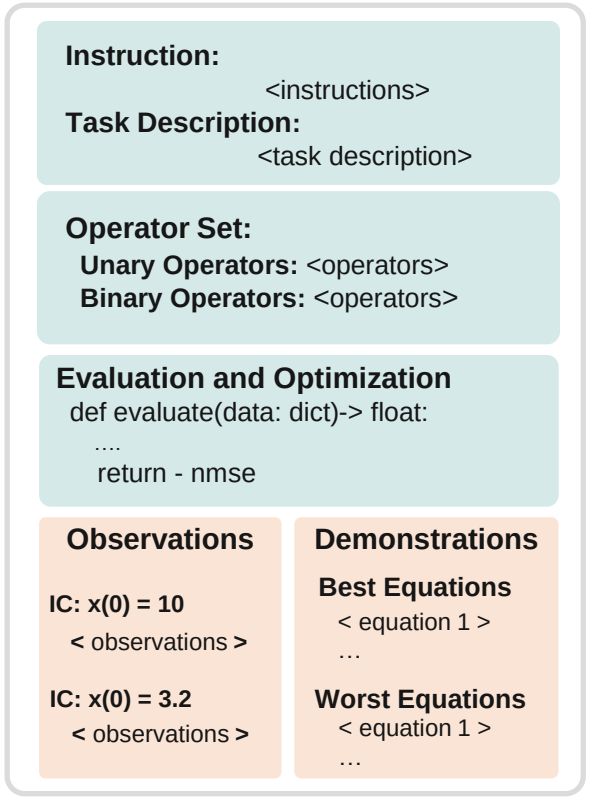


LLM-ACES | Framework



LLM-ACES | Hypothesis Construction

- **Instruction:** Define the role and instruct the LLM to generate ODEs based on the task.
- **Task description:** Description of the task, e.g., 1D ODE system.
- **Operator Set:**
  Unary Operators: [‘cos’, ‘sin’, ‘exp’, ‘log’, ‘sqrt’, ‘2^’, ‘tanh’, ….]
  Binary Operators: [‘+’, ‘*’, ‘/’, ‘-’ ‘^’]
- **Evaluation:** The evaluation function used to measure the quality of the predicted curves.
- **Demonstrations:** Helps understand positive and negative samples to guide the generation process.
- **Observations:** Initial data along with the queried samples stored after every iteration.



LLM-ACES | Hypothesis Generation

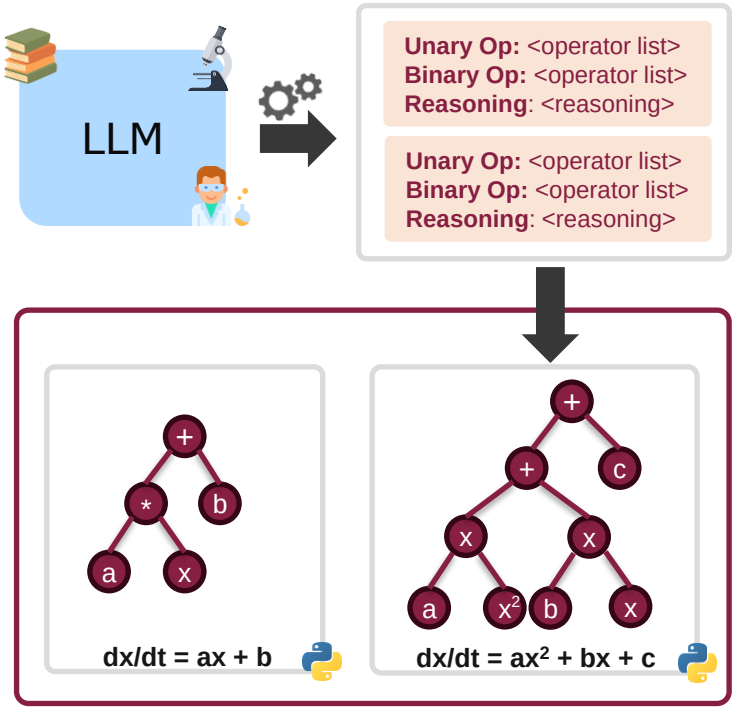
Use the LLM's **domain knowledge** and **experience buffer** to proposes K hypothesis spaces containing a set of operator priors each.

✓ **Exploitation Space:**
The LLM generates the operator priors that have previously proved to be high-performing regions.

✓ **Exploration Space:**
The LLM is then instructed to produce diverse hypotheses to explore the operator space and guide the search.

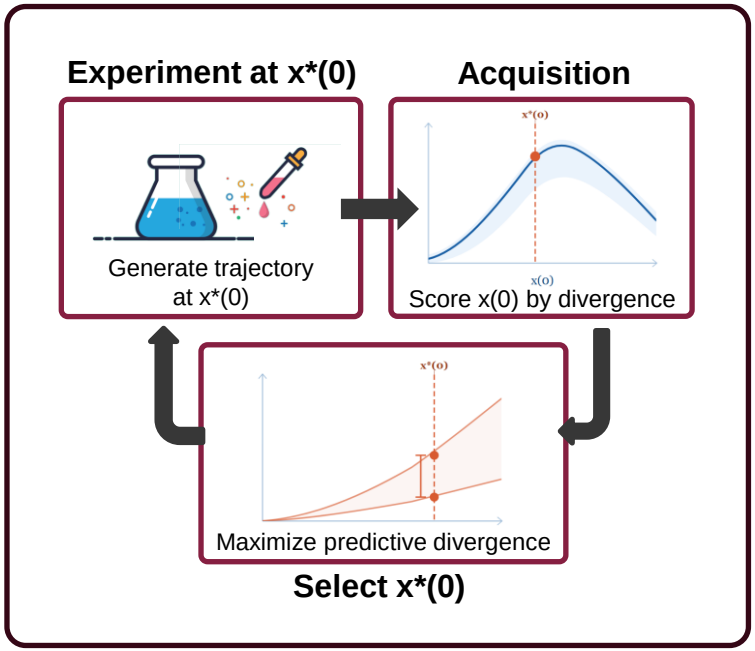
Hypothesis Generation

- The LLM generated operator space reduces the huge operator space to limit the search to only task-specific operator space.
- Symbolic regression fits candidates with the LLM generated space to generate final equations.



LLM-ACES | Active Data Acquisition

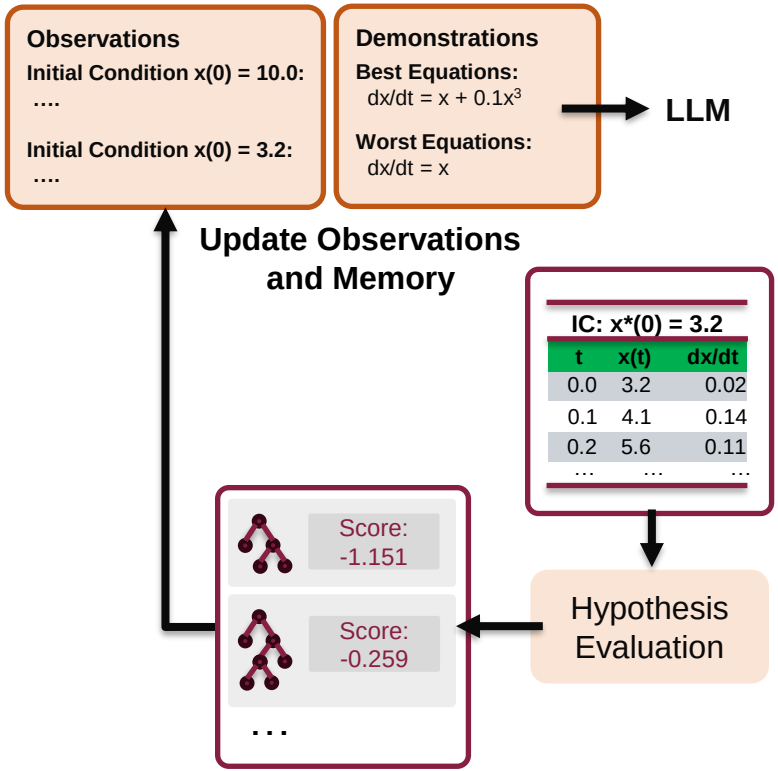
- **Score candidates**: For each candidate initial condition x(0), simulate short trajectories under every currently fitted equation and score the candidate by how much those trajectories diverge from one another.
- **Maximize predictive divergence**: A Gaussian Process models this divergence score over the candidate space to balance exploiting high-scoring regions and exploring unknown ones.
- **Select x(0)**: The candidate with the highest predicted divergence is chosen as the next experiment.
- **Query the oracle**: A real trajectory is generated producing new labeled (u, du) data.
- **Close the loop**: The new trajectory is added into the training set, the equations are refit.



LLM-ACES | Experience Management

To better navigate the search space, we store previously generated samples along with the new observations derived from the data acquisition stage.

- **Newly Acquired Data**: We acquire the data from the oracle based on the highest disagreement among the generated hypotheses.
- The dataset from the previous iteration is appended with the newly acquired data.
- **Bootstrap Hypothesis Evaluation:** For better generalization, we evaluate all the generated hypothesis on the new dataset and score the hypotheses based on their respective performance.
- **Update the Memory and Observations:** We update the memory based on the evaluation and add the new dataset to the LLM context.



LLM-ACES | Experimental Setup

Datasets

ODEBench

63 systems spanning 1D - 4D state spaces from physics, mathematics and biology

ODEBase

59 dynamical systems spanning 2D & 3D state spaces from biology

Evaluation Metrics

Normalized MSE

Measures how accurately the predicted trajectory reproduces the ground truth.

Symbolic Accuracy

Measures the functional form recovery of the predicted equation.

Complexity

Measures the parsimony and interpretability of symbolic representations.

Evaluation Protocol

Reconstruction

Train & evaluate on training initial condition, $t \in [0,1]$ for 100 points.

Generalization

Evaluate on the held-out initial condition over the same interval, $t \in [0,1]$ for 100 points.

Out-of-Distribution (OOD)

Extrapolate to $t \in (1,10]$, 150 steps, from the training initial condition.

All baselines receive the same amount of data samples

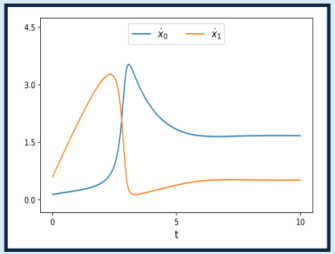
LLM-ACES | Quantitative Results

Method	ODEBench					ODEBase				
	Recon. ↓	Gen. ↓	Ext. ↓	Comp.	SA (%) ↑	Recon. ↓	Gen. ↓	Ext. ↓	Comp.	SA (%) ↑
Passive Symbolic Discovery										
SINDy	5.07e-04	5.09e-01	1.41e+00	11.7	20.1	1.06e-03	1.12e+00	3.47e-01	12.5	10.2
Operon	7.95e-05	2.57e-01	1.84e+00	14.0	4.0	1.49e-05	1.46e+00	1.41e+00	19.0	5.4
PySR	2.84e-03	8.82e-01	1.43e+00	6.6	18.7	1.37e-03	1.08e+00	1.35e+00	8.8	5.9
E2E	3.69e-01	1.49e+00	2.19e+00	52.6	0.0	1.04e+00	1.19e+01	1.98e+01	83.6	0.0
ODEFormer	4.61e-03	3.83e-01	2.04e+00	15.9	17.1	1.64e-02	1.15e+01	4.09e+00	18.5	4.8
LLM-guided Symbolic Discovery										
LLM-only	2.20e-08	3.45e+00	1.79e+00	38.9	0.0	4.52e-08	1.06e+01	1.11e+00	54.9	0.0
LLM-ODE	4.12e-05	4.72e-03	2.43e-02	25.2	9.5	7.46e-05	8.51e-01	6.59e-04	32.1	3.4
Active Symbolic Discovery										
Query-by-Committee (QBC)	1.81e-09	5.07e-09	5.69e-08	33.7	22.6	4.97e-09	7.47e-06	6.38e-06	45.2	21.5
Bayesian Optimization (BO)	1.06e-14	9.35e-13	3.47e-10	22.6	62.3	8.55e-10	4.61e-10	5.43e-08	31.3	65.0
APPS-ODE	7.52e-01	8.13e-01	1.02e+00	13.4	2.6	5.56e-01	9.12e-01	1.01e+00	10.9	15.2
LLM-ACES (GPT)	**1.33e-17**	**8.28e-17**	**2.46e-16**	17.1	**70.8**	2.54e-14	8.50e-10	3.05e-12	**33.1**	66.1
LLM-ACES (Qwen)	6.30e-16	4.18e-15	1.89e-15	**18.2**	69.2	**3.70e-15**	**4.09e-15**	**3.44e-13**	31.2	**69.5**

Table: Performance comparison of baselines on (a) ODEBench and (b) ODEBase datasets

LLM-ACES | Qualitative Results

Schnackenberg



$$\dot{x}_0 = 0.24 + x_0^2 x_1 - x_0$$
$$\dot{x}_1 = 1.43 - x_0^2 x_1$$

PySR

$$\dot{x}_0 = 0.1$$
$$\dot{x}_1 = 1.38$$

SINDy

$$\dot{x}_0 = -0.42 + 5.42x_0 - 0.39x_1$$
$$\dot{x}_1 = 2.09 - 6.42x_0 + 0.39\,x_1$$

Operon

$$\dot{x}_0 = 0.02x_0 + 0.1$$
$$\dot{x}_1 = 1.57 - 1.02x_0$$

ODEFormer

$$\dot{x}_0 = 0.72x_0 - 0.01(-0.43 - 7.14x_0)^2$$
$$\dot{x}_1 = 15.43x_0 - 1.2x_1$$

LLM-ODE

$$\dot{x}_0 = -0.19 - 0.38x_0 - 0.1x_1 - 0.04{x_0}^2 + 0.43\,x_0x_1 + 0.01{x_1}^2$$
$$\dot{x}_1 = 1.47 - 0.5x_0 + 0.09\,x_1 - 0.46\,x_0x_1$$

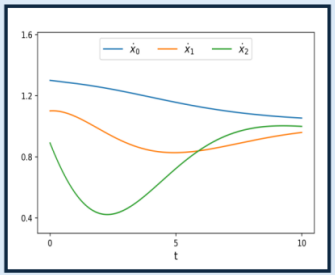
APPS-ODE

$$\dot{x}_0 = 0.1673$$
$$\dot{x}_1 = -1.4107 - 0.8515x_0^2$$

LLM-ACES

$$\dot{x}_0 = 0.24 + x_0^2x_1 - x_0$$
$$\dot{x}_1 = 1.43 - {x_0}^2\,x_1$$

Maxwell Bloch



$$\dot{x}_0 = 0.1(-x_0 + x_1)$$
$$\dot{x}_1 = 0.21(x_0x_2 - x_1)$$
$$\dot{x}_2 = 0.34(-0.31x_0x_1 + 4.1 - x_2)$$

PySR

$$\dot{x}_0 = -0.1x_0 \qquad \dot{x}_1 = 5.73 - \frac{7.43}{x_0}$$
$$\dot{x}_2 = -0.46x_2$$

SINDy

$$\dot{x}_0 = -0.01x_0 + 0.10x_1$$
$$\dot{x}_1 = -0.18 + 0.16x_0 - 0.24x_1 + 0.27x_2$$
$$\dot{x}_2 = 2.48 - 0.81x_0 - 1.39x_1 - 0.36x_2$$

Operon

$$\dot{x}_0 = 0.05x_0 - 0.08$$
$$\dot{x}_1 = 0.26\,x_2 - 0.22$$
$$\dot{x}_2 = 11.65 - 9.28\,x_0$$

ODEFormer

$$\dot{x}_0 = 16.15\sin(-1.17 + 8.26x_0) - 0.62(-1.04 - 0.08x_0)^{-1}$$
$$\dot{x}_1 = -0.06x_1 \qquad \dot{x}_2 = -0.66\,x_2^2$$

LLM-ODE

$$\dot{x}_0 = -0.09x_0 + 0.07x_1 - 0.01x_2 + 0.02x_0x_1 + 0.01x_1x_2$$
$$\dot{x}_1 = -0.03x_0 + 0.06x_1 + 0.3x_2 - 0.01x_0x_2 - 0.2x_0x_1$$
$$\dot{x}_2 = 1.55x_0 - 1.84x_1 - 0.16x^2 + 0.30x_1x_2 - 0.48x_0x_1$$

APPS-ODE

$$\dot{x}_0 = -0.3409x_0x_1$$
$$\dot{x}_1 = -x_0x_1 + 0.9585x_0$$
$$\dot{x}_2 = 0.2479$$

LLM-ACES

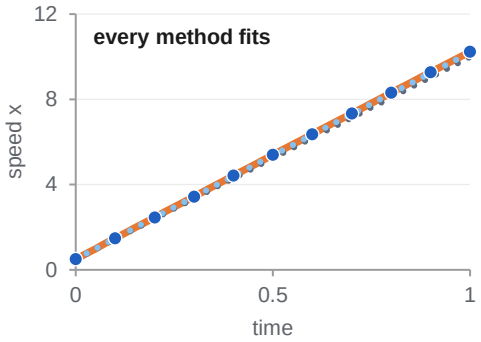
$$\dot{x}_0 = 0.10x_1 - 0.1x_0$$
$$\dot{x}_1 = 0.21x_0x_2 - 0.21\,x_1$$
$$\dot{x}_2 = -1.05x_0x_1 - 0.34x_2 + 1.38$$

Figure: LLM-ACES correctly identifies the underlying governing laws across dimensions and datasets

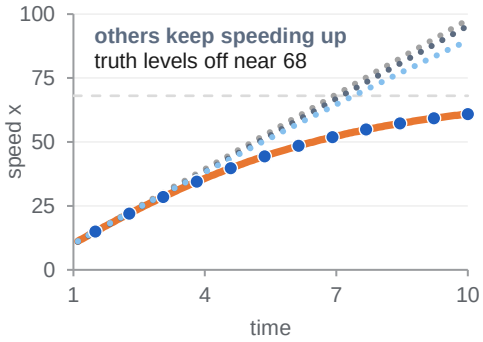
LLM-ACES | A Worked Example: Falling Object

A falling object speeds up because of gravity, but air resistance grows with speed, so it levels off at a top speed (about 68).
True law: $dx/dt = 9.81 - 0.0021x^2$ (x = speed). The initial condition determines the speed (x) at time t=0.

The first second (training data)
the only data the other methods trained on



What happens later
seconds 1–10, an extrapolation





What each method learned

Method	Learned law (dx/dt)	Verdict
PySR	9.73 (a constant)	✗ never stops speeding up
Multi-IC PySR	9.49 (a constant)	✗ never stops speeding up
LLM-only	9.8 − 0.019x	✗ drag far too weak
LLM-ACES	**$9.81 - 0.0021x^2$**	**✓ exact**

1 The trap

In the first second the object only reaches speed 10, where air resistance is tiny. **"Constant speed-up" fits the data perfectly.**

2 Ask where they disagree

The candidate laws disagree at high speed, where drag matters. **LLM-ACES runs its acquisition there** instead random conditions like in Multi-IC PySR.

3 Recover the law

The new runs show the object leveling off, which only the drag law explains. **LLM-ACES returns $9.81 - 0.0021x^2$ exactly.**

Our Findings | LLM-ACES is more efficient

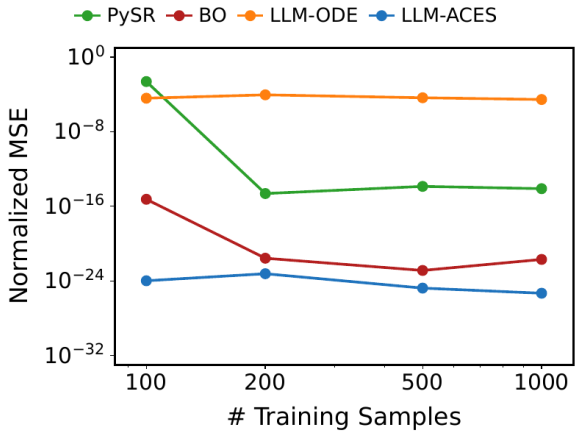


Figure: Geometric-mean NMSE over 15 ODEBench systems as the observation budget grows.

Method	Recon. NMSE	Wall-clock
PySR, 750 iterations	2.12e-03	625 s
LLM-ODE, ~1K LLM calls	3.82e-05	1,683 s
LLM-ACES (T = 1), 3 LLM calls	**1.79e-16**	**251 s**

Table: Compute efficiency comparison

Sample Efficiency

- LLM-ACES has the lowest error at every budget, and with only 100 samples it beats every baseline given up to 1,000.
- More data alone doesn't recover the law: what matters is collecting the *right* data, the trajectories that tell competing hypotheses apart.

Compute Efficiency

- LLM-ACES is more accurate at a fraction of the cost: 251 s vs. 625 s for PySR and 1,683 s for LLM-ODE.
- One acquisition round needs only 3 LLM calls (vs. ~1,000 for LLM-ODE) and still reaches 10^{-16} error, against 10^{-3} for PySR and 10^{-5} for LLM-ODE.

Findings | Where You Sample Matters Most

Predictive Divergence produces the best fit

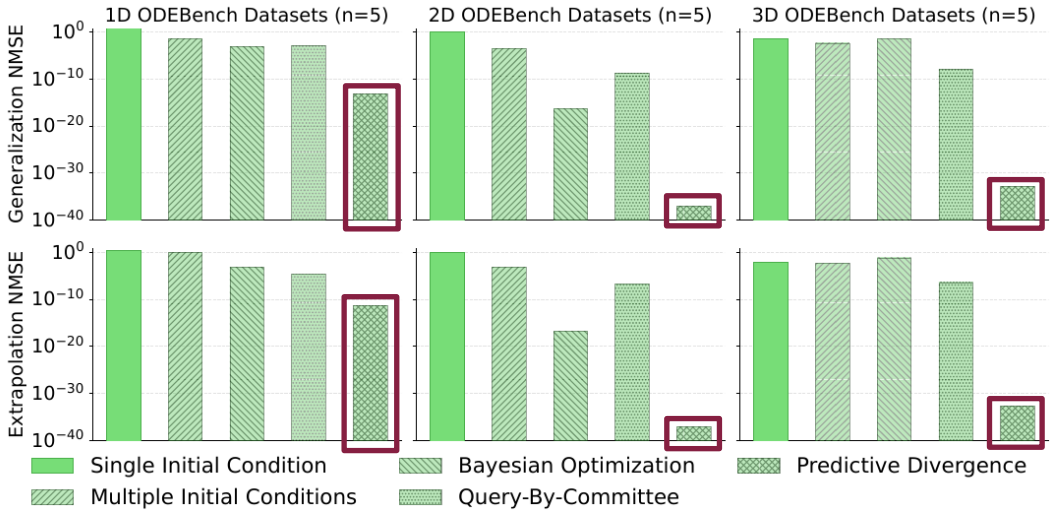


Figure: Same PySR backend and budget, five ways to choose trajectories; 5 ODEBench systems per dimension.

Random coverage helps only modestly

Spreading the budget over random initial conditions beats a single trajectory but stays orders of magnitude behind.

Discriminate, don't just explore

BO and QBC chase uncertainty. Predictive divergence seeks the experiment that separates competing laws: lowest NMSE in 1D, 2D and 3D.

Remove it and systems are lost

With uniform sampling instead, LLM-ACES has the highest median error on every split and recovers 23/30 systems (vs. 26/30).

"The central challenge is not obtaining more data but obtaining the right data."

Findings | LLM Priors Shrink the Search Space

The LLM proposes small operator subsets (2–6 unary operators per dimension) conditioned on the system and the evidence so far. **Is it the LLM, or just a smaller space?**

We provided baselines the true operator families

Method	Full operator set	Oracle-informed set
SINDy	5.07e-04	6.23e-04
PySR	2.84e-03	2.34e-03
Operon	7.95e-05	7.73e-05
LLM-ACES	**1.33e-17**	

Restriction alone barely moves them (PySR improves ~18%); all stay **>10 orders of magnitude** behind.

Replace the LLM inside LLM-ACES

Operator selection	Recon.	Gen.	Ext.
Oracle library (fixed)	2.53e-18	2.12e-17	1.02e-17
UCB bandit (adaptive)	7.67e-17	2.70e-17	6.40e-17
LLM priors (ours)	**4.01e-19**	**7.39e-18**	**2.29e-18**

Adapting from scalar rewards is not enough: the bandit is **190× worse** on reconstruction and trails even the fixed oracle library.

Operator space restriction does not work, the LLM tailors the search space to the system and the evidence.

LLM-ACES | Robustness & Consistency

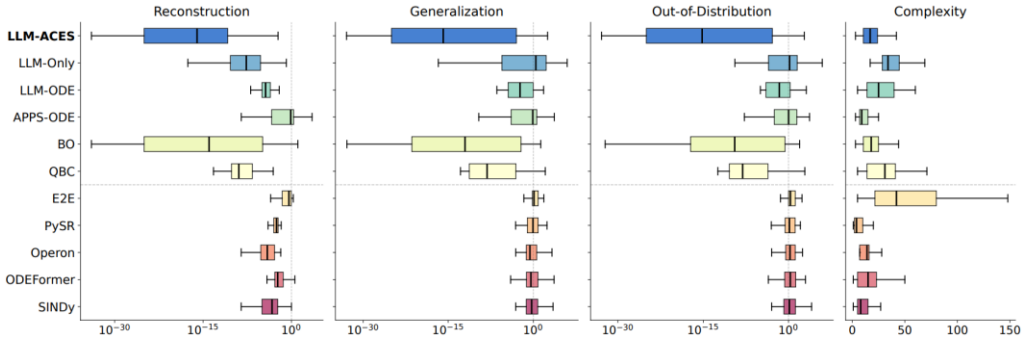


Figure: Distribution of reconstruction, generalization, out-of-distribution NMSE, and expression complexity across methods for ODEBench datasets.

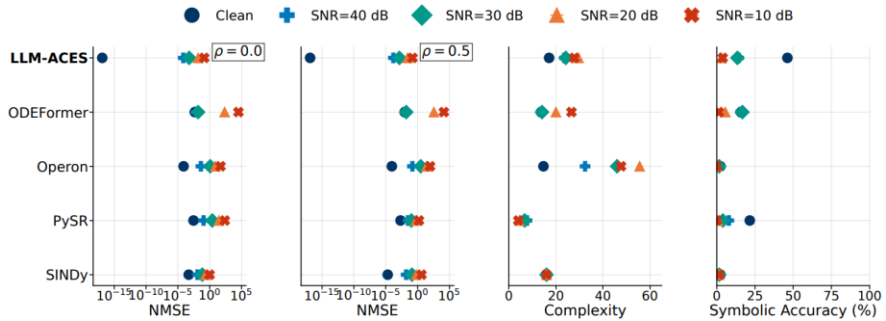


Figure: Performance across irregular sampling and noisy samples across baselines on ODEBench datasets.

38 / 63

ODEBench systems fully recovered

Every component equation correct. The strongest baseline (BO) recovers 28 / 63; the best passive method recovers only 6/63 systems.

- Robust to noise (down to 10 dB SNR)
- Robust to sparse, irregular sampling
- Broad gains: lower Q1–Q3 error
- Lowest error in every setting

Summary

- LLM-ACES **tightly couples LLM-guided hypothesis construction with adaptive trajectory acquisition,** so hypotheses and data co-evolve.
- LLMs build domain-informed operator priors that **shrink the search space**; neither an oracle operator library nor a UCB bandit reproduces the gain.
- Querying where hypotheses **disagree most** gets the right data: with 100 samples, LLM-ACES beats baselines given 5–10× more.
- Highest **symbolic accuracy** on ODEBench (70.8%) and ODEBase (69.5%), 38/63 systems fully recovered, **orders of magnitude lower NMSE**, robust to noise.

Future Directions

- Scaling acquisition to surrogate models under tighter query budgets and extending closed-loop discovery beyond ODEs to PDEs and coupled multi-physics systems.
- Expanding such dynamic discovery paradigms to more AI4Science systems to enable more interpretable, data-efficient and reliable scientific discovery.

Thank You