# TAEN:
# A Model-Constrained Tikhonov Autoencoder Network for Forward and Inverse Problems

Hai Nguyen and Tan Bui-Thanh

Probabilistic and High Order Inference, Computation, Estimation, and Simulation (PHO-ICES)

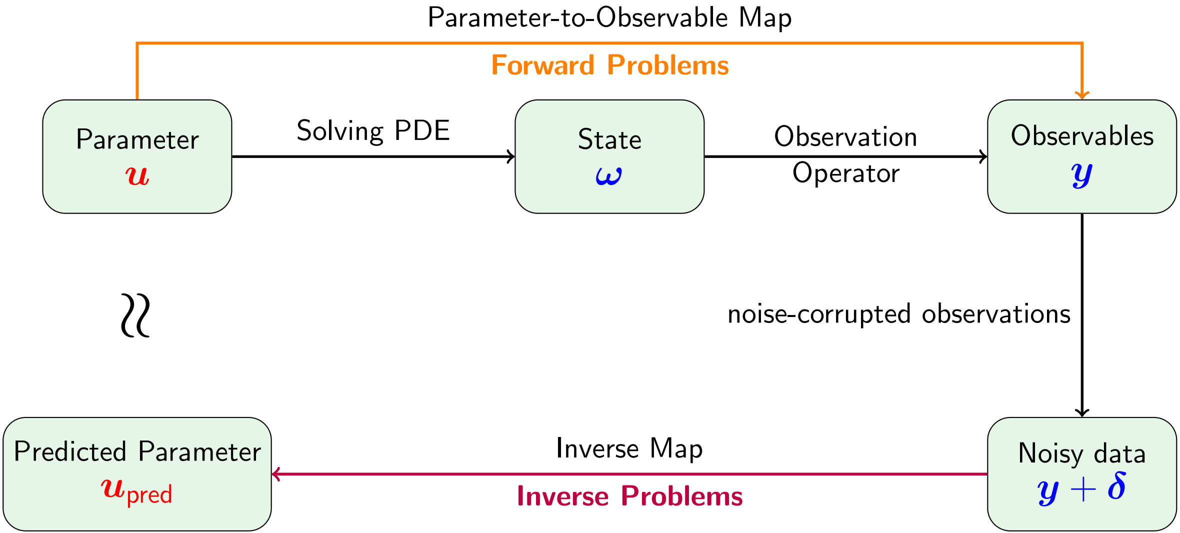
The University of Texas at Austin
Oden Institute for Computational Engineering and Sciences

The University of Texas at Austin
Aerospace Engineering and Engineering Mechanics
Cockrell School of Engineering

Our research are funded by DOE & NSF and implemented on TACC

06 March, SIAM CSE 2025

# Forward and inverse problems



# Motivation

# Overview: surrogate models for inverse and forward problems

# Problem settings

We consider

- The linear forward problem

$$\boldsymbol{y} = B \circ \underbrace{G\boldsymbol{u}}_{\omega} + \delta,$$

  where $B$ is observational operator, $G$ is linear forward map, and $\delta$ is white noise.

- The equivalent linear inverse problem

$$\boldsymbol{u} = (B \circ G)^{\dagger}\, \boldsymbol{y} = \mathrm{G}^{B\dagger} \boldsymbol{y}.$$

We want to learn linear autoencoder surrogate models

- Encoder $\Psi_{\mathrm{e}}(\boldsymbol{y}) = W_e \boldsymbol{y} + \mathbf{b}_e$ for inverse map $\mathrm{G}^{B\dagger}$.
- Decoder $\Psi_{\mathrm{d}}(\boldsymbol{u}) = W_d \boldsymbol{u} + \mathbf{b}_d$ for forward map $G$ or PtO map $G^B$.

# Autoencoder (AE) approaches

**We discuss**

1. Naive Autoencoder (`nAE`)
2. Model-constrained Autoencoder (`mcAE`)
3. Tikhonov Autoencoder Network Approach (`TAEN`)

# Naive Autoencoder (nAE) - error estimation

Given a dataset of pairs $\{U, Y\}$, we can learn forward and inverse surrogate models by

first optimizing the encoder $\Psi_{\mathrm{e}}^{*} = \min_{\Psi_{\mathrm{e}}} \frac{1}{2} \left\| \Psi_{\mathrm{e}}\left(Y\right) - U \right\|_F^2,$

then decoder with pre-trained encoder $\Psi_{\mathrm{d}}^{*} = \min_{\Psi_{\mathrm{d}}} \frac{1}{2} \left\| \Psi_{\mathrm{d}}\left(\Psi_{\mathrm{e}}^{*}\left(Y\right)\right) - Y \right\|_F^2.$

Applying 1st optimality condition for the optimal solutions $\Psi_{\mathrm{e}}^{*}$ and $\Psi_{\mathrm{d}}^{*}$, we have

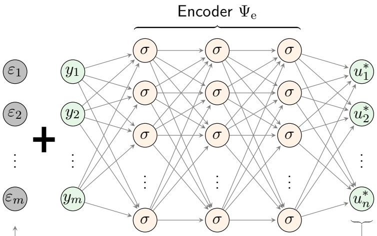
- test inverse solution error

$$\boldsymbol{\varepsilon}_{\boldsymbol{u}^{\text{test}}}^{\texttt{nAE}} = \left\| \Psi_{\mathrm{e}}^{*}(\boldsymbol{y}^{\text{test}}) - \boldsymbol{u}^{\text{test}} \right\|_2^2 = \left\| \left( \bar{U}\bar{Y}^{\dagger}\mathrm{G}^{B} - \mathbf{I} \right) \left( \boldsymbol{u}^{\text{test}} - \bar{\boldsymbol{u}} \right) \right\|_2^2.$$

- test forward solution error, with $Z = \Psi_{\mathrm{e}}^{*}(Y)$ is the encoder inverse solution,

$$\boldsymbol{\varepsilon}_{\boldsymbol{y}^{\text{test}}}^{\texttt{nAE}} = \left\| \Psi_{\mathrm{d}}^{*}(\boldsymbol{u}^{\text{test}}) - \boldsymbol{y}^{\text{test}} \right\|_2^2 = \left\| \bar{Y} \left( \bar{Z}^{\dagger} - \bar{U}^{\dagger} \right) \left( \boldsymbol{u}^{\text{test}} - \bar{\boldsymbol{u}} \right) \right\|_2^2 > 0,$$

# Model-constrained Autoencoder (mcAE) - embedding physics model

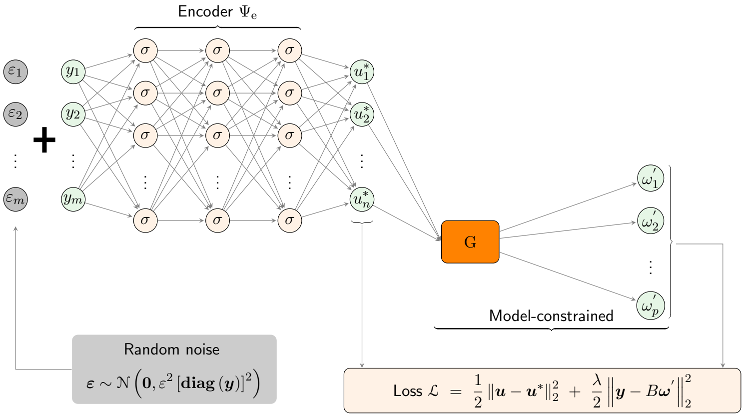



Random noise

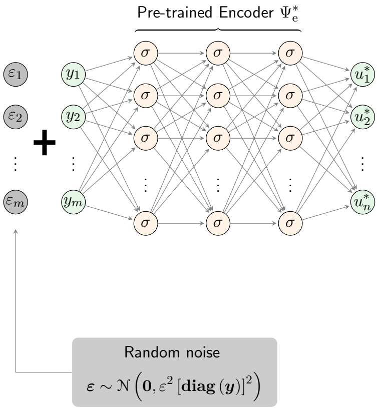
$$\boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \varepsilon^2 \left[\mathbf{diag}\left(\boldsymbol{y}\right)\right]^2\right)$$

Loss $\mathcal{L} = \frac{1}{2}\left\|\boldsymbol{u} - \boldsymbol{u}^*\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{y} - B\boldsymbol{\omega}'\right\|_2^2$

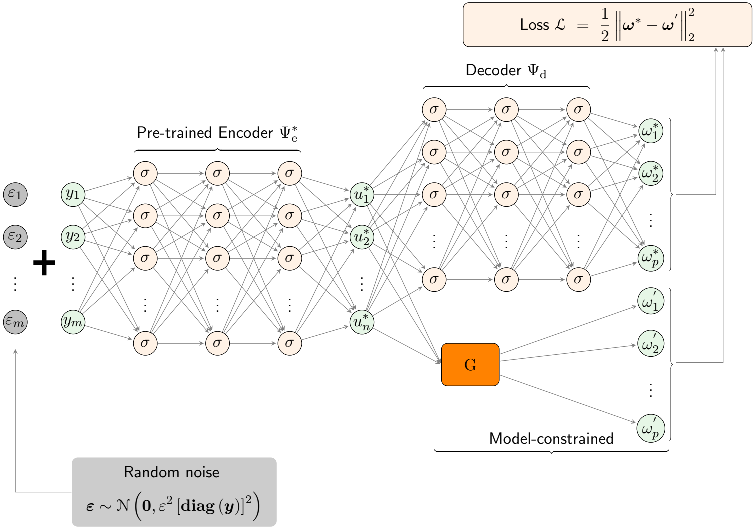
# Model-constrained Autoencoder (mcAE) - embedding physics model



# Model-constrained Autoencoder (`mcAE`) - embedding physics model



# Model-constrained Autoencoder (mcAE) - embedding physics model



# Model-constrained Autoencoder (mcAE) - error estimation

mcAE loss functions

$$\Psi_{\mathrm{e}}^{*} = \min_{\Psi_{\mathrm{e}}} \frac{1}{2} \left\| U - \Psi_{\mathrm{e}}\left(Y\right) \right\|_{F}^{2} + \frac{\lambda}{2} \left\| Y - \mathrm{G}^{B}\left(\Psi_{\mathrm{e}}\left(Y\right)\right) \right\|_{F}^{2},$$

$$\Psi_{\mathrm{d}}^{*} = \min_{\Psi_{\mathrm{d}}} \frac{1}{2} \left\| \mathrm{G}\left(\Psi_{\mathrm{e}}^{*}\left(Y\right)\right) - \Psi_{\mathrm{d}}\left(\Psi_{\mathrm{e}}^{*}\left(Y\right)\right) \right\|_{F}^{2}.$$

Applying 1st optimality condition, and **if $\bar{Y}$ is a full row rank**, we have

- test inverse solution error

$$\varepsilon_{\boldsymbol{u}^{\text{test}}}^{\mathtt{mcAE}} = \left\| \left(\mathbf{I} + \lambda {\mathrm{G}^{B}}^{T} \mathrm{G}^{B}\right)^{-1} \left(\left(\bar{U}\bar{Y}^{\dagger}\mathrm{G}^{B} - \mathbf{I}\right)\left(\boldsymbol{u}^{\text{test}} - \bar{\boldsymbol{u}}\right)\right) \right\|_{2}^{2}$$

which is smaller than

$$\varepsilon_{\boldsymbol{u}^{\text{test}}}^{\mathtt{nAE}} = \left\| \left(\bar{U}\bar{Y}^{\dagger}\mathrm{G}^{B} - \mathbf{I}\right)\left(\boldsymbol{u}^{\text{test}} - \bar{\boldsymbol{u}}\right) \right\|_{2}^{2}.$$

- test forward solution error

$$\varepsilon_{\boldsymbol{\omega}^{\text{test}}}^{\mathtt{mcAE}} = 0.$$

# Key issue of `mcAE`: Error estimation is data-dependent

Recall the test inverse solution error

$$\varepsilon^{\texttt{mcAE}}_{\boldsymbol{u}^{\text{test}}} = \left\| \left(\mathbf{I} + \lambda \mathrm{G}^{B^T} \mathrm{G}^B\right)^{-1} \left( \left( \bar{U} \bar{Y}^\dagger \mathrm{G}^B - \mathbf{I} \right) \left( \boldsymbol{u}^{\text{test}} - \bar{\boldsymbol{u}} \right) \right) \right\|_2^2$$

where

- $\bar{U}, \bar{Y}$ is centralized training data.
- $\bar{\boldsymbol{u}}$ is the average of training PoI samples.

If training data $U, Y$ is limited, surrogate models heavily bias on given training data. Hence, leading to high test inverse error.

Our TAEN approach is introduced to deal with this issue

# TAEN: Tikhonov Autoencoder Network Approach

Recall the `mcAE` loss functions

$$\Psi_{\mathrm{e}}^* = \min_{\Psi_{\mathrm{e}}} \frac{1}{2} \left\| U - \Psi_{\mathrm{e}}\left(Y\right) \right\|_F^2 + \frac{\lambda}{2} \left\| Y - \mathrm{G}^B\left(\Psi_{\mathrm{e}}\left(Y\right)\right) \right\|_F^2,$$

$$\Psi_{\mathrm{d}}^* = \min_{\Psi_{\mathrm{d}}} \frac{1}{2} \left\| \mathrm{G}\left(\Psi_{\mathrm{e}}^*\left(Y\right)\right) - \Psi_{\mathrm{d}}\left(\Psi_{\mathrm{e}}^*\left(Y\right)\right) \right\|_F^2.$$

In TAEN, we use the prior mean of PoI, $\boldsymbol{u}_0$, thus true PoI samples are no longer required.

$$\Psi_{\mathrm{e}}^* = \min_{\Psi_{\mathrm{e}}} \frac{1}{2} \left\| \boldsymbol{u}_0 \mathbb{1}^T - \Psi_{\mathrm{e}}\left(Y\right) \right\|_F^2 + \frac{\lambda}{2} \left\| Y - B \circ \mathrm{G}\left(\Psi_{\mathrm{e}}\left(Y\right)\right) \right\|_F^2,$$

$$\Psi_{\mathrm{d}}^* = \min_{\Psi_{\mathrm{d}}} \frac{1}{2} \left\| \mathrm{G}\left(\Psi_{\mathrm{e}}^*\left(Y\right)\right) - \Psi_{\mathrm{d}}\left(\Psi_{\mathrm{e}}^*\left(Y\right)\right) \right\|_F^2.$$

Applying 1st optimality condition, and **if $\bar{Y}$ is a full row rank**, we have

- test inverse solution error

$$\varepsilon_{\boldsymbol{u}^{\text{test}}}^{\texttt{mcAE}} \leq \left\| \boldsymbol{u}^{\text{test}} - \boldsymbol{u}_0 \right\|_2^2$$

- test forward solution error

$$\varepsilon_{\boldsymbol{\omega}^{\text{test}}}^{\texttt{mcAE}} = 0,$$

# Obtaining a full row rank $\bar{Y}$ from 1 observation sample $\boldsymbol{y}$

# Summary of error estimation

## Error estimation on test samples

| Approaches | forward error $\varepsilon_{\boldsymbol{y}^{\text{test}}} / \varepsilon_{\boldsymbol{\omega}^{\text{test}}}$ | Inverse error $\varepsilon_{\boldsymbol{u}^{\text{test}}}$ | Ability to learn with 1 sample $\boldsymbol{y}$ |
|---|---|---|---|
| `nAE` | inevitably $> 0$ | $\left\Vert \left(\bar{U}\bar{Y}^{\dagger}\mathrm{G}^{B} - \mathbf{I}\right)\left(\boldsymbol{u}^{\text{test}} - \bar{\boldsymbol{u}}\right)\right\Vert_2^2$ | none |
| `mcAE` | 0 | $\leq \left\Vert \left(\bar{U}\bar{Y}^{\dagger}\mathrm{G}^{B} - \mathbf{I}\right)\left(\boldsymbol{u}^{\text{test}} - \bar{\boldsymbol{u}}\right)\right\Vert_2^2$ | only forward |
| `TAEN` | 0 | $\leq \left\Vert \boldsymbol{u}^{\text{test}} - \boldsymbol{u}_0\right\Vert_2^2$ | forward and inverse |

- Please see our preprint for discussions for non-linear problems.

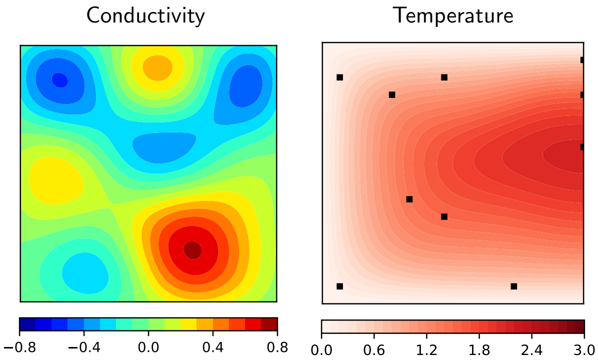
# Numerical results

## Results

- 2D Heat equation
- 2D Navier Stokes equation
- Computational cost and speedup

# 2D Heat equation

$$
\begin{aligned}
-\nabla \cdot (e^u \nabla \omega) &= 20 \quad \text{in } \Omega = (0,1)^2 \\
\omega &= 0 \quad \text{on } \Gamma^{\text{ext}} \\
\mathbf{n} \cdot (e^u \nabla \omega) &= 0 \quad \text{on } \Gamma^{\text{root}},
\end{aligned}
$$




Forward problem: Given the heat conductivity field, we aim to predict the temperature field.
Inverse problem: Given 10 observations, we aim to reconstruct the heat conductivity field.

# 2D Heat equation: relative error

## Relative error over 500 test samples ($\varepsilon = 0.1$)

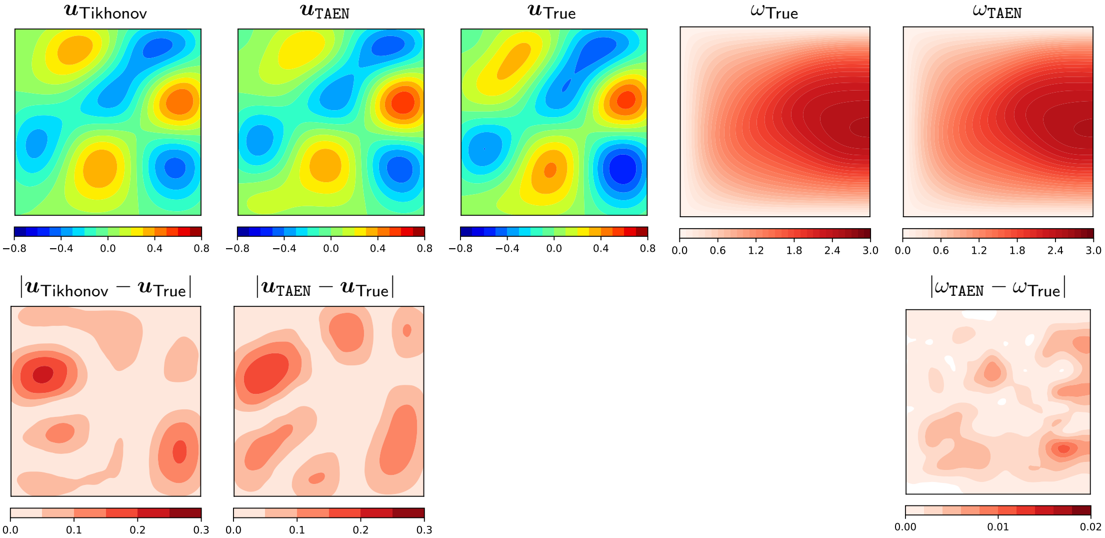
| Approach | 1 training sample | | 100 training samples | |
|---|---|---|---|---|
| | Inverse (%) | Forward | Inverse (%) | Forward |
| `nAE` | 107.55 | $2.90\times10^{-1}$ | 50.18 | $1.09\times10^{-1}$ |
| `mcAE` | 108.28 | $4.21\times10^{-2}$ | 46.32 | $4.56\times10^{-4}$ |
| `TAEN` | 45.23 | $1.57\times10^{-4}$ | 45.03 | $1.22\times10^{-4}$ |
| Tikhonov | 44.99 | | 44.99 | |

## Arbitrarily chosen samples for 1 training sample case

The statistic of inverse solution error over 10 cases

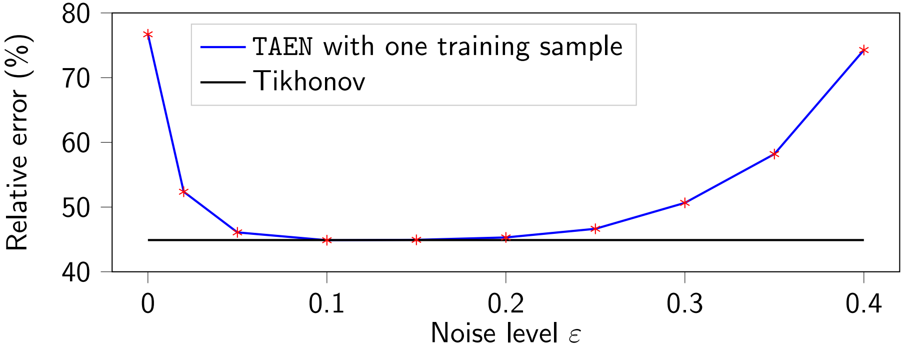
$$45.32 \pm 0.32\%$$

# 2D Heat equation: a test sample prediction



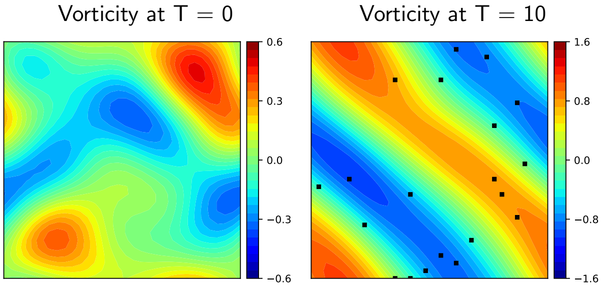
# 2D Heat equation: how much randomization noise?

Relative error of inverse solution over 500 test samples with different noise levels.



# 2D Navier–Stokes equation

$$
\begin{aligned}
\partial_t \omega(x,t) + v(x,t)\cdot\nabla\omega(x,t) &= \nu\Delta\omega(x,t) + f(x), && x\in(0,1)^2, t\in(0,T],\\
\nabla\cdot v(x,t) &= 0, && x\in(0,1)^2, t\in(0,T],\\
\omega(x,0) &= u(x), && x\in(0,1)^2,
\end{aligned}
$$




Forward problem: Given initial vorticity, we aim to predict the final vorticity.

Inverse problem: Given 20 observations at T = 10, we aim to reconstruct the initial vorticity.

# 2D Navier–Stokes equation: relative error

## Relative error over 500 test samples ($\varepsilon = 0.25$)

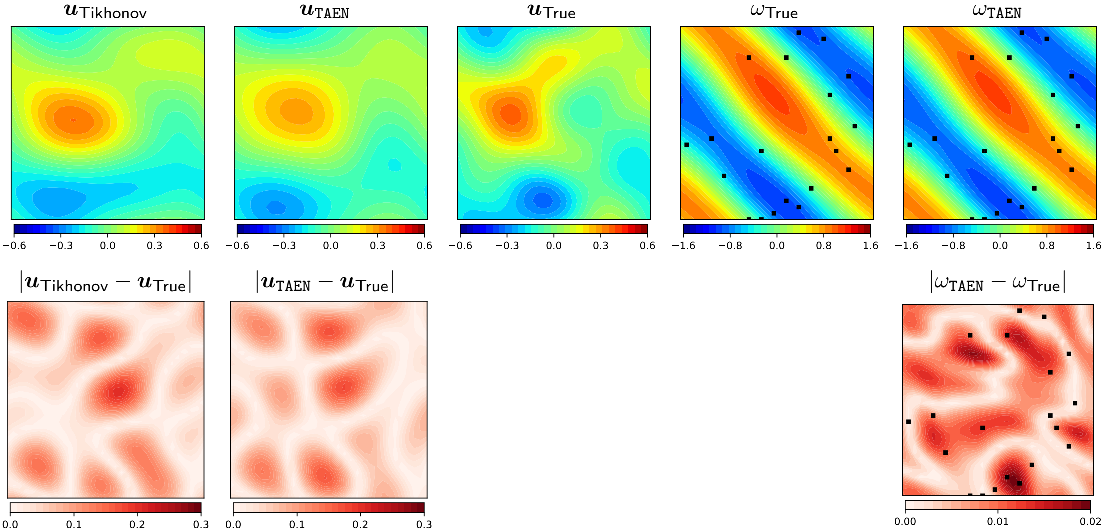
| | 1 training sample | | 100 training samples | |
|---|---|---|---|---|
| **Approach** | **Inverse (%)** | **Forward** | **Inverse (%)** | **Forward** |
| nAE | 103.94 | 5.60 | 40.20 | $5.94 \times 10^{-1}$ |
| mcAE | 46.43 | $5.15 \times 10^{-1}$ | 27.29 | $2.20 \times 10^{-3}$ |
| TAEN | 25.68 | $2.14 \times 10^{-3}$ | 24.54 | $1.49 \times 10^{-3}$ |
| Tikhonov | 22.71 | | 22.71 | |

## Arbitrarily chosen samples for 1 training sample case

The statistic of inverse solution error over 12 cases

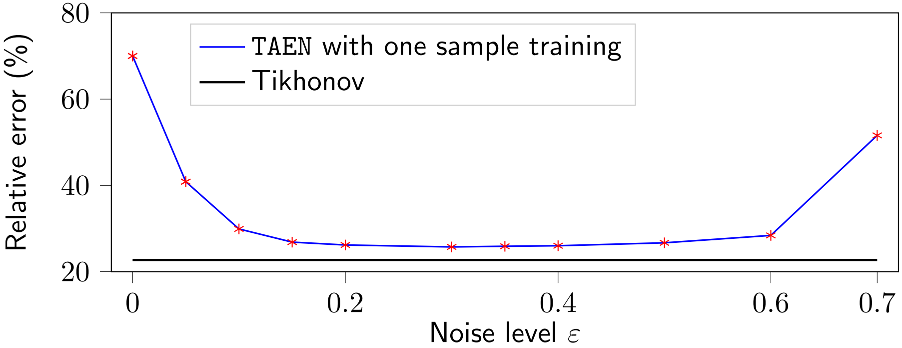
$$25.88 \pm 0.19\%$$

# 2D Navier–Stokes equation: a test sample prediction



# 2D Navier–Stokes equation: how much randomization noise?

Relative error of inverse solution over 500 test samples with different noise levels.



# Train/test computational cost & speed up for TAEN

Train/Test computation cost & Speed up for TAEN
Implemented on NVIDIA A100 GPUs (TACC)

| | | **Heat equation** | **Navier–Stokes** |
|---|---|---|---|
| Training Encoder + Training Decoder (hours) | | 2 | 16 |
| Test/Inference (second) | Inverse (Encoder) | $2.74 \times 10^{-4}$ | $2.93 \times 10^{-4}$ |
| | Forward (Decoder) | $2.86 \times 10^{-4}$ | $3.06 \times 10^{-4}$ |
| Numerical solvers (second) | Inverse (Tikhonov) | $4.36 \times 10^{-2}$ | 7.26 |
| | Forward | $3.01 \times 10^{-2}$ | 0.38 |
| Speed up | Inverse | 159 | 24,785 |
| | Forward | 105 | 1,241 |

# Conclusion

## TAEN (under review)

- is able to learn forward and inverse maps with only 1 observation sample
- is robust to a wide range of data randomization noise level
- is robust to an arbitrarily chosen observation sample
- provides real-time solvers

Drawbacks:

- has expensive training cost due to differentiable PDE solvers
- is not helpful for rich data scenarios

## Future work

- Relax the requirement for differentiable solvers
- Extend to large-scale problems
- ...