Inducing Point Operator Transformer: A Flexible and Scalable Architecture for Solving PDEs

Seungjun Lee, Taeil Oh

Alsemy, South Korea
seungjun.lee@alsemy.com, taeil.oh@alsemy.com

AAAI 2024

Introduction

Traditional Numerical Methods for Solving PDEs

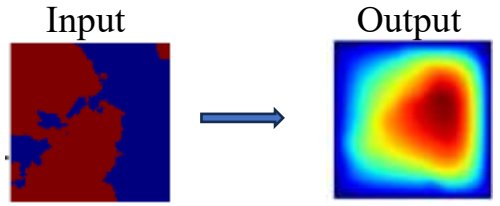
1. **Mesh Generation**: The computational domain must be discretized into grids or elements.

2. **High Computational Cost**: Solving large-scale algebraic systems is computationally expensive, especially for fine discretizations or high-dimensional problems.

3. **Re-solving for New Conditions**:
   When boundary or initial conditions change:
   - The mesh may need to be reconstructed or refined.
   - The entire PDE system must be solved again.

2D Darcy flow:

$$-\nabla \cdot (k(x)\nabla p(x)) = f(x)$$



Introduction

Learning-Based Solution Operators

Solving PDEs by learning the **solution operators** has emerged as an attractive alternative to traditional numerical methods.
However, there two main challenges:

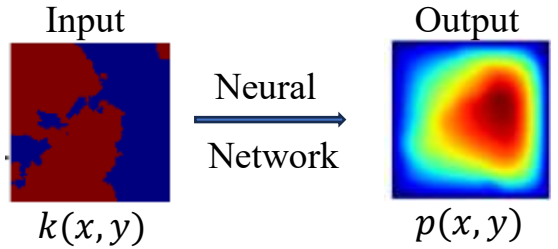
- **flexibility in handling irregular and arbitrary input and output formats**
- **scalability to large discretizations.**

To address these issues, we introduce an attention-based model called an **inducing point operator transformer (IPOT)**.

The goal of this work is to develop a **mesh-agnostic architecture** that can handle any input and output **discretizations with reduced computational complexity**.

2D Darcy flow:

$$-\nabla \cdot (k(x)\nabla p(x)) = f(x)$$



Introduction

Inducing-Point Methods

Rather than modeling the entire set of inputs X, what if we instead focused on modeling **a smaller subset** of regions of the input space?
If we were able to do this, we could exploit any redundancy and correlation in the input space to reduce computation time.

Kernel Integral Operation & Attention Mechanism

The architectures of the **neural operator**, usually consist of three components, namely lifting, iterative updates, and projection, which correspond to the encoder-processor-decoder, respectively.

$$u = \mathcal{G}_\theta(a) = (\mathcal{Q} \circ \mathcal{G}_{L-1} \circ \cdots \circ \mathcal{G}_1 \circ \mathcal{P})(a) \qquad (1)$$

The iterative updates $\mathcal{G}_l : v_l \mapsto v_{l+1}, l \in [0, L-1]$ are global transformations:

$$v_{l+1}(x) = \sigma\left(\mathcal{W}_l v_l(x) + \left[\mathcal{K}_l(v_l)\right](x)\right), \quad x \in \Omega, \qquad (2)$$

where σ are nonlinear functions, Wl are point-wise linear transformations, Kl are kernel integral operations on vl(x)

Kernel Integral Operation & Attention Mechanism

Its primary function is to capture non-local (global) interactions and long-range dependencies within the input function's domain.

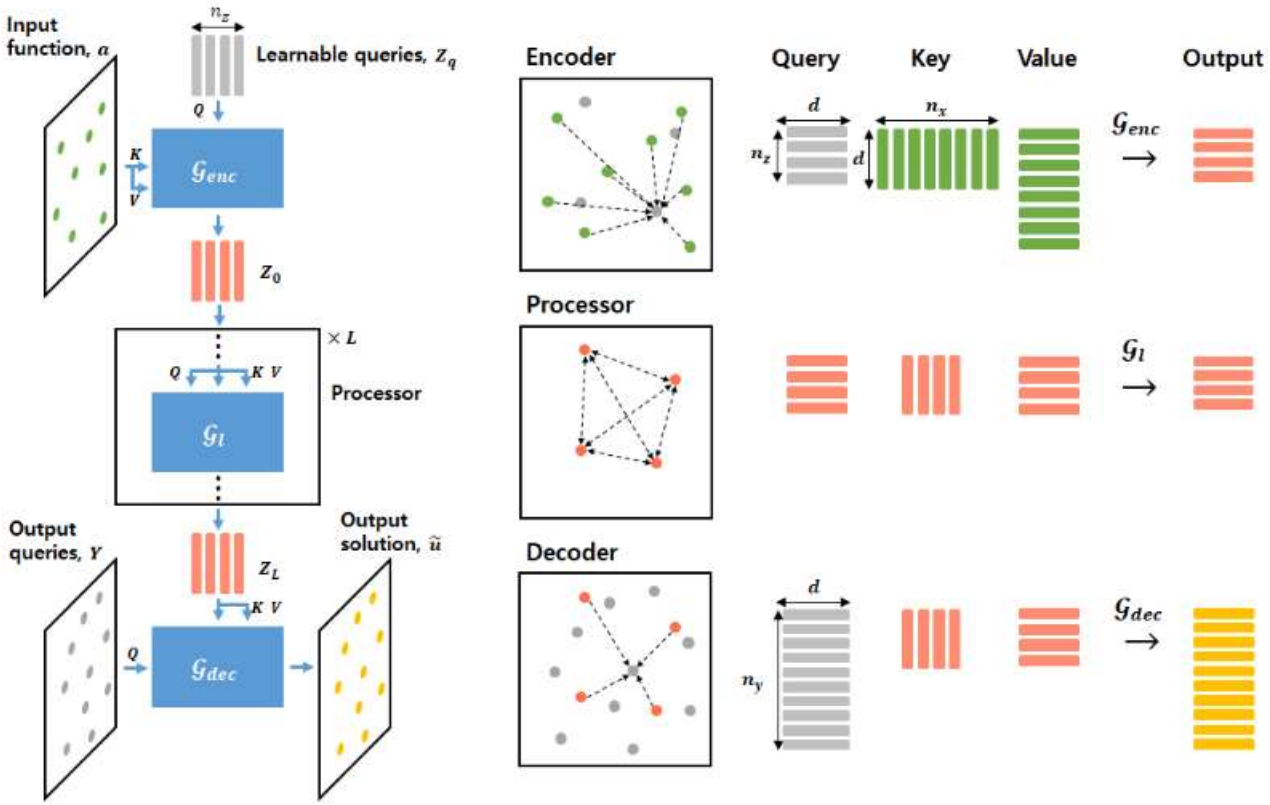
$$[\mathcal{K}(v)](y) = \int_{\Omega_x} \kappa(y,x)v(x)dx, \quad (x,y) \in \Omega_x \times \Omega_y, \tag{3}$$

It has been proved that the kernel integral operation can be successfully approximated by the attention mechanism of Transformers both theoretically and empirically (Kovachki et al. 2021; Cao 2021; Guibas et al. 2021; Pathak et al. 2022).

$$Attn(Y,X,X) = \sigma(QK^T)V \approx \int_{\Omega_x} (q(Y) \cdot k(x))v(x)dx, \tag{4}$$

Where $Q = YW^q \in \mathbb{R}^{n_y \times d}$, $K = XW^k \in \mathbb{R}^{n_x \times d}$ $V = XW^v \in \mathbb{R}^{n_x \times d}$,

Method



2D Darcy flow:

$$-\nabla \cdot (k(x)\nabla p(x)) = f(x)$$

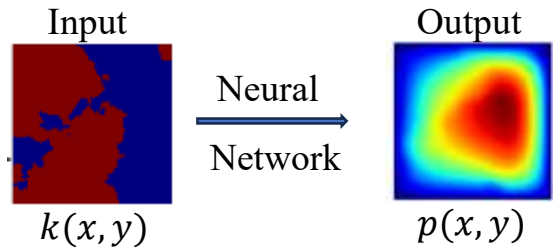


Figure 1: Inducing-Point Operator Transformer (IPOT) uses a smaller number of inducing points, enabling it to flexibly handle any discretization formats of input and output, and significantly reduce the computational costs. IPOT encodes input function discretized on $X = \{x_1, ..., x_{n_x}\}$ to a fixed smaller size n_z of learnable query vectors, and decoding them to output discretized on $Y = \{y_1, ..., y_{n_y}\}$. The number of size is varied as n_x (arbitrary) $\rightarrow$ n_z (fixed) $\rightarrow$ n_y (arbitrary).

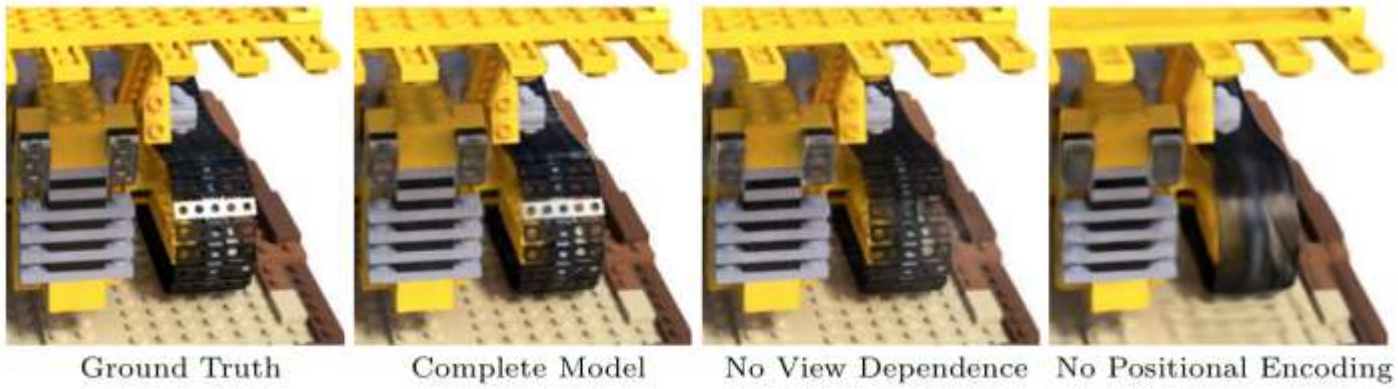
Positional Encoding

We concatenate the **Fourier Feature Position Encoding** [1] with the corresponding function values to form the input representation.

频率位置编码公式： $\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$

对于一个位置p(x,y)，我们用4种频率(如1,2,4,8)来编码，每种频率采用两种相位(sin 和cos)，那编码后的位置应该有2 x 4x2= 16维来表示原始的二维坐标向量。

假设一个位置的xo= 30，它相邻的位置x1= 31，经过r = sin(x*512)编码以后，xo编码后的位置为-0.6842，而x1编码后的位置为0.6240。



[1] NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis.

Model the time-dependent PDEs as an autoregressive process

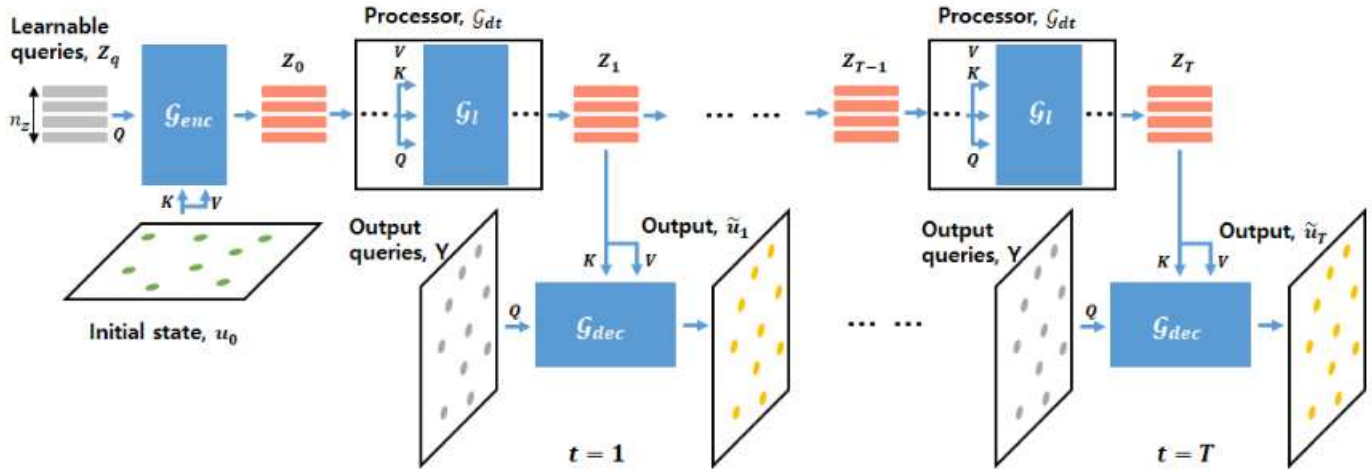


Figure 2: The state of the system at time T is denoted as $u_T = (\mathcal{G}_{dec} \circ [\mathcal{G}_{dt} \circ \cdots \circ \mathcal{G}_{dt}] \circ \mathcal{G}_{enc})(u_0)$, where $\mathcal{G}_{enc}$ is an encoder that maps the initial state u_0 to the latent state, $\mathcal{G}_{dec}$ is a decoder that maps the latent state back to the observational state, and $\mathcal{G}_{dt}$ is a processor that steps forward in time by dt implemented by a series of self-attention blocks on latent states.

Experiment

flows around complex geometries (**airfoil**), estimating stresses on irregularly sampled point clouds (**elasticity**), and predicting displacements given initial boundary conditions for plastic forging problem (**plasticity**)

Model	Dataset	Params	Runtime	Memory	Error	Dataset	Params	Runtime	Memory	Error
DeepONet	Darcy	0.42M	2.73	2.40	4.61e-2	Airfoil	0.42M	3.13	2.75	3.85e–2
MeshGraphNet		0.21M	9.51	5.57	9.67e–2		0.22M	10.92	6.38	5.57e–2
FNO		1.19M	**1.88**	1.96	1.09e–2		1.19M	2.63	2.73	4.21e–2
FFNO		0.41M	3.36	1.99	**7.70e–3**		0.41M	4.71	2.78	**7.80e–3**
OFormer		1.28M	3.63	5.71	1.26e–2		1.28M	4.17	7.97	1.83e–2
IPOT (ours)		**0.15M**	2.70	**1.82**	1.73e–2		**0.12M**	**2.15**	**2.10**	8.79e–3
DeepONet	Navier	-	-	-	-	Elasticity	1.03M	3.72	1.18	9.65e–2
MeshGraphNet		0.29M	137.17	6.15	1.29e–1		0.46M	7.36	4.04	4.18e–2
FNO		0.93M	53.73	3.09	1.28e–2		0.57M	**1.04**	1.68	5.08e–2
FFNO		0.27M	53.82	3.40	1.32e–2		0.55M	2.42	2.08	2.63e–2
OFormer		1.85M	70.15	9.90	1.04e–2		2.56M	5.58	2.98	1.83e–2
IPOT (ours)		**0.12M**	**21.05**	**2.08**	**8.85e–3**		**0.12M**	1.99	**1.13**	**1.56e–2**
DeepONet	ERA5	-	-	-	-	Plasticity	-	-	-	-
MeshGraphNet		2.07M	51.75	18.45	7.16e–2		-	-	-	-
FNO		2.37M	**9.23**	13.04	1.21e–2		1.85M	10.40	16.81	5.08e–2
FFNO		1.12M	14.39	17.06	7.25e–3		0.57M	66.47	16.86	4.70e–3
OFormer		1.85M	71.18	10.90	1.15e–2		0.49M	28.43	14.11	1.83e–2
IPOT (ours)		**0.51M**	9.83	**10.58**	**6.64e–3**		**0.13M**	**10.14**	**5.35**	**3.25e–3**

Table 1: Performance and efficiency comparisons with baselines across various datasets. The efficiencies are compared in terms of the number of parameters, time spent per epoch (seconds), and CUDA memory consumption (GB). The missing entries occur when the methods are not able to handle the datasets or when encountering convergence issues.

Experiment

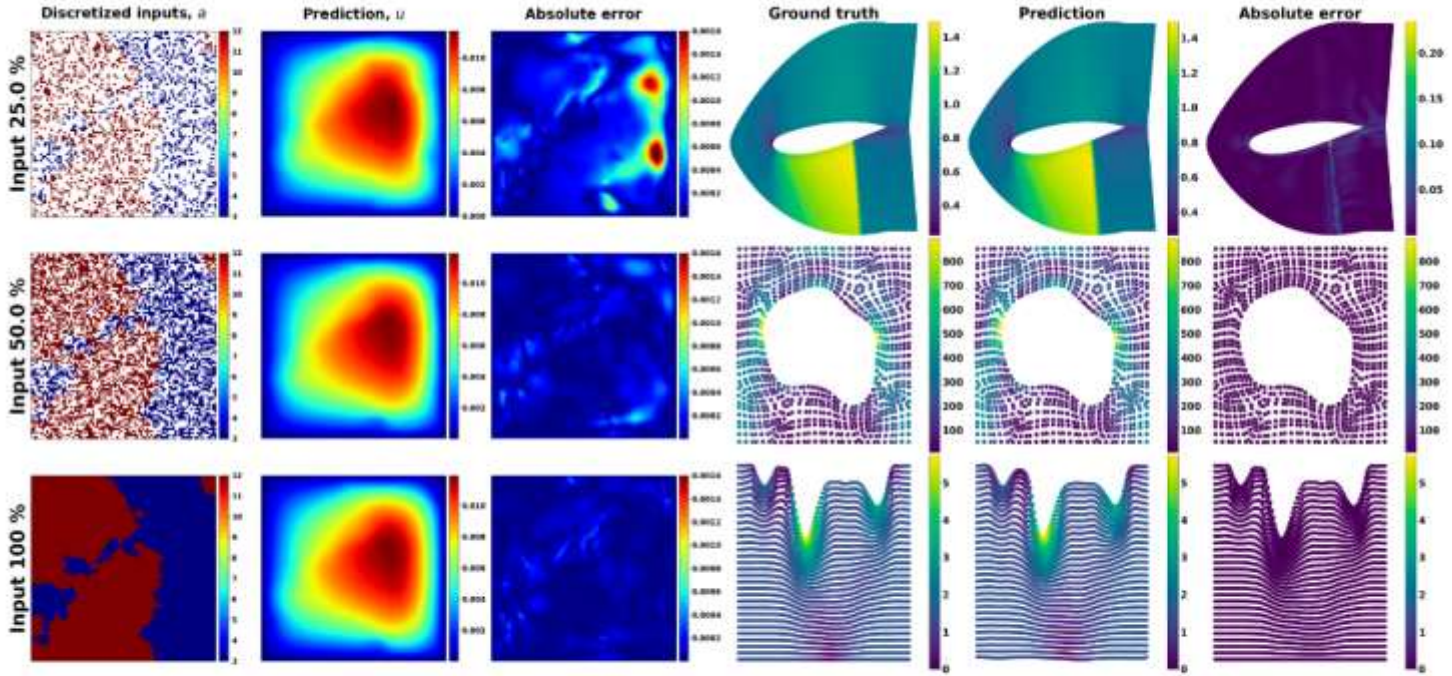


Figure 3: Predictions of IPOT on the problems of Darcy flow (left), airfoil (top right), elasticity (middle right), and plasticity (bottom right). In the case of Darcy flow, the partially observed inputs are randomly subsampled with the ratios of $\{100, 50, 25\}\%$.

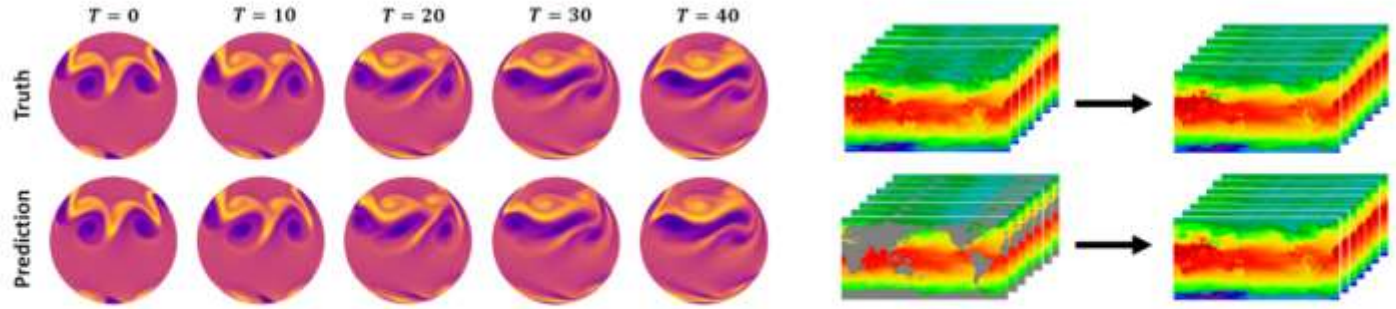


Figure 4: Long-term predictions of IPOT on spherical manifolds for the shallow-water equations (left), and on real-world weather forecasting when the inputs are spatially fully given (top right), and partially given (bottom right).

Experiment

Model	Different resolutions res= 4° / 1° / 0.25°	Partial observed Masked land
FNO	1.30e–2 / 1.23e–2 / 1.24e–2	3.10e–2
OFormer	3.66e–2 / 1.65e–2 / 1.86e–2	4.37e–2
IPOT	**8.96e–3 / 7.78e–3 / 8.66e–3**	**2.83e–2**

Table 2: Results for comparing the generalization ability on discretizations. The models are evaluated on the task of different resolutions and masked inputs for the ERA5 dataset.

Ablation Study

Model	n / n_z	Time	Error	Comp
OFormer	16.2K / 16.2K	71.18	1.15e–2	$\mathcal{O}(nd^2)$
IPOT w.o ip	16.2K / 16.2K	$\gg 100$	–	$\mathcal{O}(n^2d)$
IPOT (64)	16.2K / 64	7.44	1.45e–2	$\mathcal{O}(n_z^2d)$
IPOT (128)	16.2K / 128	7.61	1.30e–2	$\mathcal{O}(n_z^2d)$
IPOT (256)	16.2K / 256	7.91	6.87e–3	$\mathcal{O}(n_z^2d)$
IPOT (512)	16.2K / 512	9.83	6.44e–3	$\mathcal{O}(n_z^2d)$

Table 3: Performance of IPOT with varying the number of inducing points with respective of runtime (Time), error, and complexity (Comp). IPOT w.o ip denotes that IPOT without inducing points which emulates the standard Transformer.

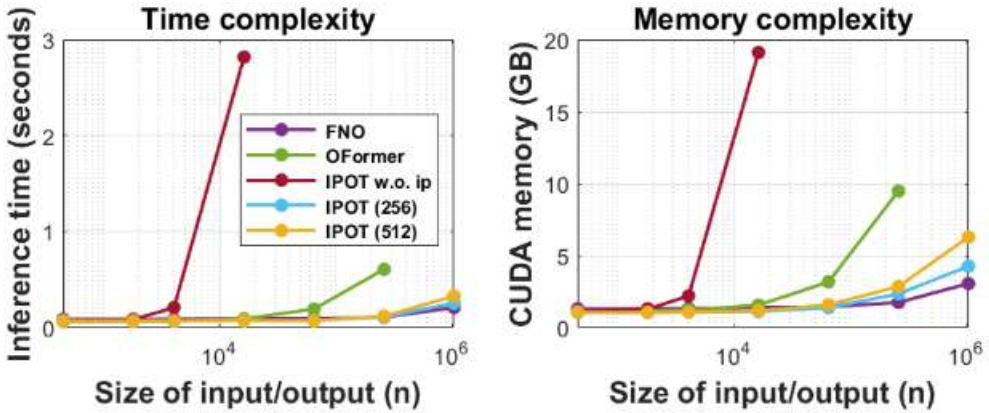


Figure 5: Complexity comparisons on different resolutions. We compare the different models in terms of inference time (left) and CUDA memory usage (right) with different sizes of input/output.