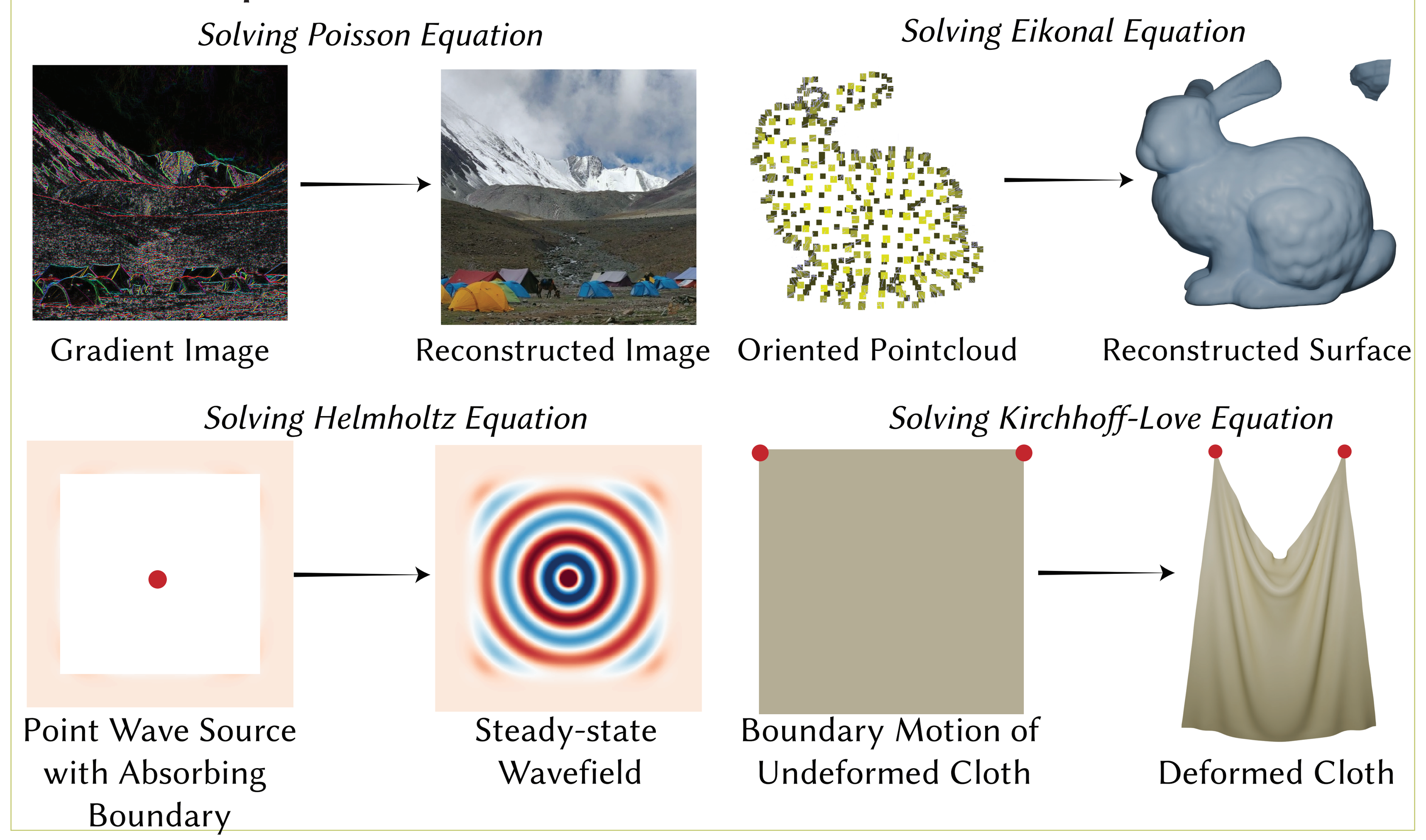
∂∞-Grid: A Neural Differential Equation Solver with Differentiable Feature Grids

MAX PLANCK INSTITUT FOR INFORMATICS [1]
[2]
Navami Kairanda[1], Shanthika Naik[2], Marc Habermann[1], Avinash Sharma[2], Christian Theobalt[1], Vladislav Golyanik[1]

ICLR
4dqv.mpi-inf.mpg.de/DInf-Grid

Overview

∂∞-Grid is a feature-grid that models signals and higher-order derivatives. Leveraging spatial structure, it solves differential equations faster than coordinate-based MLPs.



Problem Setting

Physical fields and how to represent them?

$$\mathbf{u}(\mathbf{x}) : \Omega \to \mathbb{R}^m \qquad \Omega \subset \mathbb{R}^d$$

d=1,...,4, spatio-temporal

How to recover unknown fields by solving differential equations?

$$\mathcal{F}(\mathbf{x}, \mathbf{u}, \nabla_{\mathbf{x}}\mathbf{u}, \nabla^2_{\mathbf{x}}\mathbf{u}, \ldots; g(\mathbf{x})) = 0$$

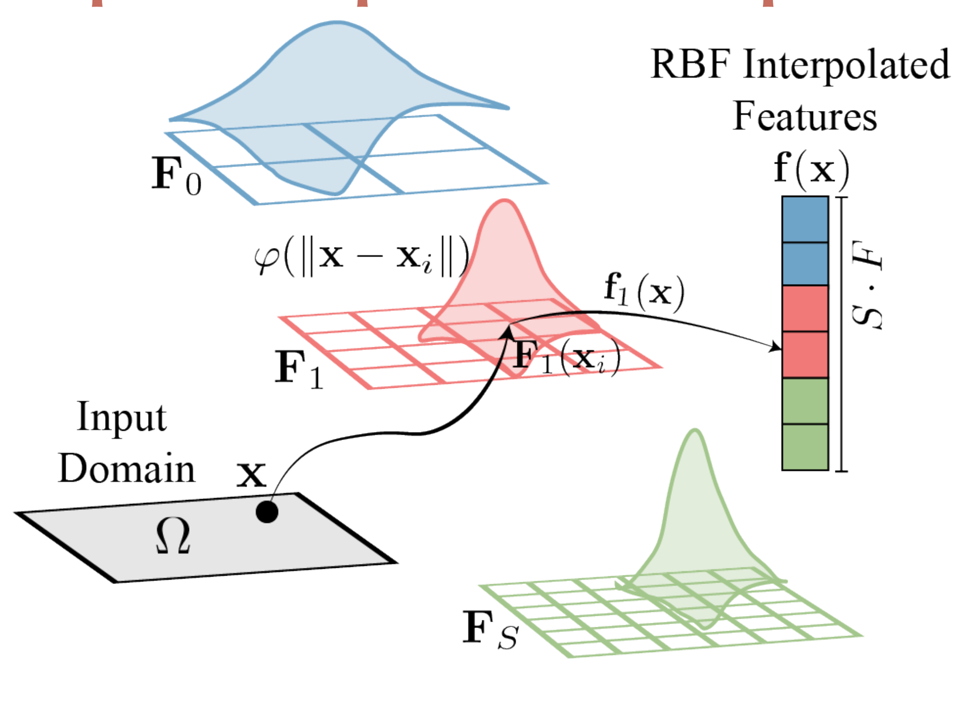
Differential operator — Known functions (eg, gravity)

- MLP-based solvers? Do not leverage spatial structure of fields
- Grid-based neural fields? Non-differentiable
- Ours ∂∞-Grid? **Continuous, differentiable, grid-based, neural DE solver**

Method

We solve differential equations $\mathcal{F}$ by representing the field $\mathbf{u}$ with multi-scale differentiable grids and minimizing the residual as the loss.

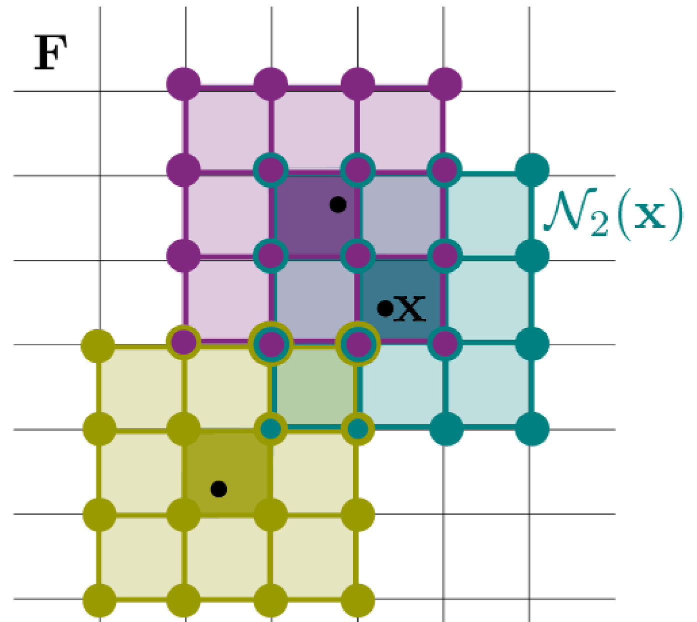
Step 1: Sample and Interpolate



Multi-resolution Feature Grid

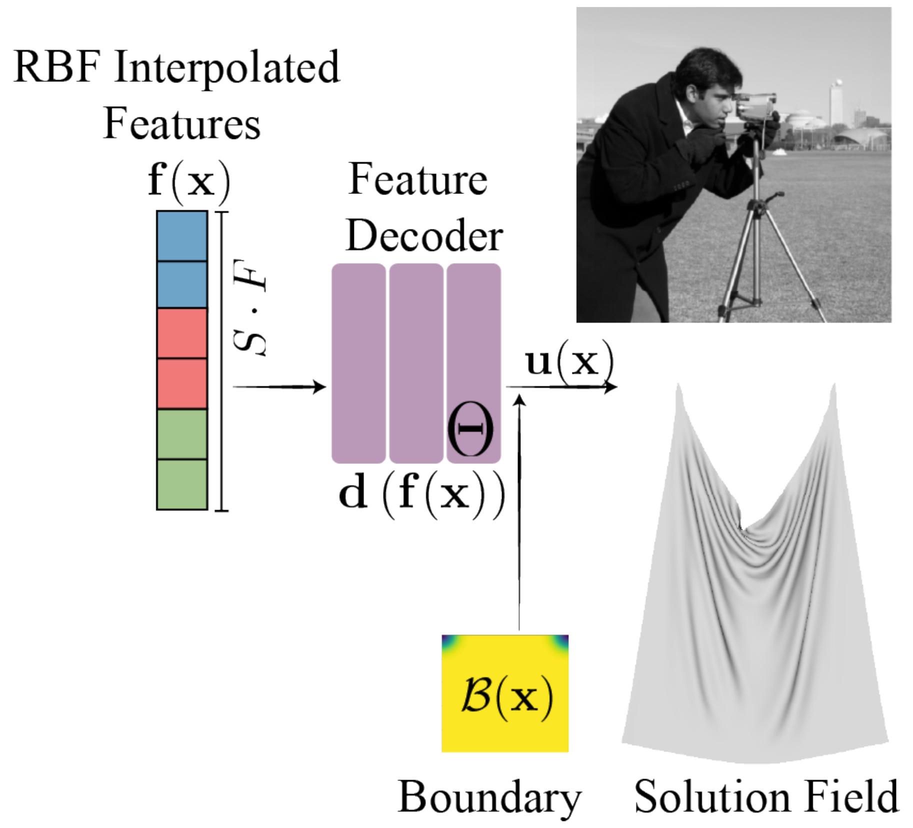
- Sample query points
- Extract smooth features using RBF interpolation across multi-scale grids
- Concatenate features.

Step 2: Neighbourhood Precomputation



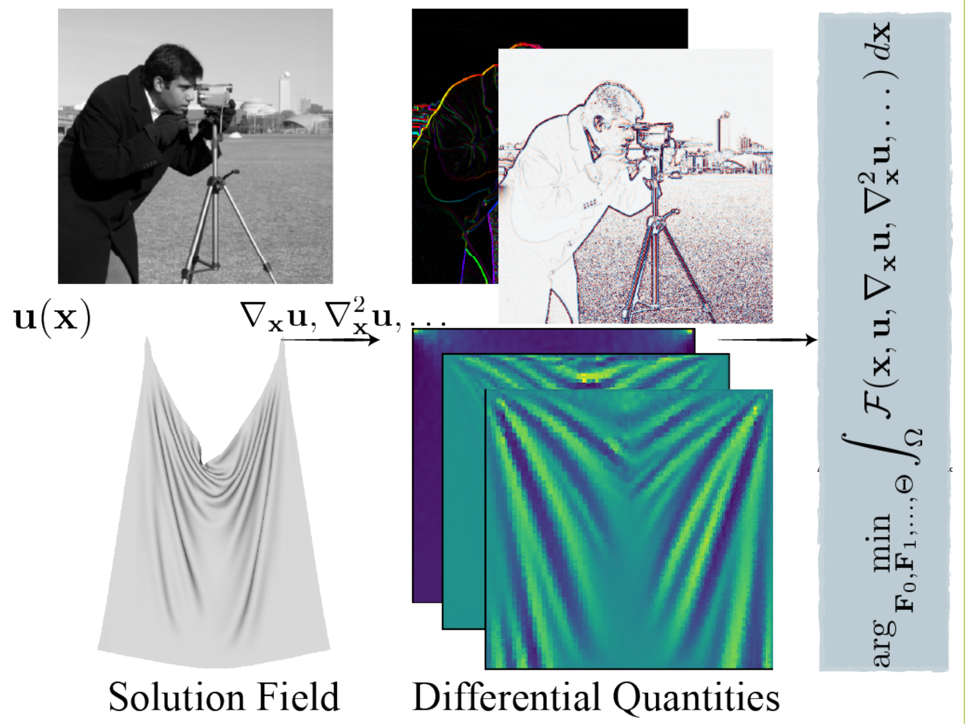
- Restrict RBF support to compact neighborhoods via stratified sampling
- Reduces cost while preserving overlap for higher-order derivatives.

Step 3: Produce the Signal



- Map features to the target signal using a decoder
- Enforce boundary/initial conditions through boundary-aware modulation

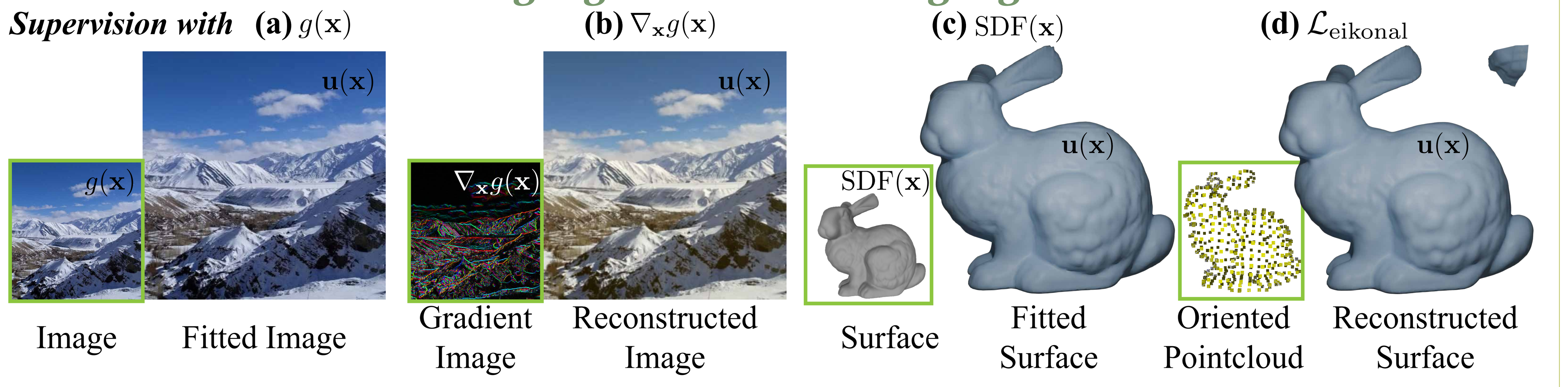
Step 4: Optimise for the Solution



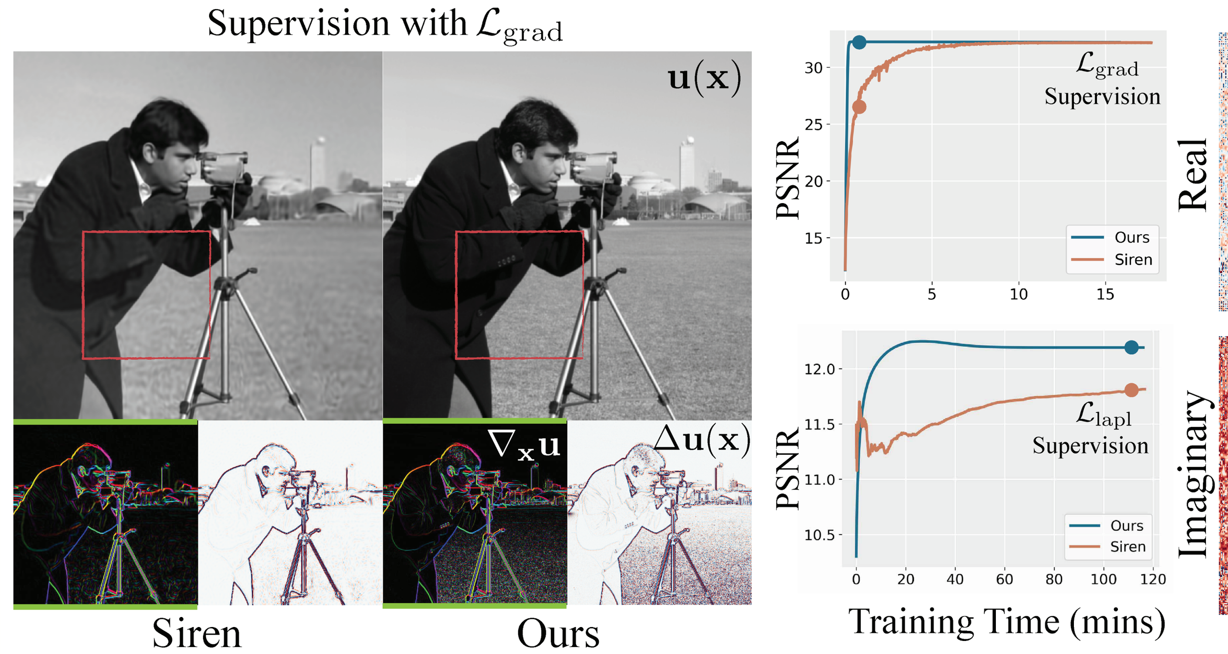
- Use automatic differentiation to compute derivatives
- Optimize model parameters by minimizing the PDE residual

Results and Comparisons

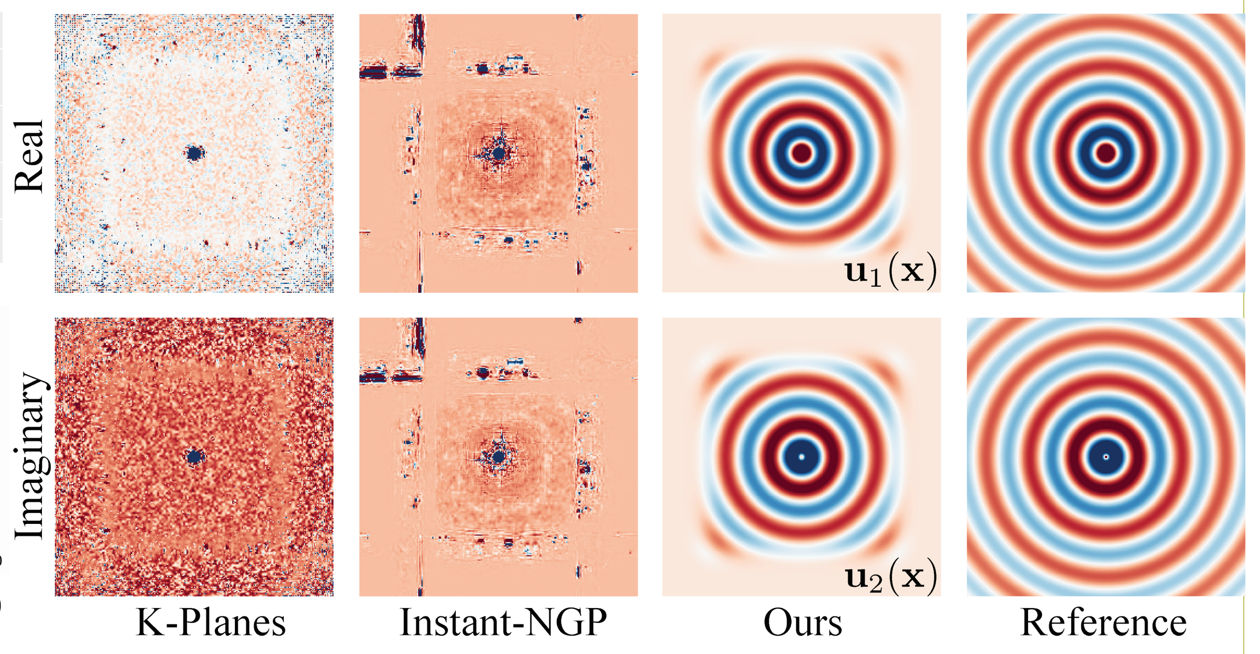
Fitting Signals and Learning Signals



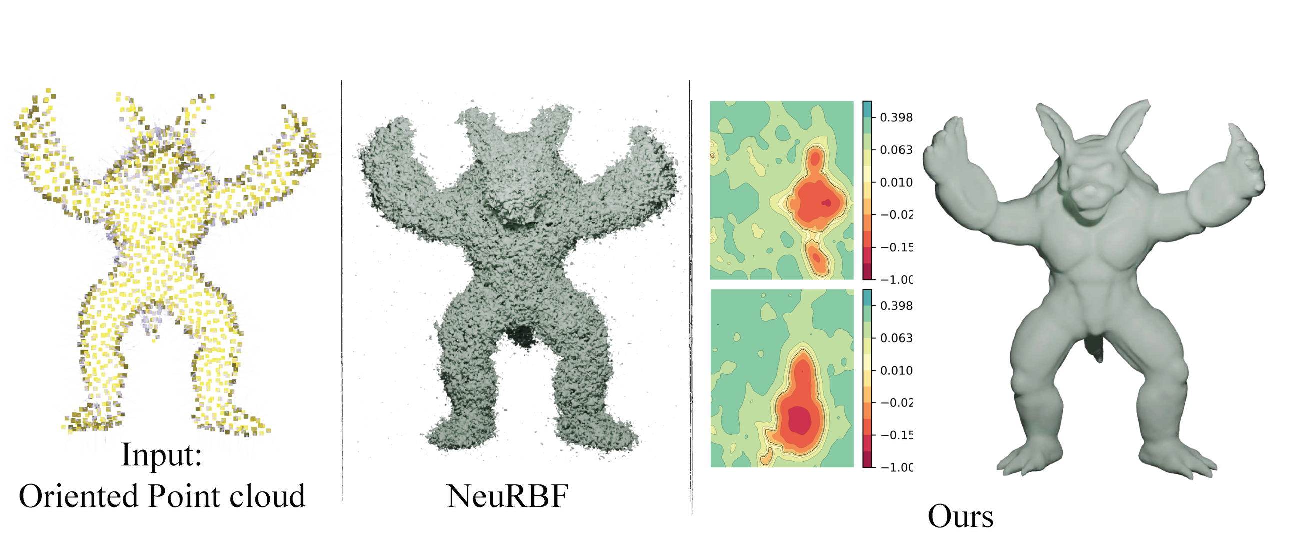
Solving Poisson Equation



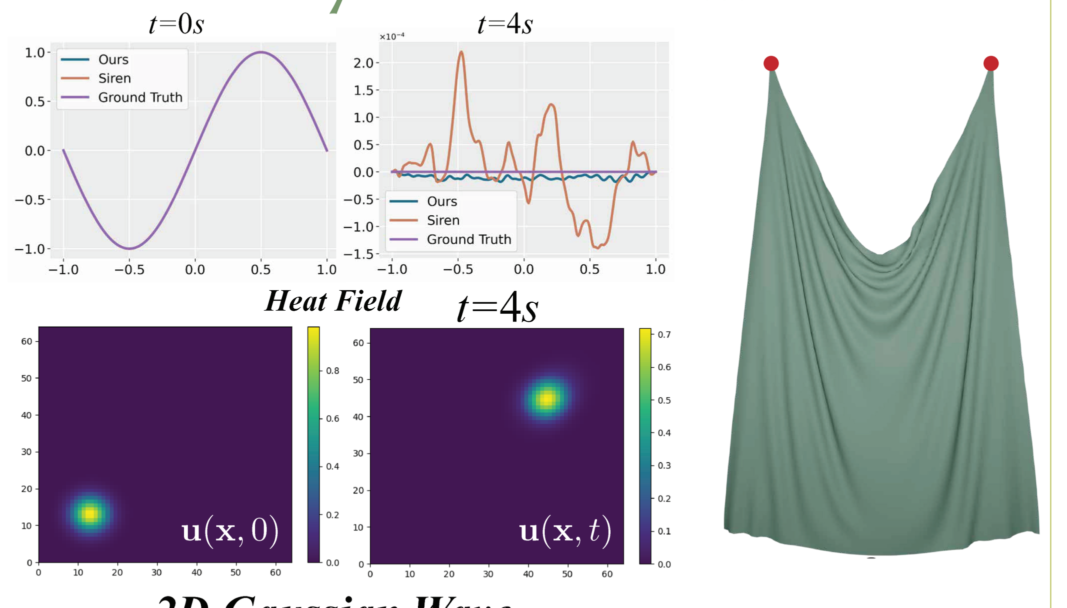
Solving Helmholtz Equation



Solving Eikonal Equation



Many more PDEs...



2D Gaussian Wave

Conclusion

- Fast and accurate solver for differential equations
- Poor scalability to higher-dimensions

References

Sitzman *et al.*, Siren, NeurIPS 2020
Mueller *et al.*, Instant-NGP, SIGGRAPH 2022
Chen *et al.*, NeuRBF, ICCV 2023
Kairanda *et al.*, NeuralClothSim, NeurIPS 2024