GITO: Graph-Informed Transformer Operator for Learning Complex Partial Differential Equations

Milad Ramezankhani[*†]
Phi Labs, Quantiphi,
Marlborough, MA 01752, USA.
milad.ramezankhani@quantiphi.com

Janak M. Patel[*†]
Phi Labs, Quantiphi,
Marlborough, MA 01752, USA.
janak.patel@quantiphi.com

Anirudh Deodhar
Phi Labs, Quantiphi,
Marlborough, MA 01752, USA.
anirudh.deodhar@quantiphi.com

Dagnachew Birru
Phi Labs, Quantiphi,
Marlborough, MA 01752, USA.
dagnachew.birru@quantiphi.com

Abstract

We present a novel graph-informed transformer operator (GITO) architecture designed for learning complex partial differential equation systems defined on irregular geometries and non-uniform meshes. GITO consists of two main modules: a hybrid graph transformer (HGT) and a transformer neural operator (TNO). HGT leverages a graph neural network (GNN) to encode local spatial relationships, an intermediate graph-transformer layer operating on selected supernodes to capture mesoscopic relational dependencies, and a global transformer to model long-range interactions. A self-attention fusion layer then integrates local, intermediate, and global signals to produce richer, more expressive graph embeddings. The TNO module employs linear-complexity cross-attention and self-attention layers to map encoded input functions to predictions at arbitrary query locations, ensuring discretization invariance and enabling zero-shot super-resolution across any mesh. Empirical results on benchmark PDE tasks demonstrate that GITO outperforms existing transformer-based neural operators, paving the way for efficient, mesh-agnostic surrogate solvers in engineering applications.

1 Introduction

Solving partial differential equations (PDEs) underpins a vast array of phenomena in engineering and physical sciences, from fluid flow and heat transfer to fracture mechanics and structural deformation. Traditional numerical methods offer rigorous error bounds and adaptable frameworks, but they often incur substantial computational costs when applied to high-dimensional, nonlinear, or time-dependent problems [Olver et al., 2014]. This computational burden can become prohibitive in real-time control and optimization tasks, motivating the search for surrogate models that deliver rapid yet accurate PDE solutions.

In recent years, deep neural network–based surrogates have emerged as a powerful alternative, demonstrating orders-of-magnitude speedups over classical solvers while maintaining competitive accuracy [Zhu and Zabaras, 2018, Bhatnagar et al., 2019]. These data-driven models can learn solution operators from precomputed simulation data, enabling instantaneous inference once trained. Physics-informed neural networks (PINNs) [Raissi et al., 2019] introduced a paradigm shift by embedding the governing PDE residual directly into the loss function, thus bypassing the need for labeled solution data. While PINNs have been successfully applied to a wide range of forward and

[*]Equal Contribution.
[†]Corresponding Authors: milad.ramezankhani@quantiphi.com, janak.patel@quantiphi.com

2nd Conference on Applied AI and Scientific Machine Learning (CASML 2025).

inverse problems, each new setting of initial conditions, boundary values, or forcing terms requires retraining from scratch, constraining their applicability to a single PDE configuration [Chen and Koohy, 2024, Ramezankhani and Milani, 2024].

Neural operators extend the concept of surrogate modeling by directly mapping infinite-dimensional input-output spaces, effectively learning solution operators for a family of PDEs. Foundational architectures such as DeepONet [Lu et al., 2021] and the Fourier Neural Operator (FNO) [Li et al., 2020a] show that a single model can generalize across varying PDE conditions and enable zero-shot super-resolution. Inspired by the success of the transformer architecture [Vaswani et al., 2017] in natural language processing and computer vision, recent works explored attention-based surrogate models to simulate physical systems. Typically, these models are trained on function samples defined over fixed discretization grids, which limits their ability to generalize across varying meshes [Cao, 2021, Han et al., 2022]. To address this, a new class of transformer-based neural operators has emerged, which enables super-resolution and discretization-invariant query of the output function [Li et al., 2022a, Hao et al., 2023, Alkin et al., 2024]. They employ cross-attention to aggregate input features and predict outputs at arbitrary spatial/temporal coordinates, regardless of the underlying input grid.

Despite these early successes, significant challenges remain in scaling transformer-based operators to realistic engineering applications. In particular, modeling systems with irregular geometries and non-uniform meshes demands more powerful mechanisms to capture complex interactions and dynamics among spatial nodes. To address these challenges, we propose a novel graph-informed transformer operator (GITO) architecture tailored for mesh-agnostic operator learning on arbitrary domains (Figure 1). Our framework comprises two core modules: a hybrid graph transformer (HGT) and a transformer neural operator (TNO). HGT combines graph neural networks (GNNs) to model intricate local spatial relations, an intermediate graph-transformer layer on selected supernodes to learn mesoscopic relational dependencies, and global transformer layers to capture long-range interactions. A dedicated fusion layer then applies self-attention to merge the local, intermediate, and global embeddings, producing richly expressive relational representations. Building on these embeddings, TNO applies cross-attention for discretization-invariant querying of the output domain, followed by self-attention to capture dependencies among enriched query embeddings. Our main contributions are: 1) a novel graph-transformer-based neural operator architecture that seamlessly integrates local and global feature learning on irregular meshes and geometries, and 2) superior performance on benchmark PDE tasks, outperforming existing transformer-based neural operators.

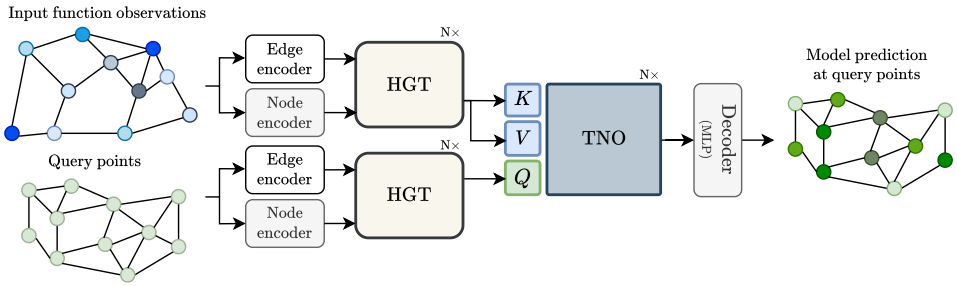

Figure 1: Overall architecture of GITO. The input function and query points are first converted into graph representations and encoded via edge and node encoders. These encoded graphs are then processed by the hybrid graph transformer (HGT) module to learn informative relational features. The output representations from the HGT are used as key/value and query inputs to the transformer neural operator (TNO) module, which integrates contextual information from input function observations to enrich the query representations. Finally, an MLP decoder maps the query embeddings to real spatial coordinates.

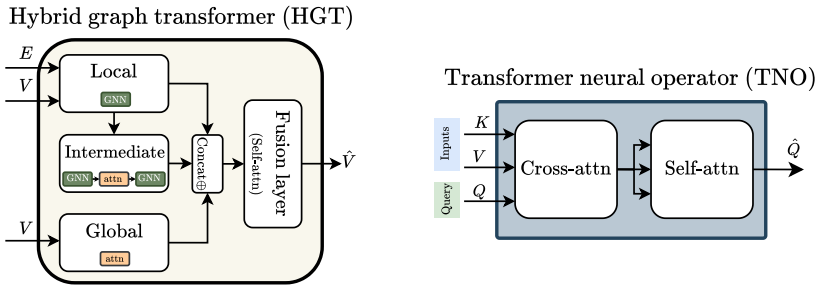

Figure 2: (Top) The hybrid graph transformer (HGT) module consists of a GNN layer, an intermediate graph-transformer layer, a self-attention global layer, and a self-attention fusion layer that jointly learn graph-based representations. (Bottom) The transformer neural operator (TNO) module employs cross-attention and self-attention mechanisms to integrate and process representations of input functions and query points. For clarity, standard components such as layer normalization, residual connections, and feed-forward networks are omitted.

2 Methodology

2.1 Graph construction and feature encoding

We represent both the input function and query points as separate graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V}$ corresponds to a spatial location (e.g., a mesh cell or a query point) and each edge $(i, j) \in \mathcal{E}$ connects node i either to its k nearest neighbors or to nodes within a specified Euclidean radius. The value of k and radius are considered as model hyperparameters. Each node feature vector V_i includes the spatial coordinates $\mathbf{x}_i$. For nodes corresponding to the input function, the observed field value $\mathbf{u}_i$ is concatenated to the node features. Edge features $\mathbf{E}_{ij}$ comprise relative displacements $(\mathbf{x}_i - \mathbf{x}_j)$, Euclidean distances $|\mathbf{x}_i - \mathbf{x}_j|$, and, in case of input function graphs, differences in solution values between connected nodes $\mathbf{u}_i - \mathbf{u}_j$ [Brandstetter et al., 2022]. Both node and edge features are passed through dedicated MLP-based encoders to generate initial embeddings, which are then fed into the HGT layers for subsequent representation learning (Figure 1).

2.2 Hybrid graph transformer (HGT) module

Local and Global Graph Representations. Despite their strengths, GNNs suffer from fundamental limitations due to sparse message passing, notably over-smoothing [Oono and Suzuki, 2019] and over-squashing [Alon and Yahav, 2020]. Graph transformers (GTs) [Dwivedi and Bresson, 2020, Ying et al., 2021, Mialon et al., 2021] address these shortcomings by allowing nodes to attend to all others in the graph; however, they often overlook edge features, hindering accurate representation learning. Hybrid architectures such as GPS Graph [Rampášek et al., 2022] and Exphormer [Shirzad et al., 2023] combine GNN and transformer layers to overcome these challenges: the GNN component captures local interactions and integrates edge information, while the transformer module models long-range and global dependencies and mitigates over-smoothing and over-squashing. Following this paradigm, we employ a GNN layer (`Local`) alongside a linear self-attention module (`Global`) to learn the graph dynamics (Figure 2).

Intermediate Relational Learning. Aside from capturing immediate local neighborhoods and distant global contexts, PDE systems defined on complex, irregular meshes exhibit rich interactions at intermediate scales that are neither purely local nor fully global—making mid-level relational features crucial for accurate operator learning Fortunato et al. [2022], Lam et al. [2023], Wen et al. [2025]. By modeling these mesoscopic dependencies, a surrogate can propagate information across varying ranges and resolutions, effectively overcoming the bottlenecks inherent in single-scale GNNs. Building on these insights, our architecture also inserts an intermediate layer, a hybrid module combining GNN and graph transformer blocks, operating on a purpose-designed mid-level graph. This layer learns relational patterns at intermediate scales, seamlessly bridging local continuity with

global coherence before the final decoding stage. In particular, the intermediate layer `Intermediate` constructs a set of supernodes, a randomly chosen subset of the original graph's nodes, through which mesoscopic information propagation is performed (Figure 2). First, using a pre-specified radius, features flow from the full graph to each supernode via GNN message-passing, effectively downsampling the representation. Next, these aggregated supernode embeddings engage in inter-supernode communication through a graph-transformer module, leveraging self-attention to avoid information bottlenecks at the intermediate scale. Finally, the refined supernode representations are upsampled back to the original graph via a second GNN message-passing phase, and a concluding GNN layer ensures that this enriched information seamlessly permeates all nodes. This hybrid downsample–transform–upsample design enables our model to learn mid-level relational patterns along with capturing local details or global dependencies.

Multi-Scale Fusion Layer. In addition, we introduce a dedicated fusion layer, `Fusion`, which leverages multi–head self-attention to seamlessly weave together local, intermediate, and global signals. This joint aggregation allows the model to dynamically weight information from all three scales—neighborhood, supernode, and global—producing richer, more discriminative graph embeddings that capture complex relational patterns across the entire domain. (Figure 2). In the HGT module, node representations are updated by concatenating the outputs of the `Local`, `Global` and `Intermediate` layers, followed by processing the combined embedding through the `Fusion` layer:

$$V_G, E = \texttt{Local}(V, E) \tag{1}$$

$$V_T = \texttt{Global}(V) \tag{2}$$

$$V_I = \texttt{Intermediate}(V_G, E) \tag{3}$$

$$\hat{V} = \texttt{Fusion}\left(V_G \oplus V_T \oplus V_I\right). \tag{4}$$

The modularity of the hybrid graph transformer enables seamless integration of diverse GNN architectures and transformer modules, allowing the model to be tailored to specific application requirements.

2.3 Transformer neural operator (TNO) module

To empower zero-shot super-resolution and fully decouple input and output representations, we integrate a cross-attention layer capable of querying the output domain at arbitrary spatial locations (Figure 2). This design parallels the branch and trunk networks in the DeepONet [Li et al., 2020b], seamlessly fusing input function embeddings with output queries to achieve discretization-invariant evaluation, regardless of the underlying input mesh [Li et al., 2022a]. The cross-attention layer takes as input the query embeddings and the input function representations generated by the HGT modules. The cross-attention enriches the query embeddings with the information from the input functions. A subsequent self-attention module then captures interactions and dependencies among the enriched query points. Finally, an MLP decoder translates the resulting embeddings into the target physical output values.

2.4 Model implementation details

To efficiently learn operators for large-scale physical systems with numerous input and query locations, we adopt the linear-complexity attention mechanism proposed by Hao et al. [2023]. Similar to Fourier and Galerkin attention mechanisms [Cao, 2021], this approach can capture complex dynamics while avoiding the quadratic computational cost of softmax-based attention. We adopt a "Norm-Attn-MLP-Norm" with residual connections for all attention layers. To handle cases with multiple input functions, we use a dedicated encoder for each input function. These encoded representations are then processed by the cross-attention module in TNO, specifically designed to handle multiple key-value (K/V) combinations, enabling efficient interaction across heterogeneous inputs. We incorporate a mixture of experts module following each attention mechanism. The gating network assigns weights to the experts based on the spatial location of the query points, effectively promoting a form of *soft* domain decomposition, which has been shown to enhance the learning of physical systems in prior work [Chalapathi et al., 2024, Hao et al., 2023]. In the HGT module, we use the Graph Attention Network (GATv2) [Brody et al., 2021] as the GNN layer and apply the same linear-complexity attention mechanism as in TNO for the global, intermediate and fusion layers. The graph construction strategies are detailed in ablation studies section. In this work, we choose to use the HGT module only for query points for learning more expressive relational features. For the intermediate layer, We

select approximately 10% of the nodes as supernodes for all case studies and use the same k or radius as the original graph for supernodes upsampling and downsampling.

3 Experimental results

3.1 Datasets and baseline models

Datasets. To evaluate GITO's scalability on complex geometries, we test it on four challenging datasets: Navier-Stokes [Hao et al., 2023], Heat Conduction [Hao et al., 2023], Airfoil [Li et al., 2022b] and Elasticity [Li et al., 2022b]. A brief overview of the datasets is provided below:

- **2D steady-state Navier-Stokes (NS)**: This dataset involves steady 2D fluid flow governed by Navier-Stokes equations in a rectangular domain with varying cavity positions . The goal is to predict velocity components u, v, and pressure p from the input mesh.
- **Multilayer 2D Heat Conduction (Heat)**: This dataset models heat conduction in composite media with multiple boundary shapes and spatially varying boundary conditions . The task is to predict temperature T from multiple input functions.
- **Airfoil**: This dataset involves the Mach number M distribution over different airfoil shapes . The task is to predict M from the input mesh and the geometry of the airfoil.
- **Elasticity**: This dataset simulates solid mechanics governed by elastokinetics equations . The domain is a unit square containing an irregular cavity, and the objective is to predict the stress field given the input mesh.

Baseline Models. We benchmark our model against both conventional neural operator architectures, including FNO [Li et al., 2020a], Geo-FNO [Li et al., 2022b], and MIONet [Jin et al., 2022], as well as recently developed transformer-based operators, namely, GNOT [Hao et al., 2023], Galerkin Transformer (GKT) [Cao, 2021], and OFormer [Li et al., 2022a]. To ensure a fair comparison, we re-implement GNOT and evaluate it under the same experimental settings as our model, using a comparable or slightly larger number of parameters. For all the datasets, we directly report the baseline performances from Hao et al. [2023].

3.2 Results

Table 1 summarizes the mean relative L^2 error across all test datasets for the compared models, with lower values indicating higher prediction accuracy. The relative L^2 error is defined as $\frac{|\hat{y}-y|_2}{|y|_2}$, where $\hat{y}$ is the model prediction and y is the ground truth. This metric provides a normalized measure of prediction accuracy that is consistent across datasets with varying magnitudes. GITO consistently achieves the lowest error across all datasets and variables, outperforming existing neural operator baselines.

Dataset	Subset	MIONet	FNO	GKT	Geo-FNO	OFormer	GNOT	**GITO (Ours)**	Improvement
NS	u	2.74e-2	6.56e-2	1.52e-2	1.41e-2	2.33e-2	<u>1.05e-2</u>	**8.19e-3**	22 %
	v	5.51e-2	1.15e-1	3.15e-2	2.98e-2	4.83e-2	<u>2.33e-2</u>	**2.02e-2**	13.3 %
	p	2.74e-2	<u>1.11e-2</u>	1.59e-2	1.62e-2	2.43e-2	1.23e-2	**1.07e-2**	3.6 %
Heat	T	1.74e-1	–	–	–	–	<u>5.42e-2</u>	**3.79e-2**	30.07 %
Airfoil	M	–	–	<u>1.18e-2</u>	1.38e-2	1.83e-2	1.80e-2	**6.29e-3**	46.7 %
Elasticity	σ	9.65e-2	5.08e-2	2.01e-2	2.20e-2	1.83e-2	<u>9.04e-3</u>	**8.38e-3**	7.3%

Table 1: Comparison of GITO with existing operator learning methods on the NS, Heat, Airfoil and Elasticity datasets. The metric used for this comparison is relative L^2 error, with lower scores indicating better performance. The top **first** and <u>second</u> best results are highlighted. For all datasets, baseline results are taken directly from Hao et al. [2023] except GNOT. For a fair comparison, we trained a smaller GNOT model to match GITO's model size. Missing values in the table indicate that the respective models were unable to handle the specific challenges posed by the dataset, such as irregular geometries, multiscale patterns, or multiple input functions, as discussed in Hao et al. [2023].

4 Conclusion

In this work, we introduced GITO, the Graph-Informed Transformer Operator, a novel architecture that unifies graph neural networks with transformer attention to learn mesh-agnostic PDE solution operators for arbitrary geometries. By combining hybrid message-passing at local, intermediate, and global scales, discretization-invariant cross-attention, and scalable linear-complexity attention mechanisms, GITO delivers zero-shot super-resolution and outperforms existing transformer-based operators across diverse benchmarks. These results underscore GITO's promise as an accurate and efficient surrogate model for complex engineering applications.

References

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64:525–545, 2019.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34: 24924–24940, 2021.

Nithin Chalapathi, Yiheng Du, and Aditi Krishnapriyan. Scaling physics-informed hard constraints with mixture-of-experts. *arXiv preprint arXiv:2402.13412*, 2024.

Yanlai Chen and Shawn Koohy. Gpt-pinn: Generative pre-trained physics-informed neural networks toward non-intrusive meta-learning of parametric pdes. *Finite Elements in Analysis and Design*, 228:104047, 2024.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter Battaglia. Multiscale meshgraphnets. *arXiv preprint arXiv:2210.00612*, 2022.

Xu Han, Han Gao, Tobias Pfaff, Jian-Xun Wang, and Li-Ping Liu. Predicting physics in mesh-reduced space with temporal attention. *arXiv preprint arXiv:2201.09113*, 2022.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, pages 12556–12569. PMLR, 2023.

Pengzhan Jin, Shuai Meng, and Lu Lu. Mionet: Learning multiple-input operators via tensor product. *arXiv preprint arXiv:2202.06137*, 2022.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *arXiv preprint arXiv:2207.05209*, 2022b.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3 (3):218–229, 2021.

Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

Peter J Olver et al. *Introduction to partial differential equations*, volume 1. Springer, 2014.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Milad Ramezankhani and Abbas S Milani. A sequential meta-transfer (smt) learning to combat complexities of physics-informed neural networks: Application to composites autoclave processing. *Composites Part B: Engineering*, 283:111597, 2024.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pages 31613–31632. PMLR, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shizheng Wen, Arsh Kumbhat, Levi Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. Geometry aware operator transformer as an efficient and accurate neural surrogate for pdes on arbitrary domains. *arXiv preprint arXiv:2505.18781*, 2025.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447, 2018.