Scalable Multi-GPU Training of Neural Operators: Advancing Generalization in High-Dimensional Physical Systems

Luis Felipe dos Santos, PhD Candidate

Dibakar Roy Sarkar, PhD Student

Deane Roehl, PhD

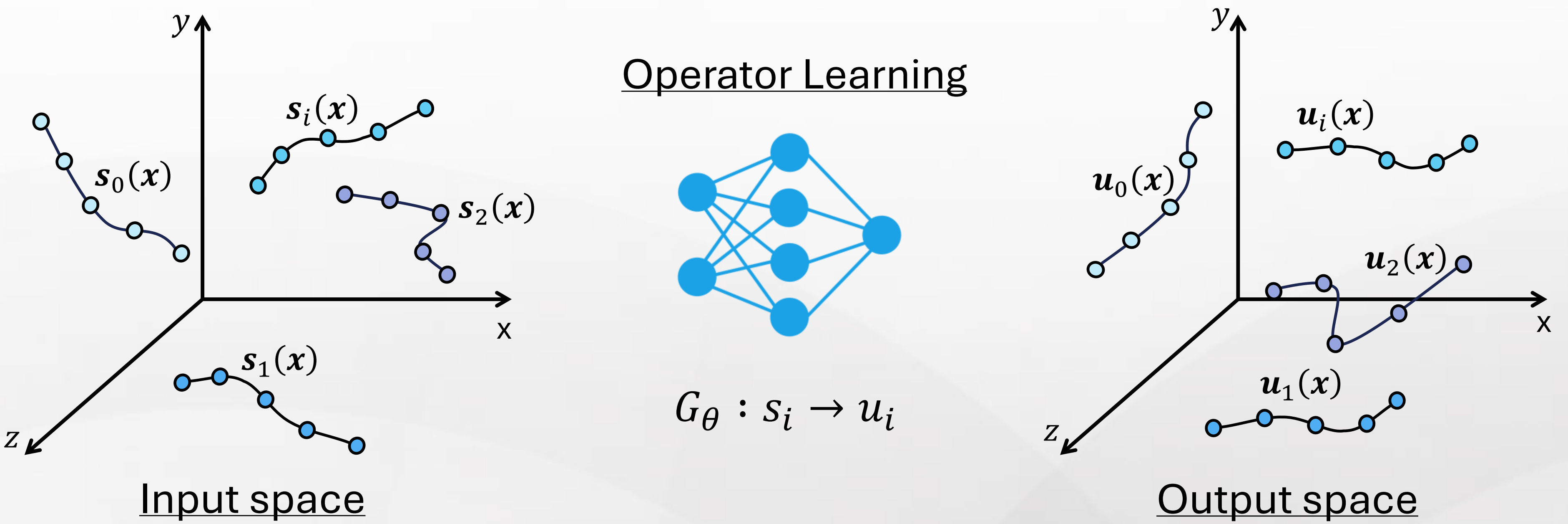
Somdatta Goswami, PhD

AGENDA

- INTRODUCTION
- MOTIVATION
- DATA PARALLEL OPERATOR LEARNING
- RESULTS
- FUTURE PERSPECTIVES

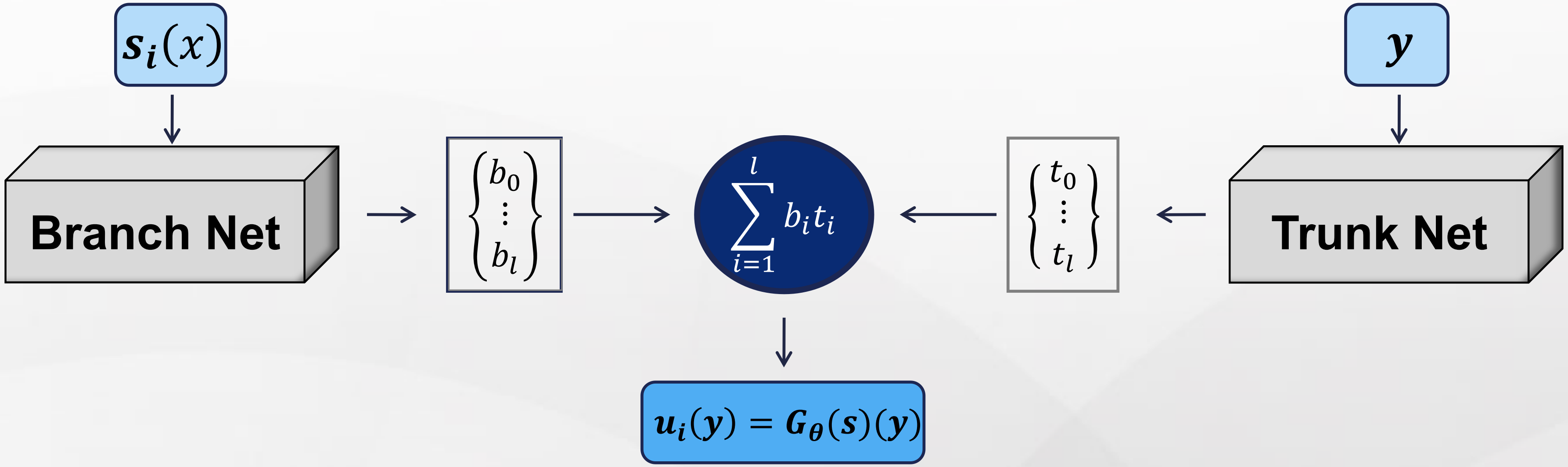
INTRODUCTION

- Infinite - dimensional mapping between spaces



INTRODUCTION

- Leverages the universal approximation capabilities of neural networks to enable efficient Deep Operator Network learning with a flexible architecture.



MOTIVATION

ADVANTAGES

- Real-time inference
- Reliable surrogate model

CHALLENGES

- Down-sampling input-output spaces miss fine details, reducing accuracy.
- Latent space learning is ineffective for non-dissipative systems.
- Small mini-batches limit the network's generalization ability.

MOTIVATION

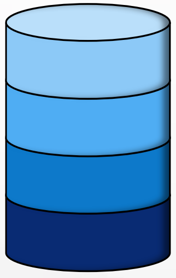
- SOLUTION
  - Developing a scalable framework for data-parallel operator learning to handle high-dimensional PDEs using JAX, optimized for multi-node and multi-GPU HPC

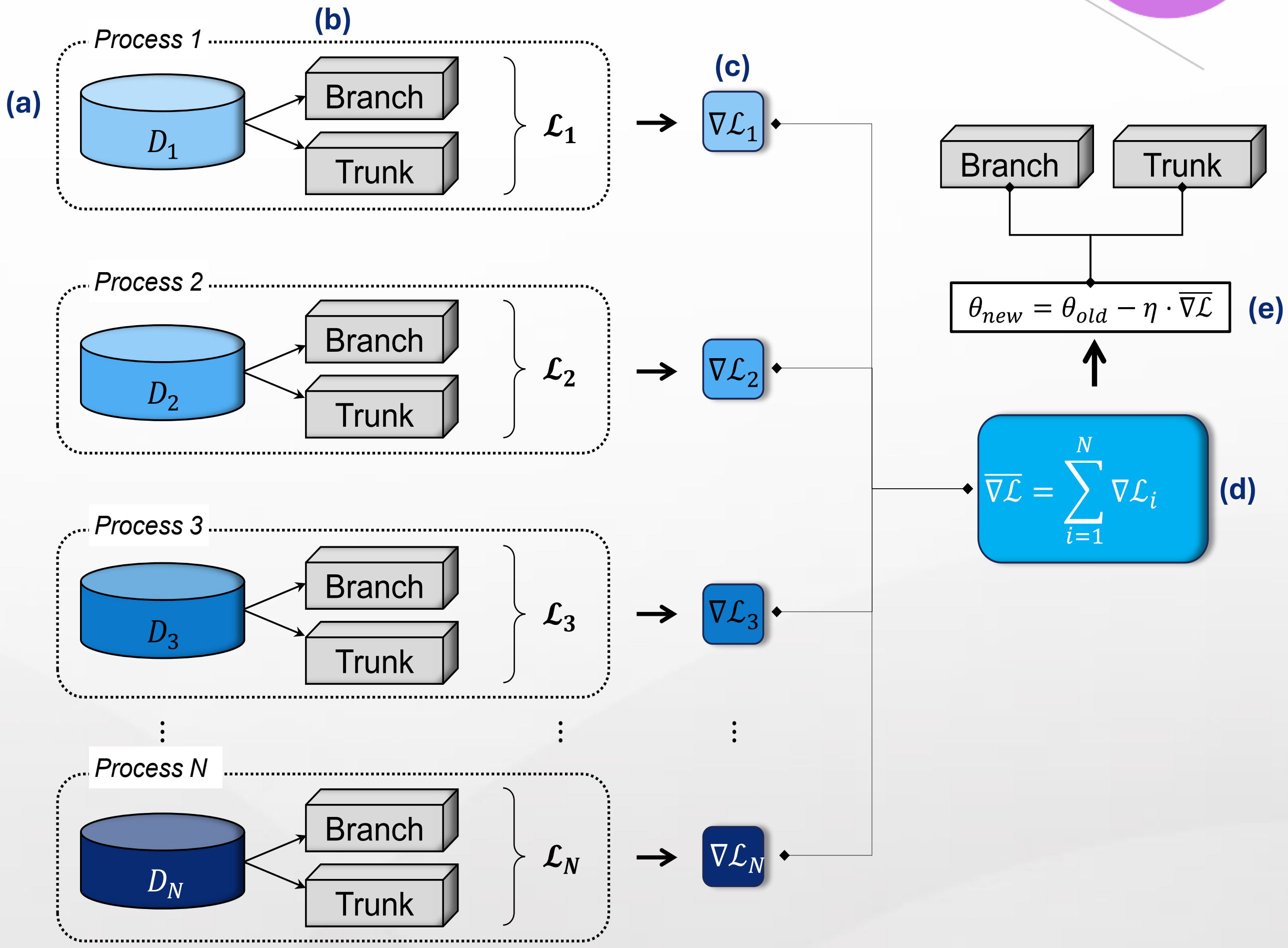
DATA PARALLEL OPERATOR LEARNING

MACH 2025

GENERAL FRAMEWORK:

- Model:
  - Create and replicate trunk and branch states
- Train step:
  - i. Data treatment:
    - Sharding inputs and outputs
  - ii. Get local predictions – losses – gradients
  - iii. Averaging gradients and sum losses
    - All-Reduce mean and Reduce sum
  - iv. Local parameters updating
  - v. Go to step i. and repeat





DATA PARALLEL OPERATOR LEARNING

SCALING METRICS:

- Speed up (S_{up})

$$S_{up} = \frac{T_1}{T_N}$$

- Efficiency (E_f)

$$E_f = N \cdot S_{up} = \frac{N \cdot T_1}{T_N}$$

T_1: Time for processing considering one device

T_N: Time for processing considering N devices

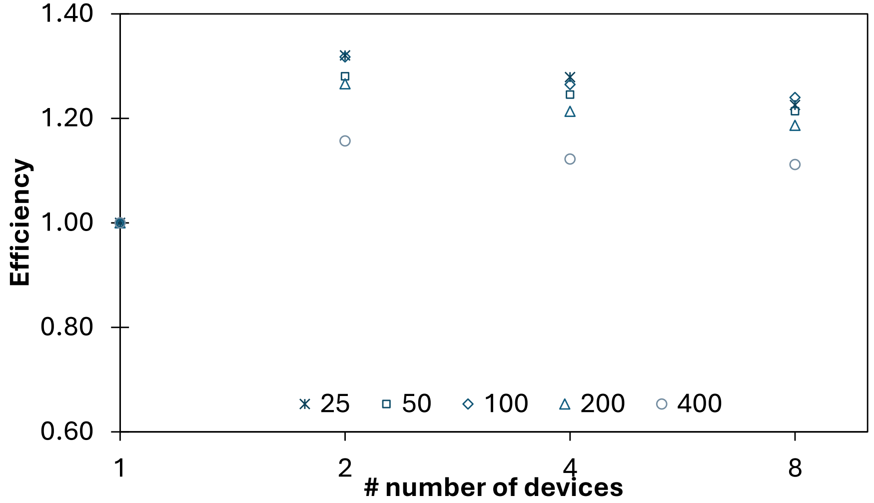
RESULTS

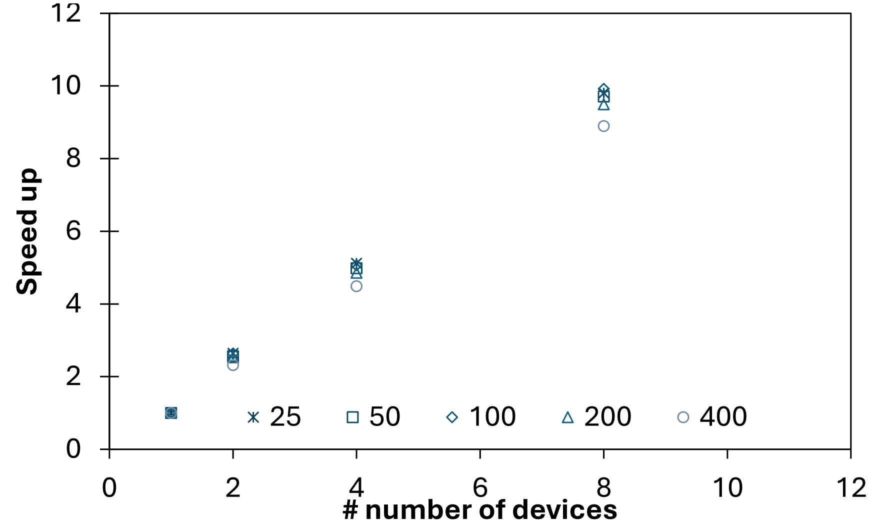
- Results for **Branch Parallel** training (**Physics-informed** Burgers):

$$\frac{\partial \boldsymbol{u}}{\partial t} + u\frac{\partial \boldsymbol{u}}{\partial x} = \nu\frac{\partial^2 \boldsymbol{u}}{\partial^2 t}$$

- **Input space** $\rightarrow u(t=0, \boldsymbol{x}) \;\therefore\; (2000, 101)$
- **Output space** $\rightarrow u(t, \boldsymbol{x}) \;\therefore\; (2000, 101, 101)$

$$G_\theta(u(t=0,\boldsymbol{x}))(t,\boldsymbol{x}) \rightarrow u(t,\boldsymbol{x})$$





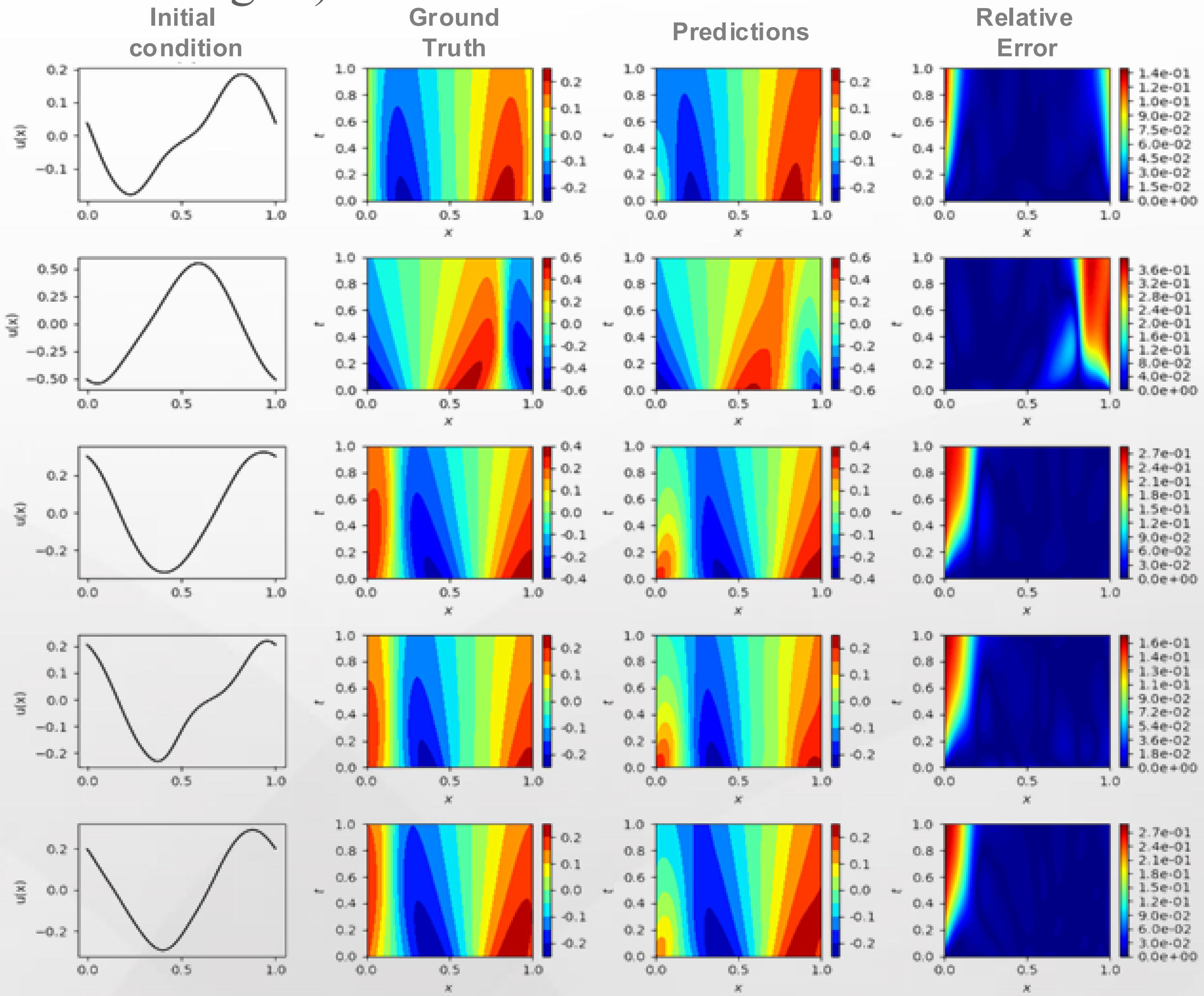
RESULTS

- Results for **Branch Parallel** training (**Physics-informed** Burgers):

$$\frac{\partial \boldsymbol{u}}{\partial t} + u\frac{\partial \boldsymbol{u}}{\partial x} = \nu\frac{\partial^2 \boldsymbol{u}}{\partial^2 t}$$

- **Input space** $\rightarrow u(t=0,\boldsymbol{x}) \;\therefore (2000, 101)$
- **Output space** $\rightarrow u(t,\boldsymbol{x}) \;\therefore (2000, 101, 101)$

$$G_\theta(u(t=0,\boldsymbol{x}))(t,\boldsymbol{x}) \rightarrow u(t,\boldsymbol{x})$$



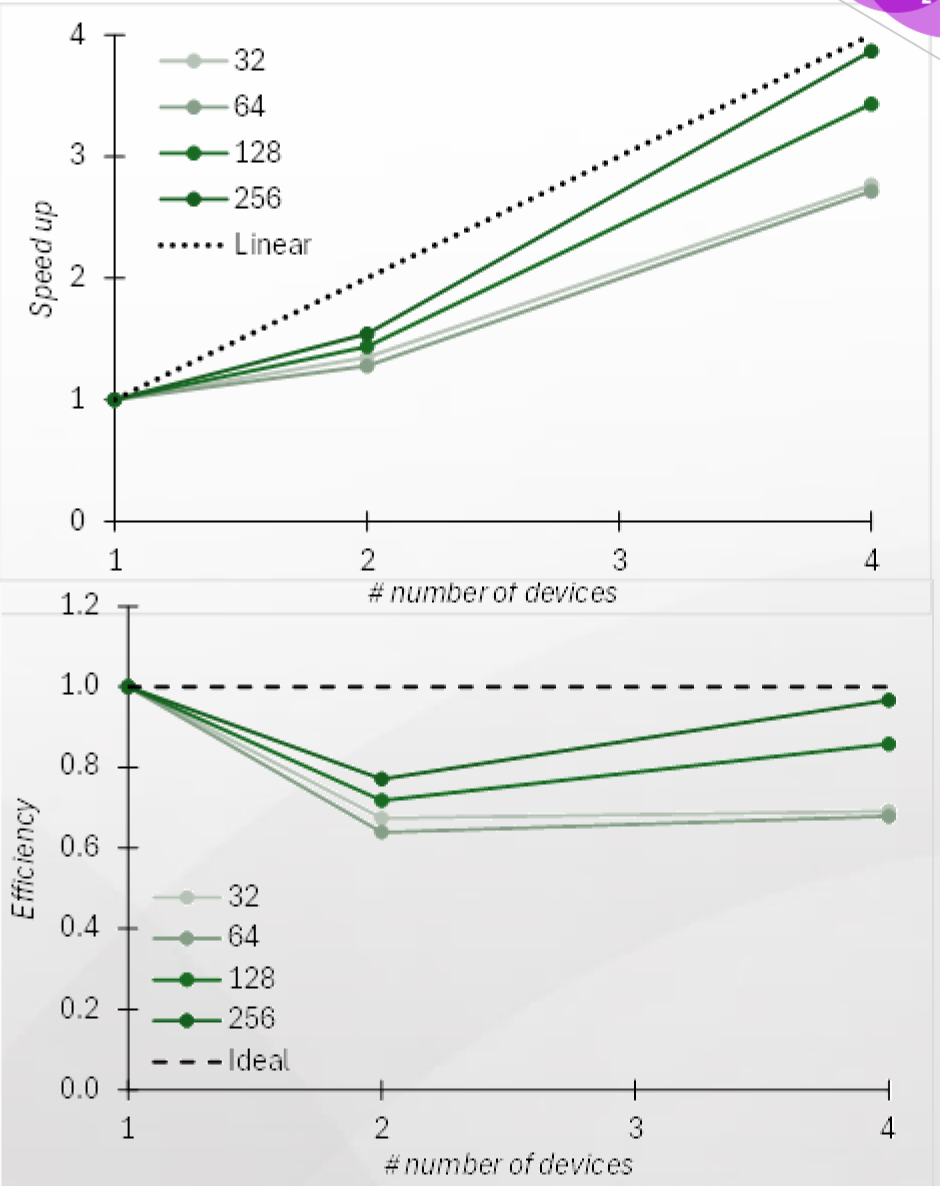
RESULTS

- Results for **Trunk Parallel** training (Data-driven Darcy):

$$\frac{\partial \boldsymbol{P}}{\partial t} - K(x,y) * \boldsymbol{\nabla P} = \boldsymbol{q}$$

- **Input space** → $K(x,y) \therefore (1000, 100, 100)$
- **Output space** → $\boldsymbol{P}(t,x,y) \therefore (1000, 72, 100, 100)$

$$G_\theta(K(x,y))(t,x,y) \rightarrow \boldsymbol{P}(t,x,y)$$



RESULTS

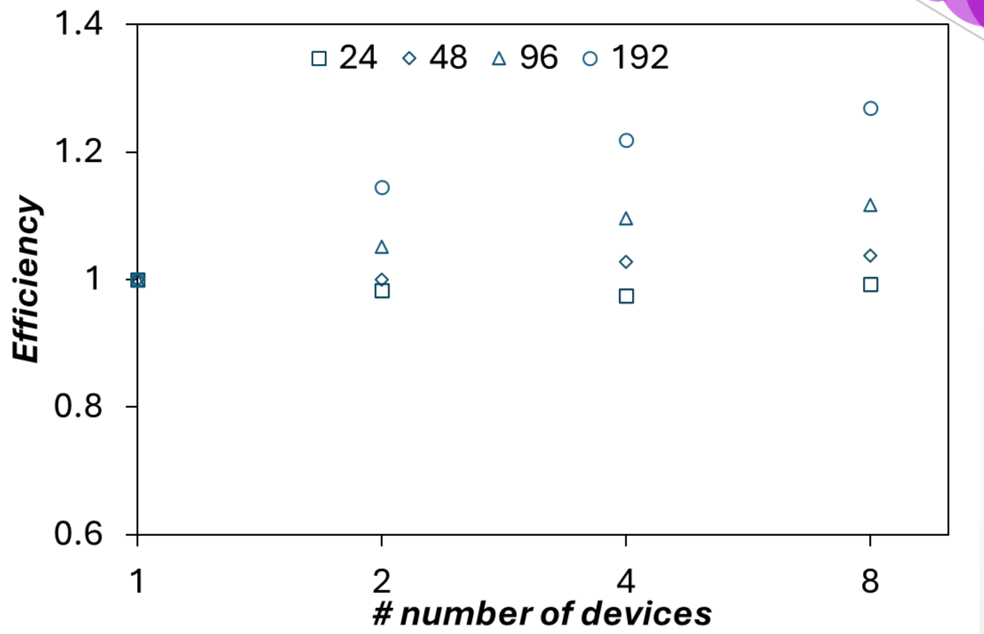
- Results for **Branch and Trunk Parallel** training (Data-driven shallow water):

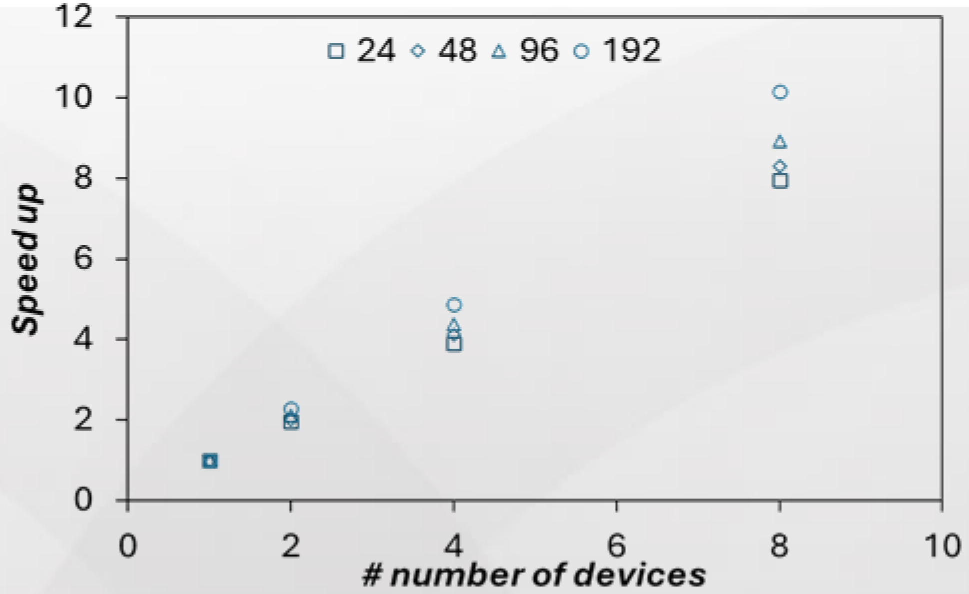
$$\frac{D\boldsymbol{V}}{Dt} = -f\boldsymbol{k} \times \boldsymbol{V} - \boldsymbol{g}\nabla\boldsymbol{h} + \nu\nabla^2\boldsymbol{V}$$

$$\frac{Dh}{Dt} = -\boldsymbol{h}\nabla \cdot \boldsymbol{V} + \nu\nabla^2 h$$

- Input space → $\boldsymbol{h}(t = 0, \boldsymbol{x}) \;\therefore (256, 256)$
- Output space → $\boldsymbol{V}(t, \boldsymbol{x}) \;\therefore (72, 256, 256)$

$$G_\theta(h(t = 0, \boldsymbol{x}))(t, \boldsymbol{x}) \rightarrow \boldsymbol{V}(t, \boldsymbol{x})$$





RESULTS

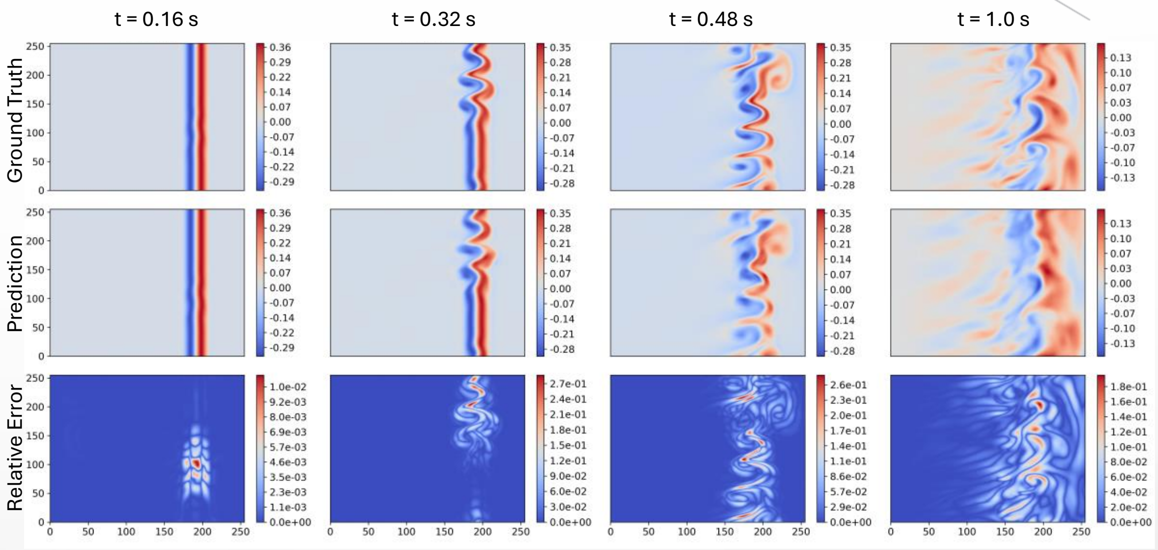
- Results for **Branch and Trunk Parallel** training (Data-driven shallow water):

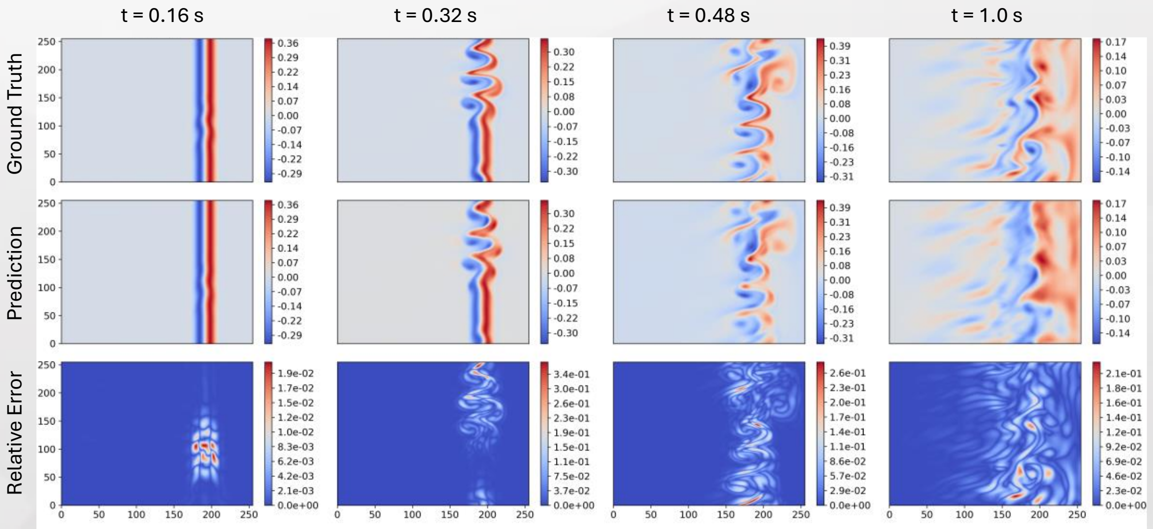
$$\frac{D\boldsymbol{V}}{Dt} = -f\boldsymbol{k} \times \boldsymbol{V} - g\nabla \boldsymbol{h} + \nu\nabla^2 \boldsymbol{V}$$

$$\frac{Dh}{Dt} = -\boldsymbol{h}\nabla \cdot \boldsymbol{V} + \nu\nabla^2 h$$

  - Input space $\rightarrow \boldsymbol{h}(t=0,\boldsymbol{x}) \;\therefore (256, 256)$
  - Output space $\rightarrow \boldsymbol{V}(t,\boldsymbol{x}) \;\therefore (72, 256, 256)$

$$G_\theta(h(t=0,\boldsymbol{x}))(t,\boldsymbol{x}) \rightarrow \boldsymbol{V}(t,\boldsymbol{x})$$





FUTURE PERSPECTIVES

- The **efficiency** of the proposed DP training is evident in its superior performance, which **exceeds the minimum admissible (0.8)** for all cases and is particularly pronounced (superior to 1) in the majority of the evaluated scenarios.
- The **speed-up** consistently **surpasses** the **ideal (linear) scaling** and exhibits a marked enhancement as the number of devices increases, thereby demonstrating the effective management of high-dimensional problems.
- **Designed** to be **generic**, allowing **easy extension** to various types of **neural networks** and **operator learning** being **adaptable to** different numbers of **nodes and GPUs**, ensuring **scalability** and **flexibility**. Available at https://github.com/Centrum-IntelliPhysics
- **Future directions** include the integration of **domain decomposition** with the proposed data parallel operator learning.

ACKNOWLEDGMENTS

Centrum IntelliPhysics @JHU

Thank You!

FUNDING

NSF NAIRR Pilot CAPES CNPq