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

Seungjun Lee, Taeil Oh

Alsemy, South Korea

Motivations – flexibility and scalability

Architecture for solving partial differential equations (PDEs) faces two main challenges: flexibility in handling arbitrary discretization formats and scalability to large discretization. Most existing architectures are limited by their desired structure or infeasible to scale large inputs and outputs.

Contributions

- Our attention-based model is designed to handle any input function and output query while capturing global interactions in a computationally efficient way.
- Inspired by inducing points method, our model offers flexibility in processing arbitrary discretization and scales linearly with the size of inputs/outputs.
- Our experimental results demonstrate that our model achieves strong performances with manageable computational complexity on an extensive range of PDE benchmarks and real-world weather forecasting scenarios, compared to state-of-the-art models.

Preliminaries

Neural operators

The objective of operator learning is to minimize the empirical loss $E_{a\sim\mu}[\mathcal{L}(\mathcal{G}_\theta(a),u)] \approx \frac{1}{N}\sum_{i=1}^{N}\|u_i - \mathcal{G}_\theta(a_i)\|^2$ to learn a mapping $\mathcal{G}_\theta : \mathcal{A} \to \mathcal{U}$.

The architecture consists of encoder, processor, and decoder:

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

where $\mathcal{P}$ and $\mathcal{Q}$ are the encoder and decoder. Iterative updates $\mathcal{G}_l : v_l(x) \mapsto v_{l+1}(x)$ capture the interactions between the elements implemented by

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

where $\mathcal{K}_l$ are kernel integral operations on $v_l(x)$.

Kernel integral operation and attention mechanism

Kernel integral operations are generally implemented by

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

where the kernel κ represent the pairwise interactions between the elements on input domain $x \in \Omega_x$ and output domain $y \in \Omega_y$. It's discretization can be approximated by the attention mechanism when input vectors $X \in R^{n_x \times d_x}$ and output query vectors $Y \in R^{n_y \times d_y}$,

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

where, $Q = YW^q \in R^{n_y \times d}, K = XW^k \in R^{n_x \times d}, V = XW^v \in R^{n_x \times d}$ are query, key and value matrix.

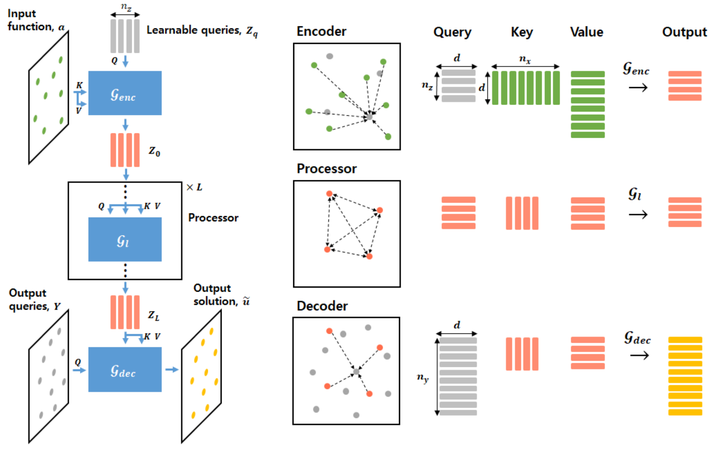
Quadratic complexity of attention modules

When size of input and output discretization is $n_x \approx n_y \approx n$.

- Self-attention: $Attention(X,X,X) \sim \mathcal{O}(n^2 d)$
- Cross-attention: $Attention(Z,X,X) \sim \mathcal{O}(nn_z d) \qquad (n_z \ll n)$

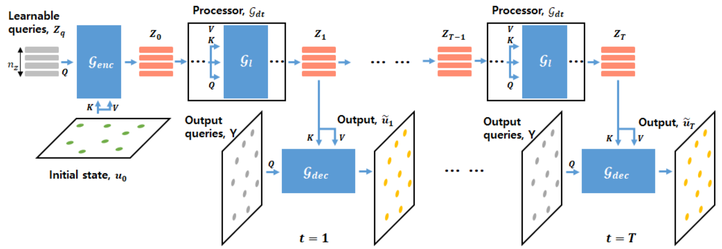
Cross-attention can project observational space into smaller latent space

Inducing point operator transformer (IPOT)



IPOT uses a smaller number of inducing points, enabling it to flexibly handle any discretization formats of input and outputs, and significantly reduce the computational costs.

Time-stepping through latent space

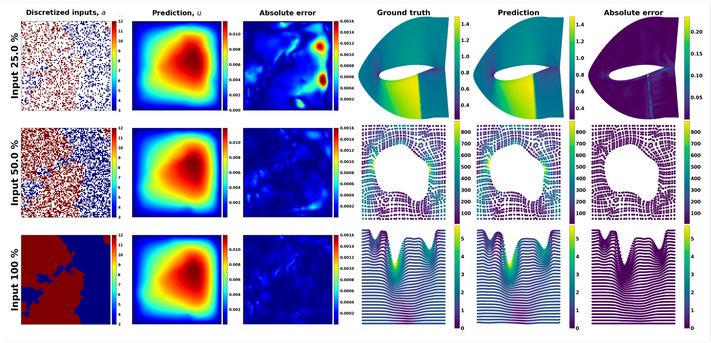


We model the time-dependent PDEs as an autoregressive process. By encoding the initial state into the latent space, we can significantly reduce the the computational costs of subsequent processing compared to processing in the observational space.

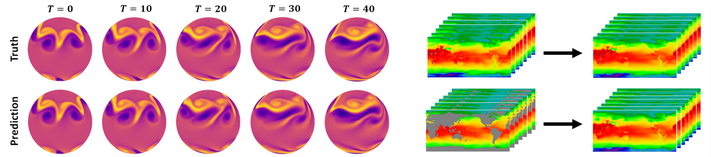
Computational complexity

Model	n	n_z	L	Runtime	Error	Complexity
OFormer	16.2K	16.2K	19	71.18	1.15e–2	$\mathcal{O}(Lnd^2)$
IPOT w.o ip	16.2K	16.2K	28	$\gg 100$	–	$\mathcal{O}(Ln^2 d)$
IPOT (64)	16.2K	64	28	7.44	1.45e–2	$\mathcal{O}(Ln_z^2 d)$
IPOT (128)	16.2K	128	28	7.61	1.30e–2	$\mathcal{O}(Ln_z^2 d)$
IPOT (256)	16.2K	256	28	7.91	6.87e–3	$\mathcal{O}(Ln_z^2 d)$
IPOT (512)	16.2K	512	28	9.83	6.44e–3	$\mathcal{O}(Ln_z^2 d)$

Flexible to arbitrary discretization formats

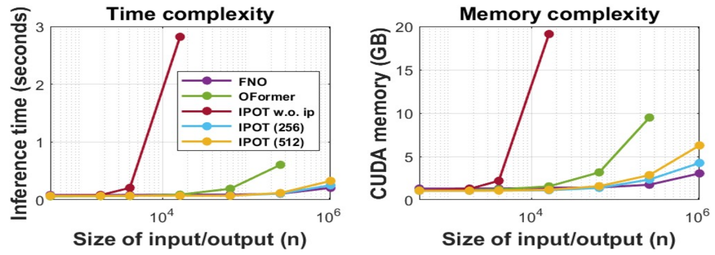


Predictions of IPOT on the problems of Darcy flow (left), airfoil (top right), elasticity (middle right), and plasticity (bottom right).



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).

Scalable to large discretization



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.

References

Z. Li, et. al. (2021). "Fourier Neural Operator for Parametric Partial Differential Equations." In: International Conference on Learning Representation, 2021.

S. Cao. (2021). "Choose a Transformer: Fourier or Galerkin." In: Advances in neural information processing systems, 2021.

Z. Li, et. al. (2023). "Transformer for Partial Differential Equations Operator Learning." In: Transactions on Machine Learning Research, 2023.

A. Jaegle, et. al. (2021). "Perceiver: General Perception with Iterative Attention", In: International Conference on Machine Learning, 2021.