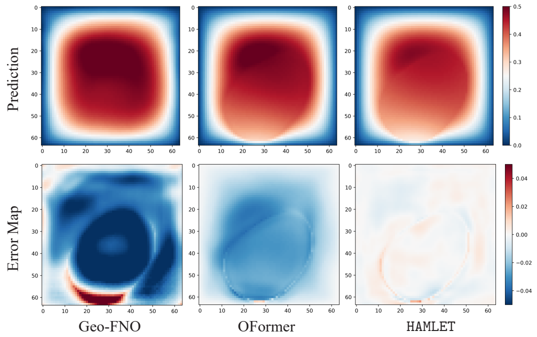
UNIVERSITY OF CAMBRIDGE



HAMLET

GrapH TrAnsforMer NeuraL OpEraTor for Partial Differential Equations

Andrey Bryutkin, Jiahao Huang, Zhongying Deng, Guang Yang, Carola-Bibiane Schönlieb, Angelica Aviles-Rivero

Outline

- Section 1: Understanding the Significance of Partial Differential Equations
- Section 2: Limitations of Current Numerical Techniques
- Section 3: Introducing HAMLET: Graph Transformer Neural Operator for PDEs
- Section 4: Methodology and Experimental Results
- Section 5: Implications and Future Directions

Importance of Partial Differential Equations

Problems in science and engineering reduce to PDEs

Two main cases: Initial condition problems

Heat equation:

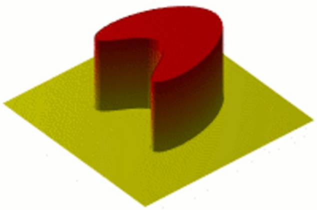
Spatial domain: $D = [0,1] \times [0,1]$

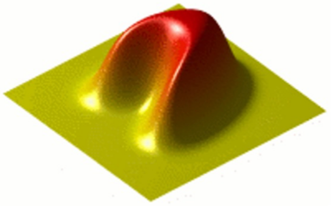
Time domain: $T = [0,1]$

Points: $\vec{x} = (x, y) = (x_1, x_2)$

Solution function (Temperature): $u: T \times D \rightarrow \mathbb{R}$

Equation: $u_t = \Delta u$





Initial value problem: given the temperature at time $t = 0$, What is the temperature a time $t = 1$?

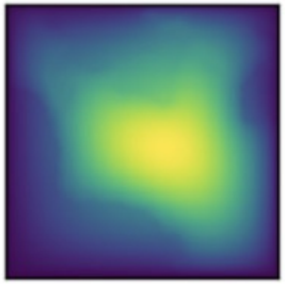
Given $u(0, x)$what is $u(1, x)$?

Two main cases: Boundary condition problems

For example a two dimensional elliptic PDE:

$$-\nabla \cdot \big(a(x)\nabla u(x)\big) = f(x), \qquad x \in D$$

Input: a(x)



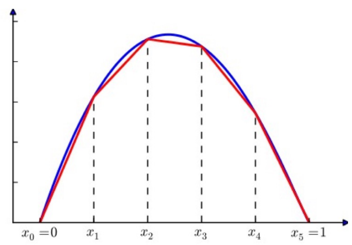
Output: u(x)

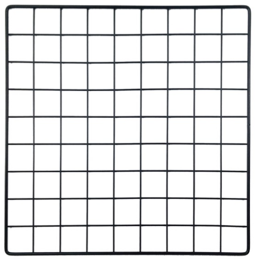
Given $a(x)$, what is $u(x)$?

Numerical Solver

Pipeline

1. Discretize the space
2. Write out a system
3. Solve the system





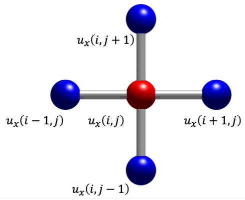
Commong numerical methods:

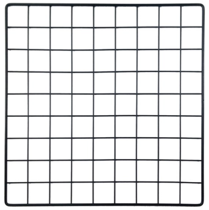
- Finite difference methods
- Finite element methods
- Spectral methods
- Iterative methods

$$u_x(x) = \lim_{dx \to 0} \frac{[u(x+dx) - u(x)]}{dx}$$

$$u_x(x) \sim \frac{[u(x+dx) - u(x)]}{dx}$$

$$u_x(i,j) \sim \frac{[u(i+1,j) - u(i,j)]}{dx}$$





Trade-off: finer grids are more accurate, but also more expensive

Role of Deep Learning in PDEs

- Physics informed neural networks (PINN)
  - Underlying differential equation as a loss-function
- Operator Learning
  - Fourier Neural Operator (FNO)
  - Physics Informed Neural Operator (PINO)
  - Graph Neural Operator (GNO)
  - FNO Transformer
- Encoder-Decoder Networks
- Graph Neural Networks

The Need for Innovative Solutions

Adapting to Varied Geometries:

- The demand for PDE solutions that can adapt to arbitrary geometries and diverse input formats has driven the search for innovative neural network architectures.

Addressing Computational Challenges:

- There is a growing need for models that can handle increasing data complexity and noise while maintaining computational efficiency.

Challenges in Current Numerical Approaches

Limited Generalizability:

- Existing techniques may struggle to generalize across multiple PDE instances and lack discretization invariance, impacting their adaptability to diverse scenarios.

Inability to Generalize Beyond Specific Resolutions:

- Current methods may face limitations in generalizing beyond specific resolutions and geometries observed during training, hindering their applicability to varied scenarios.

Formulation of the Problem

$$P : \mathcal{P} \times \mathcal{D} \times \mathcal{V} \times \mathbb{R}^m \times \ldots \times \mathbb{R}^m \to \mathbb{R}^\ell, \quad \mathcal{D} \subset \mathbb{R}^n, \mathcal{V} \subset \mathbb{R}^m, \mathcal{P} \subset \mathbb{R}^p$$

$$P\left(\theta, x, u, \partial_{x_1} u, \ldots \partial_{x_n} u, \ldots, \partial_{x_1}^{\alpha_1} \cdots \partial_{x_n}^{\alpha_n} u\right) = 0.$$

For most of the problems at least one of the conditions is fulfilled:

$$u(x) = u_0(x), \quad x \in \mathcal{D}_\theta \times \{T_0\}, \quad \mathcal{D}_\theta \subset \mathbb{R}^{n-1}$$

$$u(x) = u_b(x), \quad x \in \partial\mathcal{D}_\theta \times \mathcal{T}, \quad \mathcal{T} \subset \mathbb{R}_{\geq 0}$$

Formulation as operator learning:

$$\mathcal{S} : \mathcal{P} \times \mathcal{D} \times \mathbb{R}^m \times \ldots \times \mathbb{R}^m \times \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathcal{V}$$

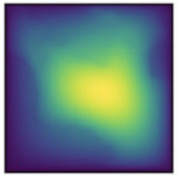
$$(\theta, x, \partial_{x_1} u, \ldots \partial_{x_n} u, \ldots, \partial_{x_1}^{\alpha_1} \cdots \partial_{x_n}^{\alpha_n} u, u_b, u_0) \mapsto u$$

Two examples from before

- Darcy flow PDE:

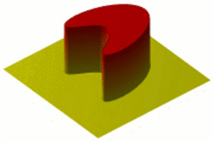
$$-\nabla \cdot \big(a(x)\nabla u(x)\big) = f(x) \qquad S : a \longmapsto u$$

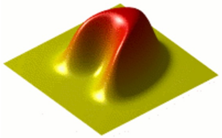

Input: a(x)


Output: u(x)

- Heat equation:

$$u_t = \Delta u \qquad S : u_0 \longmapsto u(t)$$





A discretization is needed

Addressing Limitations with Neural Operators

- Idea: We want to learn the solution operator directly from data
- Approximate the solution operator with a neural network -> Neural Operator
- The two main examples are the FNO (Fourier Neural Operator) and the GNO (Graph Neural Operator)

$\mathcal{S} : \mathcal{D} \to \mathcal{V}$

$\mathcal{D}_L$ be an L-point discretisation of $\mathcal{D}$.

$\tilde{\mathcal{S}}_\mu$ the approximate solution operator.

$$R_K\left(\mathcal{S}, \tilde{\mathcal{S}}_\mu, \mathcal{D}_L\right) = \sup_{\theta \in K} \left\| \hat{\mathcal{S}}\left(\mathcal{D}_L, \theta|_{\mathcal{D}}\right) - \mathcal{S}(\theta) \right\|_{\mathcal{V}}$$

$$\lim_{N \to \infty} R_K\left(\mathcal{S}(\cdot, \theta), \tilde{\mathcal{S}}^{(N)}(\cdot, \cdot, \theta), \mathcal{D}^{(N)}\right) = 0$$

Discretization invariance

Fourier Neural Operator

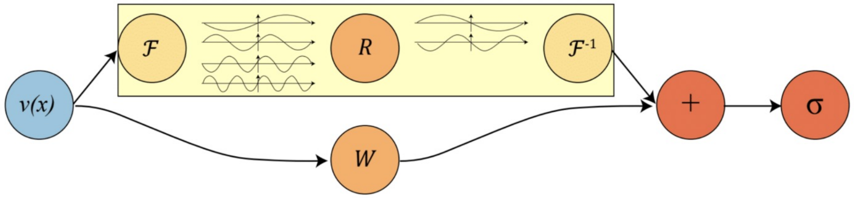


Figure 1 – layer of the Fourier neural operator

$$\tilde{\mathcal{S}}_\mu := \mathcal{Q} \circ (W_L + \mathcal{K}_L + b_L) \circ \cdots \circ \sigma (W_1 + \mathcal{K}_1 + b_1) \circ \mathcal{P}$$

Integral Kernel Operators

$$\mathcal{K}_l : \{\mathcal{D} \to \mathbb{R}^{d_l}\} \to \{\mathcal{U} \to \mathbb{R}^{d_{l+1}}\} \;:\; \mathcal{F}^{-1}\Big(R_\phi \cdot (\mathcal{F} v_t)\Big)(x) \qquad \forall x \in D$$

Depending on discretization and uniform grids

Graph Neural Operator [1]

$$v_{t+1}(x) = \sigma\left(W v_t(x) + \int_D \kappa_\phi(x, y, a(x), a(y)) v_t(y)\, \nu_x(dy)\right)$$

$$v_{t+1}(x) = W v_t(x) + \sum_{y \in N(x)} \kappa_\phi\big(e(x,y)\big) v_t(y)$$

[1] Li et al. [2020] https://arxiv.org/abs/2003.03485

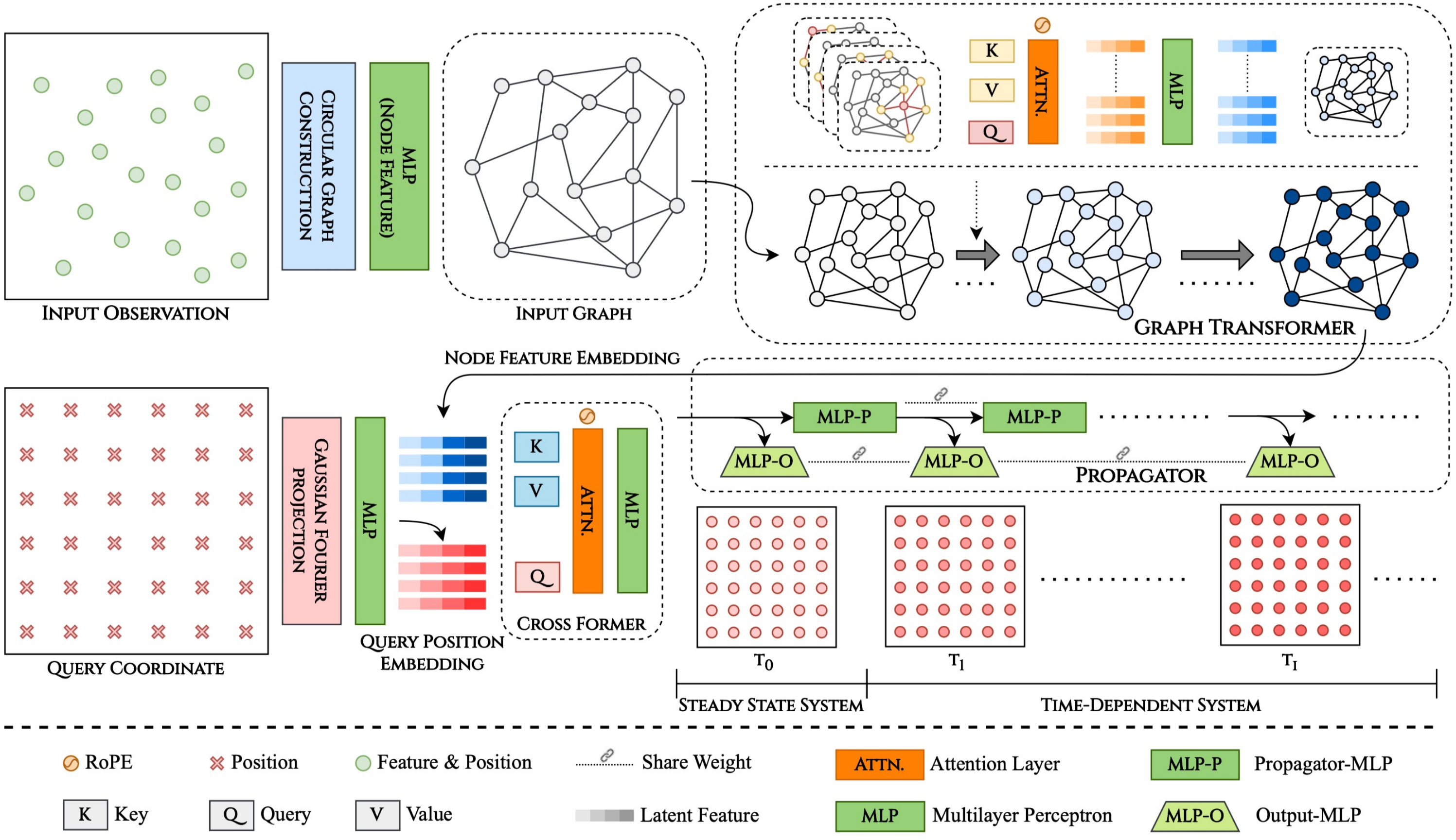
The Role of Graph Transformer Architectures

- Can we make a neural operator, which is adaptable to any grid realizations and can work with a multi-modality framework?
- Can we break down the efficiency of the neural operator and enhance the information in the data (data limitation)?
- Can we create a flexible approach with easy extendibility?
- A Graph Transformer indeed represents a neural operator

Outlook for the new architecture

- Include information for properties of the PDE directly in the architecture using multihead attention (with two output channels)
  - boundary conditions
  - Initial value conditions
- Use the flexibility and efficiency of graph transformer in the architecture
- Different graph structures are used to increase details resolutions
- Propagate the solution in the latent space and transform it afterwards into a graph structure

Introducing HAMLET



Results

Table 1. Numerical comparison of our approaches vs. existing techniques. The values reflect the nRMSE. The best-performing results are highlighted in green.

Dataset Setting		Normalised RMSE (nRMSE)							
Dataset	Param.	GNOT	U-Net	FNO	DeepONet	OFormer	GeoFNO	MAgNet	HAMLET
Darcy Flow	$\beta = 0.01$	–	1.10E+00	2.50E+00	3.25E-01	3.04E-01	1.03E+00	7.71E-02	**3.11E-01**
Darcy Flow	$\beta = 0.1$	–	1.80E-01	2.20E-01	1.67E-01	1.15E-01	3.13E-01	8.10E-02	**7.65E-02**
Darcy Flow	$\beta = 1.0$	–	3.30E-02	6.40E-02	5.12E-02	2.05E-02	6.34E-02	1.03E-01	**1.40E-02**
Darcy Flow	$\beta = 10.0$	–	8.20E-03	1.20E-02	3.97E-02	6.34E-03	2.51E-02	1.62E-01	**4.77E-03**
Darcy Flow	$\beta = 100.0$	–	4.40E-03	6.40E-03	3.64E-02	4.19E-03	2.04E-02	1.95E-01	**3.46E-03**
Shallow Water	–	4.16E-03	8.30E-02	4.40E-03	2.35E-03	2.90E-03	6.70E-03	–	**2.04E-03**
Diffusion Reaction	–	8.22E-01	8.40E-01	1.20E-01	8.42E-01	3.28E+00	7.72E+00	–	**9.02E-02**

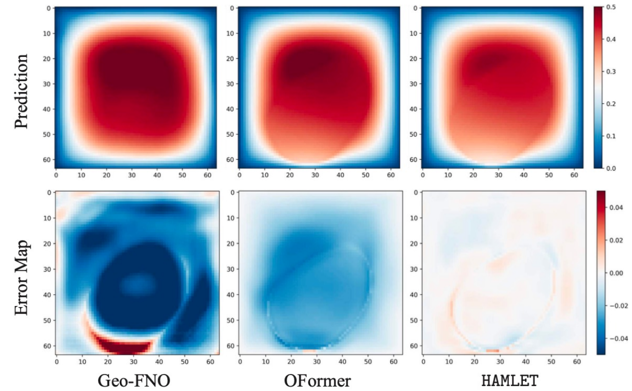
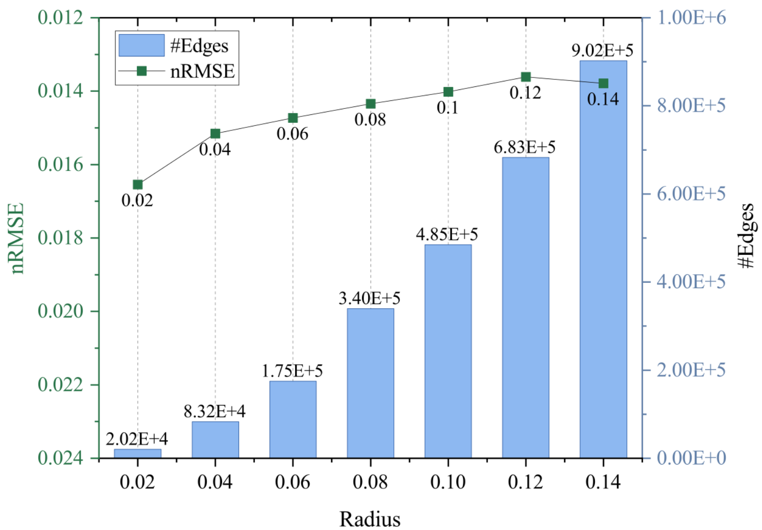


Figure 4. Predictions and corresponding error maps for Geo-FNO, OFormer, and HAMLET models using Darcy Flow ($\beta = 1.0$).

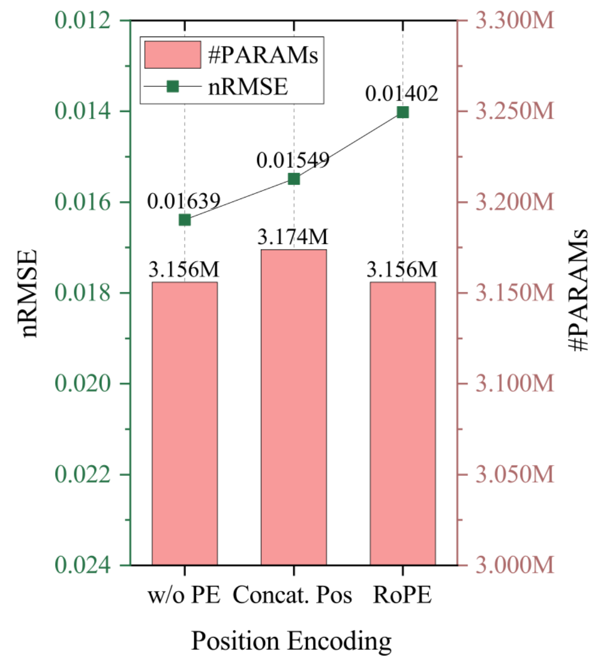
Scalability and Robustness



(A) Ablation Study
Radius for Circular Graph Construction

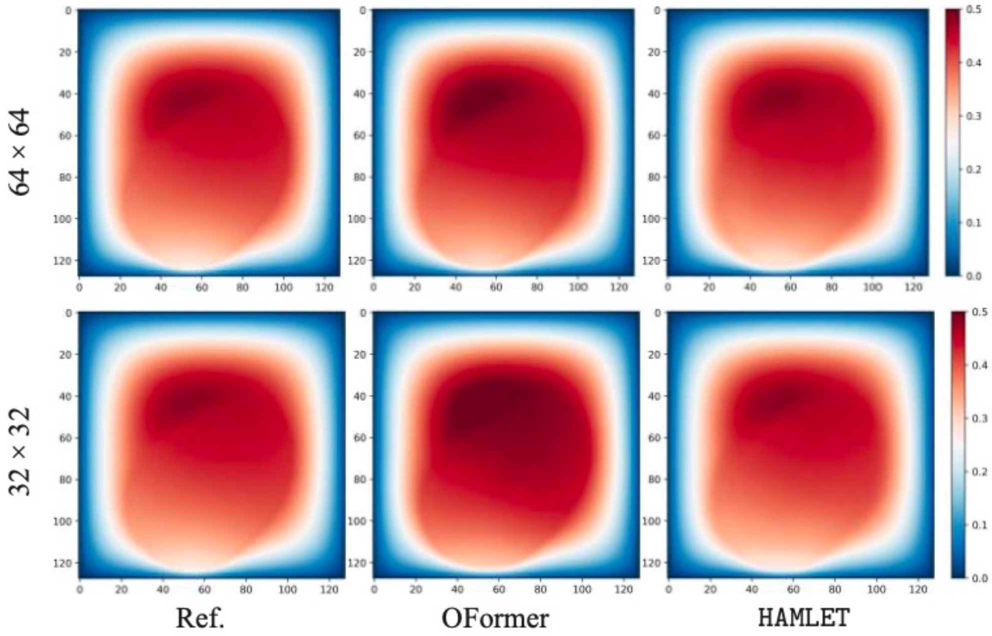
Table 2. Ablation studies for graph constructions on Darcy Flow 2D ($\beta = 1.0$) in 64×64 grid.

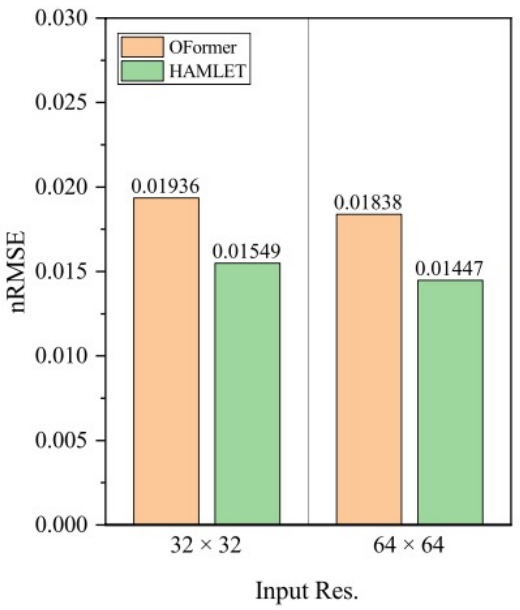
	KNN				Circular
	$k = 21$	$k = 51$	$k = 101$	$k = 151$	
nRMSE	0.02018	0.01996	0.02054	0.02094	**0.01402**



(B) Ablation Study
Position Encoding

Impact of Data Size on Model Performance





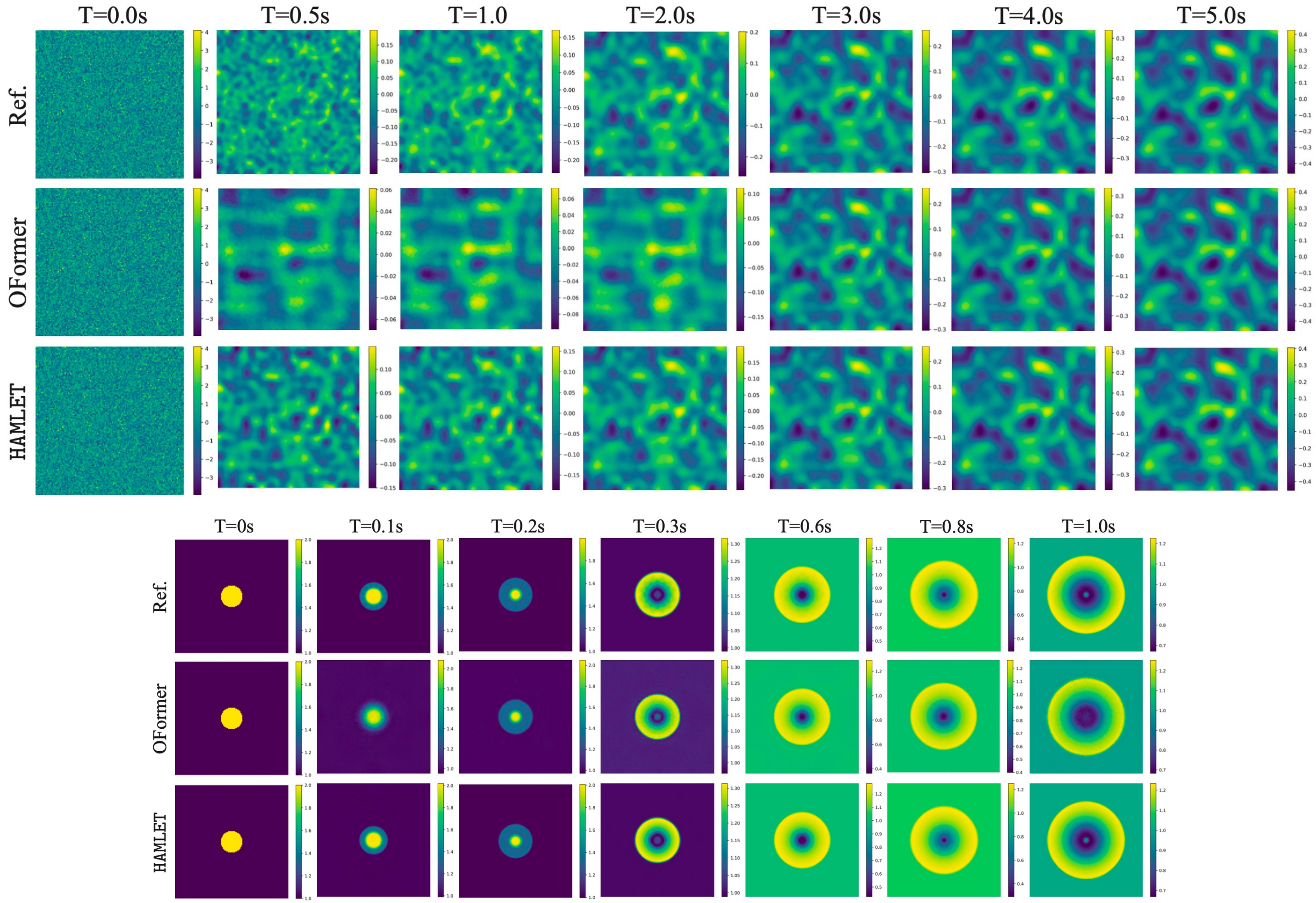
Darcy Flow (β = 1.0), nRMSE

#Training Data	9K	5K	2K	1K
OFormer	2.048E-02	2.093E-02	2.674E-02	3.321E-02
HAMLET	1.402E-02	1.642E-02	2.211E-02	2.779E-02

Shallow Water, nRMSE

#Training Data	900	500	200	100
OFormer	2.900E-03	1.190E-02	2.310E-02	2.910E-02
HAMLET	2.044E-03	2.320E-03	3.255E-03	4.746E-03

Evaluation Protocol in time dependent problems



Conclusion and Future Directions

- Offers a flexible and robust solution adaptable to various geometries and conditions. employs modular input encoders and establishes new benchmarks in scenarios with limited data availability.

Limitation:

- Graph construction time is a limitation, common in graph-based approaches. An increased number of parameters and lack of interpretability
- Not adaptable to every problem

Future Work:

- Integration of Lie-symmetry preservation and augmentation.
- Extension to handle higher-dimensional PDEs, including 3D problems.
- Dedicated exploration to refine the model architecture.
- Multiple parameter meshes and estimations

Call to Action

- Establish and promote the use of standardized benchmarks for evaluating neural operators of PDEs. Provide a common ground for comparison, ensuring consistent evaluation metrics.

- Rethink current evaluation methods to include a broader range of problem settings and conditions. Standardize the reporting of results to include crucial error analysis. Compare to existing methods like FEM, FDM, spectral methods…

- Encourage the publication and discussion of frameworks and approaches that did not succeed.

- Investigate the integration of physical laws and symmetries into neural operator frameworks. Explore hybrid models that combine traditional numerical methods with neural operators.

- Aim for the development of general-purpose neural operators capable of solving a wide variety of PDEs. Less parameter tuning and adaptability for datasets.

Thank you very much for your attention

References

Nikola B. Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. CoRR, abs/2108.08481, 2021. URL https://arxiv.org/abs/2108.08481.

Li, Z., Meidani, K., and Farimani, A. B. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023.