Differentiable Physics Simulations for Deep Learning

(*... and Beyond*)

Nils Thuerey

Physical Phenomena

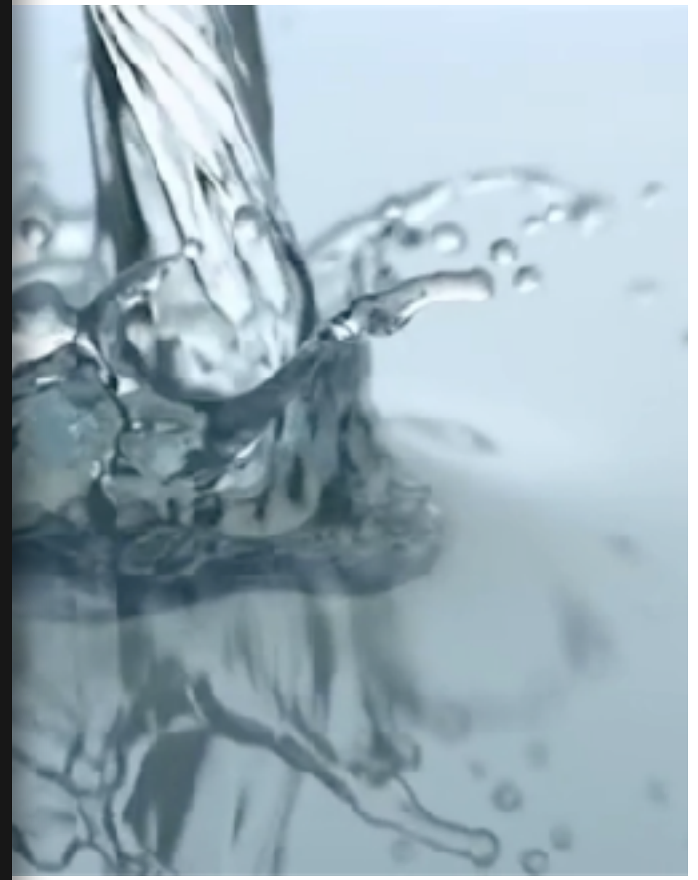
Everywhere around us…

- Fluid Mechanics

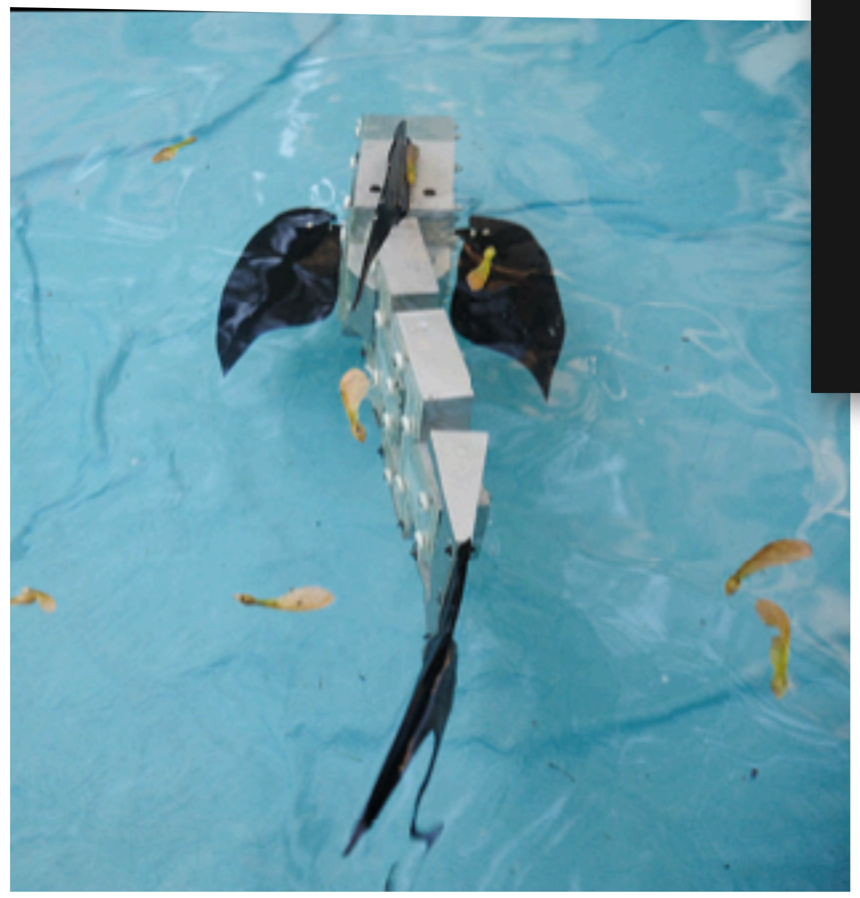
Physical Phenomena

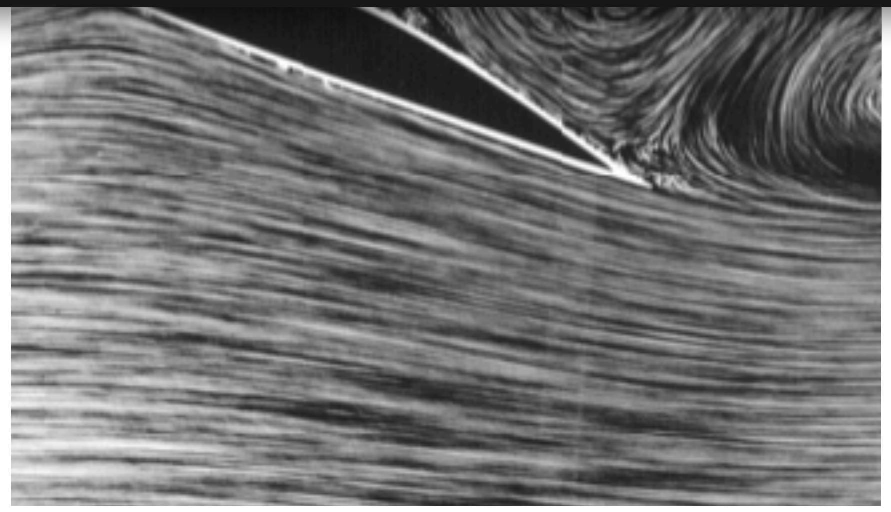
Everywhere around us…

- Fluid Mechanics
- Robotic Control
- Thermodynamics
- Plasma Physics
- Medical Simulations
- Many more…



Tremendous success of numerical simulations

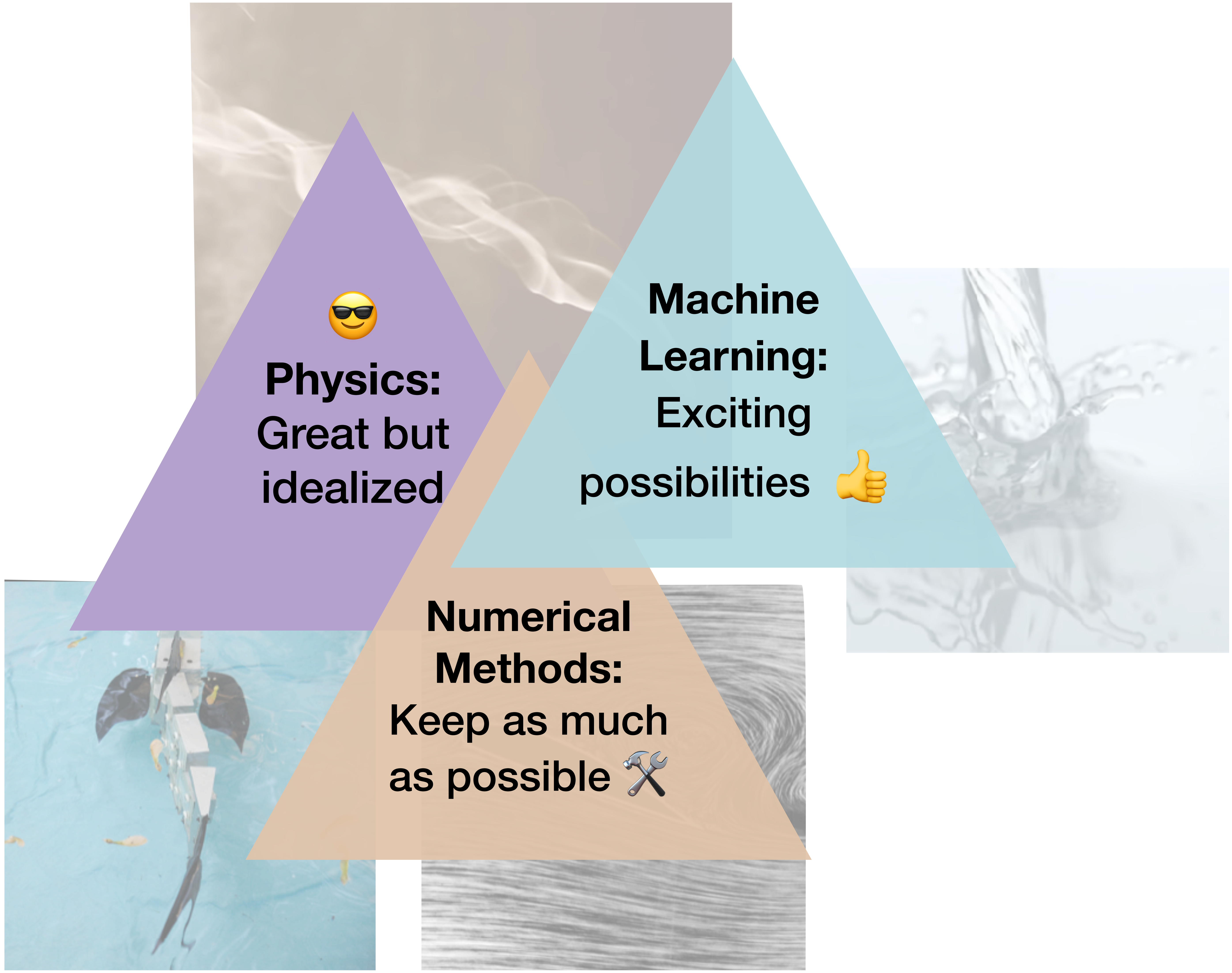




Physical Phenomena

Everywhere around us…

- Fluid Mechanics
- Robotic Control
- Thermodynamics
- Plasma Physics
- Medical Simulations
- Many more…



Related Work

- *Schnell et. al*: Half-Inverse Gradients for Physical Learning
- *Holl et. al*: (Scale-invariant) Physical Gradients for Deep Learning
- *Um et. al*: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers
- *Bar-Sinai et. al*: Learning data-driven discretizations for partial differential equations
- *Raissi et. al:* Hidden physics models: Machine learning of nonlinear partial differential equations
- *Chen et. al.*: Neural ordinary differential equations
- *Morton et. al:* Deep dynamical modeling and control of unsteady fluid flows
- …

Differentiable Simulations in a Nutshell

Discretized PDE $\mathcal{P}$ with phase space states $\mathbf{s}$

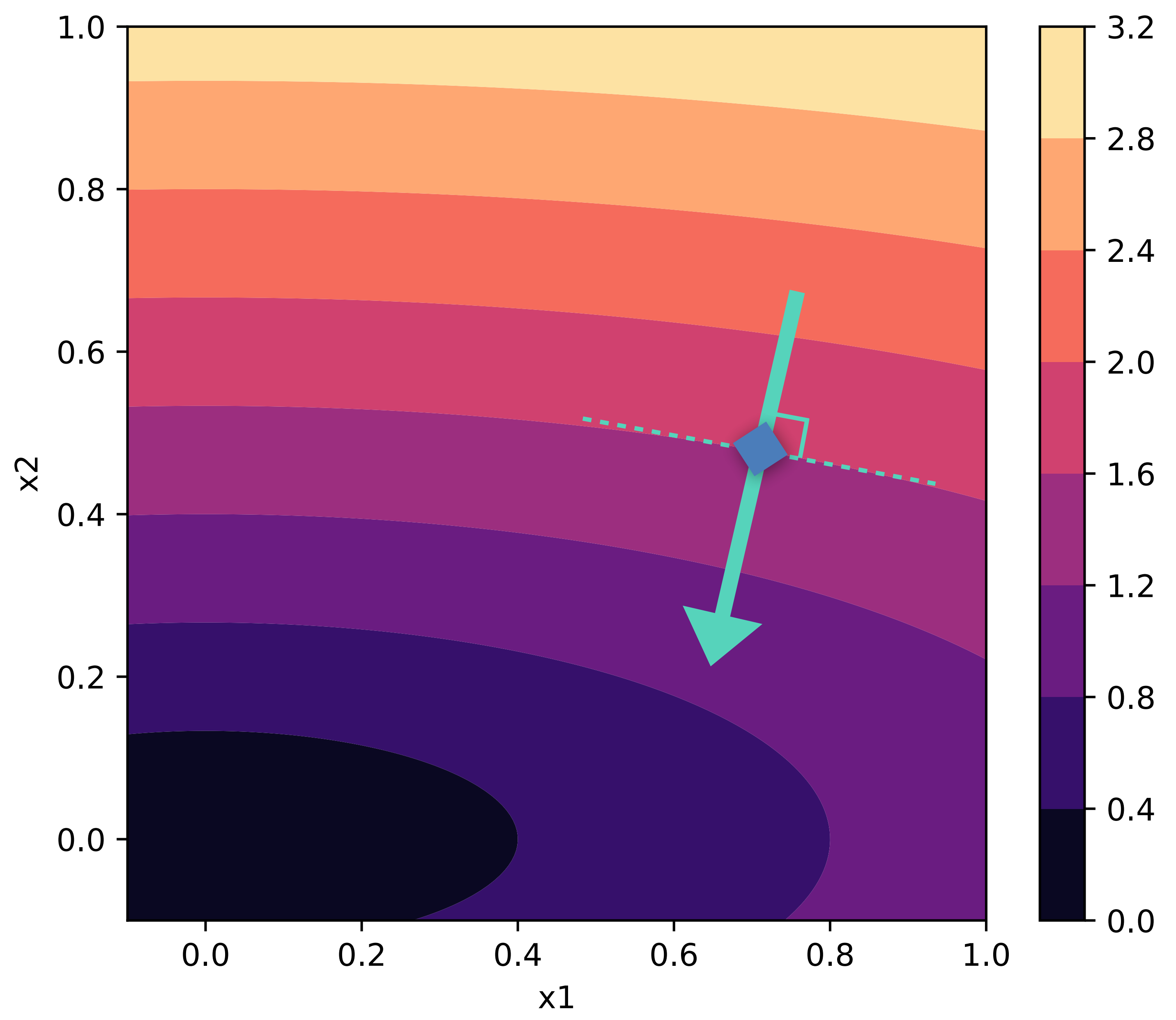
Learn via gradient $(\partial \mathcal{P} / \partial \mathbf{s})^T$

E.g., with loss L and $\mathbf{s} = \mathrm{NN}(\mathbf{x} \,|\, \theta)$

Gradient is $-\eta \frac{\partial \mathbf{s}}{\partial \theta}^T \frac{\partial \mathcal{P}}{\partial \mathbf{s}}^T \frac{\partial L}{\partial \mathcal{P}}^T$

Requires differentiable physics simulator for $\mathcal{P}$

→ Tight integration of numerical methods and learning process

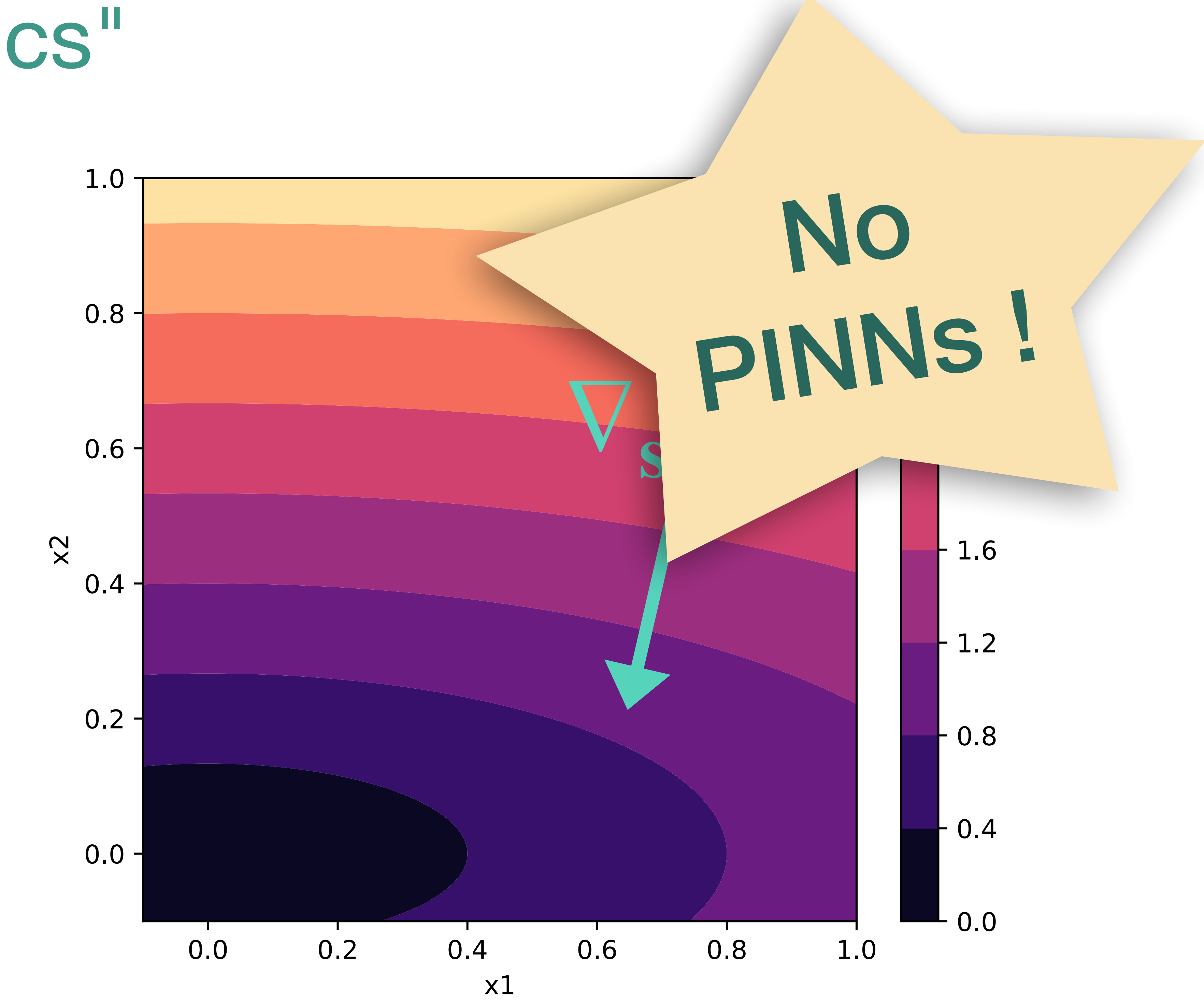


Differentiable Simulations - Terminology

Differentiable PDE solver for $\mathscr{P}$ = “differentiable physics"

Equivalent:

- *Adjoint method / differentiation*
- *Reverse-mode / backward differentiation*
- *Backpropagation*



D. Kahneman: System 1 & 2

System 1 / NN Component	System 1 / NN Component	System 1 / NN Component	System 1 / NN Component	System 1 / NN Component	System 1 / NN Componen
System 2 / Simulation	System 2 / Simulation	System 2 / Simulation	System 2 / Simulation	System 2 / Simulation	System 2 / Simulation

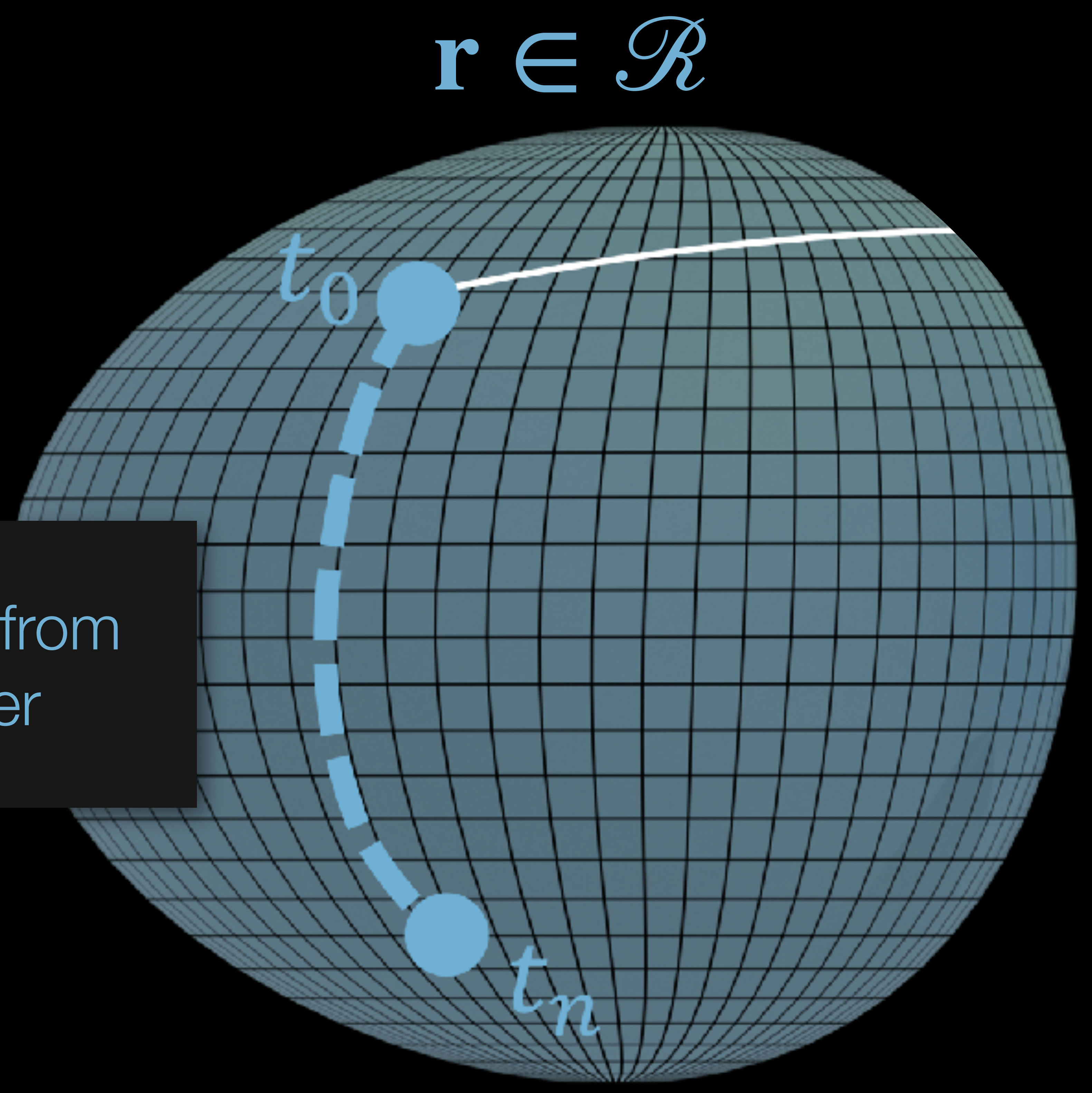
Differentiable Physics in Action

Um et. al: Solver-in-the-Loop: Learning from Differentiable Physics to Interact with PDE-Solvers
List et. al: Learned Turbulence Modelling with Differentiable Fluid Solvers

Reducing Numerical Errors

"Solver-in-the-Loop"

PDE: $\mathscr{P}$

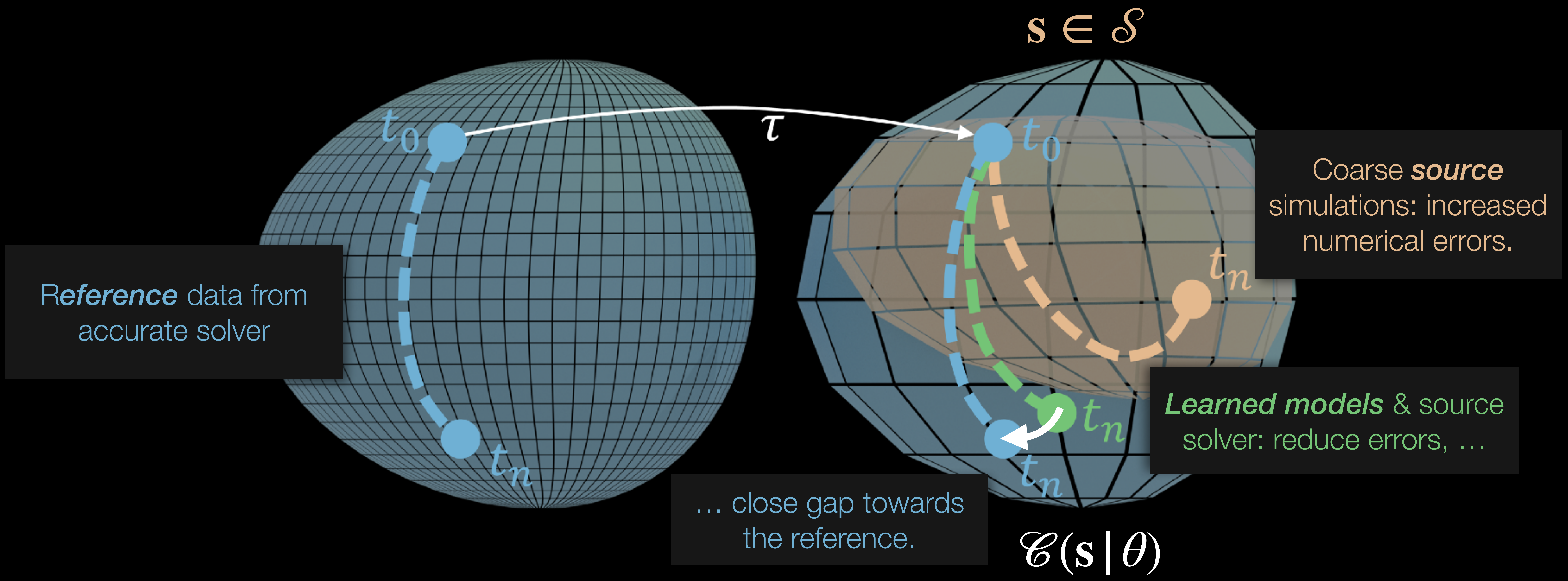


Reference data from accurate solver

Um et. al: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers

Reducing Numerical Errors

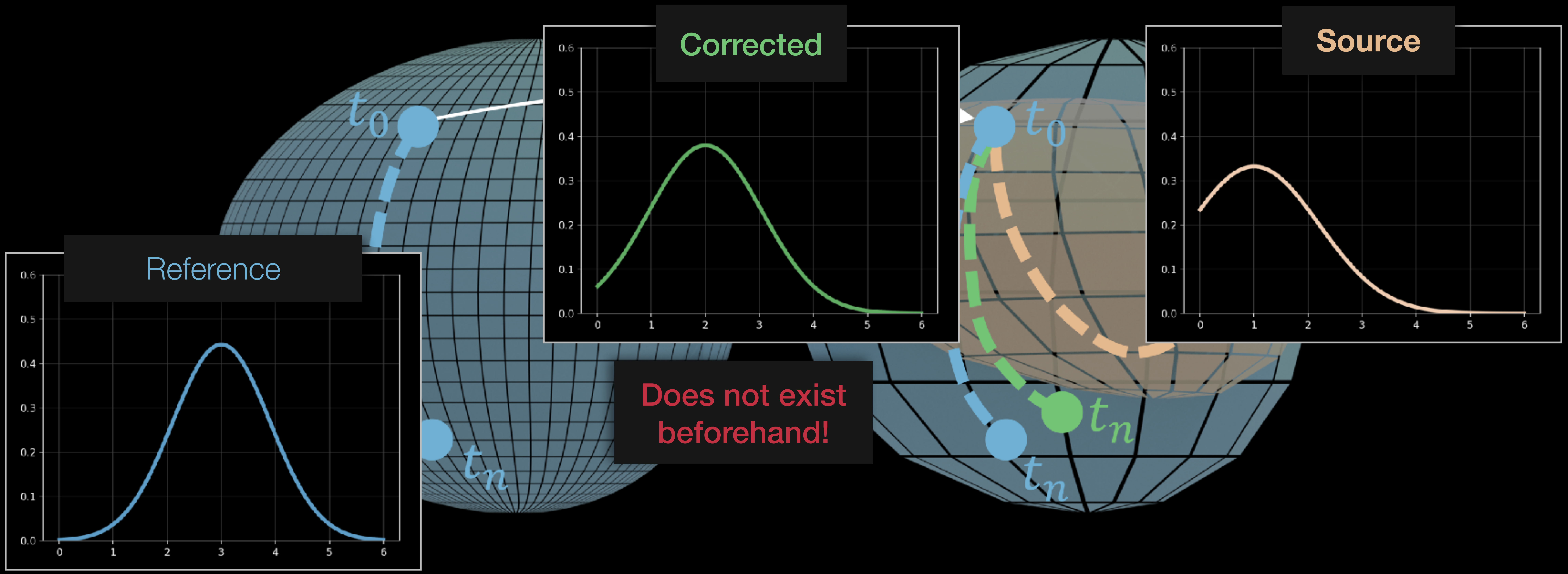
"Solver-in-the-Loop"



Um et. al: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers

Reducing Numerical Errors

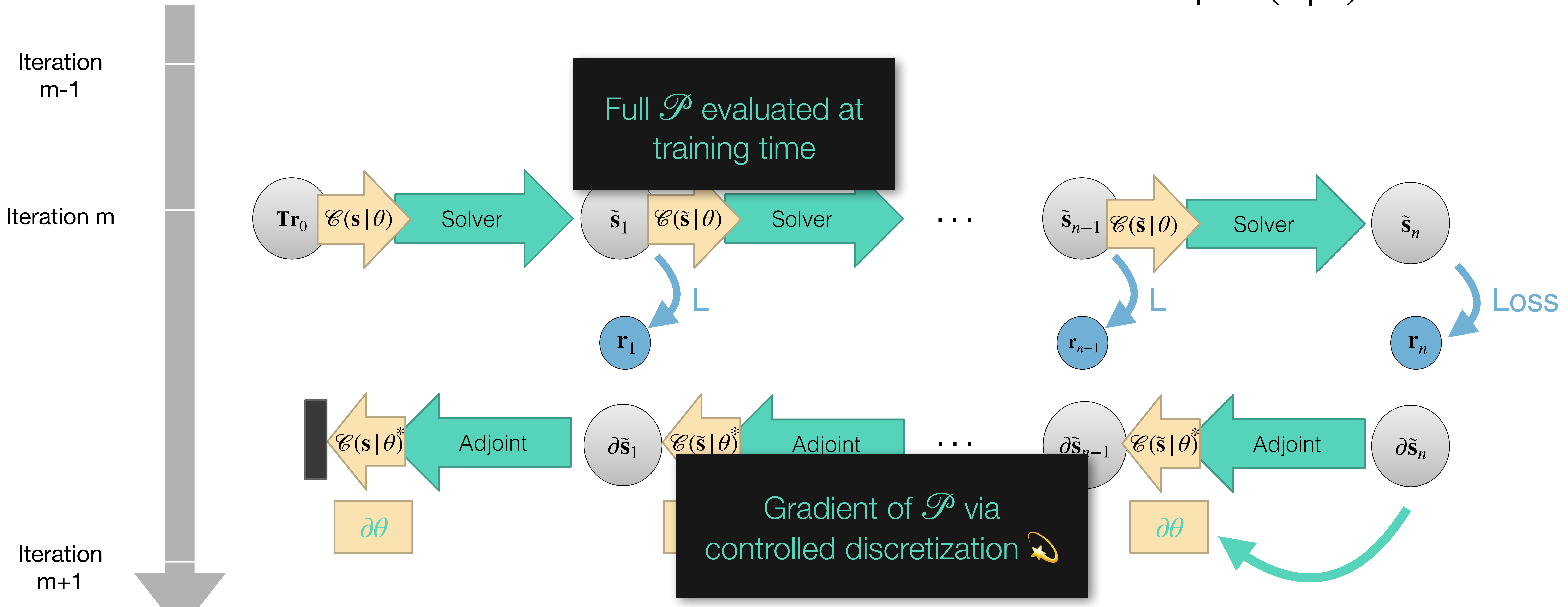
Shift of Input Feature Distributions



Reducing Numerical Errors

Learning via Differentiable Physics

Correction via network for each unrolled simulation step $\mathscr{C}(\tilde{\mathbf{s}} \,|\, \theta)$

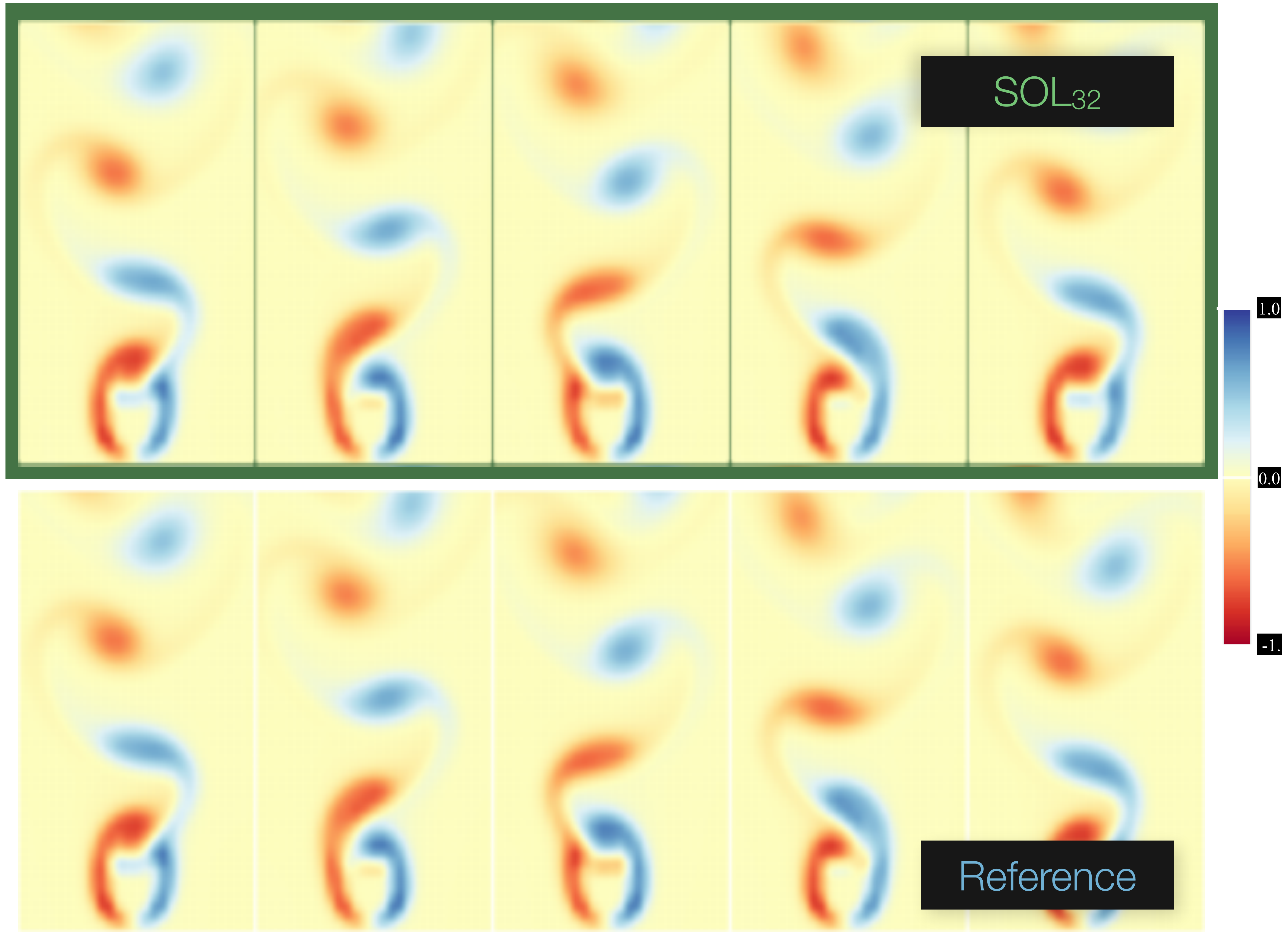


Um et. al: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers

A few more Details…

Unsteady Wake Flow in 2D

- Setup: Reference is 4x
- 3000 frames training data, Re $\in \{98 \;..\; 3125\}$
- Test data: new Re Nr.s
- Source MAE: 0.146
- SOL_{32} MAE: 0.013
- More than 10x reduction



Re=2343.8

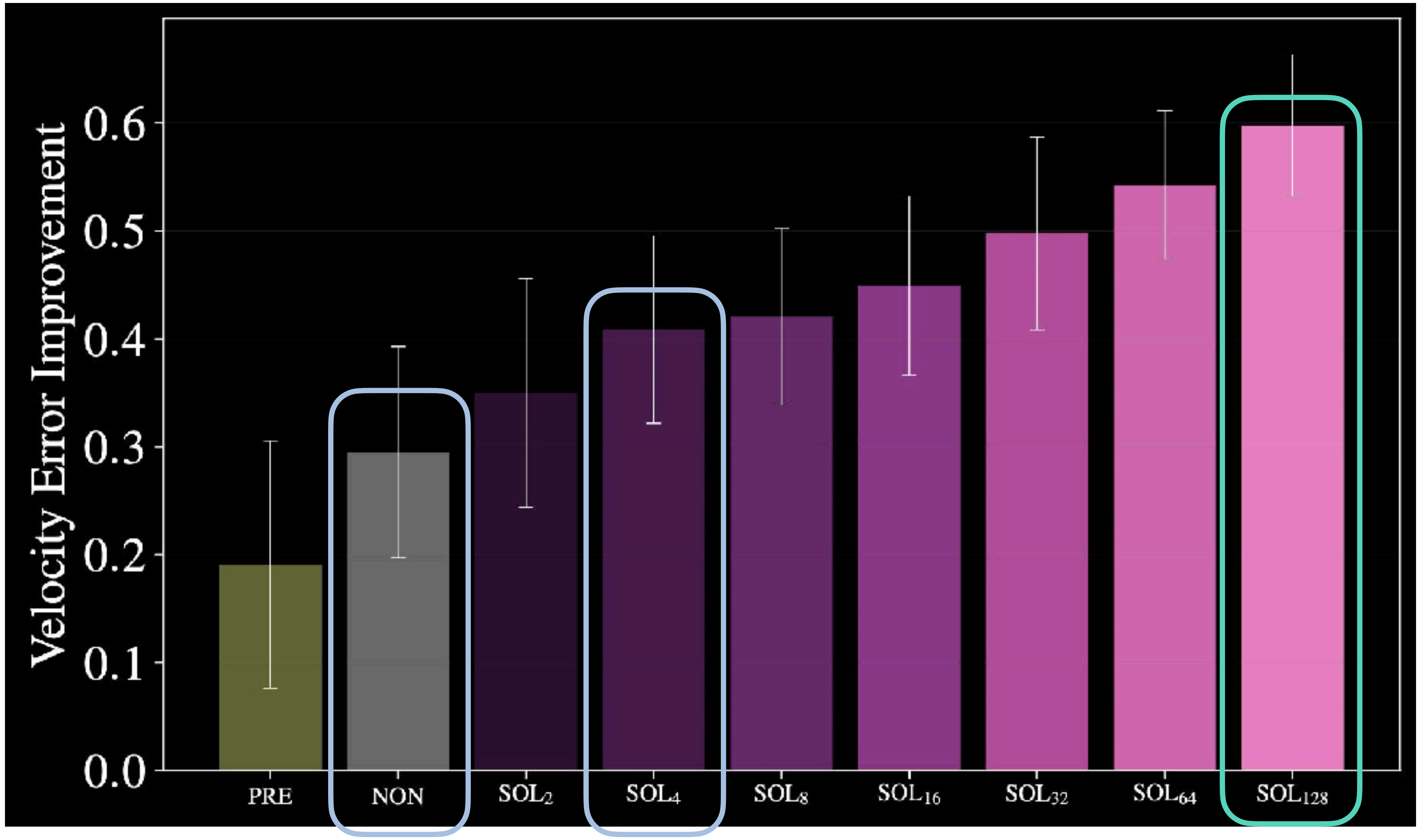
Um et. al: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers

Looking into the Future

Learning via a Large Number of Simulation Steps

Evaluation:

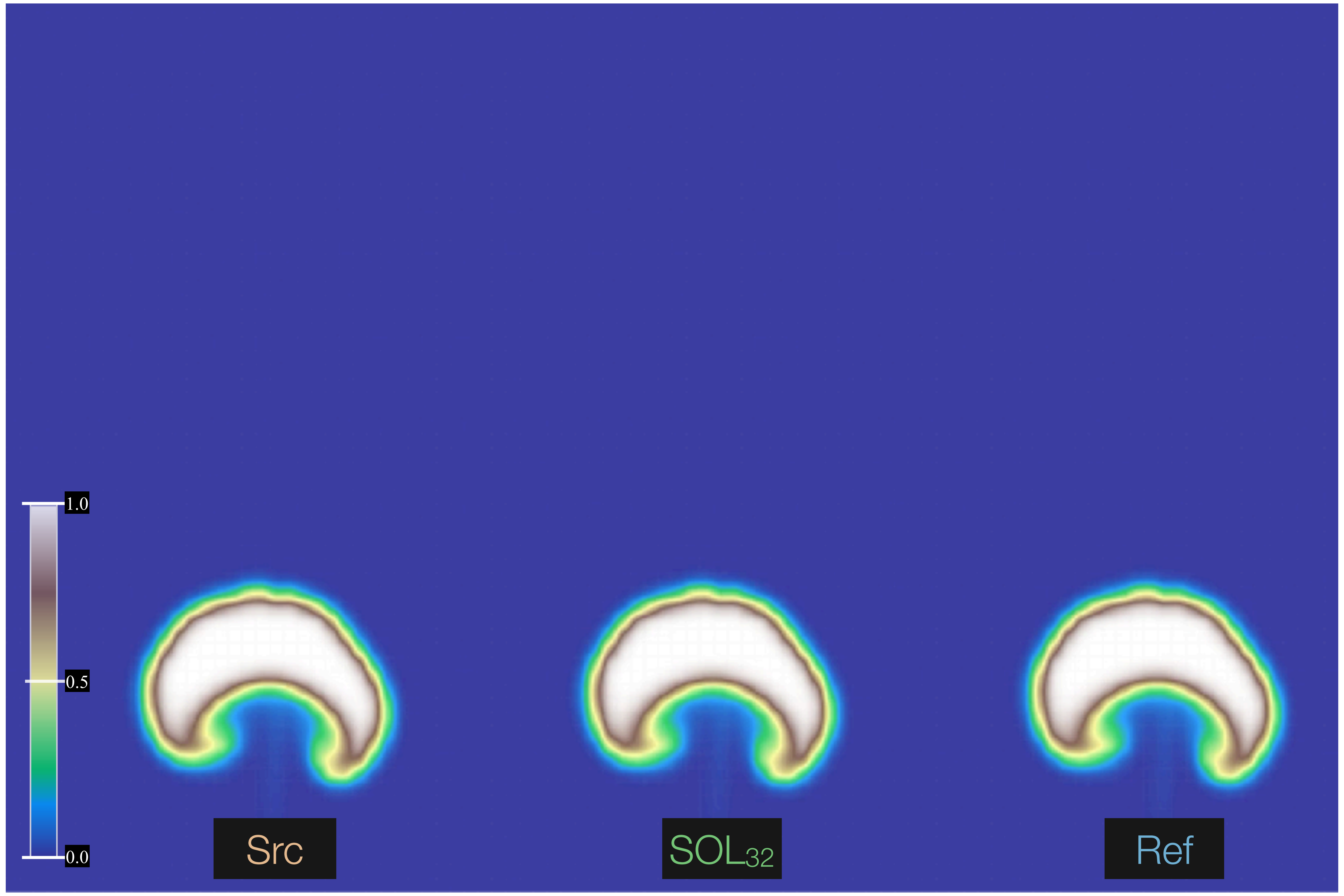
- MAE Improvement over Src
- Supervised training: 29%
- D.P. with 4 steps: 41%
- D.P. with 128 steps: 60%



Generalization

Improved generalization due to varied, gradient-based training feedback

- Better performance for previously unseen inputs
- Flexible due to combination with source solver

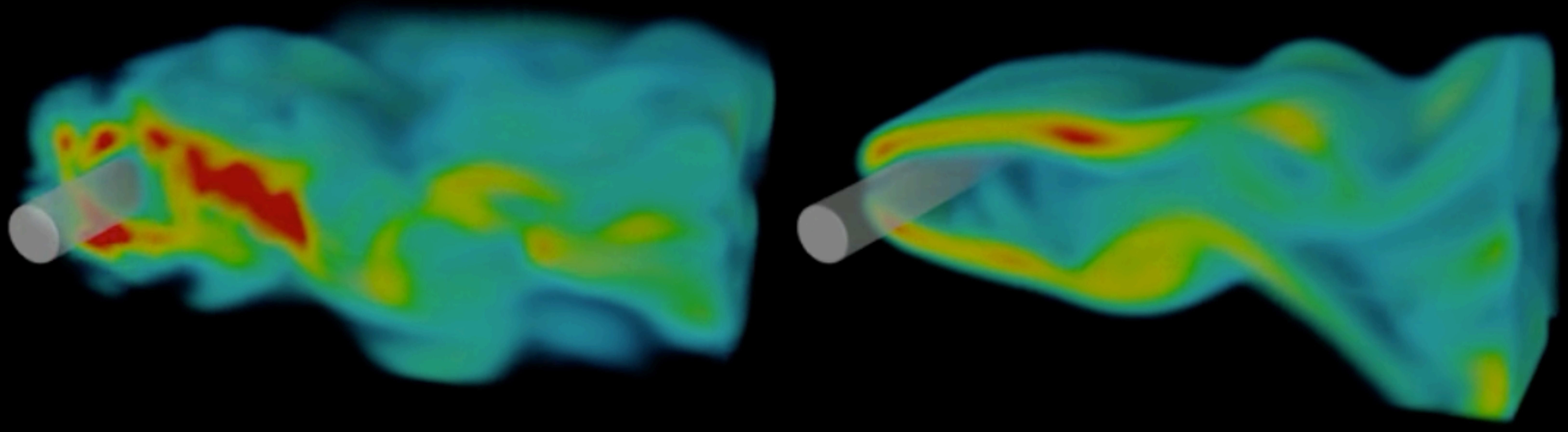


Um et. al: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers

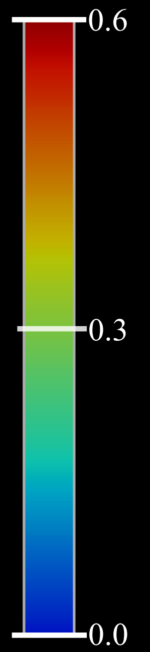
Long-term Stability

Unsteady Wake Flow (250 time steps)

3D Test Case, Re=468.8



NON	SOL_{16}
MAE=0.144	MAE=0.130



Um et. al: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers

3D Results

Unsteady Wake Flow in 3D, Re=546.9

Um et. al: Solver-in-the-Loop, Learning from Differentiable Physics to Interact with PDE-Solvers

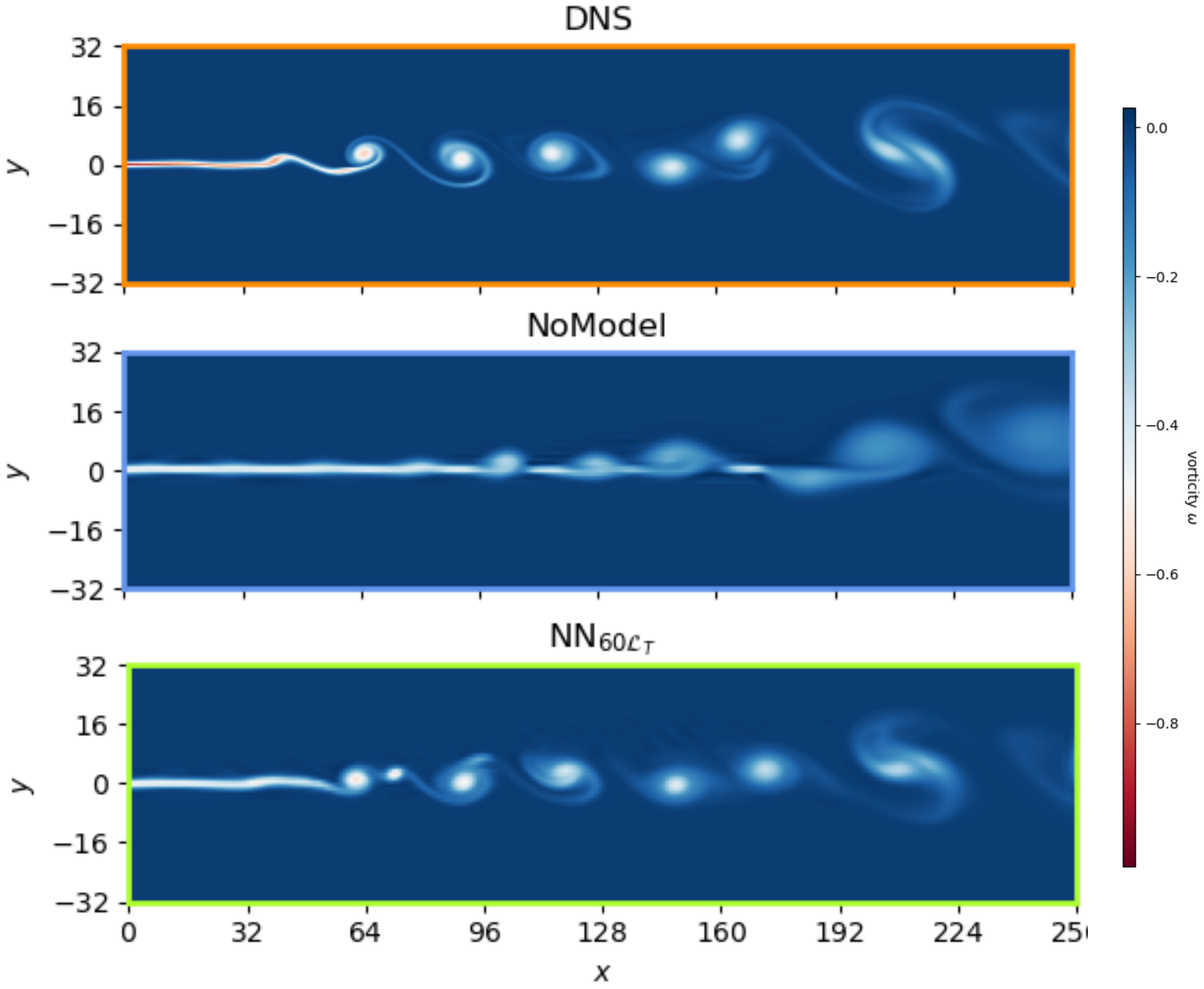
Differentiable Physics

Wide Range of Applications

- Error reduction for (generic) PDEs
- Control problems
- Plasma simulations
- Model completion (reacting flows)
- Turbulence

Turbulence: Spatial Mixing Layer

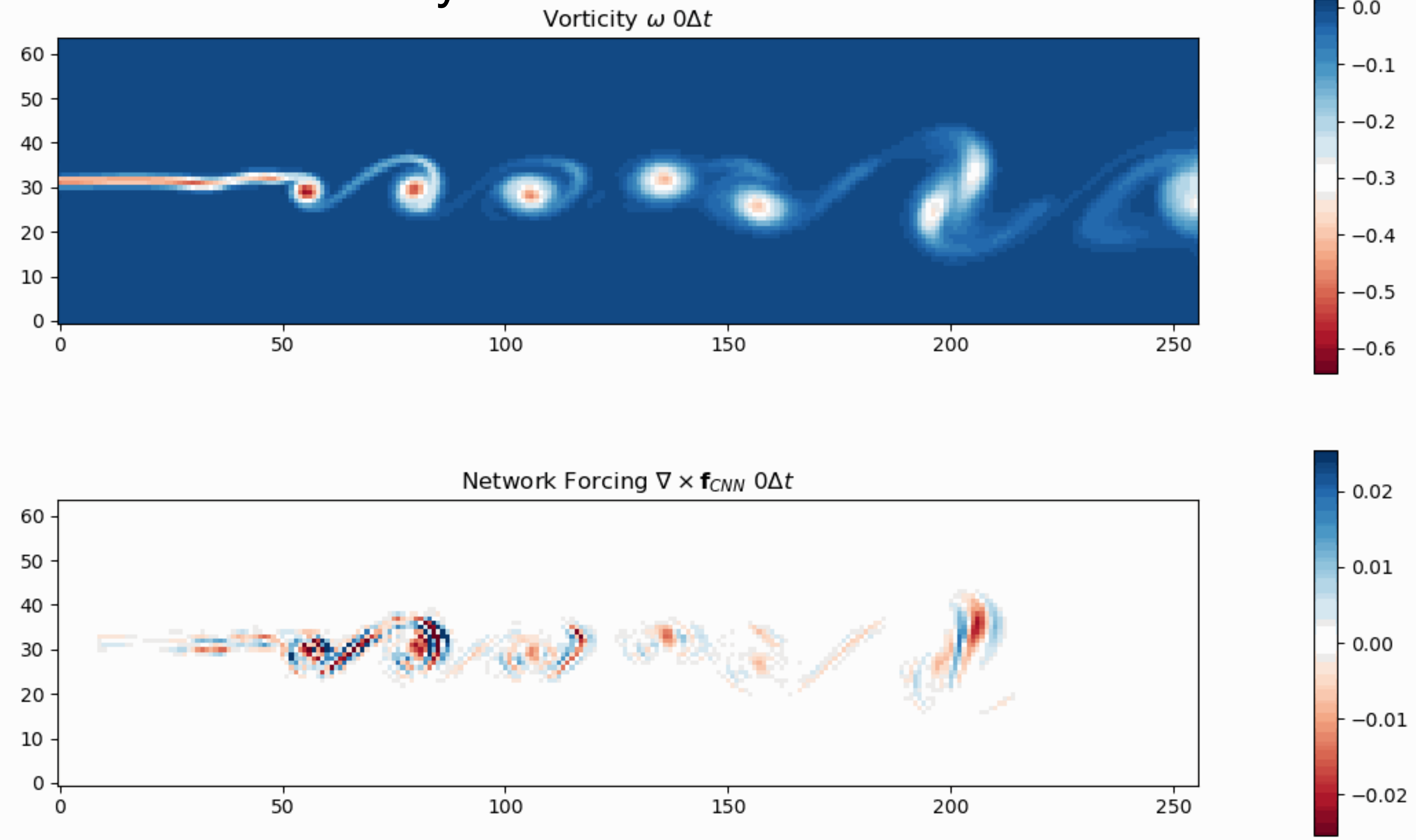
- Semi-implicit PISO solver (2nd order in time)
- Shear layer with vorticity thickness Re = 500
- Evaluate on test set of unseen perturbation modes



List et. al: Learned Turbulence Modelling with Differentiable Fluid Solvers

Turbulence: Spatial Mixing Layer

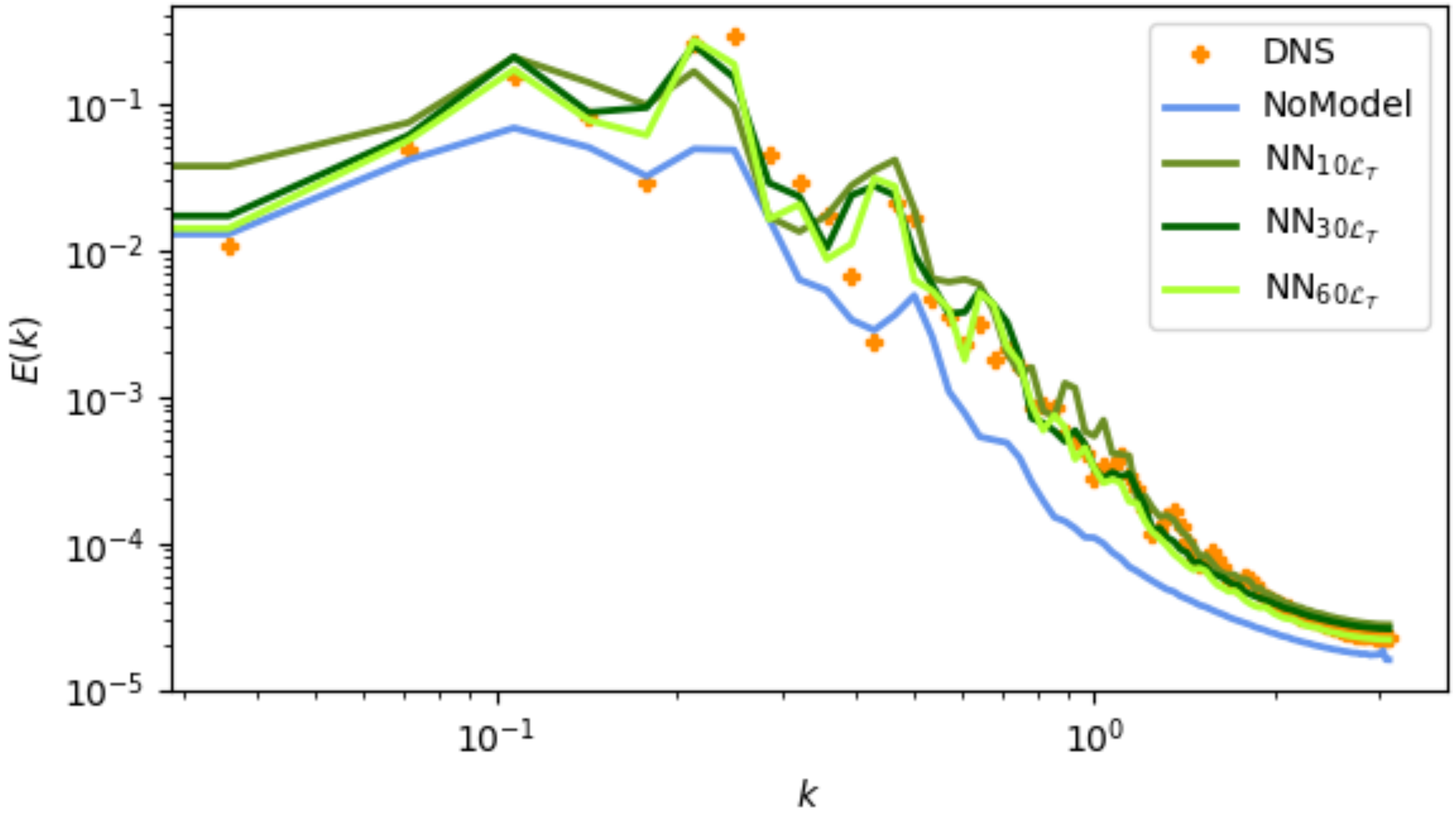
Learned Simulator only:



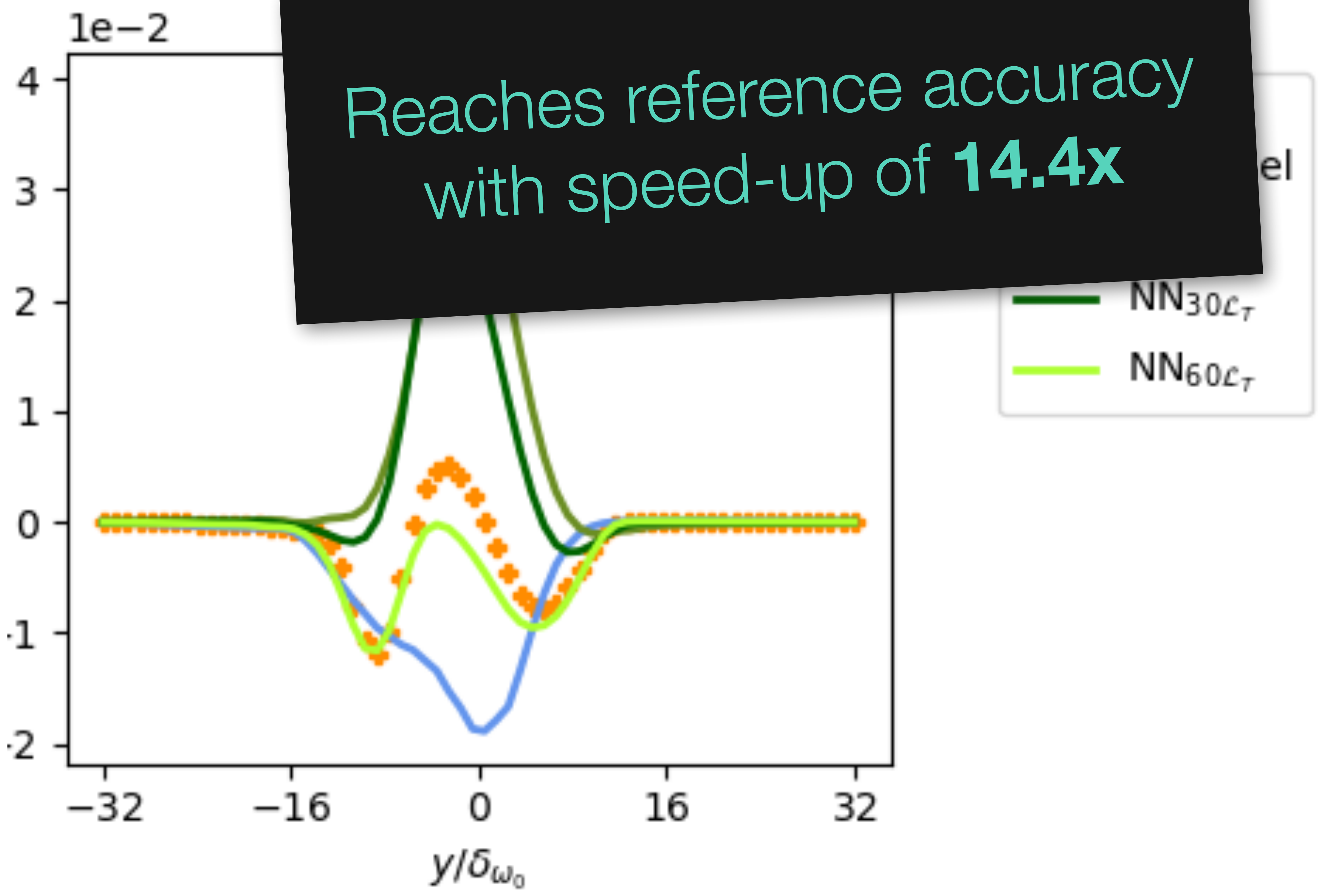
List et. al: Learned Turbulence Modelling with Differentiable Fluid Solvers

Turbulence: Spatial Mixing Layer

Closely matches DNS turbulence statistics (steady state over 2500 steps)



Energy spectrum



Reaches reference accuracy with speed-up of **14.4x**

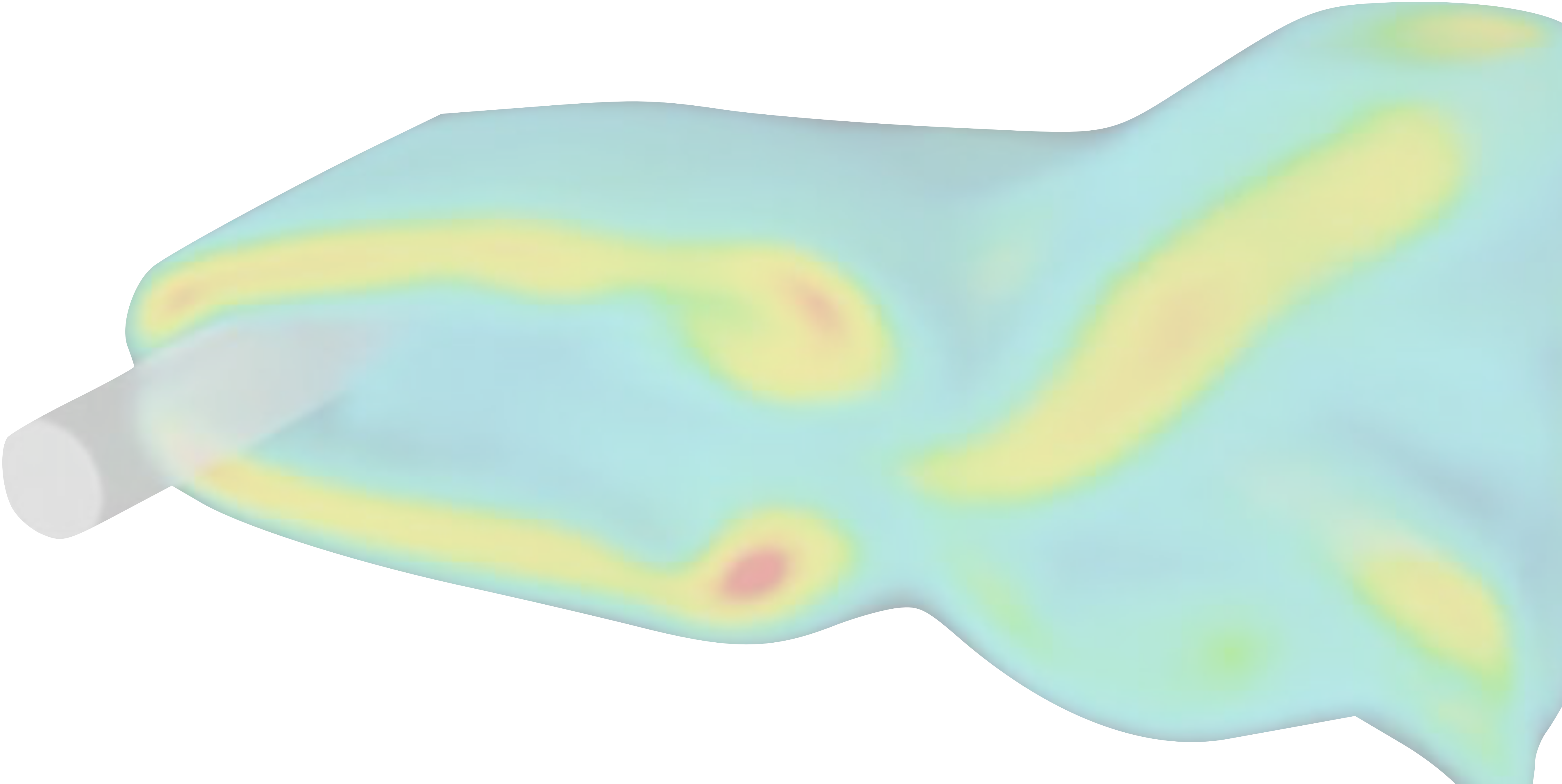
Reynolds stresses

List et. al: Learned Turbulence Modelling with Differentiable Fluid Solvers

Training via Differentiable Physics

Numerous Advantages:

- Generalization
- Runtime performance
- Improved accuracy



Improved Updates via Inversion

Holl et. al: Scale-invariant / Physical Gradients for Deep Learning
Schnell et. al: Half-Inverse Gradients for Physical Learning

Improved Learning Updates

Motivation

So far taken for granted: deep learning paradigm of optimizing via $\partial \mathscr{P} / \partial \mathbf{s}^T$

Has fundamental problems in physical settings

- Units are wrong: deep learning gradient for **s** was: $-\eta \frac{\partial \mathscr{P}}{\partial \mathbf{s}}^T \frac{\partial L}{\partial \mathscr{P}}^T$
- Update should have units of **s**, as in Newton's method: $-\eta \left(\frac{\partial^2 L}{\partial \mathbf{s}^2} \right)^{-1} \frac{\partial L}{\partial \mathbf{s}}^T$

➡ Scaling problems & unstable training

Improved Learning Updates

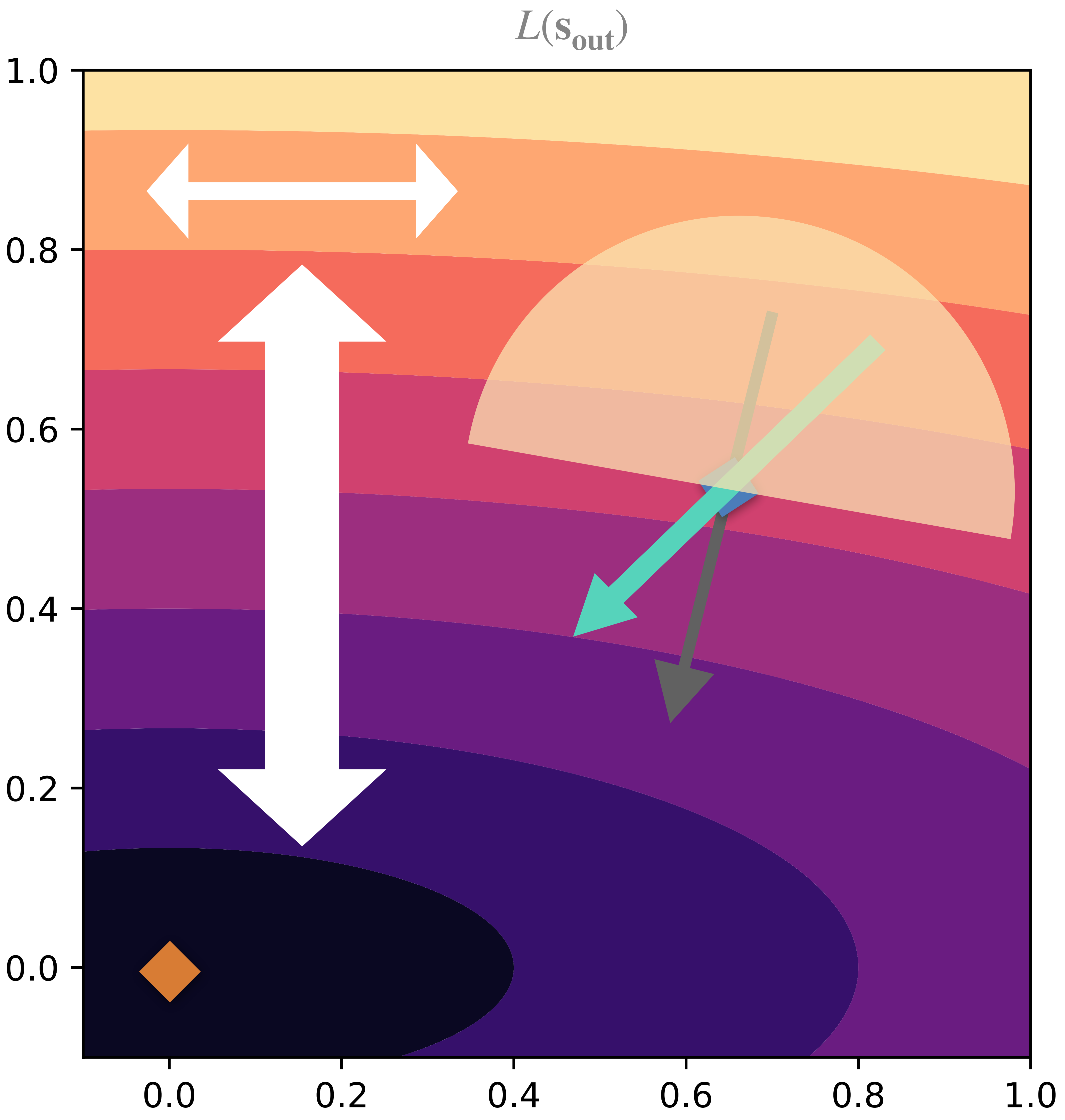
Inversion is Crucial

Steepest direction via $\frac{\partial L}{\partial \mathbf{s}}^T$ not optimal

Example: steep dimension severely limits steps

Inversion accounts for rescaling, e.g. $\left(\frac{\partial L}{\partial \mathbf{s}}^T\right)^{-1}$

Either *numerical or analytical* inversion



Holl et. al: Scale-invariant / Physical Gradients for Deep Learning

Physical Gradients

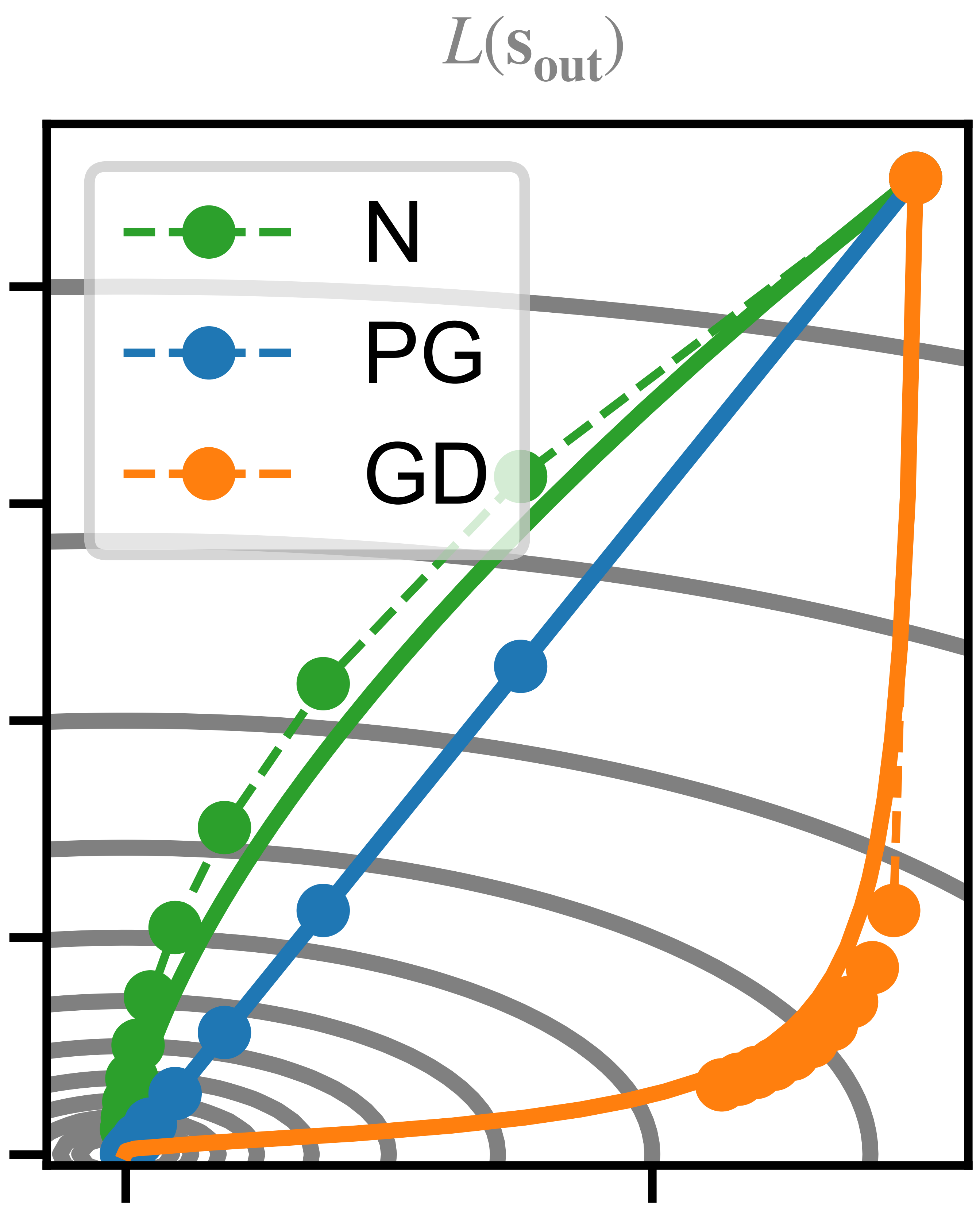
Leverage Inverse Solver for Update Step

Employ (custom) inverse solver $\mathscr{P}^{-1}$

Compute NN *update step* via proxy-L2 loss

$$\Delta\theta_{\text{PG}} = -\eta \frac{\partial \mathbf{s}}{\partial \theta}^T \left(\mathbf{s} - \mathscr{P}^{-1}(\mathbf{s}_{out}) \right)$$

Update step integrates gradient w.r.t. outputs $\mathbf{s}_{out}$

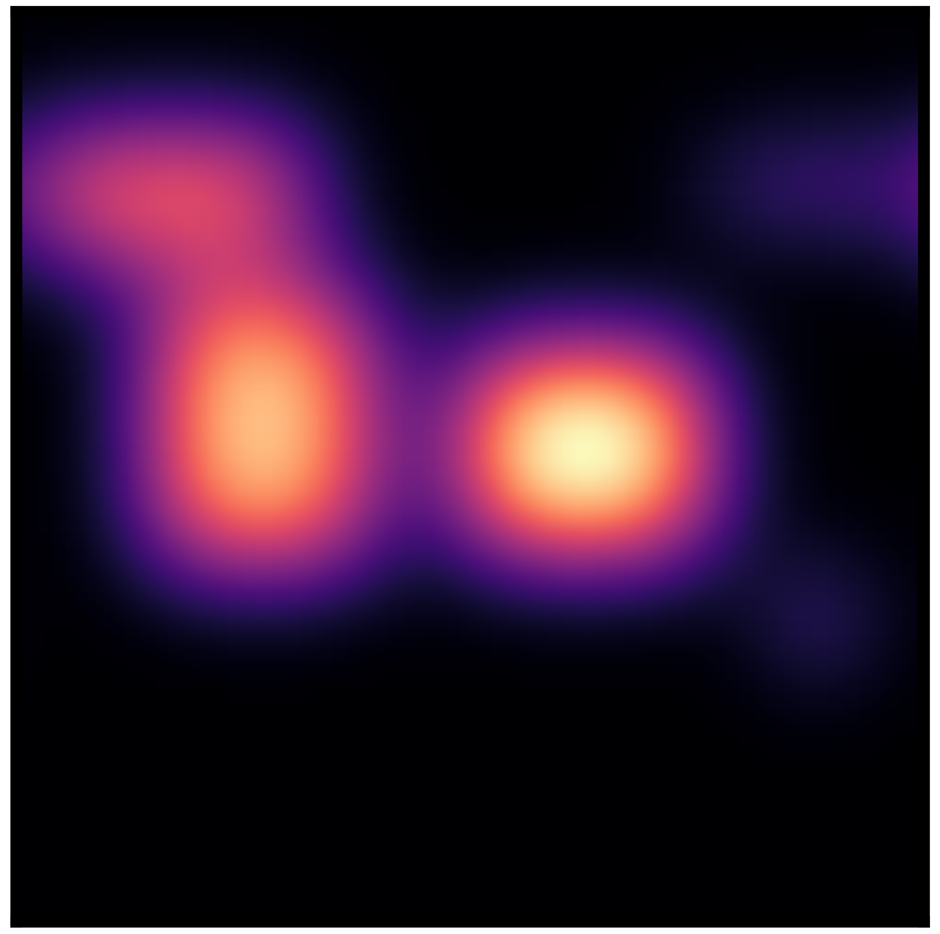


Holl et. al: Scale-invariant / Physical Gradients for Deep Learning

Scale-Inverse Physics Gradients

NN Solving Inverse Problem with Heat Diffusion

y^*



Observation

Only difference: training method (Adam or Adam + PG)

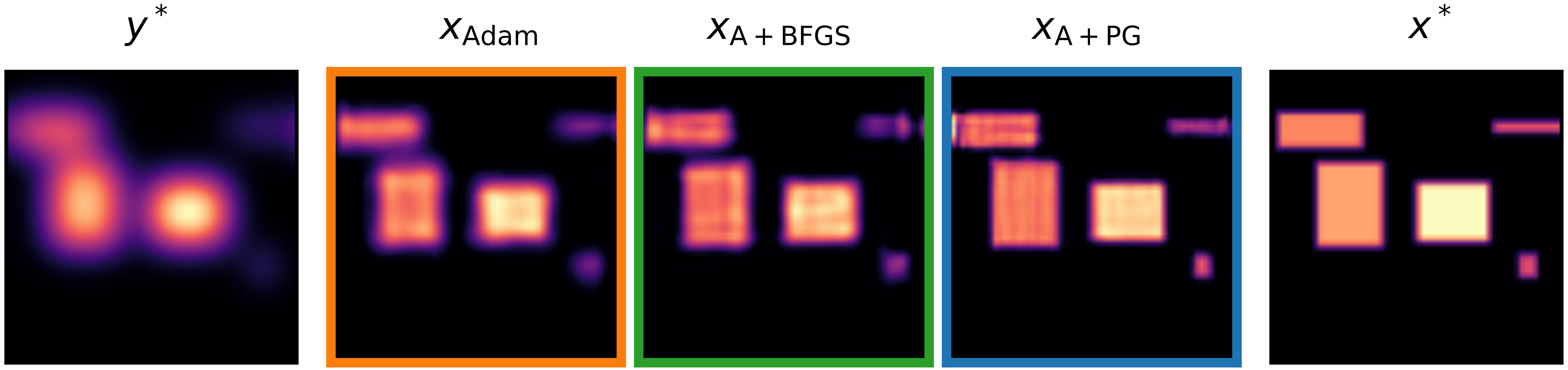
x^*

Reconstruct Input

Holl et. al: Scale-invariant / Physical Gradients for Deep Learning

Scale-Inverse Physics Gradients

NN Solving Inverse Problem with Heat Diffusion



Identical NNs!

Half-inverse Gradients

Joint Inversion of Physics and Network

Partially invert Jacobian from NN and simulator jointly

Resulting update step $\Delta\theta_{\text{HIG}} = -\eta\left(\frac{\partial\mathscr{P}}{\partial\theta}\right)^{-1/2}\left(\frac{\partial L}{\partial\mathscr{P}}\right)^{\top}$

Over all samples of a mini-batch

Update represents optimal & scale respecting first-order step

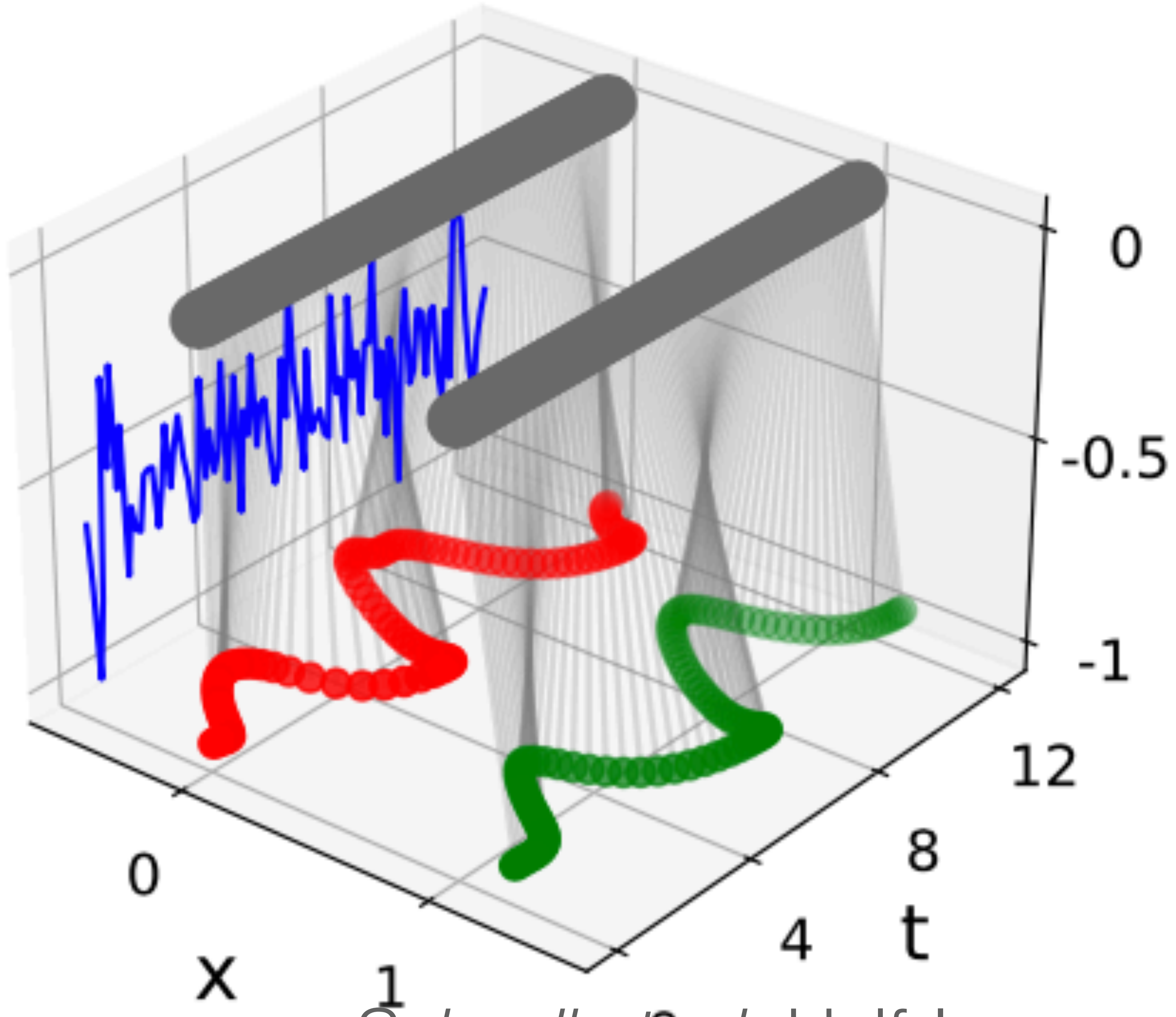
Schnell et. al: Half-Inverse Gradients for Physical Learning

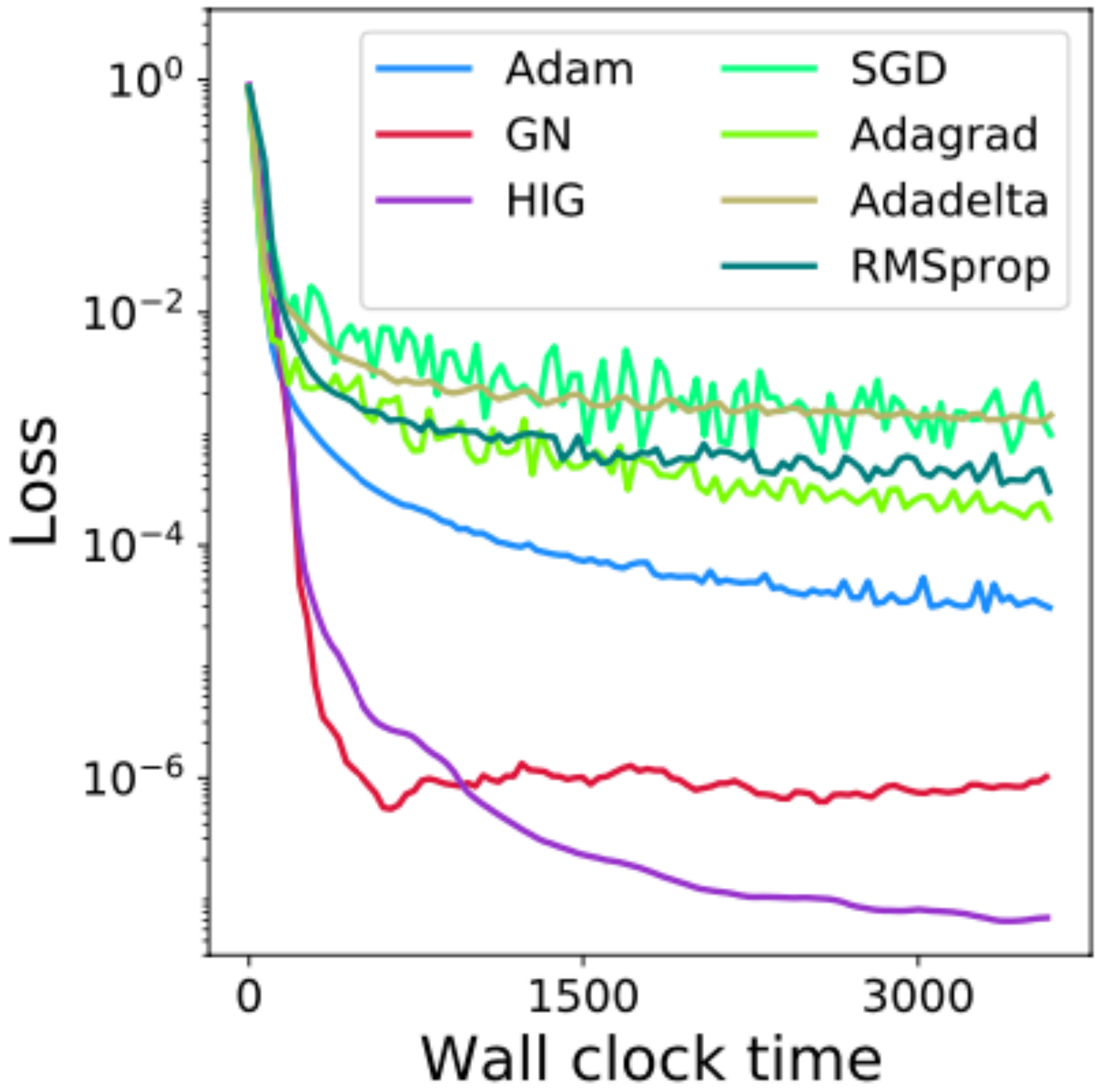
Half-inverse Gradients

Non-linear Oscillator

Classical problem setup with non-linear force term

Backprop through 96 time integration steps (RK4)





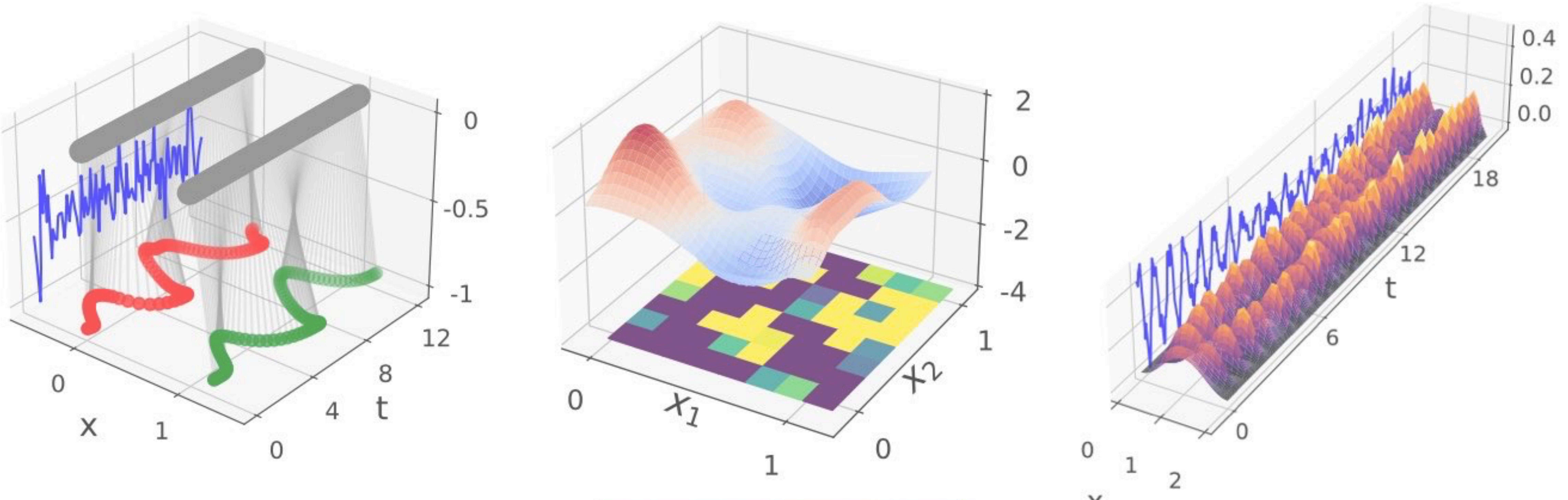
Schnell et al: Half-Inverse Gradients for Physical Learning

Improved Gradients - Summary

Fundamentally improved learning directions

Yields neural network states that are unreachable with simpler methods

→ Illustrates potential gains from going *beyond* 1st-order gradients

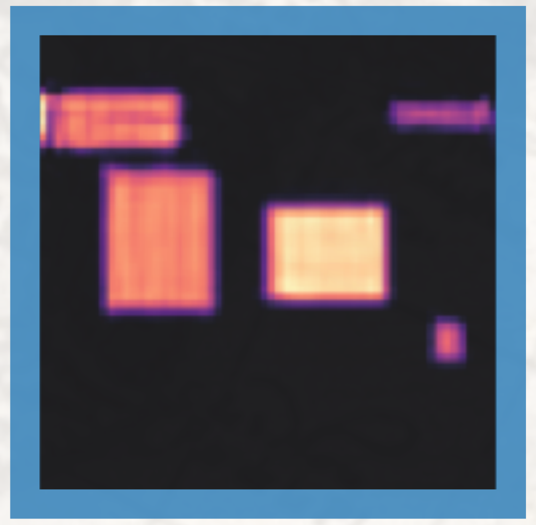


Summary & Outlook

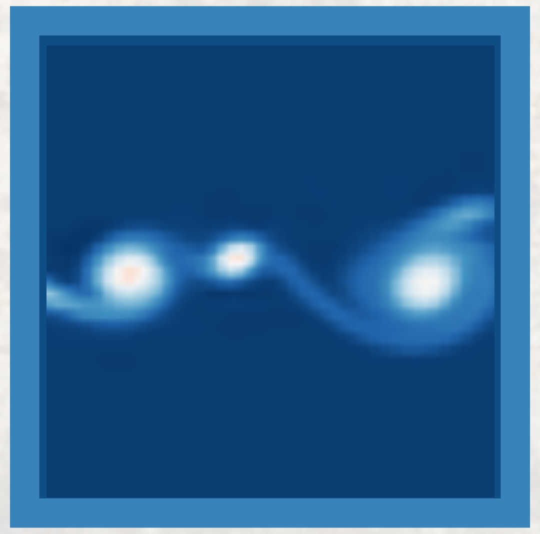
Summary

Differentiable Simulations and Inversion

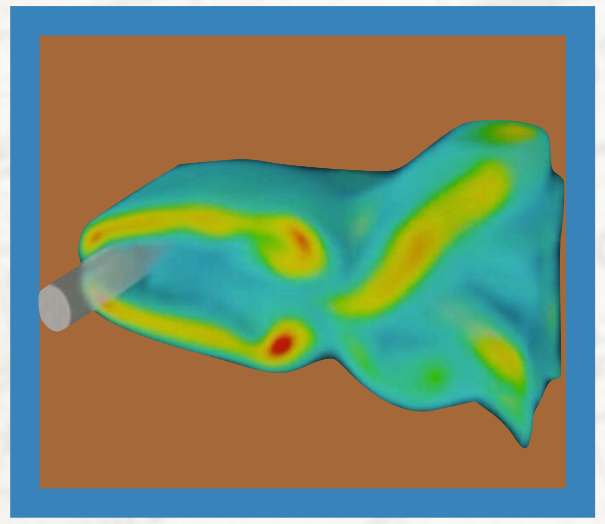
as Tools to bridge Physics & Learning 🤗



Improved Updates

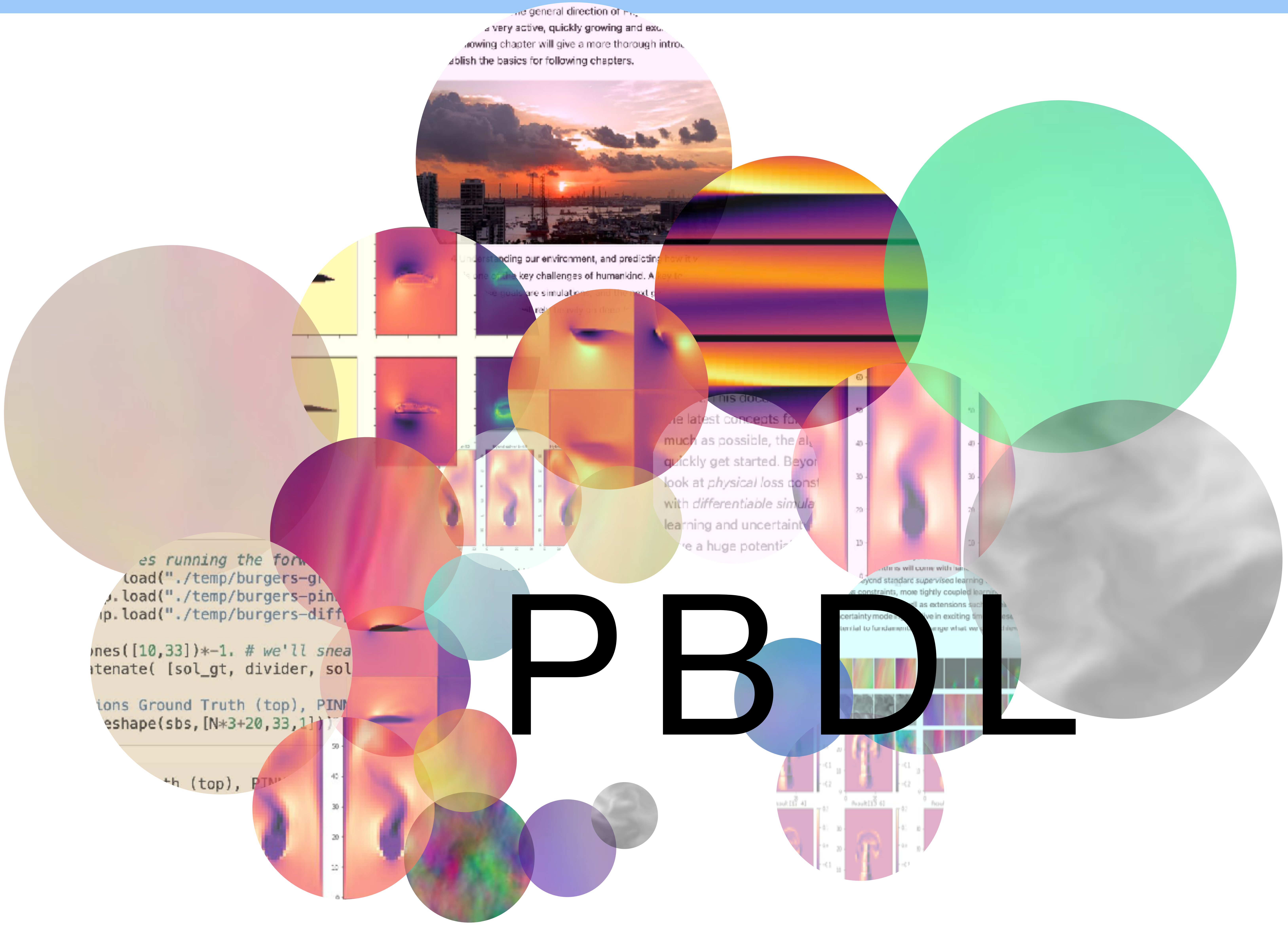


Turbulence Modeling



Error Correction

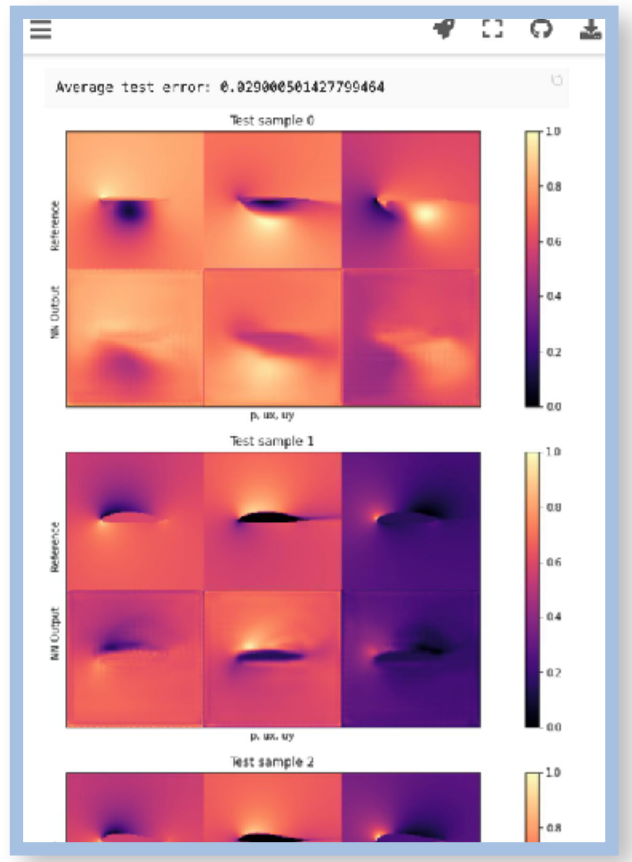
Physics-based Deep Learning - the Book

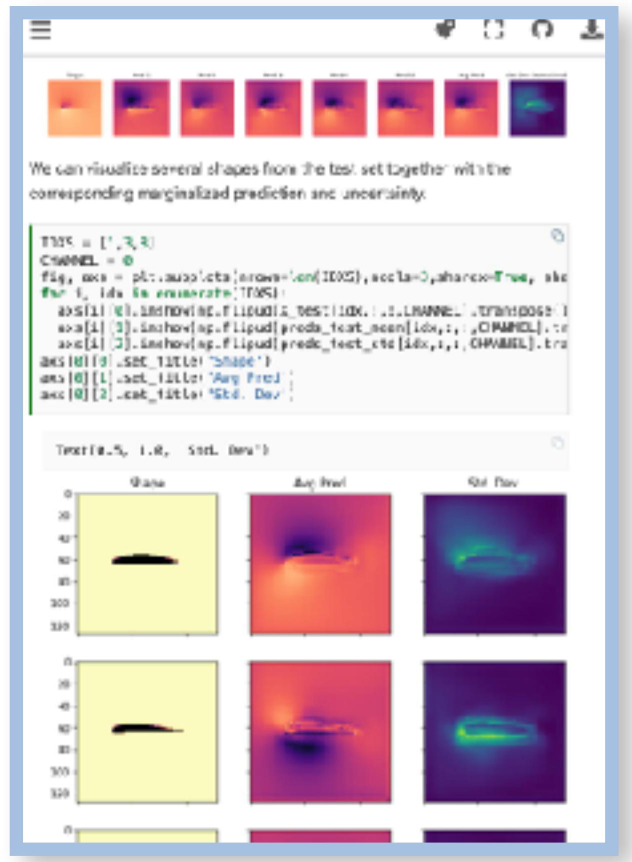


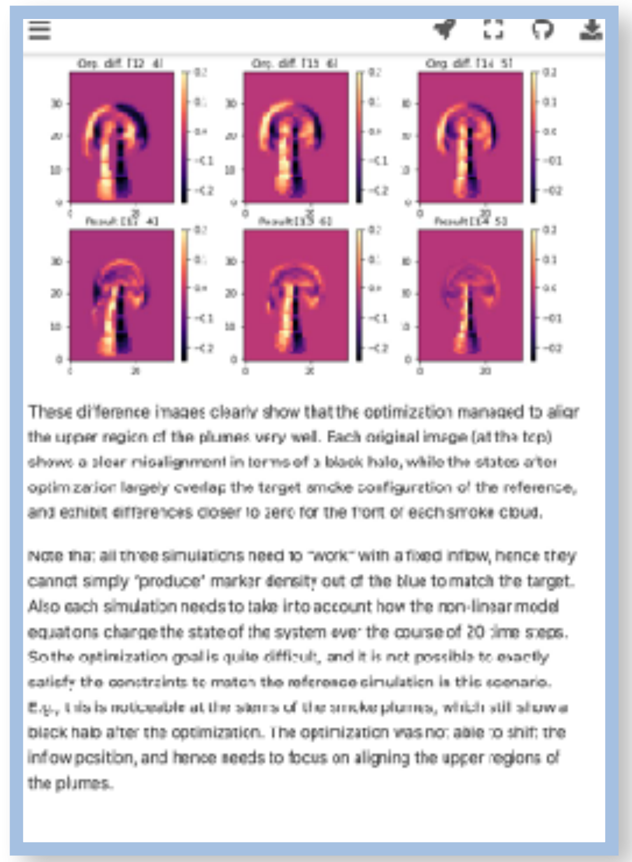
PBDL

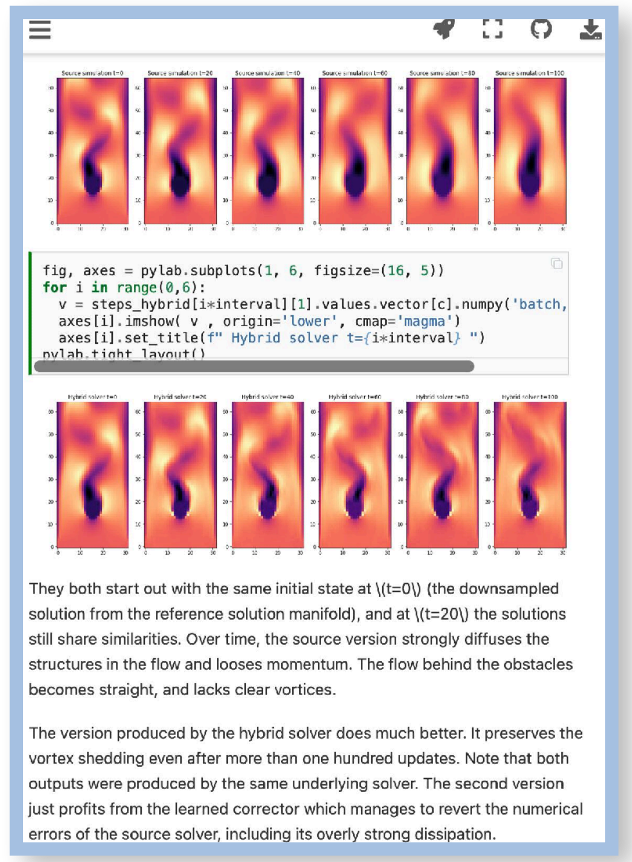
Physics-based Deep Learning - the Book

- Intro to physical simulations & deep learning
- Comprehensive overview
- Hands-on code examples, run on the spot
- Among others: supervised learning, tightly coupled differentiable simulations, reinforcement learning and uncertainty modeling…









Thanks for Listening!

https://physicsbaseddeeplearning.org

Acknowledgements: P. Holl, K. Um, V. Koltun, R. Fei, B. List, L. W. Chen