Discretization Invariant Operator Learning

Yong Zheng Ong

Department of Mathematics
National University of Singapore

September 28, 2022
SIAM Conference of Mathematics of Data Science (MDS22)

Collaborators

Haizhao Yang

Department of Mathematics
University of Maryland College Park

Zuowei Shen

Department of Mathematics
National University of Singapore

Outline

Operator Learning

Learn a map between function spaces.

- Denote $\mathcal{X}$ and $\mathcal{Y}$ as appropriate function spaces.
- E.g., $\mathbb{R}$-valued functions over $\Omega \subset \mathbb{R}^d$.
- Suppose that $\Psi : \mathcal{X} \to \mathcal{Y}$ denotes a mapping between $\mathcal{X}$ and $\mathcal{Y}$.

The objective of operator learning is to learn the underlying mapping Ψ.

Example

Consider the Burgers Equation given by

$$\begin{aligned} \partial_t u(x,t) + \partial_x(u^2(x,t)/2) &= \nu \partial_{xx} u(x,t), \qquad x \in (0,1), t \in (0,1] \\ u(x,0) &= u_0(x) \end{aligned}$$

with periodic boundary conditions, and ν representing the viscosity. An example of operator learning is to learn the mapping $\Psi : u_0(\cdot, 0) \to u(\cdot, 1)$.

Why Operator Learning

Broad Applications:

- Predictive Modelling.
- Solving parametric PDEs.
- Solving Inverse Problems.
- Image Processing.
- Signal Processing.

Deep Neural Network

$$y = \Psi^n(x;\theta) = \sigma \circ T \circ h^L \circ h^{L-1} \circ \cdots \circ h^1(x)$$

- $h^i(x) = \sigma({W^i}^T x + b^i)$
- σ is an activation function, e.g., $ReLU(x) = \max\{0, x\}$.
- $T(x) = V^T x$.
- $\theta = \{W^1, \cdots, W^L, b^1, \cdots, b^L, V\}$.

Approach: Learn $\Psi : \mathcal{X} \to \mathcal{Y}$.

- Use deep neural network $\Psi^n(\cdot;\theta)$ to parameterize Ψ.
- Learn Ψ^n via supervised learning, given samples $\{(f_i, g_i)\}_{i=1}^n$, $f_i \in \mathcal{X}$ and $g_i \in \mathcal{Y}$.

Why Data Driven

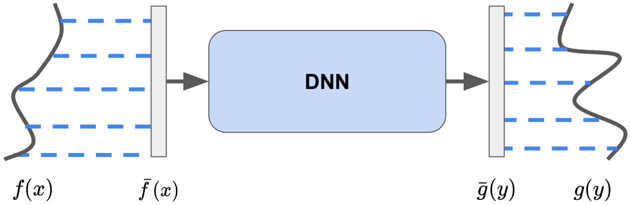
Conventional Methods

- Years of design needed to develop methods.
- Accurate but maybe slow.

Data Driven Methods

- Learn to solve from data.
- After model is trained, evaluation is fast.

Discretization Invariance



NNs were originally proposed to learn mappings between a vector space $X \subset \mathbb{R}^{d_x}$ to another vector space $Y \subset \mathbb{R}^{d_y}$.

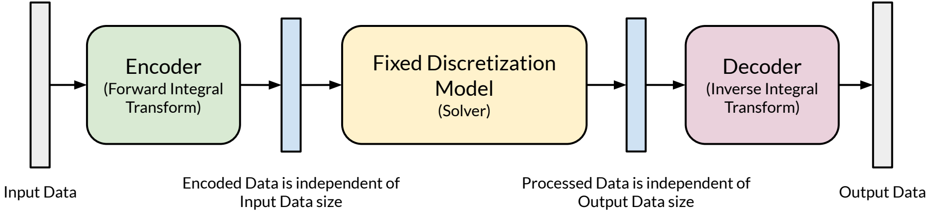
- $\bar{f} \in X$ and $\bar{g} \in Y$ comes from the discretization of functions $f \in \mathcal{X}$ and $g \in \mathcal{Y}$.
- We cannot apply a NN trained on X and Y to other discrete spaces.

Why Discretization Invariance

- Apply trained model to other resolutions.
- Expensive re-training is required to retrain a new NN for different discretization formats.
- Difficult to obtain training samples from certain discrete formats, e.g., high resolution.

Integral Autoencoder (IAE)

(Ong, Shen, Y., 2022)



Map between an Input/Output Data of arbitrary discretization to a representation of a fixed discretization.

- **Data-driven integral transforms**
- Captures intrinsic dimension (sparsity).

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Integral Transform

(Ong, Shen, Y., 2022)
A non-linear integral transform:

$$(Tf)(z) = \int_{\Omega_x} K(f(x), x, z) f(x) dx, \qquad z \in \Omega_z.$$

- f function defined on Ω_x.
- K kernel function.
- Tf function on Ω_z, independent of Ω_x.

Idea: Design K as a NN $\phi_1(\cdot; \theta_{\phi_1})$.

$$v(z) = \int_{\Omega_x} \phi_1(f(x), x, z; \theta_{\phi_1}) f(x) dx, \qquad z \in \Omega_z.$$

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Backward Integral Transform

(Ong, Shen, Y., 2022)
Backward integral transform:

$$g(y) = \int_{\Omega_z} \phi_2(u(z), x, z; \theta_{\phi_2}) u(z) dz, \qquad y \in \Omega_y.$$

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Backward Integral Transform

(Ong, Shen, Y., 2022)
Backward integral transform:

$$g(y) = \int_{\Omega_z} \phi_2(u(z), x, z; \theta_{\phi_2}) u(z) dz, \qquad y \in \Omega_y.$$

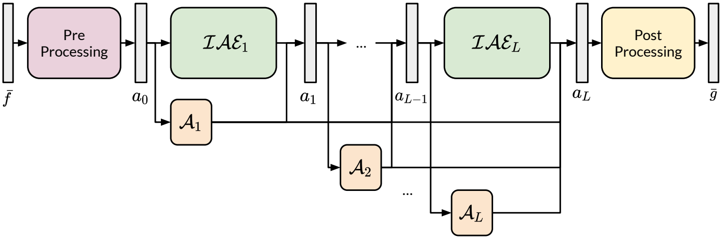
Introduce ϕ_0 to map $v(z)$ to $u(z)$.

$$f(x) \xrightarrow{\phi_1} v(z) \xrightarrow{\phi_0} u(z) \xrightarrow{\phi_2} g(y)$$

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Multi-Layer Structure

(Ong, Shen, Y., 2022)



- Compose consecutive $\mathcal{IAE}$-Blocks, each containing an IAE.
- $\mathcal{IAE}_i$ takes as input $a_0, \ldots, a_{i-1}$ from previous $\mathcal{IAE}$-Blocks.
- Pre and Post Processing handle dimensions.
- $\mathcal{A}_i$ are pointwise affine transforms performing rescaling.

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Why Multiple Layers

(Ong, Shen, Y., 2022)

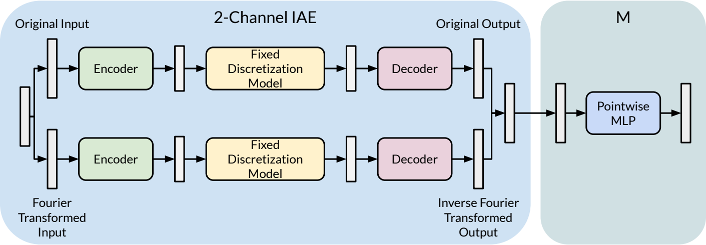
$$\bar{f} \xrightarrow{\text{Pre Process}} a_0 \xrightarrow{\mathcal{IAE}_1} a_1 \xrightarrow{\mathcal{IAE}_2} \dots \xrightarrow{\mathcal{IAE}_L} a_L \xrightarrow{\text{Post Process}} \bar{g}.$$

- Multi-layer structure incorporates multiple encoders.
- Each focus on specific features.
- Use previous approximations $a_0, \dots, a_{i-1}$ to learn a_i.

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Multi-Channel Structure

(Ong, Shen, Y., 2022)



- Multiple IAEs are assembled in parallel, using known integral transforms as a guide.
- Different features are captured in different domains.

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Data Augmentation

(Ong, Shen, Y., 2022)

- IAE-Net uses data-driven integral kernels.
- Does not assume any known structure in the data.

Question: How do we guide IAE-Net to learn to encode data from different resolutions?

Data Augmentation!

$$\min_{\theta_{\Psi^n}} \mathbb{E}_{(\bar{f},\bar{g})\sim p_{data}} \mathbb{E}_{I_T\sim\mathcal{I}}[L(\Psi^n(\bar{f};\theta_{\Psi^n}),\bar{g}) + \lambda \mathbf{L}(\boldsymbol{\Psi}^{\mathbf{n}}(\mathbf{I_T}(\bar{\mathbf{f}});\theta_{\boldsymbol{\Psi}^{\mathbf{n}}}),\mathbf{I_T}(\bar{\mathbf{g}}))],$$

$I_T \sim \mathcal{I}$ denotes a randomly sampled interpolation operator from a set of interpolator functions $\mathcal{I}$.

Y. Z. Ong, Z. Shen, and H. Yang. IAE-Net: Integral Autoencoders for Discretization-Invariant Learning (2022).

Baseline Models

- UNet (Ronneberger et al., 2015)
- DeepONet (Lu et al., 2019)
- Fourier Neural Operator (FNO) (Li et al., 2020)
- Fourier Transformer (FT), Galerkin Transformer (GT) (S. Cao, 2021)

UNet: O. Ronneberger, P. Fischer, T. Brox. U-net: Convolutional networks for biomedical image segmentation (2015).
DeepONet: L. Lu, P. Jin, G. E. Karniadakis. Deeponet: Learning nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems (2019).
Fourier Neural Operator (FNO): Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, A. Anandkumar. Fourier neural operator for parametric partial differential equations (2020).
Fourier Transformer (FT), Galerkin Transformer (GT): S. Cao. Choose a transformer: Fourier or galerkin (2021).

Predictive Modelling

Example 1: Burgers Equation

$$\partial_t u(x,t) + \partial_x(u^2(x,t)/2) = \nu \partial_{xx} u(x,t), \quad x \in (0,1), t \in (0,1]$$
$$u(x,0) = u_0(x)$$

- Periodic Boundary Condition.
- ν denotes a given viscosity coefficient.
- **Operator:** Mapping from $u(x,0) \to u(x,1)$.

Predictive Modelling

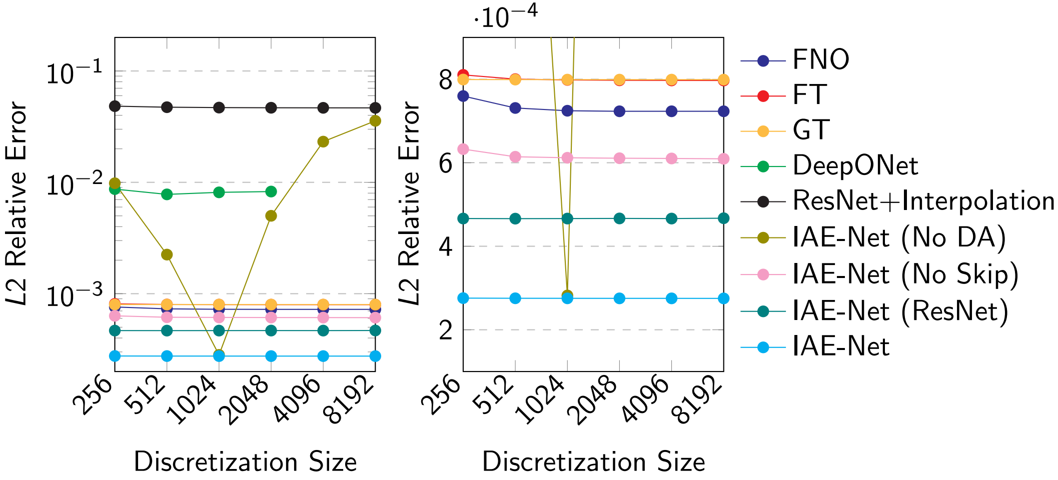


Figure: $L2$ relative error on the Burgers data set with $\nu = 1e^{-1}$ (Left) and its closeup (Right). Models are trained with $s = 1024$ and tested on the other resolutions.

Forward Problem

Example 2: Darcy Flow

$$
\begin{aligned}
-\nabla \cdot (a(x)\nabla u(x)) = f(x), \quad x \in (0,1)^2 \\
u(x) = 0, \quad x \in \partial(0,1)^2,
\end{aligned}
$$

- $f(x)$ denotes a given forcing function.
- **Operator:** Mapping from $a(x) \to u(x)$.

Forward Problem

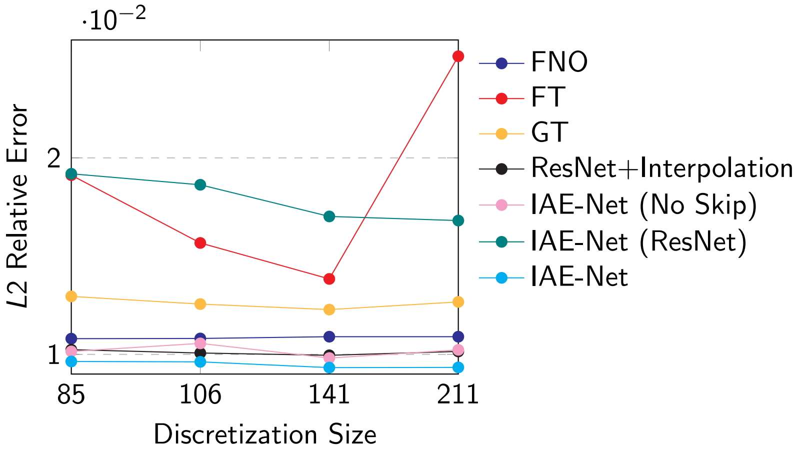


Figure: $L2$ relative error on the benchmark Darcy data set. Models are trained with $s = 141$ size training data and tested on the other resolutions.

Inverse Problem

Example 3: Scattering Problem
Helmholtz Equation

$$(-\Delta - \frac{\omega^2}{c^2(x)})u = 0$$

In most applications, a known background velocity $c_0(x)$ exists. Introduce

$$\frac{\omega^2}{c(x)^2} = \frac{\omega^2}{c_0(x)^2} + \eta(x) \qquad L_0 = -\Delta - \frac{\omega^2}{c_0^2(x)}$$

Parametric PDE

$$(L_0 - \eta(x))u(x) = 0$$

- **Operator:** Mapping between $u(x)$ and $\eta(x)$.

Inverse Problem

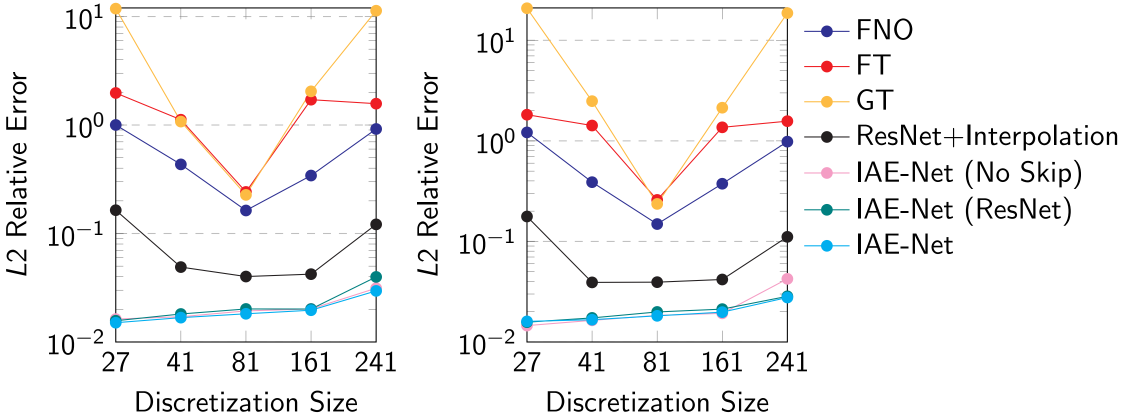


Figure: $L2$ relative error on the scattering data set for the forward (Left) and inverse (Right) problem. Model is trained with $s = 81$ and tested on different resolutions.

Example 4: ECG Signal Separation

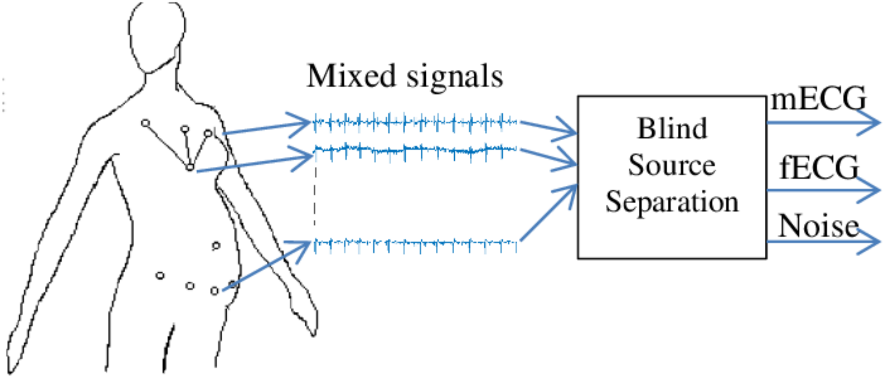


Image From: K. Bensafia, A. Mansour, S. Haddab. Blind Source Subspace Separation and Classification of ECG Signals (2017).

Signal Processing

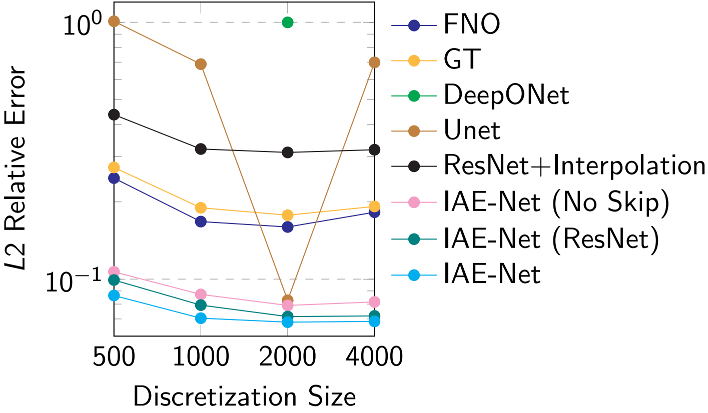


Figure: $L2$ relative error on the fecgsyndb data set. Model is trained with s=2000 and tested on different resolutions.

Summary

Proposed IAE-Net for Discretization Invariant Operator Learning.

- Learn a **data-driven kernel** function using DL and **Data Augmentation** to achieve discretization invariance.
- **Multi-Layer**, **Multi-Channel** helps improve the learning process.

Numerically, we show that

- Discretization invariance is achievable for mathematically well-posed problems even with simple interpolation techniques, but IAE-Net achieves state-of-the-art accuracy.
- Existing methods fail to achieve discretization invariance for ill-posed and highly oscillatory problems, while IAE-Net succeeds with reasonably good accuracy.

Paper: `https://arxiv.org/abs/2203.05142`
Code: `https://github.com/IAE-Net/iae_net`
Email: e0011814@u.nus.edu

Thank you for your attention!