

## 1 A Technical Appendices and Supplementary Material

2 Please refer to the Technical Appendices and Supplementary Material in the main paper.

## 3 B Additional Supplementary Material

### 4 B.1 AnalyzerVLM’s Hyperparameter Initialization for Optimization

5 As reported in [2] and [3], hyperparameter optimization for Gaussian process regression is a non-  
6 convex problem, making good initialization crucial for effective model discovery. To address this  
7 problem, [2] utilized random initialization with hyperparameter inheritance over rounds, and [3]  
8 has utilized random restarts with strong prior, sampling the hyperparameters from certain prior  
9 distribution.

10 In our case, we utilize AnalyzerVLM to propose model structures and suggest initialization point  
11 based on its analysis. In particular, we initialize the period and lengthscale of the periodic kernel and  
12 the lengthscale of the squared exponential kernel, using values suggested by AnalyzerVLM, then  
13 start optimizing in the first round. Then, such well-estimated hyperparameter values proposed by  
14 AnalyzerVLM can be carried over rounds, enabling the construction of progressively more complex  
15 and refined model structures initialized with strong hyperparameter estimates.

16 Fig. S1 shows two examples of optimized models with AnalyzerVLM proposal initialization and  
17 random initialization. As shown, with the AnalyzerVLM initialization, the model can be optimized to  
18 appropriate hyperparameters, while random initialization fails at finding the appropriate hyperparam-  
19 eters and leads to the whole kernel structure’s failure.

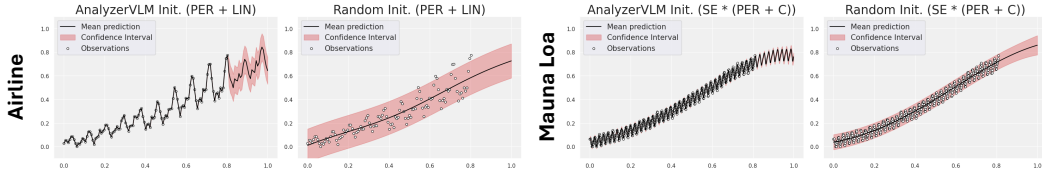


Figure S1: **Visualization of the optimized model with different initializations.** Optimized model initialized by AnalyzerVLM proposal reliably identifies appropriate parameter settings through data-driven analysis, whereas random initialization frequently leads to suboptimal configurations.

### 20 B.2 Function Composition and Implementation Details for Symbolic Regression

21 In our experiment for symbolic regression, we have defined our basis function and base grammars and  
22 experiments for function composition following [6]. We have conducted the iteration for 20 rounds,  
23 and the parameter optimization is done through Scipy’s optimize curve fit. For EvaluatorVLM, we  
24 have utilized  $\alpha$  to 0.05, which can balance the original symbolic regression score function and the  
25 visual evaluation. Our function composition is done through simple composition grammar(+, x) and  
26 basis functions. We prompted AnalyzerVLM to propose the function composition based on the below  
27 basis functions and grammar, so our model search space of symbolic regression  $\Sigma$  is:

$$\Sigma = \bigcup_{n=1}^{\infty} \{f \mid f \in \mathcal{B}, f = \oplus_{i=1}^n (b_i), b_i \in \mathcal{B}, \oplus_i \in \mathcal{O}\}, \quad (1)$$

$$\mathcal{B} ::= x \mid \sin(x) \mid \cos(x) \mid \tan(x) \mid \sinh(x) \mid \cosh(x) \quad (2)$$

$$\mathcal{O} ::= + \mid \times \mid \sqrt{\phantom{x}} \mid \exp \mid \log \mid \text{abs} \quad (3)$$

### 30 B.3 Symbolic Regression Results at Real-World Dataset

31 We report the result of symbolic regression to the real-world univariate datasets [4], including Airline  
32 Passenger, Solar Irradiance, Mauna Loa, Wheat, Call-Center, Radio, and Gas Production. For the  
33 experiments, we conducted 20 rounds for each method. As shown in Table S1, SR-based methods  
34 require a large number of trials to identify appropriate models. Unlike AnalyzerVLM which proposes

functions based on a detailed analysis of the data, symbolic regression-based methods like SGA [5] and LLM-SR [8] generate naive proposals without such insight, often leading to ineffective results. Although ICSR [6] enhances function proposal by utilizing visualization of data, enabling it to show better results SGA and LLM-SR, it still falls short of achieving the same level of performance as ours, due to the absence of precise, data-driven analysis. Also, symbolic regression-based methods underperform in real-world datasets since they do not explicitly model the observation noise, while Gaussian process regression does.

Interestingly, we observe that symbolic regression’s function composition can be a good starting point for the Gaussian process kernel discovery. To leverage this, we introduce a hybrid model discovery framework, denoted as Ours (SR + GP) in Table S1. We utilized simple SR framework [1] to generate the initial function composition and apply the top-3 function compositions’ corresponding kernel structure with its initial parameters. Then, we conducted our GP kernel discovery pipeline from the kernels for 2 rounds. With this, our hybrid model discovery results in superior performance across all evaluated datasets. Moreover, it highlights the potential synergy between symbolic model discovery and probabilistic modeling for interpretable and accurate forecasting. Fig. S2 shows the qualitative results. As shown, utilizing the hybrid model discovery framework can enhance the performance and find the appropriate model across the dataset.

Table S1: **Quantitative results in symbolic regression.** We compare our pipeline with competing methods on the train and test region, reporting RMSE as the evaluation metric. On average, our pipeline achieves superior performance across datasets. **Bold** stands for the best, and underline for the second best.

Method	Dataset														Avg.	
	Airline		Solar		Mauna		Wheat		Call		Radio		Gas			
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test		
SGA [5]	0.0700	0.1668	0.1451	0.2652	0.0332	0.0354	0.0599	<b>0.1338</b>	0.0689	0.8583	0.2859	0.8025	0.0574	0.3424	0.1029	0.3720
LLM-SR [8]	0.0692	0.3304	0.1039	1.7142	0.0317	0.2096	<b>0.0492</b>	0.1438	0.0438	21.028	0.1875	0.1798	0.0490	0.6521	0.0763	3.4659
ICSR [6]	0.0420	0.1029	0.1807	0.3845	0.0347	0.0343	0.0495	1.4430	0.0548	0.4725	0.1799	0.2427	0.0497	0.1672	0.0844	0.4067
Ours (SR + GP)	<b>0.0194</b>	<b>0.0354</b>	<b>0.0297</b>	<b>0.3345</b>	<b>0.0037</b>	<b>0.0166</b>	<b>0.0150</b>	0.1801	<b>0.0095</b>	<b>0.1088</b>	<b>0.0491</b>	<b>0.0515</b>	<b>0.0159</b>	<b>0.0581</b>	<b>0.0203</b>	<b>0.1121</b>

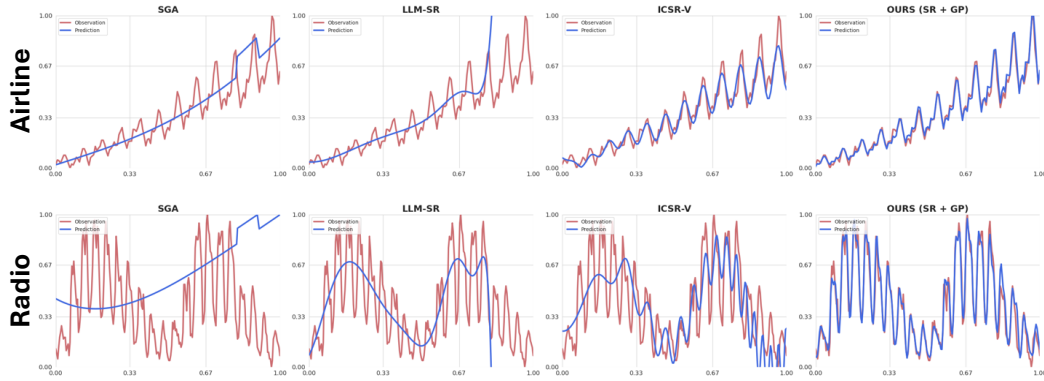


Figure S2: **Qualitative result in symbolic regression.** Although conventional symbolic regression methods often exhibit limited performance on real-world datasets, our hybrid model discovery framework demonstrates notable improvements.

## B.4 Prompt for AnalyzerVLM and EvaluatorVLM at Symbolic Regression

We report the prompts used for AnalyzerVLM and EvaluatorVLM for symbolic regression experiments at Table S2. Our implementation of prompts follows the scheme of [7], utilizing our multi-step reasoning for AnalyzerVLM and visual evaluation for EvaluatorVLM.

## B.5 Broader Impact

Our proposed pipeline for automatic model discovery has the potential to significantly lower the barrier to advanced data analysis by enabling non-experts to obtain high-quality models from domain-specific raw data. By leveraging Vision-Language Models (VLMs), our system can interpret and analyze the data in a human-like manner, making modeling more accessible, interpretable, and adaptable across domains such as science, engineering, and healthcare. However, attention must

Table S2: **AnalyzerVLM and EvaluatorVLM prompts for symbolic regression.** We report action choosing prompt used by AnalyzerVLM and fitness & generalizability evaluation prompt employed by EvaluatorVLM in symbolic regression.

---

**AnalyzerVLM: Action choosing prompt.**

---

Your task is to give me a list of five new potential functions that are different from all the ones reported below, and have a lower error value than all of the ones below. Before the function generation, please first analyze the given data points and reported functions first(e.g., visualization, or get the statistics). Guess and list up which this function would be. If you generate the Python code that includes your analysis, I will execute and give you the result. You can use sympy for checking the function prediction. For saving the visualization, please avoid using plt.show(), and use plt.savefig('./ztmpimgs/imagename') when imagename is any visualization you made. Before using the data points, please sort them first.  
Please give me only python code for now. Code:

```
'''python
Python code goes here
'''
```

Please try to build upon the function with the smallest error, then generating different ones too. Generate as diverse as diverse functions!

---

**EvaluatorVLM: Fitness evaluation prompt.**

---

You are an intelligent chatbot designed for evaluating two graph's similarity.  
You will evaluate the structure similarity of the two graph, data graph and predicted mean graph. Assign a score from 0 to 50.  
Evaluate the Structure Similarity Between Real Data and Mean Prediction.  
Please check the real data graph is similar to predicted mean graph. Please check below:  
- Mean graph is similar with sample graph (20-50 points).  
- Predicted mean graph is linear line while it shares trend with data graph (10-20 points)  
- Mean graph is linear and it does not share the trend at all(0-10 points).  
Please evaluate how similar the two graphs are. First is data's line plot, and second is predicted value's line plot. Please evaluate how well the predicted value fits to the data. Output should be the score for the function1. Please generate the response in the form of a Python dictionary string with keys of function name. score is in INTEGER, not STRING.  
function1:

---

**EvaluatorVLM: Generalizability evaluation prompt.**

---

You are an intelligent chatbot designed for evaluating the correctness of each functions.  
You will evaluate how well the predicted value (red line) fits based on the below criteria:  
- Evaluate the structure similarity of middle of the graph and the ends of the graph.  
- Check the blue line's structure similarity of the middle maintains at the left and right end of the graph.  
- If it was following the data well but suddenly changes to the constant line at the ends of the graph, assign low score for structure similarity score.  
But if structure similarity is maintained, assign 40-50 score.  
Please generate the response in the form of a Python dictionary string with keys of function name. 'score for structure similarity' are in INTEGER, not STRING.

---

62 be taken to ensure the responsible use of such automated systems, as overreliance on the model  
63 suggestions without human oversight may lead to misinterpretation or misuse.

## 64 **References**

- 65 [1] Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv*  
66 *preprint arXiv:2305.01582*, 2023.
- 67 [2] David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani.  
68 Structure discovery in nonparametric regression through compositional kernel search. In *International*  
69 *Conference on Machine Learning (ICML)*, 2013.
- 70 [3] Hyunjik Kim and Yee Whye Teh. Scaling up the automatic statistician: Scalable structure discovery using  
71 gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 575–584.  
72 PMLR, 2018.
- 73 [4] James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani.  
74 Automatic construction and natural-language description of nonparametric regression models. In *AAAI*  
75 *Conference on Artificial Intelligence (AAAI)*, 2014.
- 76 [5] Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B Tenenbaum, Daniela Rus, Chuang  
77 Gan, and Wojciech Matusik. Llm and simulation as bilevel optimizers: A new paradigm to advance physical  
78 scientific discovery. In *International Conference on Machine Learning (ICML)*, 2024.
- 79 [6] Matteo Merler, Katsiaryna Haitsiukevich, Nicola Dainese, and Pekka Marttinen. In-context symbolic  
80 regression: Leveraging large language models for function discovery. *arXiv preprint arXiv:2404.19094*,  
81 2024.
- 82 [7] Samiha Sharlin and Tyler R Josephson. In context learning and reasoning for symbolic regression with large  
83 language models. *arXiv preprint arXiv:2410.17448*, 2024.
- 84 [8] Parshin Shojaei, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K Reddy.  
85 Llm-sr: Scientific equation discovery via programming with large language models. *arXiv preprint*  
86 *arXiv:2404.18400*, 2024.