VLM4Physics: Equation Discovery Using Multimodal Inputs

Qianshu Ye [1] **Jian Cheng Wong** [2] **Chin Chun Ooi** [2] **Yew-Soon Ong** [1]

[1]*College of Computing and Data Science, Nanyang Technological University* [2]*Institute of High Performance Computing, A*STAR*. Correspondence to: Qianshu Ye qye010@e.ntu.edu.sg.

1. Introduction

Scientists study natural phenomena through hypothesis formulation, analytical reasoning, and empirical validation — a longstanding yet time-intensive process. As problems grow in scale and complexity, AI models show potential to accelerate equation discovery. Compared to traditional genetic programming, which primarily accommodates structural priors [1, 2], Large Language Models (LLMs) can incorporate domain context, integrate multimodal inputs, and efficiently explore a richer hypothesis space. This flexibility greatly improves their search efficiency and efficacy beyond symbolic regression [3].

That said, in complex multivariable systems—especially under noise—LLMs can be unstable and underperform symbolic approaches. This limits their usefulness in realistic settings. Most prior work emphasizes contextual information [4, 5], and visual reasoning has not been effectively integrated in prior work [6]. We argue that vision language modeling (VLM) can further improve graphical fit and optimization stability.

We introduce VLM4Physics (Fig. 1), a multimodal equation discovery framework built on three principles: (1) VLM-based reasoning to capture structural nuances; (2) Informed coefficient initialization to stabilize optimization; (3) Explicit contextual priors to guide hypothesis generation.

Together, these mechanisms construct a cohesive search process that provides better starting points, improved convergence, and more reliable structural recovery, as demonstrated on 3 nonlinear and coupled dynamical systems.

2. Discussion

2.1 Related Work

Recent LLM-based methods frame equation discovery as program synthesis, where symbolic structures are proposed under highly specific contexts followed by numerical optimization [3, 4]. Our approach identifies key factors that hinder convergence under empirical conditions, and incorporate multimodal inputs to improve inferential robustness for complex dynamical systems.

2.2 Methodology

VLM4Physics follows an iterative generation–verification–refinement pipeline. A structured prompt instructs the LLM to generate executable equation skeletons under explicit constraints (e.g., no loops or conditionals), restricting the symbolic search space to physically plausible forms.

Part 1: Hypothesis Generation

At each iteration, a structured metaprompt conditions the LLM/VLM to generate a batch of candidate equation skeletons with unknown coefficients. The specific additional inputs include: (1) time series and phase plots, (2) coefficient initialization guidance, and (3) contextual priors on the physical meaning of the system (e.g., predator–prey populations). The full prompt is provided in Appendix B.

The output space is strictly constrained to syntactically valid functions of ordinary differential equations (ODEs). This promotes meaningful hypotheses and allow seamless implementation within the LLM-guided evolutionary loop.

Part 2: Optimization and Evaluation

Candidate equation skeletons are evaluated by solving the corresponding initial value problems to compared predicted trajectories against observed data, and equation coefficients are optimized with BFGS or Nelder Mead to minimize the mean squared error (MSE). LLM/VLM-informed initialization effectively prevents stiffness during convergence, and enhances the ranking and refinement of hypotheses.

High performing candidates are stored in an experience log and reused in subsequent iterations, forming an Optimization by PROmpting (OPRO) loop. The predicted graphs of the highest performing equation are also provided to the VLM. Such a framework allows equation structures to evolve while maintaining diversity and parsimony.

2.3 Evaluation and Results

We benchmark VLM4Physics on 3 canonical systems from the Feynman dataset. Equations 1 and 2 describe damped oscillators, while Equation 3 represents the Lotka Volterra(L-V) system [7]. Full equation details are provided in Appendix A. All experiments use GPT 4.1 with a fixed temperature of 0.8.

We conduct an ablation study across three types of configurations: (1) contextual priors, including variable units and a brief system description; (2) VLM reasoning, which adds unlabeled trajectory and phase plots; and (3) coefficient initialization, where the LLM/VLM proposes starting values. Components are introduced individually and incrementally to isolate their effects on the performance, resulting in seven experiments in total (Table 1).

Each experiment runs for five iterations across three trials. We evaluate structural convergence (recovery of key equation terms) and graphical fit of trajectories (with a quantitative loss threshold of

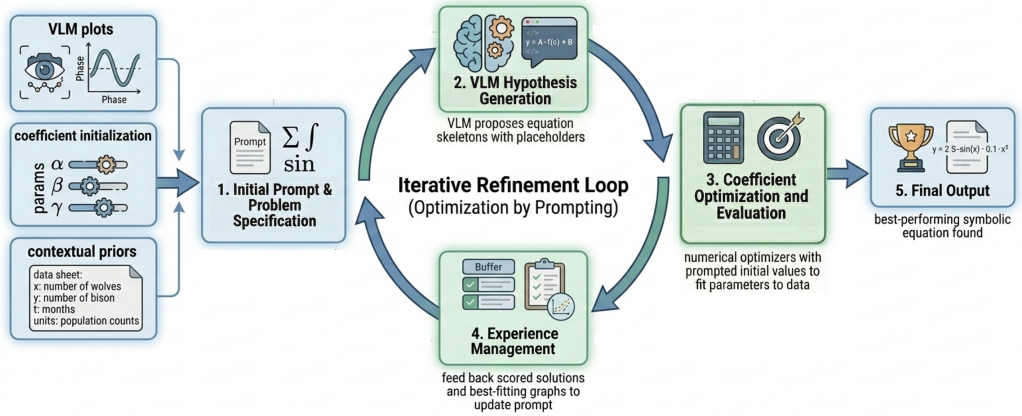

Fig. 1: Overview of VLM4Physics workflow. The VLM extracts structural cues from data visualizations to inform equation terms (Step 1) and provides feedback for iterative refinement (Step 4). Textual priors guide hypothesis generation/evolution (Step 2), while suggested initializations help enhance coefficient optimization (Step 3).

Experiment	Contextual Priors	VLM Reasoning	Coefficient Init	Equation 1		Equation 2		Equation 3	
				Struct.	Graphic	Struct.	Graphic	Struct.	Graphic
A				●	●	●	●	●	●
B	✓			●	●	●	●	●	●
C		✓		●	●	●	●	●	●
D			✓	●	●	●	●	●	●
E	✓		✓	–	–	●	●	●	●
F		✓	✓	–	–	●	●	●	●
G	✓	✓	✓	–	–	●	●	●	●

Table 1: Ablation results with “Structural Accuracy” and “Graphic Fit” reported under varying settings.

1×10^{-2}). Green markers indicate successful convergence, while red markers denote failure.

2.4 Discussion

VLM is critical for structural recognition and informative evolution. In Figure 2, for the L-V system, only Experiments F and G achieve graphical fit, i.e. after adding VLM reasoning to coefficient initialization. This demonstrates that visual inputs are essential for capturing interaction patterns and latent coupling structures. The visual modality reinforces structural mutations and steers the search toward geometrically coherent hypotheses, thereby promoting stable convergence in complex systems.

Coefficient initialization is necessary but not sufficient for convergence. In Figure 2 and Figure 6 (Appendix B), Experiment A–C show stagnant losses across iterations, visible as flat dotted trajectories compared to the decreasing solid curves. Notably, Exp B and C recover correct expressions, yet evolution does not progress. This is especially pertinent to nonlinear systems, where coefficient-sensitive terms can induce instability or vanishing gradients, thereby complicating effective optimization. However, without evolving contextual information, good starting values alone cannot guarantee successful recovery.

Additional observations further clarify these effects. In Figure 2, Experiment E does not reach graphical fit , yet its loss stably converges by iteration seven, illustrating effective but slower evolution. Equation 1 has simple structures with the correct skeleton identified early, and the ground truth equation is immediately recovered once initialization is introduced (Table 1). These results reinforce that initialization enables evolution, while VLM enables discovery in structurally complex systems. Further details are discussed in Appendix C.

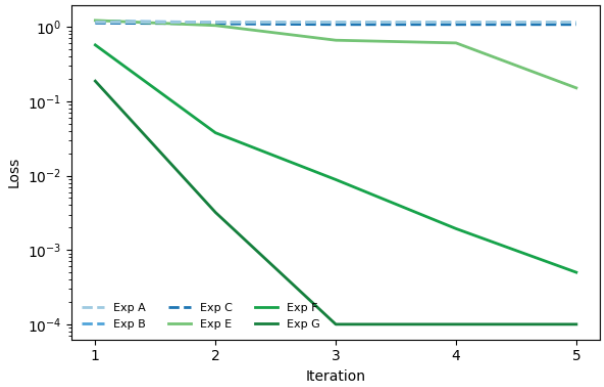

Fig. 2: Loss Trajectories of Equation 3, with results of Experiment A, B, C in dotted blue lines, and E, F, G in solid green lines.

Acknowledgments

This project was supported by Nanyang Technological University under the URECA Undergraduate Research Programme.

The authors acknowledge the use of a large language model tool to support language refinement and readability improvement in this manuscript. The LLM was used solely for editorial assistance.

References

[1] Jie Zhong, Liang Feng, Wen Ting Cai, and Yew-Soon Ong. Multifactorial genetic programming for symbolic regression. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–14, 2018.

[2] Ming-Yang Zheng, Y. Wang, J. Zhong, and J. Zhang. Discovering Infinite Recursive Conjectures Through Genetic Programming. *IEEE Transactions on Evolutionary Computation*, 2025.

[3] Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific Equation Discovery via Programming with Large Language Models. *arXiv preprint arXiv:2404.18400*, 2024. ICLR 2025 Oral.

[4] Zhilong Song, Qionghua Zhou, Chunjin Ren, Chongyi Ling, Minggang Ju, and Jinlan Wang. LLM-Feynman: Leveraging Large Language Models for Universal Scientific Formula and Theory Discovery. *arXiv preprint arXiv:2503.06512*, 2025.

[5] Yimeng Chen, Piotr Piękos, Mateusz Ostaszewski, Firas Laakom, and Jürgen Schmidhuber. PhysGym: Benchmarking LLMs in Interactive Physics Discovery with Controlled Priors. *arXiv preprint arXiv:2507.15550*, 2025.

[6] Matteo Merler, Katsiaryna Haitsiukevich, Nicola Dainese, and Pekka Marttinen. In-Context Symbolic Regression: Leveraging Large Language Models for Function Discovery. *arXiv preprint arXiv:2404.19094*, 2024.

[7] Peter J Wangersky. Lotka-volterra population models. *Annual Review of Ecology and Systematics*, 9:189–218, 1978.

Appendix A: Equation Setup

Here are the three equations used to generate the dataset and train the model. In setting up the prompt, each equation is solved by calling the solve_ivp function from SciPy. The solution data points were then organized and formatted into a string to pass into the model. For Experiment C,F, and G, the solution were also presented as time series and/or phase plots, which were also fed to the model.

$$\ddot{x} = -x - \alpha v \tag{1}$$

$$params = [0.1] \tag{1.1}$$

$$\ddot{x} = F\sin(\omega t) - \alpha v^3 - \beta x v - \delta x e^{\gamma x} \tag{2}$$

$$params = [0.3,\ 0.5,\ 1.0,\ 5.0,\ 0.5,\ 1.0] \tag{2.1}$$

$$\begin{cases} \dot{x} = \alpha x - \beta xy \\ \dot{y} = \delta xy - \gamma y \end{cases} \tag{3}$$

$$params = [1.5,\ 1.0,\ 1.75,\ 1.0] \tag{3.1}$$

Appendix B: Initial Prompt

The prompt used to generate the equation skeletons is reported in Figure 3. Additional instructions are inserted accumulatively as conditions are applied. The incremental prompt to activate coefficient initialization is recorded in Figure 4, while the contextual priors are recorded in Figure 5.

Sample Initial Prompt

```
Instruction:
You are a helpful research assistant tasked with
discovering mathematical function structures
for scientific systems.  Complete the equation
function below, considering the physical meaning
and relationships of inputs.Output ONLY code.

Problem specification:  Find the mathematical
function skeletons that govern a system, given
observations of the variables.  The goal is to
infer the forms of the time derivatives of each
state variable based on the data.

Constraints:
· No conditionals
· No loops inside equation
· use Numpy

Output Format:
· Generate EXACTLY 5 distinct equation skeletons
· Each skeleton must be a separate Python code
block
· Do not include any text outside the quoted code
blocks
· Return all 5 code blocks in a single array
· each item in the array is a single executable
Python code block
· The function name must be ‘equation‘
· avoid duplicate equation skeletons

Evaluation:
Fitness = mean squared error between predicted
and observed x,y values.

IMPORTANT:
You must improve the top performing equations
using mutation and crossover.

Task:
Generate 5 distinct equation skeletons for the
system.  Use params[i] as placeholders.  Avoid
duplicate structures.
```

Fig. 3: Sample initial prompt used in our experiments

Coefficient Initialization

```
After each code block, include a line:  params
= [p0, p1, ..., pN] with specifically chosen
numeric initial values that best fits the data.
E.g:
params = [1.0, 0.5, 0.2]
```

```
Propose more precise initial values based on the
performance/graphs of past equations.
```

Fig. 4: Sample prompt added for coefficient initialization

Contextual Priors

```
Contextual priors:
x:  displacement (meters)
v:  velocity (m/s)
t:  seconds
Description:  this system describes a driven
oscillator.
```

Fig. 5: Sample prompt added for contextual priors

Appendix C: Loss Trajectories and Results

Figure 6 shows the loss trajectories of Equation 2, the nonlinear driven oscillator, and reveals slightly different responses to the ablations compared to Equation 3. Notably, Experiment D achieved lower losses than F, despite the latter having more accurate skeletons. This may be because Equation 2 is nonlinear and sensitive to coefficient values; thus, poorly initialized nonlinear terms likely led to instability and worse performance than uninformed guesses. For similar reasons, experiments with contextual priors (E and G) seem to perform better than those using VLM (F), although VLM remains more practically applicable in settings where full contextual information is unavailable.

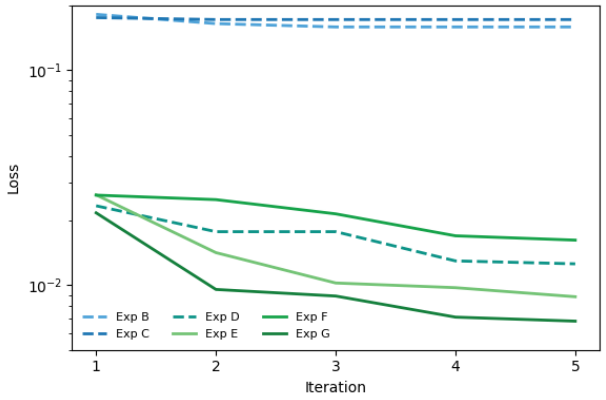

Fig. 6: Loss Trajectory of Equation 2

Nonetheless, all three equations were correctly inferred by Experiment G within five iterations, as evidenced by the graphical fit between predicted and actual variable values in Figures 7, 8, and 9.

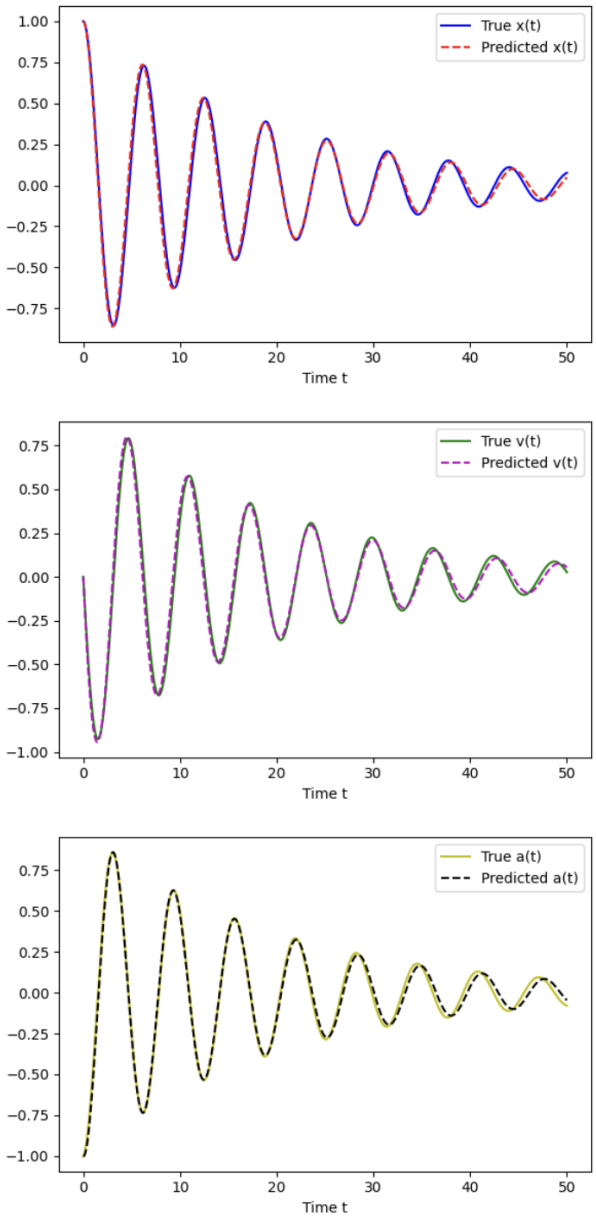

Fig. 7: Predicted vs Ground Truth Plots of Equation 1

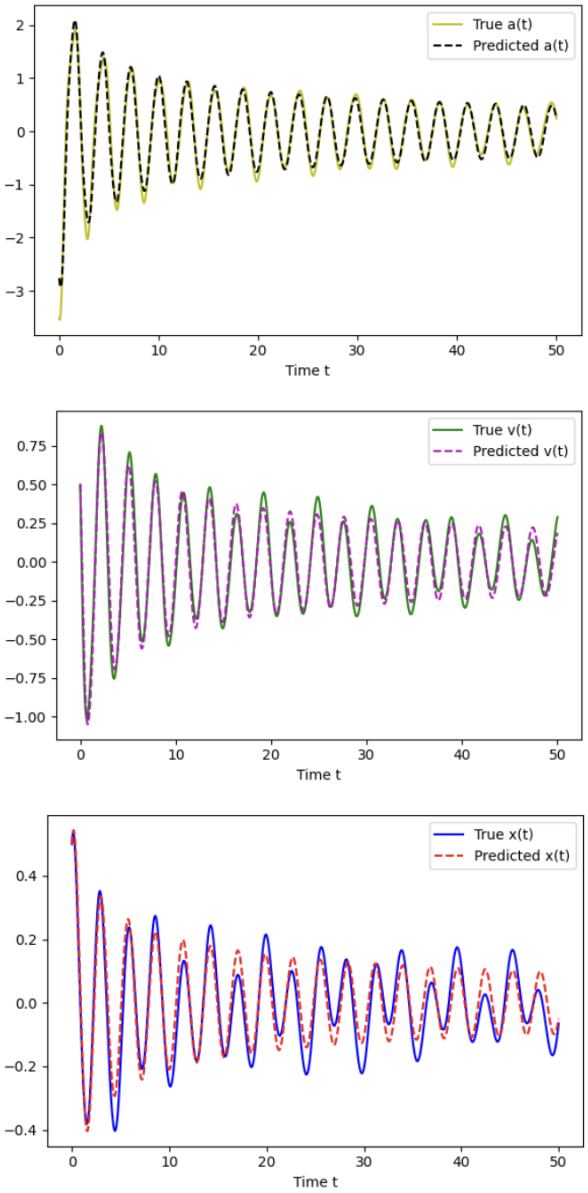

Fig. 8: Predicted vs Ground Truth Plots of Equation 2

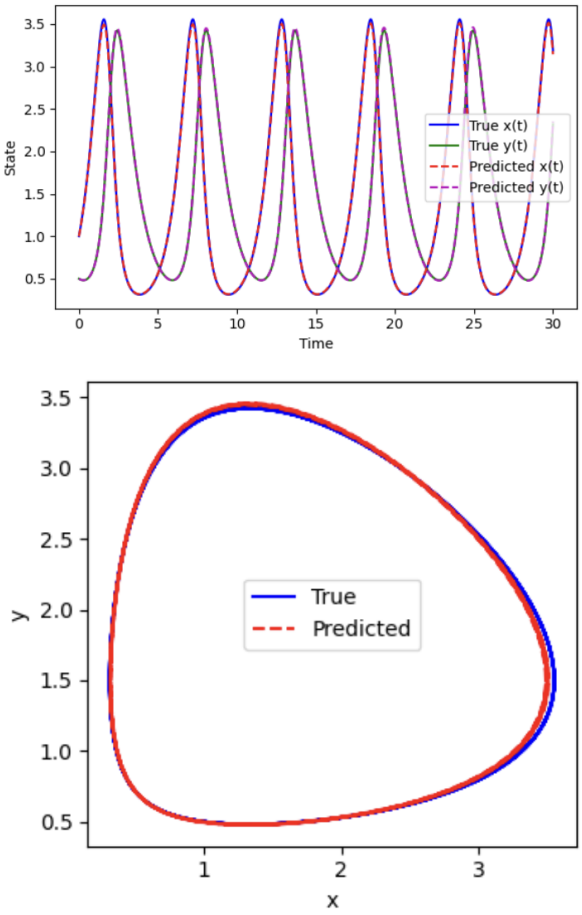

Fig. 9: Predicted vs Ground Truth Plots of Equation 3