STANFORD COMPUTATIONAL IMAGING LAB
WWW.COMPUTATIONALIMAGING.ORG

Learning to Solve PDE-constrained Inverse Problems with Graph Networks

Qingqing Zhao, David B. Lindell, Gordon Wetzstein
Stanford University

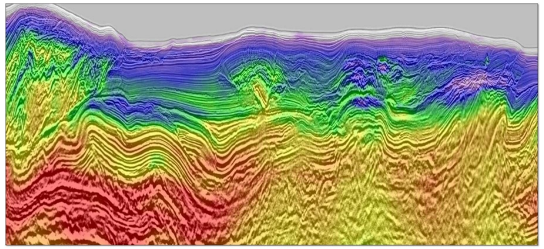
ICML 2022

PDE-constrained Inverse Problem

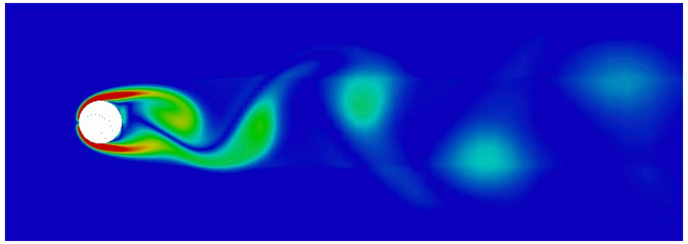
- **Goal**: Inferring knowledge from observation data by leveraging simulation and mathematical models (PDEs)

Sparse measurements Underlying PDEs (partial differential equations)	⇒	Internal structures, physics parameters Initial States, etc.

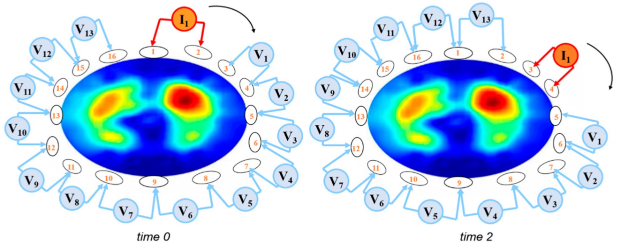
Seismic Imaging



Navier-Stokes



Electrical impedance tomography



PDE-constrained Inverse Problem

- **Goal**: Inferring knowledge from observation data by leveraging simulation and mathematical models (PDEs)

ill-posed problem

Sparse measurements

Underlying PDEs
(partial differential equations)

Internal structures,
physics parameters
Initial States, etc.

We need
- Fast Simulator
- Priors / regularization

Estimated Physics Parameter

Simulator

Predicted Dynamics

Sparse Measurements

$\sum_t \mathcal{L}$ (Observations, Predictions)

PDE-constrained Inverse Problem

- **Goal**: Inferring knowledge from observation data by leveraging simulation and mathematical models (PDEs)

ill-posed problem

Sparse measurements

Underlying PDEs
(partial differential equations)

Internal structures, physics parameters Initial States, etc.

We need
- Fast Simulator
- Priors / regularization

Estimated Physics Parameter

Learned Generative Prior

Simulator

GNN solver

Predicted Dynamics

Sparse Measurements

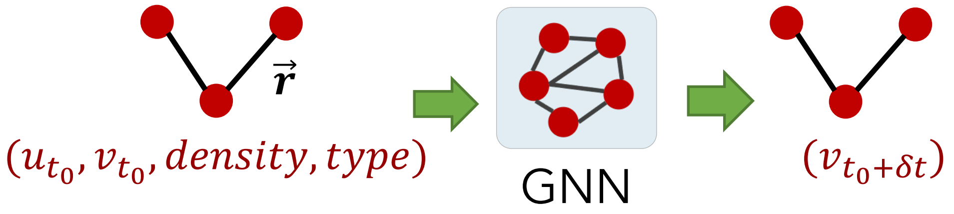
$\sum_t \mathcal{L}\,(\text{Observations}, \text{Predictions})$

Fast Solver

Priors

⟹ GNN-based Simulator

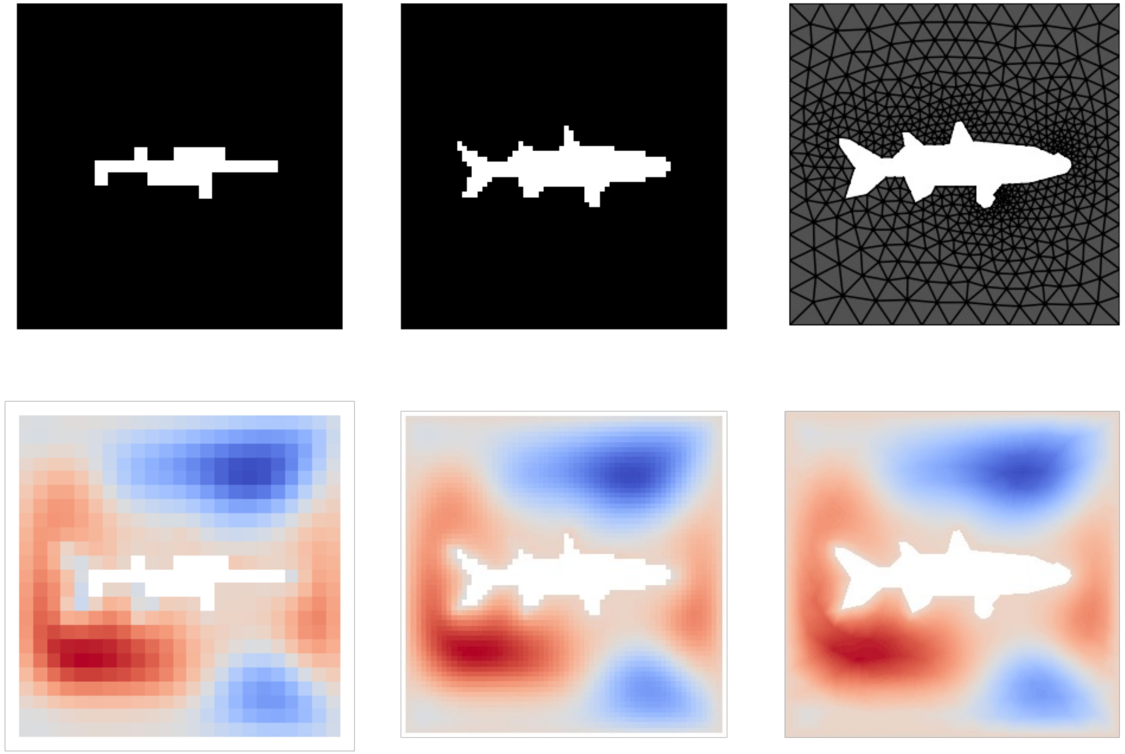
- Fast
  - Learn larger timesteps
  - Learn on coarse meshes



⟹ Generative Prior

LEARNING MESH-BASED SIMULATION WITH GRAPH NETWORKS, Tobias Pfaff et. al, 2021

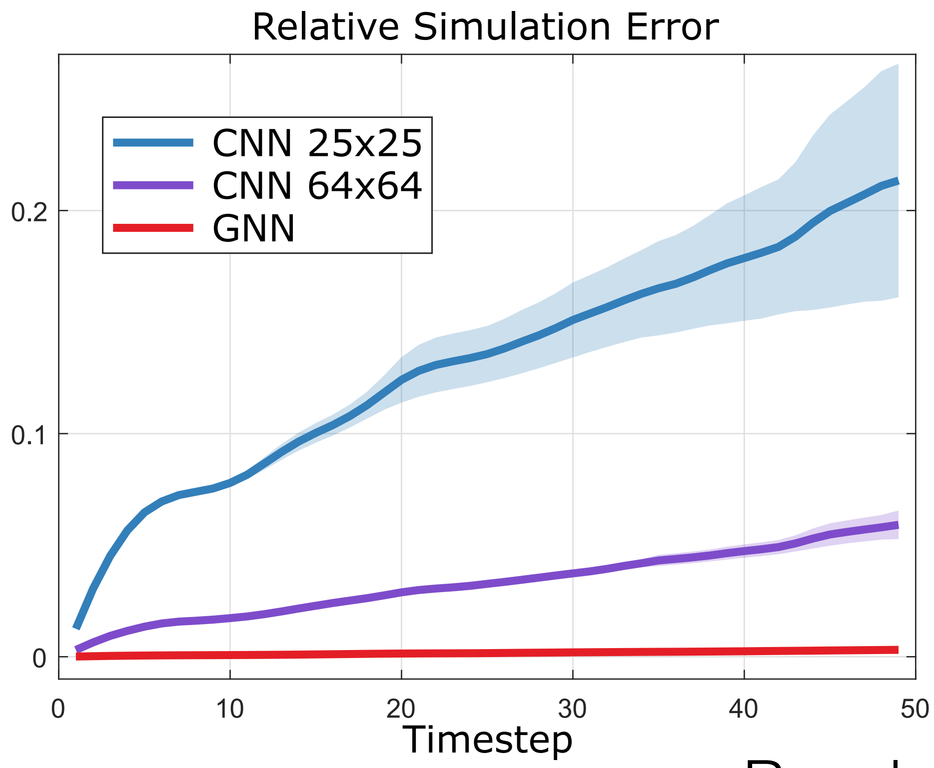
GNN based simulator VS CNN based simulator

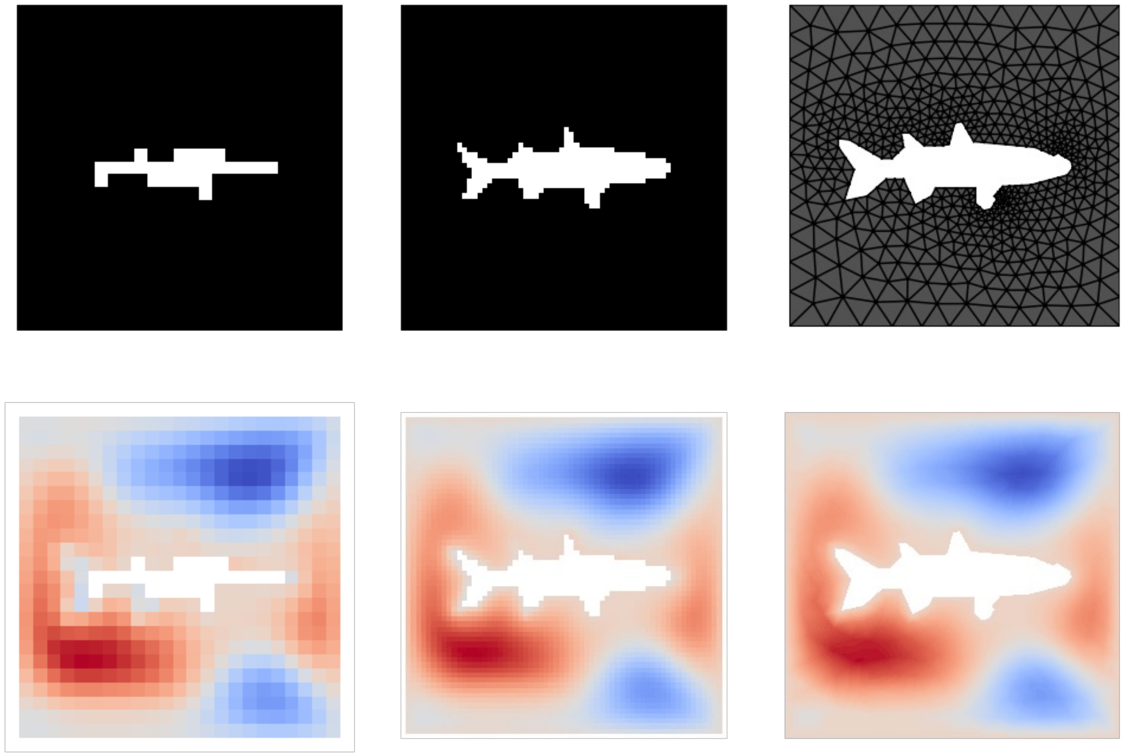


Resolution: CNN 25x25   CNN 64x64   GNN (611)

GNN operates on irregular meshes – which is a more efficient representation

GNN based simulator VS CNN based simulator





Resolution: CNN 25x25 CNN 64x64 GNN (611)

GNN operates on irregular meshes – which is a more efficient representation

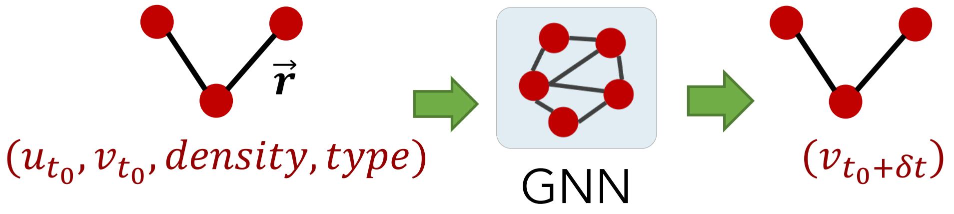
GNN has better long-term accuracy

Fast Solver

Priors

$\Longrightarrow$ GNN-based Simulator

- Faster
- More accurate compared to other learned simulators
- Better scaling due to adaptive resolution

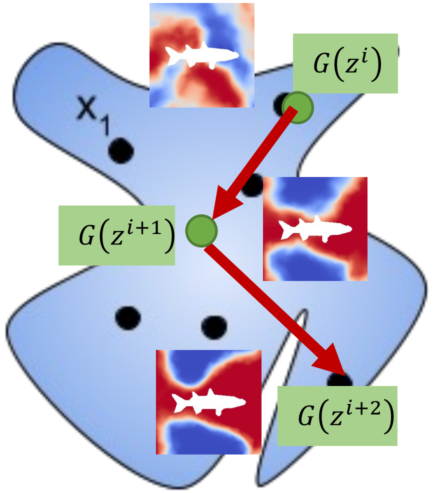


$\Longrightarrow$ Generative Prior

Priors

⟹ Generative Prior

- we constrained the solution space to the manifold learned by G, and avoid bad local optima lie far outside the dataset distribution



Dataset distribution

Priors

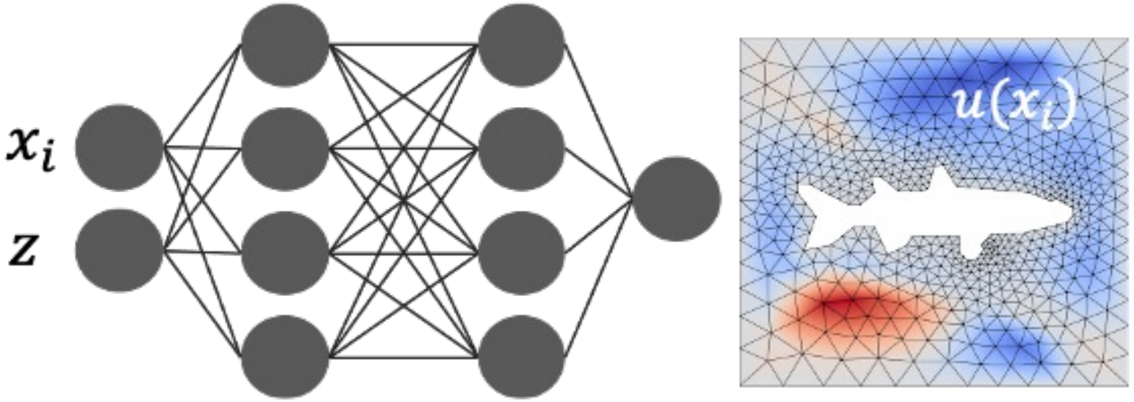
⟹ Generative Prior

- we constrained the solution space to the manifold learned by G, and avoid bad local optima lie far outside the dataset distribution

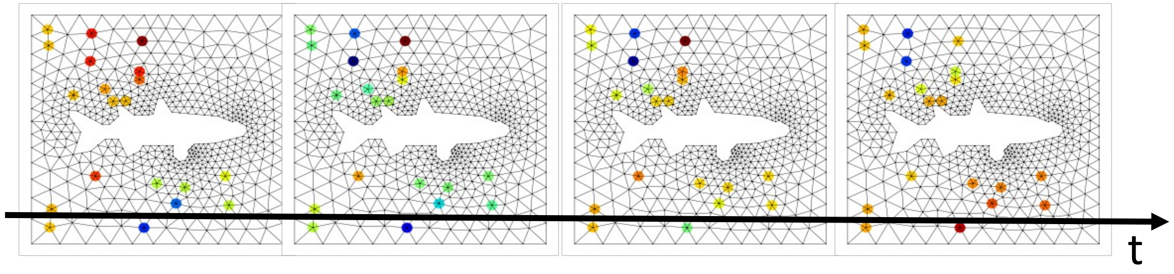
Dataset distribution

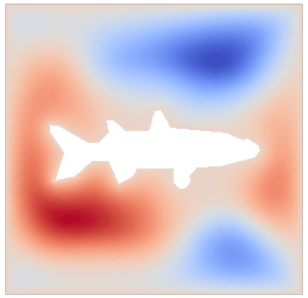
Coordinate Network

- Map points x_i and latent z to the field value at that point.
- Independent of meshes



Inverse problem setup





u_{init}

Sparse measurements

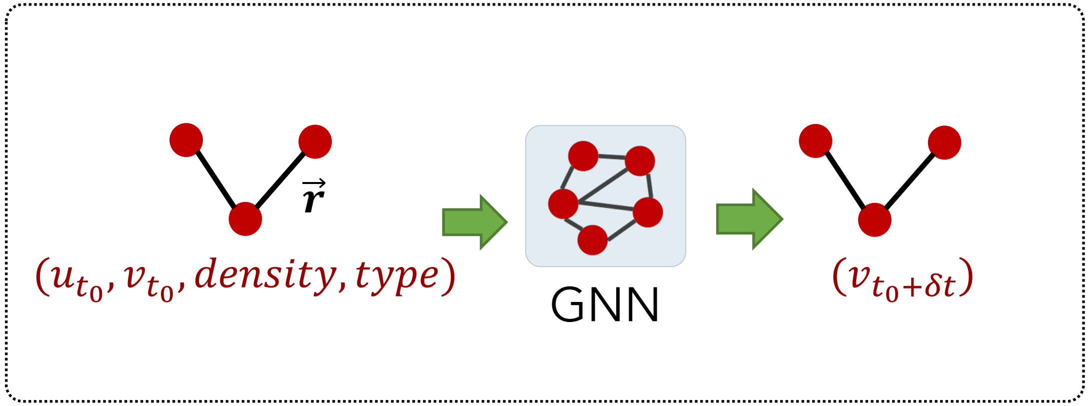
Underlying PDEs
(partial differential equations)

Internal structures,
physics parameters
Initial States, etc.

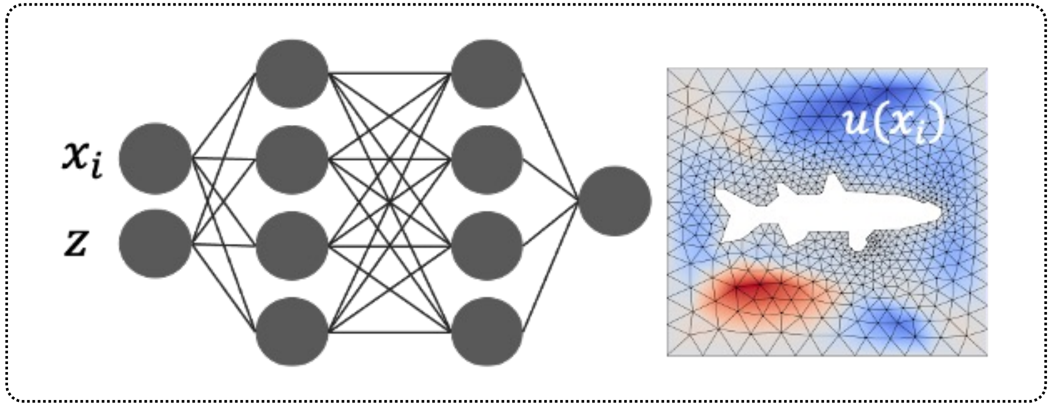
$$\frac{\partial^2 u}{\partial t^2} - c(x)^2 \frac{\partial^2 u}{\partial x^2} = 0, \quad u_0 = \boxed{u_{\text{init}},} \quad u_0' = \frac{\partial u}{\partial t}|_{t=0} = 0.$$

Pipeline

- Pretrain GNN-based simulator and Learned Prior

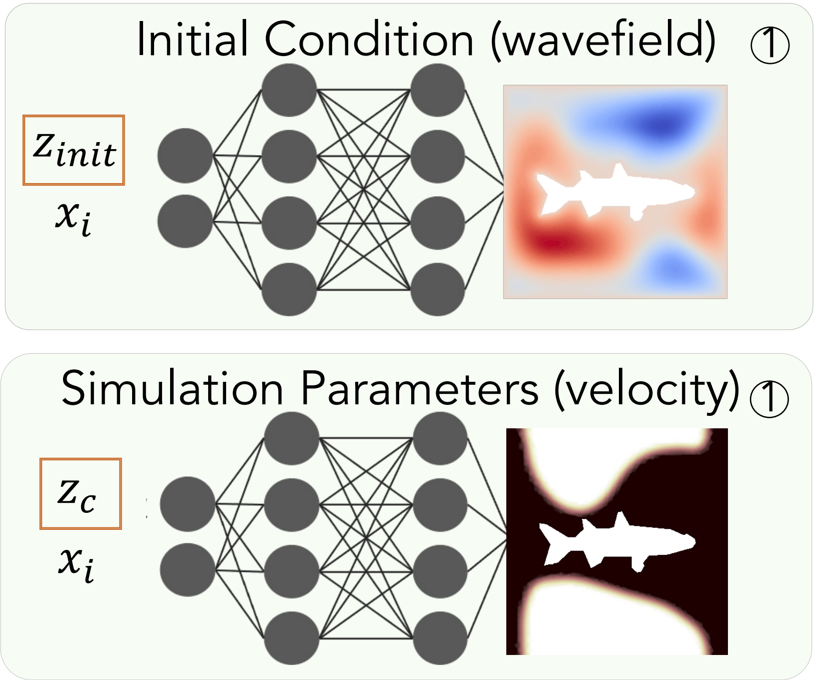


GNN-based simulator



Learned Prior

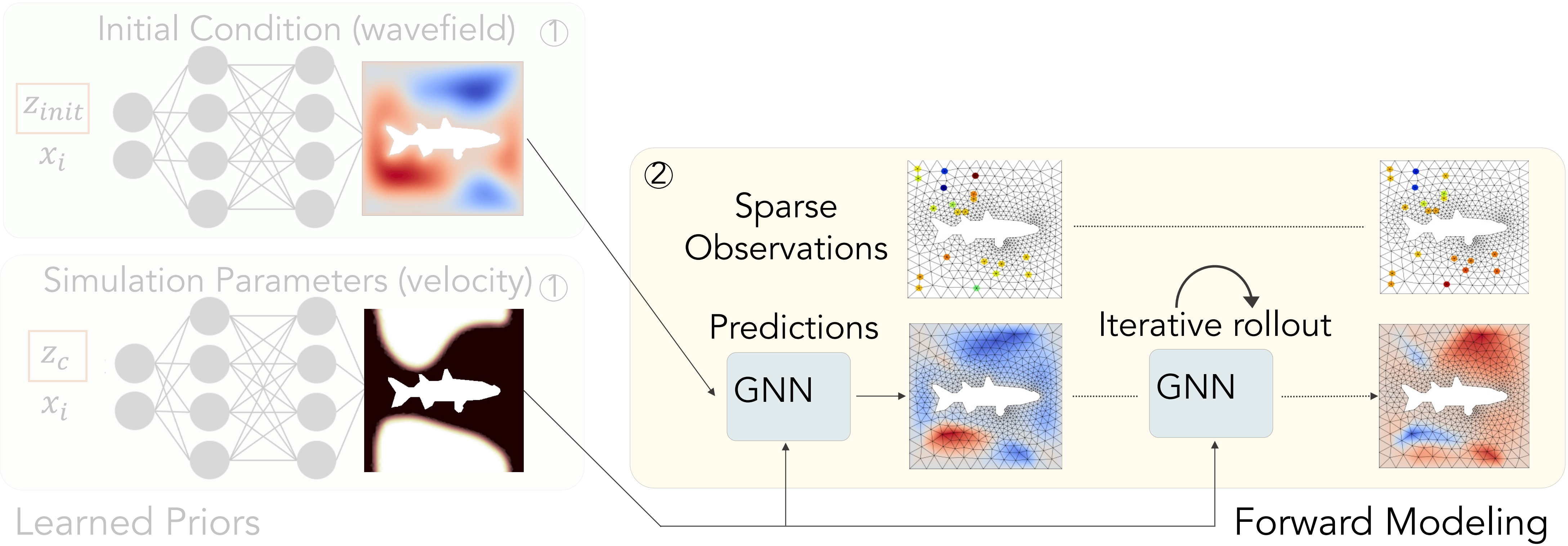
Pipeline



Learned Priors

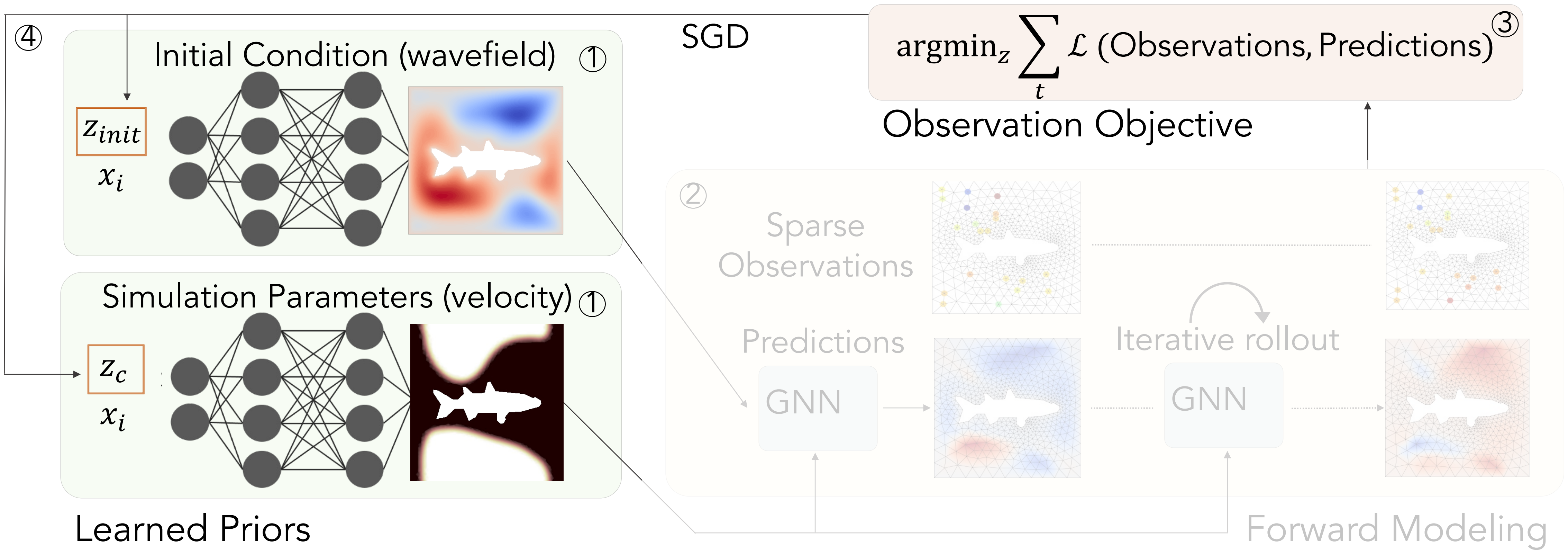
Given current estimated z, we use learned prior to decode it to physics parameters

Pipeline



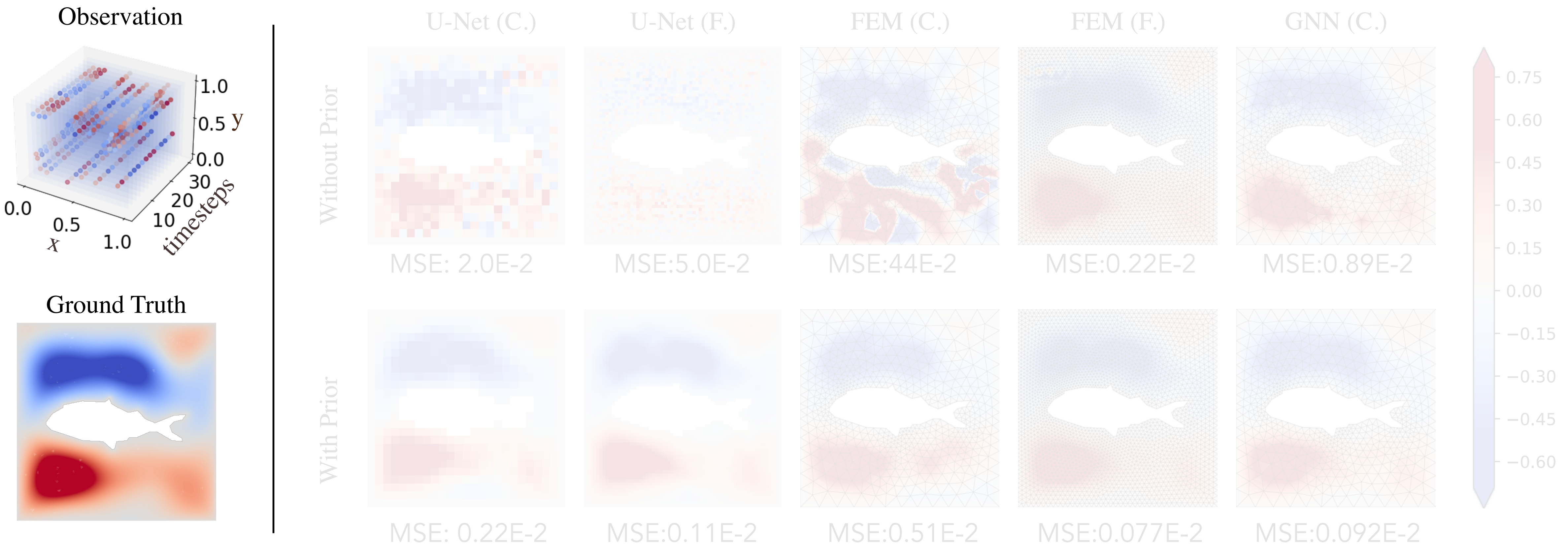
Decoded parameters pass to the GNN for forward modeling to obtain predicted dynami

Pipeline



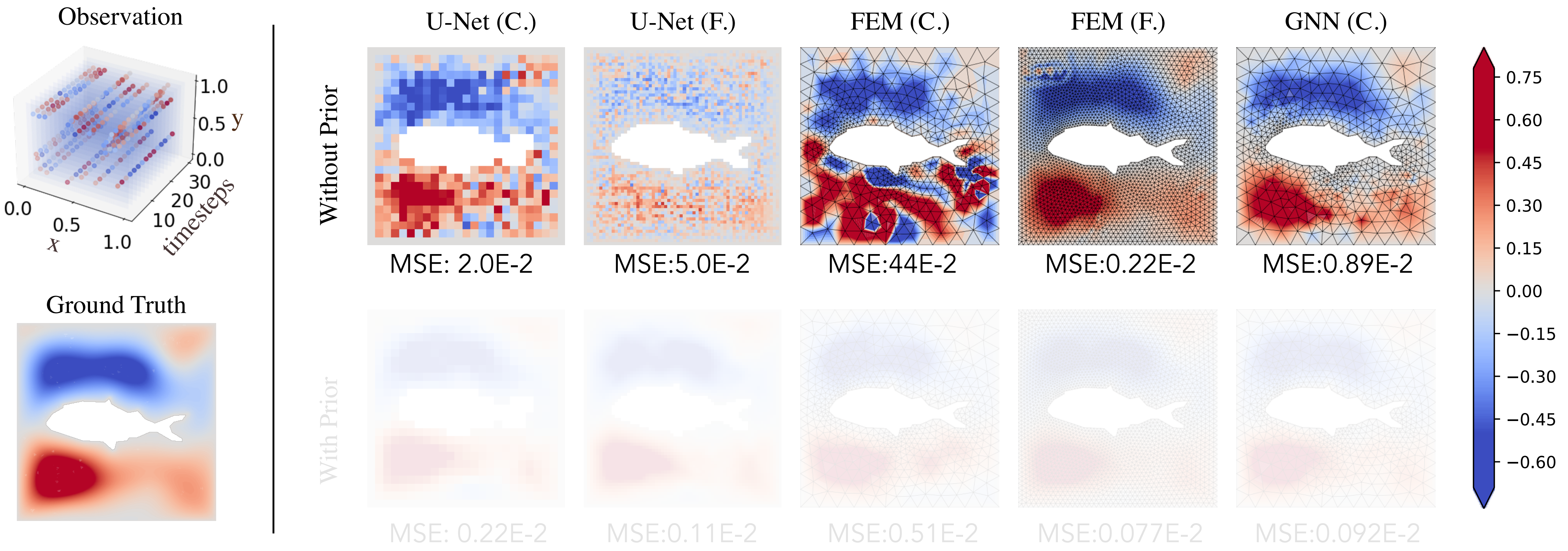
Latent code z_{init} and z_c is optimized to minimize the observation objective

Results

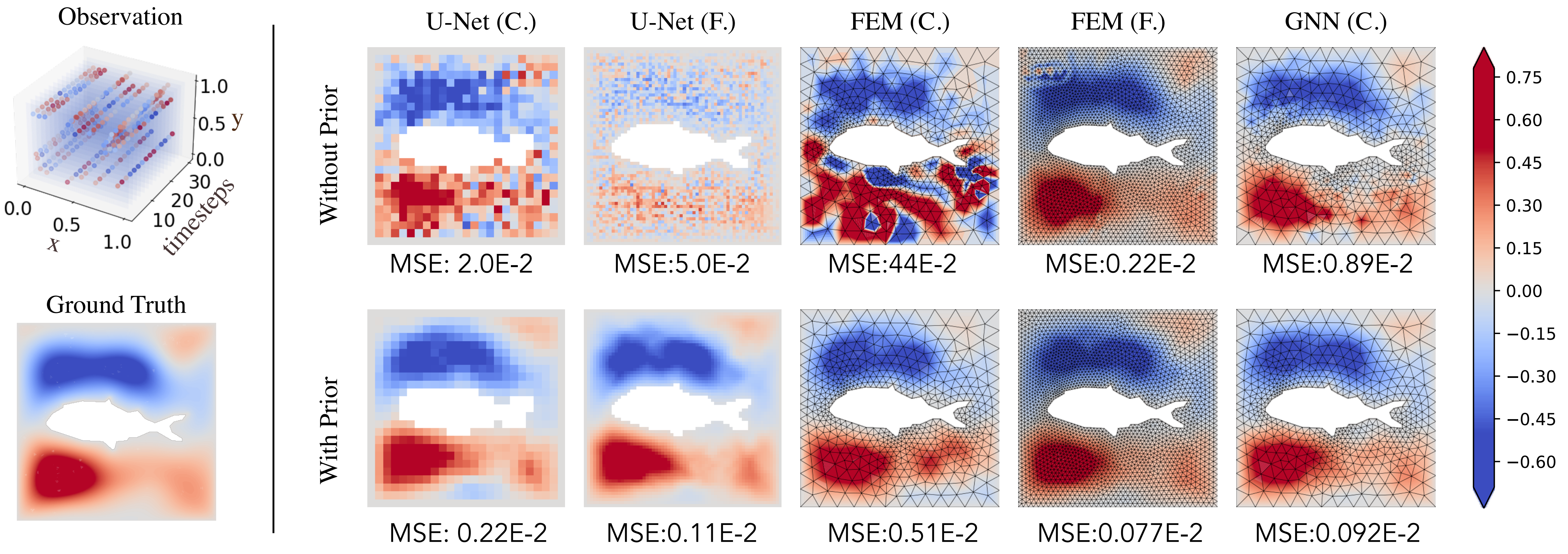


Inverse Problem:

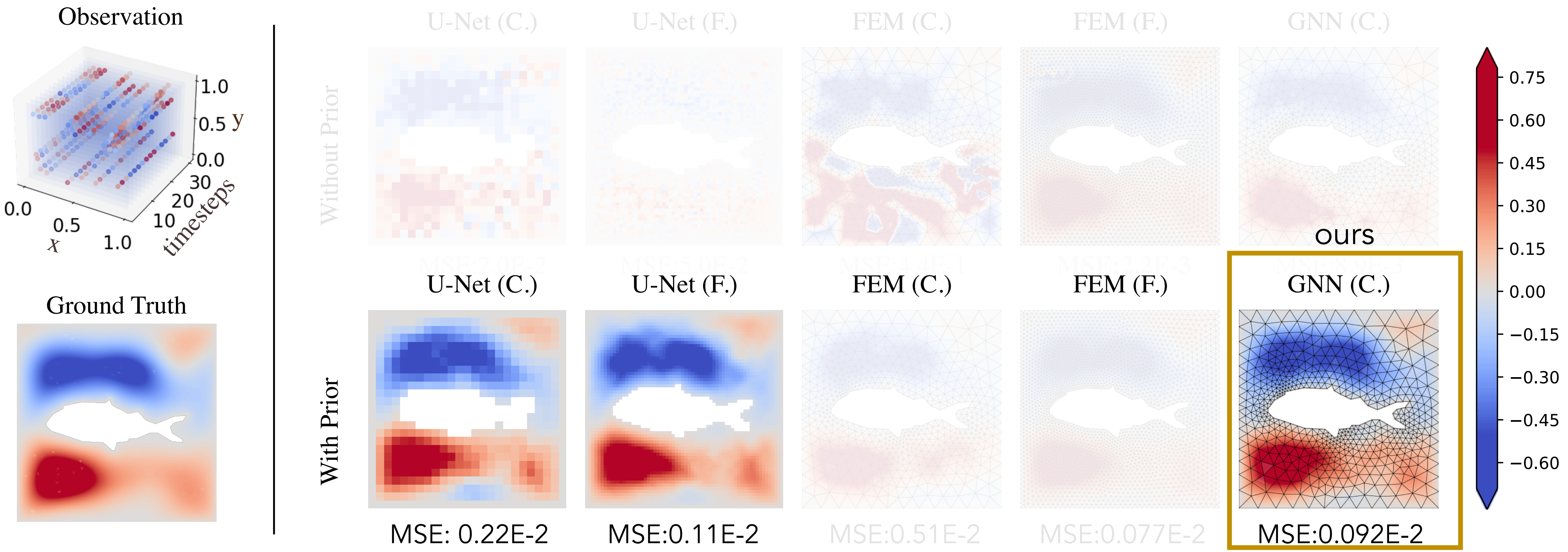
Given sparse observations $u_t(x)$, where $x \in$ {sensor locations} time$\in \{0, 2\delta t, \cdots\ 30\ \delta t\}$, we want to infer the field $u_{t=0}$



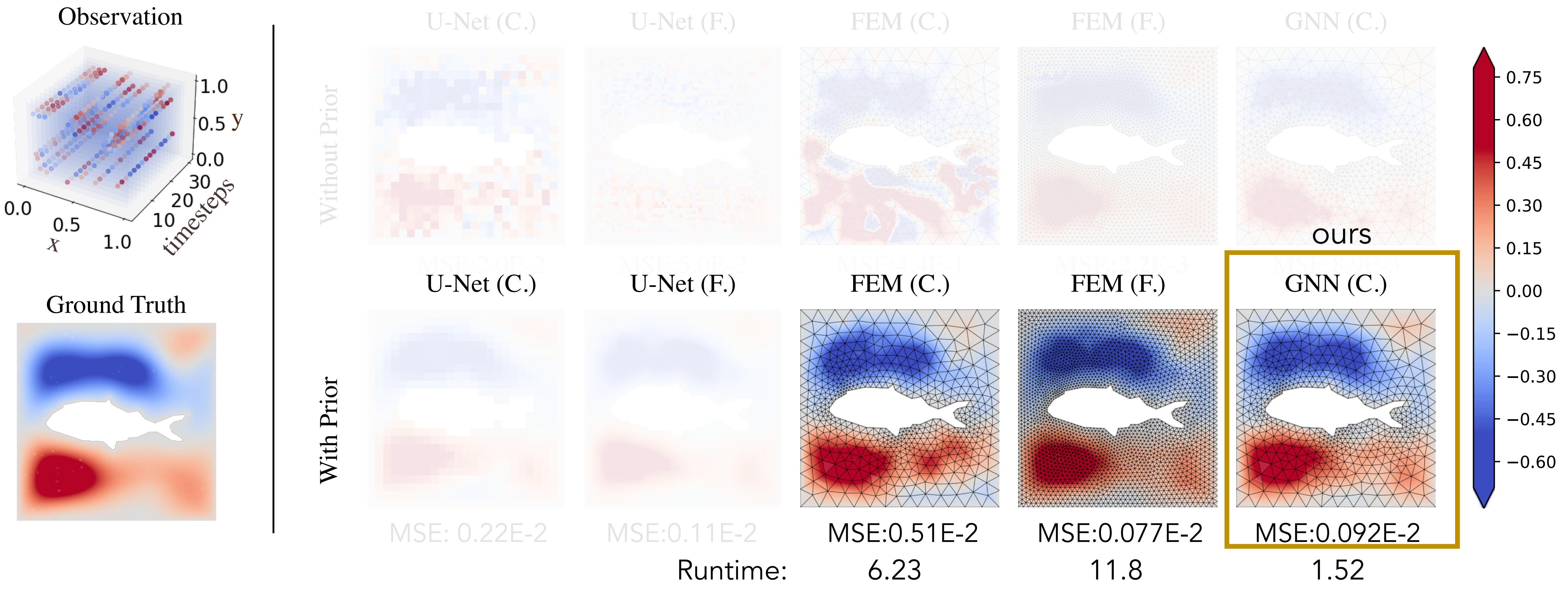
- With prior significantly out-perform without-prior cases



- With prior significantly out-perform without-prior cases

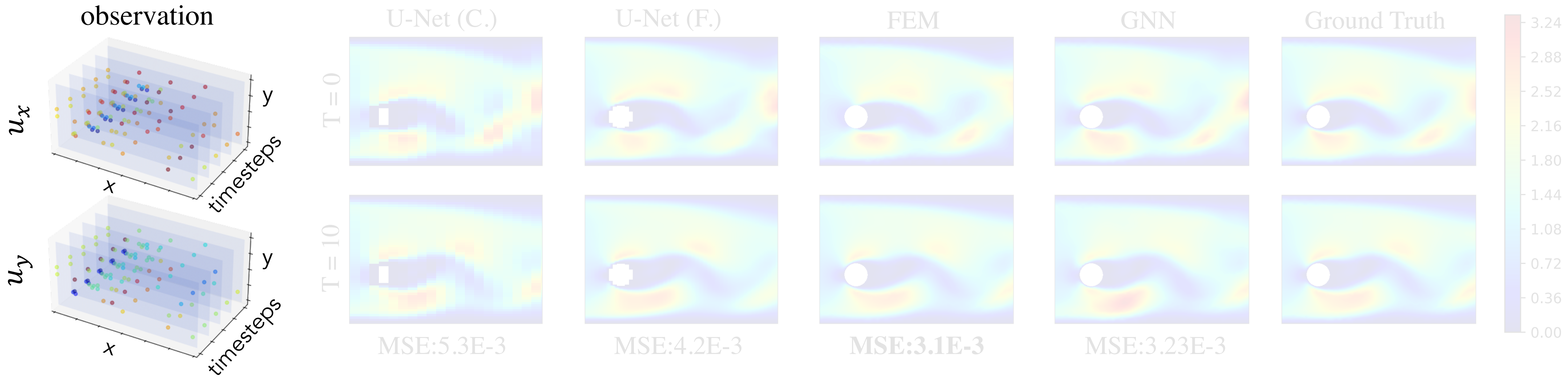


- With prior significantly out-perform without-prior cases
- GNN outperform other learning-based method (CNN)



- With prior significantly out-perform without-prior cases
- GNN outperform other learning-based method (CNN)
- GNN outperform Classical Solver at coarse resolution;
- GNN is slightly less accurate than Classical Solver at fine resolution but is ~8x faster

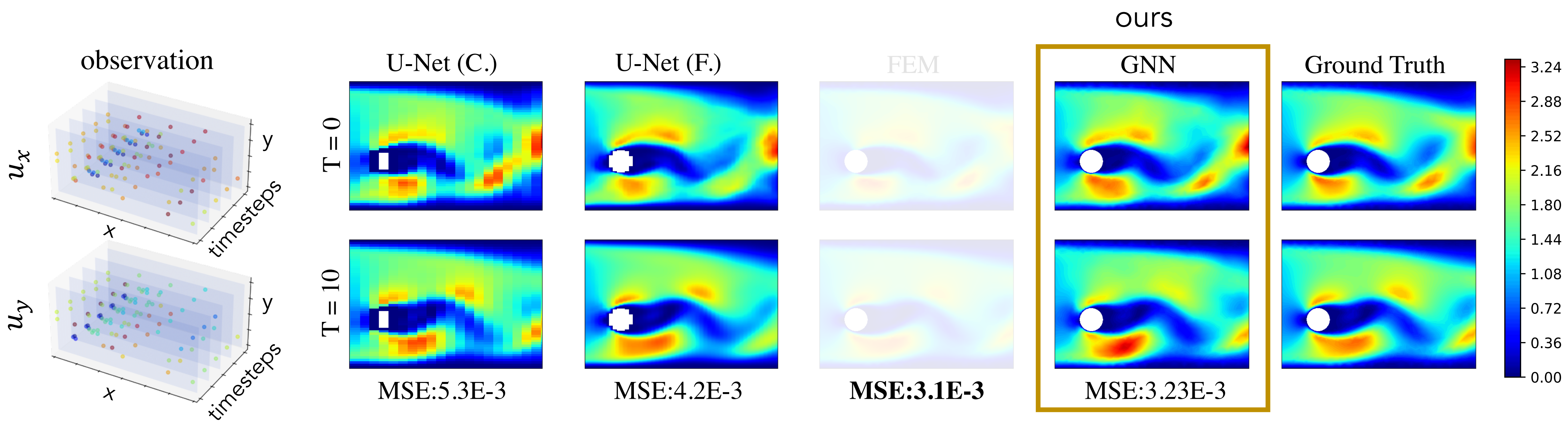
Results – Flow Assimilation



Inverse Problem:

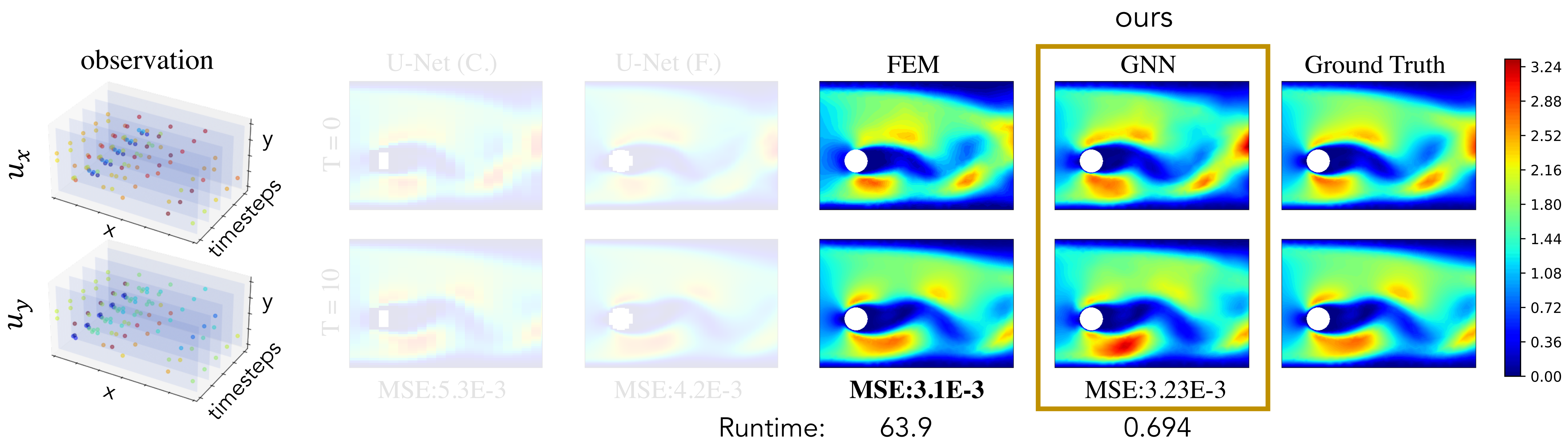
Given sparse observations $\boldsymbol{u}_t(\mathbf{x})$, where $\mathbf{x} \in$ {sensor locations} time$\in \{0, 2\delta t, \cdots\ 30\ \delta t\}$, we want to infer the field everywhere

Results – Flow Assimilation



- GNN outperform other learning-based method (CNN)

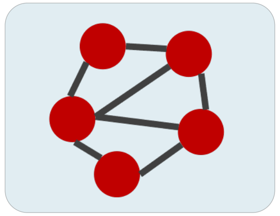
Results – Flow Assimilation

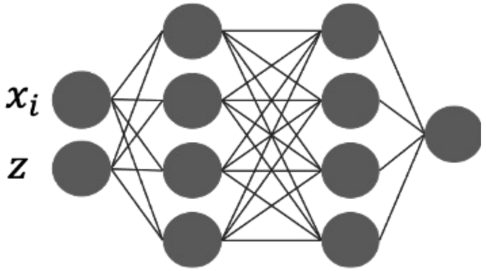


- GNN outperform other learning-based method (CNN)
- GNN is slightly less accurate than Classical Solver at fine resolution but is ~90x faster

Summary

- We develop a new physics-based graph neural network incorporating generative model for solving physics constrained inverse problem <u>faster</u> and <u>more accurate</u>



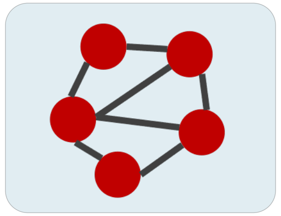


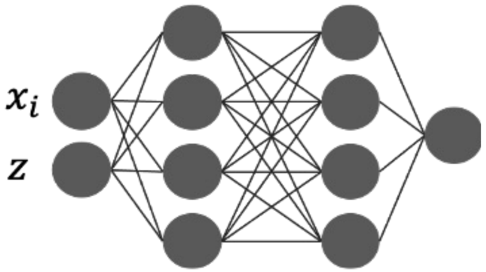
GNN based solver + Learned Prior

Summary

- We develop a new physics-based graph neural network incorporating generative model for solving physics constrained inverse problem <u>faster</u> and <u>more accurate</u>

- Limitation/Future research direction
    - memory requirement due to unrolling
        - gradient checkpoints
    - generalizing learned physics solvers across problem settings





GNN based solver + Learned Prior

- Checkout our paper and project page for more!
  https://cyanzhao42.github.io/LearnInverseProblem