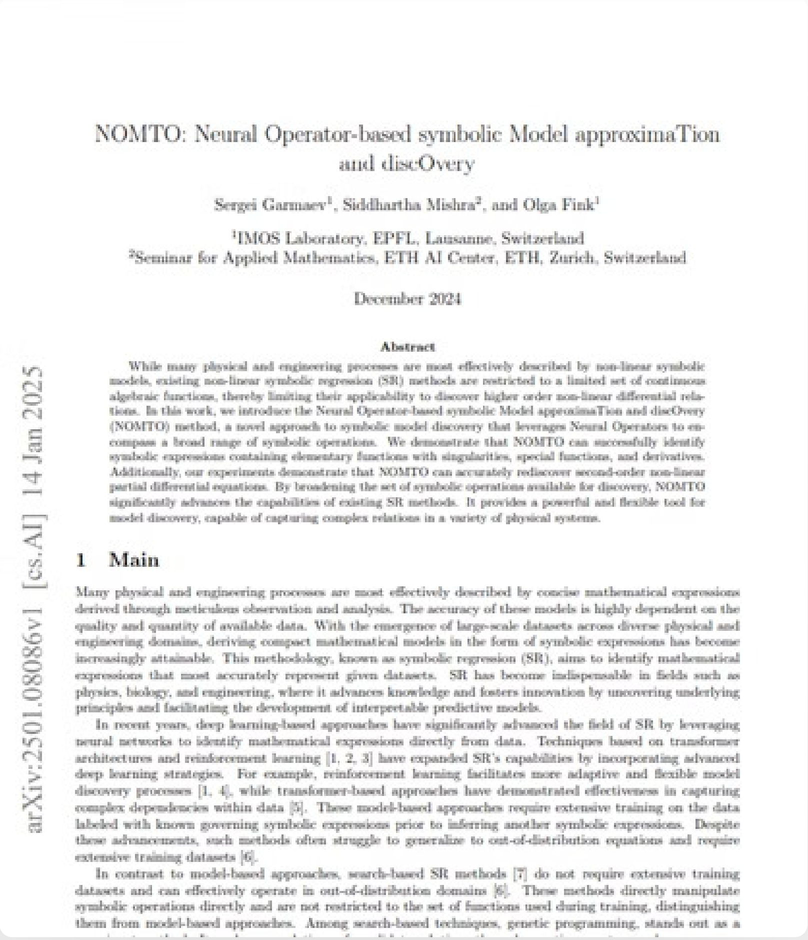
# From Symbolic Regression to Neural Operators: The Path to NOMTO

## Discovering Symbolic Models in Plasticity Problems

### Presented by: Yasser El Kouhen

My github summary (7 pages) : [https://github.com/Yasser-EL-KOUHEN/Projects/blob/main/Machine%20Learning%20Projects%20-%20Yasser%20El%20Kouhen/Theory%20-%20Symbolic%20Regression%20and%20Neural%20Operators%20-%20NOMTO.pdf](https://github.com/Yasser-EL-KOUHEN/Projects/blob/main/Machine%20Learning%20Projects%20-%20Yasser%20El%20Kouhen/Theory%20-%20Symbolic%20Regression%20and%20Neural%20Operators%20-%20NOMTO.pdf)

y par yasser el kouhen



NOMTO: Neural Operator-based symbolic Model approximaTion and discOvery

Sergei Garmaev[1], Siddhartha Mishra[2], and Olga Fink[1]

[1]IMOS Laboratory, EPFL, Lausanne, Switzerland
[2]Seminar for Applied Mathematics, ETH AI Center, ETH, Zurich, Switzerland

December 2024

arXiv:2501.08086v1 [cs.AI] 14 Jan 2025

**Abstract**

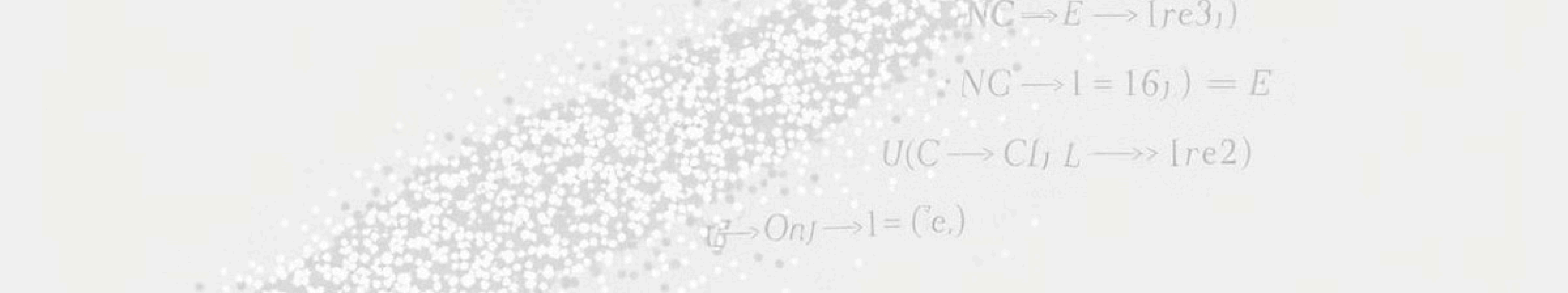
While many physical and engineering processes are most effectively described by non-linear symbolic models, existing non-linear symbolic regression (SR) methods are restricted to a limited set of continuous algebraic functions, thereby limiting their applicability to discover higher order non-linear differential relations. In this work, we introduce the Neural Operator-based symbolic Model approximaTion and discOvery (NOMTO) method, a novel approach to symbolic model discovery that leverages Neural Operators to encompass a broad range of symbolic operations. We demonstrate that NOMTO can successfully identify symbolic expressions containing elementary functions with singularities, special functions, and derivatives. Additionally, our experiments demonstrate that NOMTO can accurately rediscover second-order non-linear partial differential equations. By broadening the set of symbolic operations available for discovery, NOMTO significantly advances the capabilities of existing SR methods. It provides a powerful and flexible tool for model discovery, capable of capturing complex relations in a variety of physical systems.

**1 Main**

Many physical and engineering processes are most effectively described by concise mathematical expressions derived through meticulous observation and analysis. The accuracy of these models is highly dependent on the quality and quantity of available data. With the emergence of large-scale datasets across diverse physical and engineering domains, deriving compact mathematical models in the form of symbolic expressions has become increasingly attainable. This methodology, known as symbolic regression (SR), aims to identify mathematical expressions that most accurately represent given datasets. SR has become indispensable in fields such as physics, biology, and engineering, where it advances knowledge and fosters innovation by uncovering underlying principles and facilitating the development of interpretable predictive models.

In recent years, deep learning-based approaches have significantly advanced the field of SR by leveraging neural networks to identify mathematical expressions directly from data. Techniques based on transformer architectures and reinforcement learning [1, 2, 3] have expanded SR's capabilities by incorporating advanced deep learning strategies. For example, reinforcement learning facilitates more adaptive and flexible model discovery processes [1, 4], while transformer-based approaches have demonstrated effectiveness in capturing complex dependencies within data [5]. These model-based approaches require extensive training on the data labeled with known governing symbolic expressions prior to inferring another symbolic expressions. Despite these advancements, such methods often struggle to generalize to out-of-distribution equations and require extensive training datasets [6].

In contrast to model-based approaches, search-based SR methods [7] do not require extensive training datasets and can effectively operate in out-of-distribution domains [6]. These methods directly manipulate symbolic operations directly and are not restricted to the set of functions used during training, distinguishing them from model-based approaches. Among search-based techniques, genetic programming, stands out as a



# Why Learn Equations Instead of Just Predicting?

1. **Black-box Approaches**

   Traditional machine learning models are **black-box** approaches.

2. **Outcome Prediction**

   They predict **outcomes** without explaining **underlying laws**.

3. **Example: Pendulum Motion**

   A neural network can predict **pendulum motion**, but it won't reveal **Newton's Equations**.

## Key Idea:

1. Instead of just **fitting a function**, we want to **discover the governing equation itself**.
2. This is the goal of **Symbolic Regression**: Learning **explicit mathematical expressions** from data.

🚀 This brings us to EQL (Equation Learner).

# EQL: A Neural Network for Symbolic Regression

Unlike standard neural networks, EQL learns **interpretable mathematical expressions**.

EQL's hidden layers contain **specific mathematical functions**:

- **Identity** ( $f(x) = x$ )
- **Sinusoidal functions** ( $f(x) = \sin(x), \cos(x)$ )
- **Multiplication** ( $f(x, y) = x \cdot y$ )

## Advantage:

**Generates explicit equations instead of black-box predictions.**

## Limitation:

Cannot handle **division**, crucial for **many physical laws** (e.g., fluid flow, physics equations).

🚀 EQL needed improvement... which led to EQL÷.

# Solving EQL's Limitation: Adding Division

### EQL÷ Extension

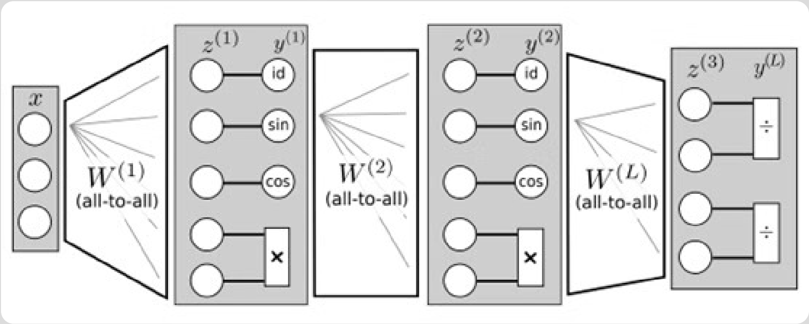
EQL÷ **extends EQL** by adding **division operations** to the network:

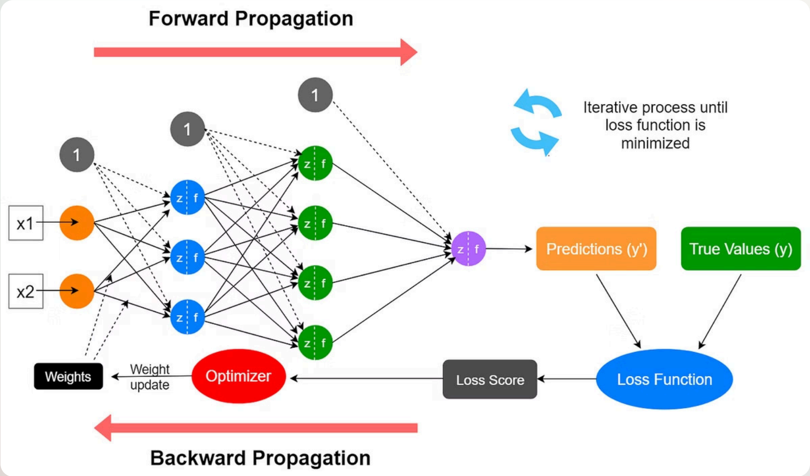
```
h_θ(a, b) = {
   a/b, if b > θ
   0,   otherwise
}
```

### Why is this important?

- Many physical equations involve **division** (e.g., inverse relationships).
- Improves **model selection** by **eliminating unnecessary terms**.
- **More robust** training with regularized division.

🚀 EQL÷ is a step forward... but still limited!






# Why Do We Need More Powerful Models?

## Complex Functions

Struggles with **complex functions** (e.g., differential operators).

## PDEs

Does not handle **partial differential equations (PDEs)** well.

## Limited Generalization

Still **dependent on specific symbolic functions**, limiting generalization.

**We need an approach that can learn more than just algebraic functions.**

🚀 A natural direction: Neural Operators!

# A New Paradigm: Learning Function Mappings

Traditional neural networks **map finite-dimensional inputs to outputs**.

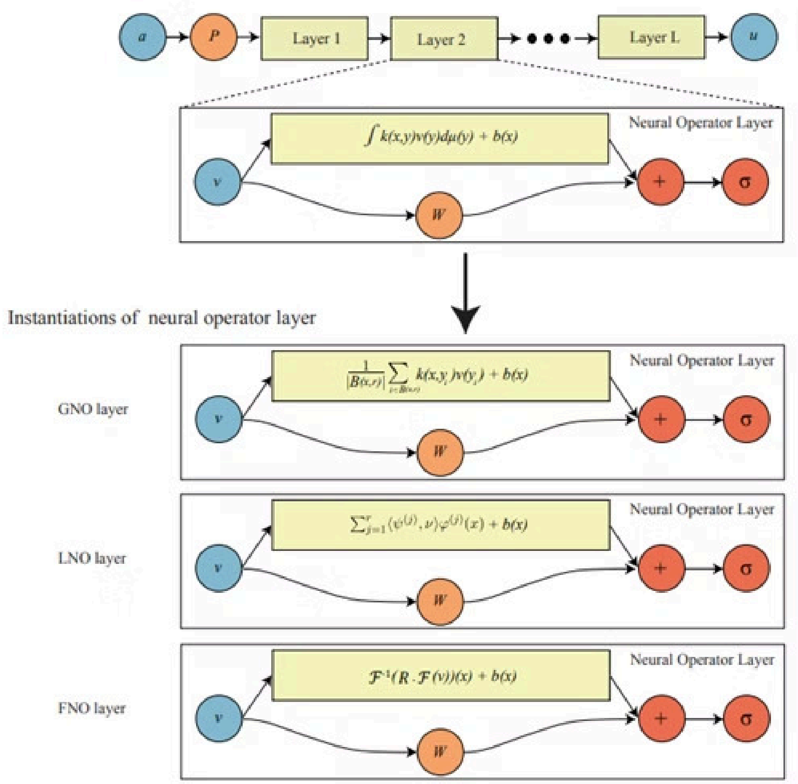
**Neural Operators** generalize this idea by learning **mappings between function spaces** (i.e., PDE solvers).

### Key properties of Neural Operators:

- **Discretization-invariant** (work on any resolution).
- **Approximate continuous operators** (universal function approximators).
- **Much faster than traditional PDE solvers** (e.g., Navier-Stokes equations).

🚀 **Neural Operators provide a new tool for learning mathematical structures.**

❓ Can we combine them with Symbolic Regression?




# NOMTO: Neural Operator-Based Symbolic Regression

1

## NOMTO Definition

**NOMTO = Neural Operators + Symbolic Regression.**

2

## Approach

Instead of learning functions **heuristically**, it builds **interpretable symbolic expressions** using **Neural Operators**.

3

## Capabilities

- Can handle **differential equations, singularities, and special functions.**
- Learns equations **directly from data** (instead of predefined function libraries).
- **Bridges the gap** between deep learning and symbolic AI.

**A breakthrough: We can now use AI to extract governing equations from data!**

🚀 Let's see how NOMTO works.

# How Does NOMTO Discover Equations?

### Train Neural Operators

Approximate **basic mathematical operations**.

### Build Computational Graph

**Build a computational graph** linking these operations.

### Optimize Graph

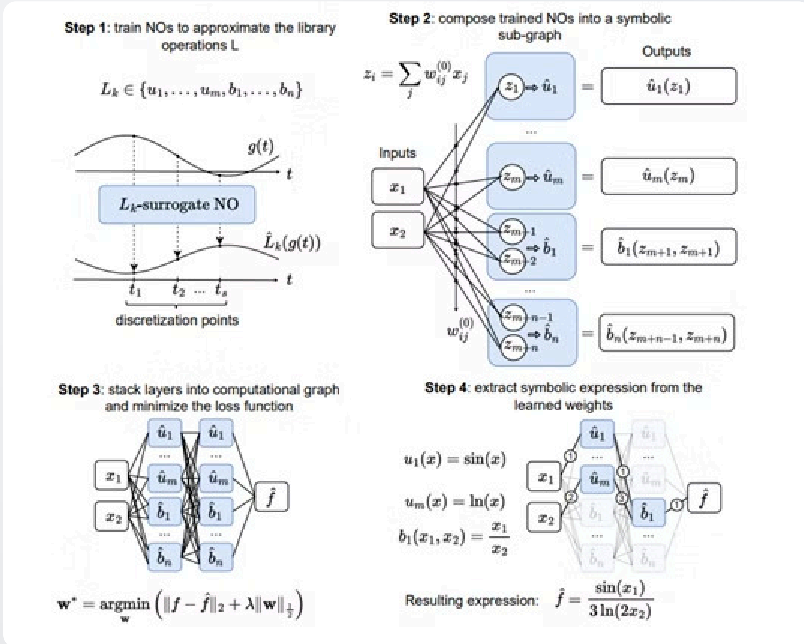
**Optimize the graph** to find the **simplest possible equation**.

### Extract Symbolic Model

**Extract the final symbolic model** that best explains the data.

Unlike traditional symbolic regression, **NOMTO can handle PDEs and special functions.**

🚀 This unlocks real-world applications!




# Where Can We Use NOMTO?

### Physics & Engineering:

- Automatically **discover governing equations** (e.g., **heat transfer, turbulence models**).
- Outperforms standard **symbolic regression methods.**

### Robotics & Control:

- Learns **equations for dynamic systems** (e.g., cart-pendulum, robotic arms).
- Enables **real-time control** with minimal data.

### Mathematical Discovery:

- First AI model to recognize **special functions (Gamma, Airy functions, etc.)**.
- Extracts equations that are **interpretable & generalizable**.

🚀 NOMTO is a powerful tool for AI-driven scientific discovery!

# Using NOMTO for Plasticity Problems

## Our research question:

- How can we discover **symbolic models** for **plasticity problems** in materials science?
- Can we use **NOMTO** to extract equations describing **material deformation**?

## Why NOMTO fits this problem:

- Plasticity problems involve **PDEs**, making standard symbolic regression **insufficient**.
- Neural Operators can **efficiently learn complex function mappings**.
- Symbolic regression ensures **interpretability**, crucial for material science.

NOMTO provides a data-driven way to discover governing laws in plasticity models.