Scientific Equation Discovery using Modular Symbolic Regression via Vision-Language Guidance

Sepideh Fatemi[1], Abhilash Neog[1], Emma Marchisin[2], Amartya Dutta[1], Medha Sawhney[1], Paul C Hanson[2], Anuj Karpatne[1]

[1]Virginia Tech, [2]University of Wisconsin Madison

Problem Statement

- ❑ Symbolic Regression (SR) seeks explicit analytic equations that explain data.
- ❑ Traditional SR treats equations as monolithic and lacks structure.
- ❑ Can we model complex physical systems which require equations that are **interpretable**, **scientifically grounded**, and **computationally efficient?**

Motivation

❑ Why Vision-Language Models?

Vision–Language Models translate the process diagram into physics-aware seed equation.

❑ Why Large-Language Models?

LLM-powered Symbolic Regression (LLM-SR[1]) pairs the reasoning of large language models with symbolic regression to rapidly uncover physics-consistent equations.

❑ Why Modular?

A modular design breaks the final equation into reusable sub-functions, so each physical process can be inspected, a swapped with other processes.

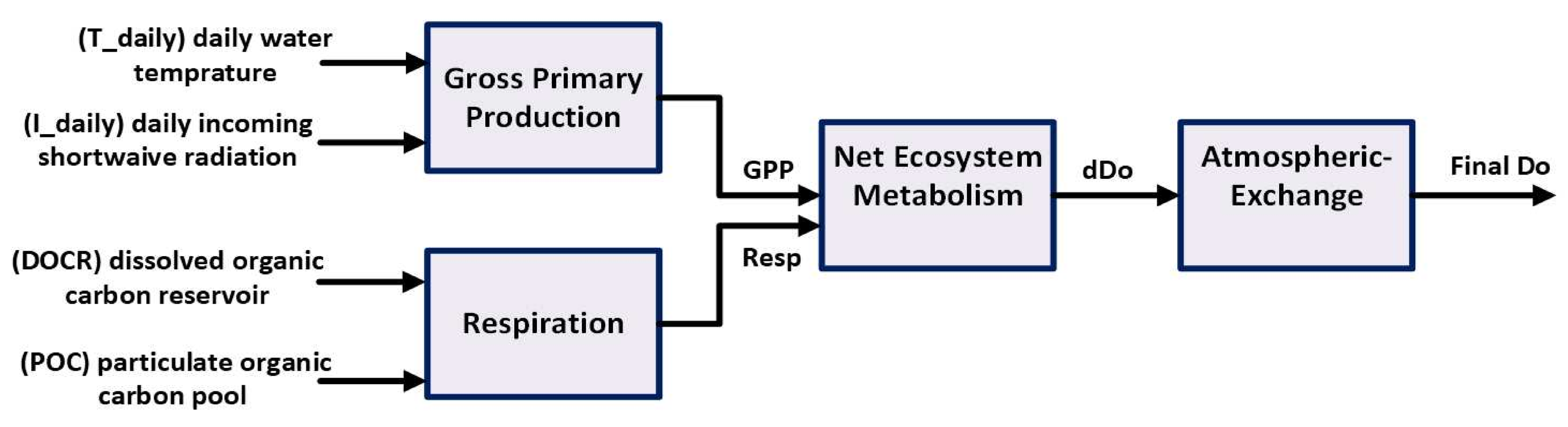
Why it matters — Accurately modeling complex physical systems demands equations that are interpretable, scientifically sound, and computationally tractable. Traditional methods rarely deliver all three.

Datasets

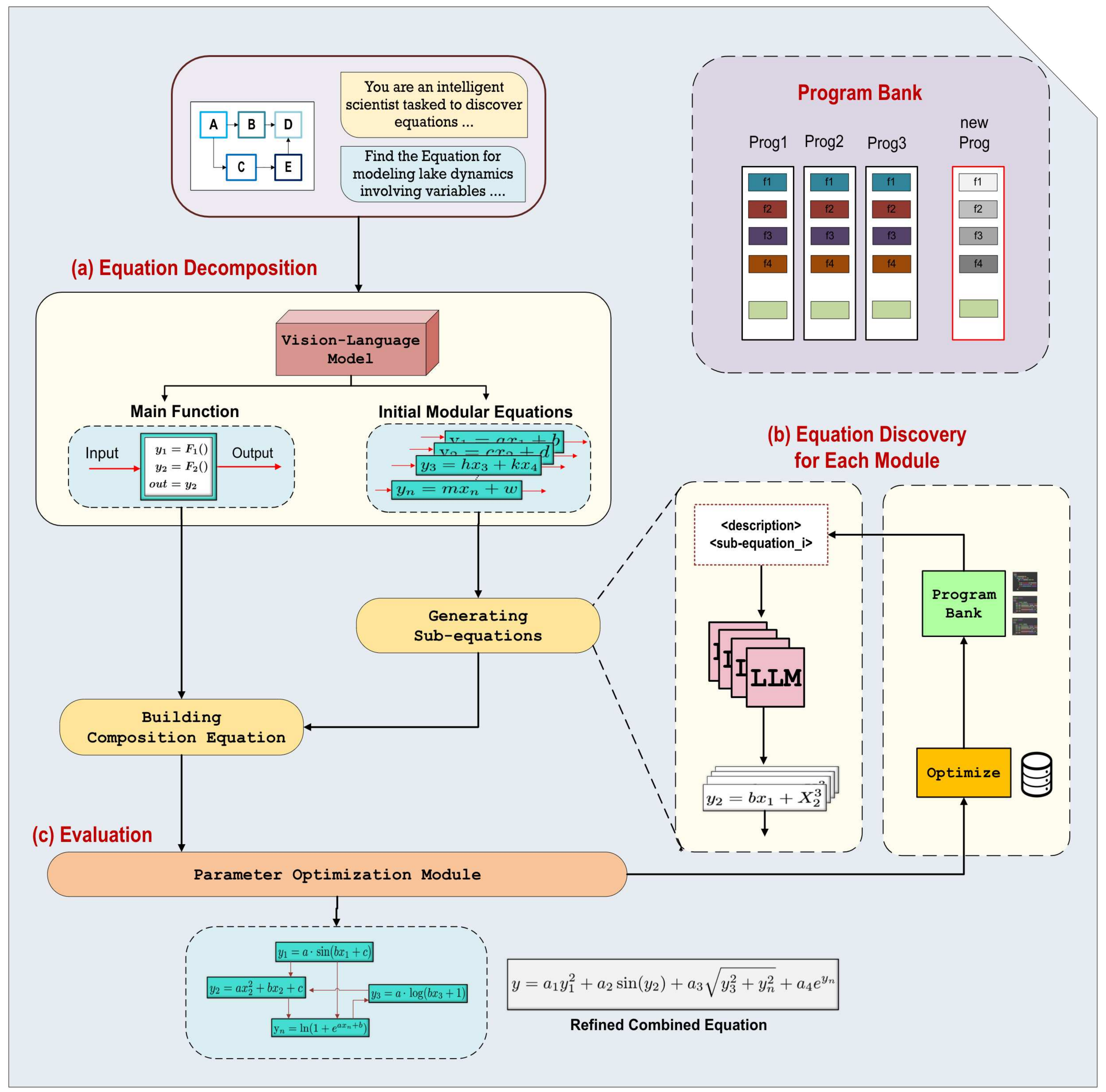
Synthetic:
Math logic

Real-world:
Lake DO simulation (environmental dynamics)

Expert-provided box and arrow diagram:



Proposed Approach



Key Contribution:

- ❑ **Modular Structure:** Decompose equations into interpretable components.
- ❑ **Integration of Domain Knowledge:** Enhance interpretability and accuracy with scientific insights.
- ❑ **Scalable Discovery:** Handle larger solution spaces via modular decomposition.
- ❑ **Vision-Language Guidance:** Multimodal models aid equation discovery and validation.

Results

Interpretability: Each equation module corresponds to a real-world scientific process, making the final model easy to interpret and validate.

Step-by-Step Reasoning: The system builds equations incrementally, reasoning one sub-process at a time, just like how scientists break down complex systems.

VLM generated output

Gross Primary Production

```
def gross_primary_production(T_daily, I_daily, params):
    return params[0] * T_daily + params[1] * I_daily +
    params[2]
```

Respiration

```
def respiration(DOCR, POC, params):
    return params[0] * DOCR + params[1] * POC + params[2]
```

Net Ecosystem Metabolism

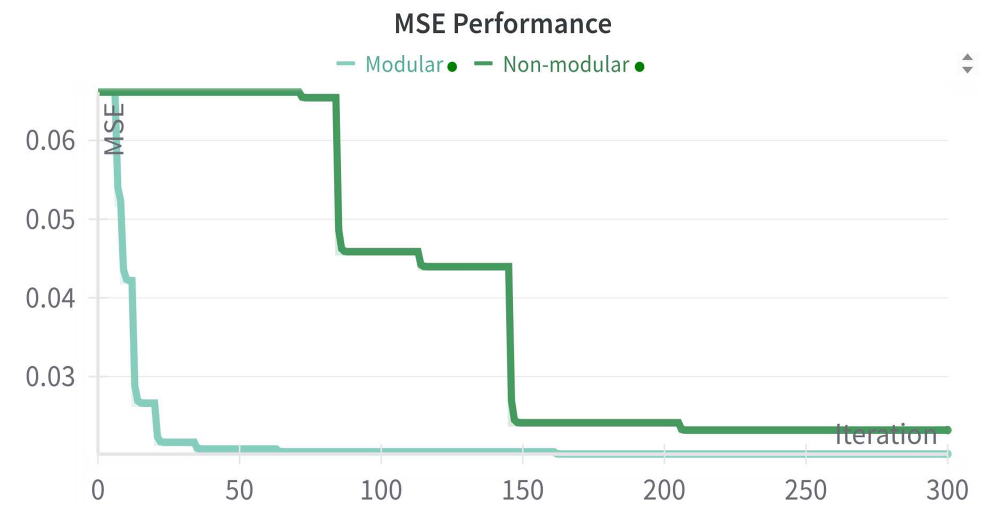
```
def net_ecosystem_metabolism(GPP, Resp, params):
    return params[0] * GPP - params[1] * Resp + params[2]
```

Atmospheric Exchange

```
def atmospheric_exchange(dDO, params):
    return params[0] * dDO + params[1]
```

Main Equation

```
def main_equation(I_daily, T_daily, POC, DOCR, params):
    GPP = gross_primary_production(T_daily, I_daily,
    params[0])
    Resp = respiration(DOCR, POC, params[1])
    dDO = net_ecosystem_metabolism(GPP, Resp, params[2])
    DO_exchange = atmospheric_exchange(dDO, params[3])
    return DO_exchange
```



Comparing Modular vs Non-modular on different datasets.

Model	Lake Modeling	Math Logic
VLM(gpt-4o)	0.0661	0.1885
Modular-SR w/o modules	0.0241	0.01560
Modular-SR	**0.0201**	**2.8e10-12**

Future Work & Discussion

- ✓ Extend to other equations and domains
- ✓ Evaluate and refine individual modules
- ✓ Improve explainability and scientific alignment
- ✓ Benchmark across diverse LLMs and VLMs
- ✓ Optimize for time and computational efficiency

Acknowledgement: This work was supported in part by NSF awards 2239328 and 2213549, and 2213550. We are grateful to the Advanced Research Computing (ARC) Center at Virginia Tech for providing access to GPU compute resources for this project.

Refrences: [1] Shojaee, Parshin, 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).