MXM Midsemester Presentation

Neural Network Parameterization for PDE-Constrained Optimization

Mark Miller, Pritam Kayal, Yifan Yang, PJ Clementson
Advisor: Prof. Yukun Yue
Graduate Student Mentor: Martin Guerra

November 11, 2025

What is PDE-Constrained Problems

PDE-constrained optimization is a special class of optimization problems where the optimization objective is constrained by partial differential equations (PDEs).

- General Form

$$\min_{f} J(u, f) \quad \text{subject to} \quad \mathcal{L}(u, f) = 0$$

The Inverse Problem

Consider the heat equation:

$$\Delta_x u - \partial_t u = f \text{ in } \Omega \times I$$

$$u = u_0 \text{ in } \partial\Omega \times I$$

Forward Problem: Given f, solve the PDE to get u
Inverse Problem: Given some observations about u, called u^{obs}, find f

Methods:

- Adjoint Method (Manual)
- Automatic Differentiation
- Automatic Differentiation + Neural Network Parameterization

Primary Idea

Idea: Frame it as a PDE constrained optimization problem.

$$J(u; f) = ||u^{\text{obs}} - u||_2 + \lambda ||f||_2$$

Vanilla:

$$\min_f J(u; f)$$

$$\text{s.t } \Delta_x u - \partial_t u = f$$

Neural Net:

$$\min_\theta J(u; f_\theta)$$

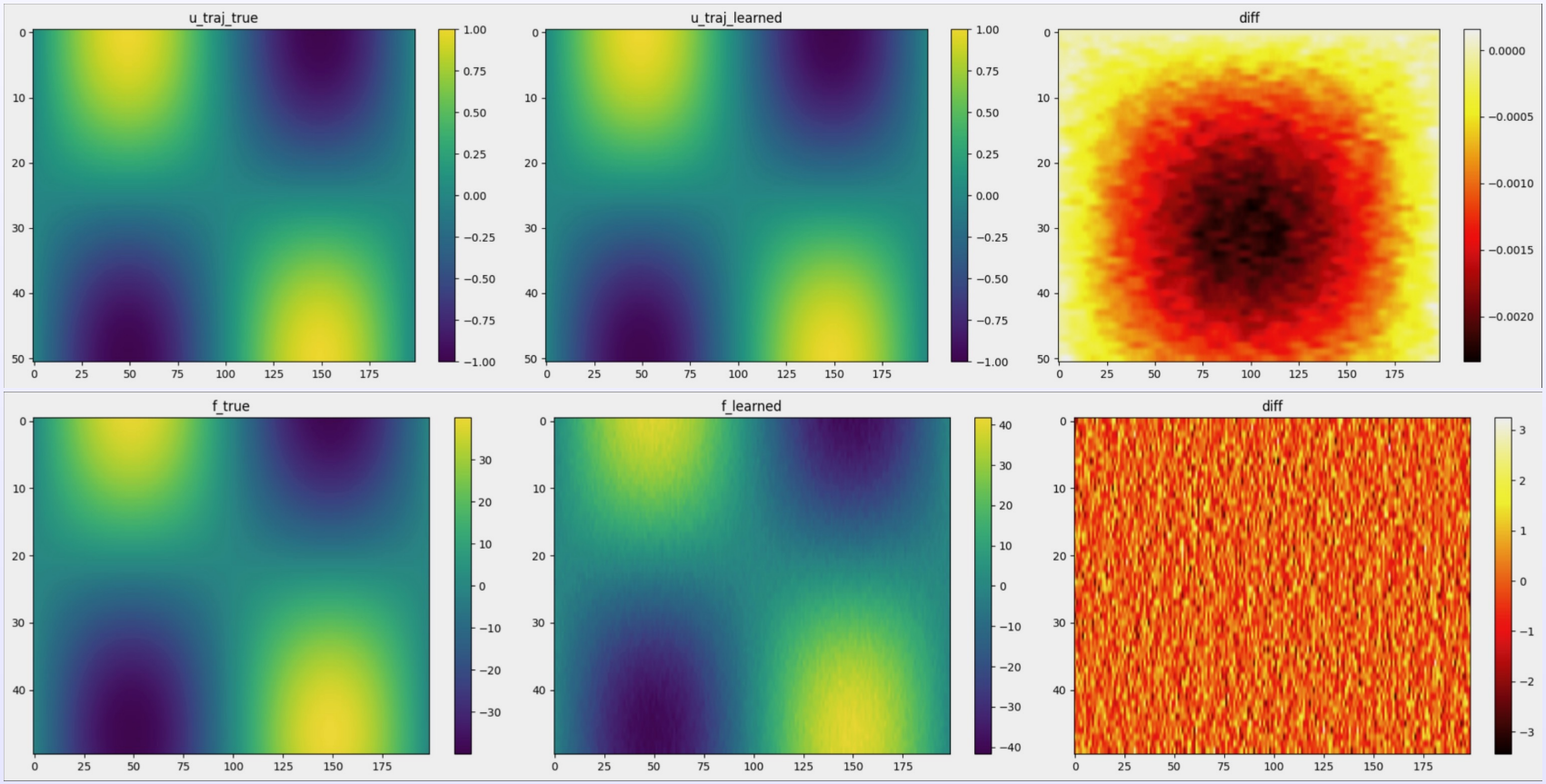
$$\text{s.t } \Delta_x u - \partial_t u = f_\theta$$

Results

Vanilla, $u = \sin(\omega\pi x)\cos(\omega\pi t)$

$X = [0, 1], \quad T = [0, 0.5], \quad n_x = 200, \quad n_t = 50$

After 8000 Epochs

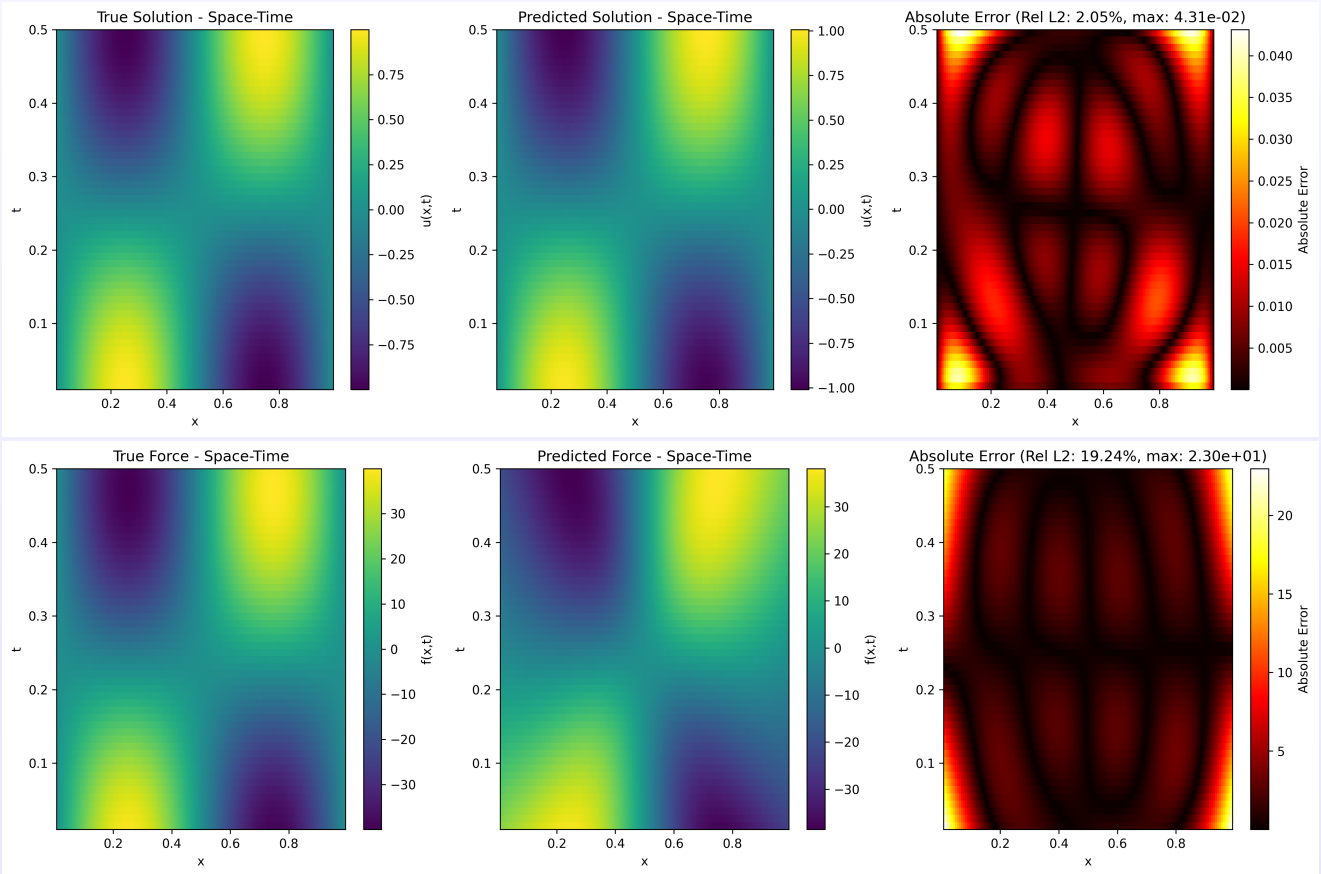


Results

Neural Network Parameterization, $u = \sin(\omega\pi x)\cos(\omega\pi t)$

$X = [0, 1], \quad T = [0, 0.5], \quad n_x = 200, \quad n_t = 50$

After 6000 Epochs

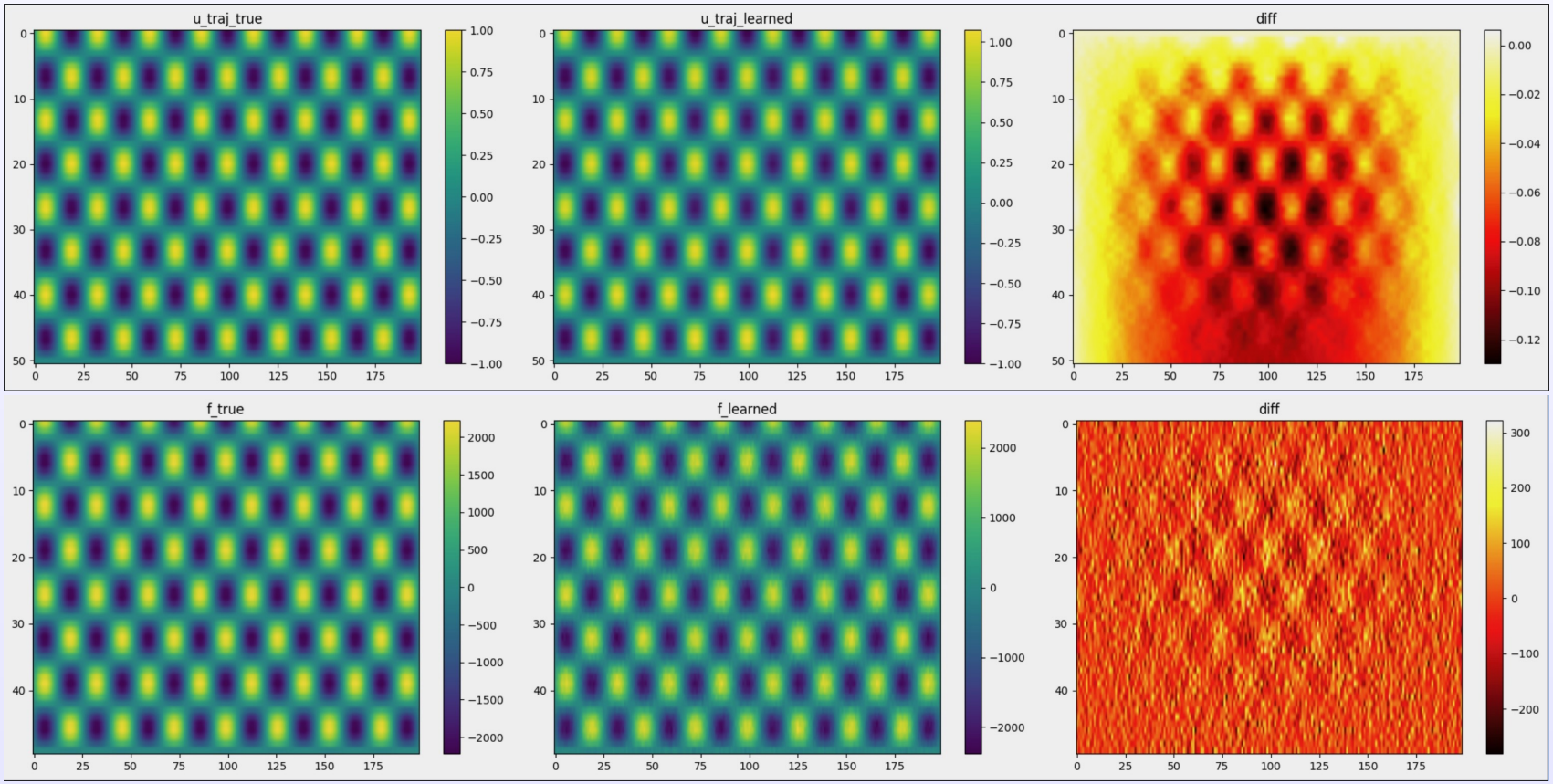


Results

Vanilla, $u = \sin(15\omega\pi x)\cos(15\omega\pi t)$

$X = [0, 1], \quad T = [0, 0.5], \quad n_x = 200, \quad n_t = 50$

After 80000 Epochs

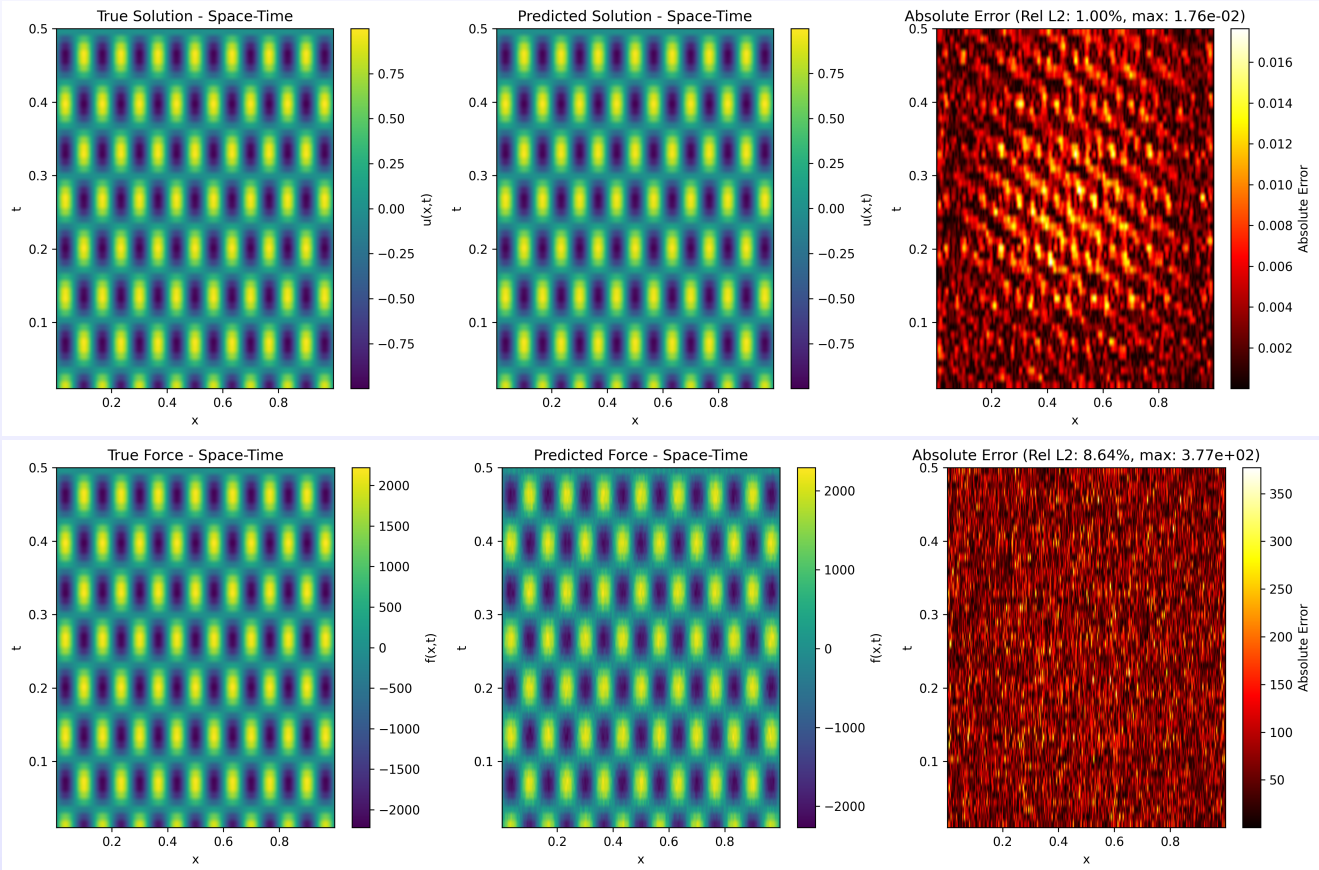


Results

Vanilla, $u = \sin(15\omega\pi x)\cos(15\omega\pi t)$

$X = [0, 1], \quad T = [0, 0.5], \quad n_x = 200, \quad n_t = 50$

After 3000 Epochs



Next Steps

What if there are certain constraints like $f \geq 0$?

- Adjoint method: Labor intensive
- Vanilla Auto-differentiation method: Some regularization? Some parameterization?
- Neural Net + Auto-differentiation method: ReLU layers

THANK YOU!