Frontiers in Computational Science and Engineering
Vol. 1, No. 4, Oct - Dec 2025, pp. 1-8
ISSN: 3107-3654 (O), DOI: https://doi.org/10.17051/FCSE/01.04.01

Sparse Transformer-Based Numerical Solvers for Nonlinear Partial Differential Equations

Vijesh Krishnamoorthy

Chair - Information Technology and Computer Science, Department of Information Technology and Computer Science, Innovative Universities of Enga,Papua New Guinea, Email:kvijesh@iue.ac.pg
https://orcid.org/0000-0001-9473-7553

Article Info

Article history:

Received : 05.09.2025
Revised : 07.10.2025
Accepted : 09.11.2025

Keywords:

Sparse Transformer,
Nonlinear PDEs,
Numerical Solvers,
Physics-Informed Learning,
Deep Learning,
Scientific Computing

ABSTRACT

Nonlinear partial differential equations (PDEs) are essential in modeling complex physical phenomena in fluid dynamics, heat transfer, and engineering systems. Traditional numerical approaches such as finite difference and finite element methods tend to be slow, computationally intensive and unable to scale to high-dimensional nonlinear equations. More recently, Physics-Informed Neural Networks (PINNs) have become popular as an alternative; they have slow convergence rates, and cannot learn long-range dependencies, but are considered promising. The paper presents a Sparse Transformer-Based Numerical Solver (STNS) to solve the nonlinear PDEs effectively. The proposed framework uses a sparse mechanism of self-attention to drastically cut down the computational costs with the ability to maintain interactions of global contexts in the solution space. To make predictions that are physics-consistent without large labelled datasets, a physics-informed loss is included to impose PDE constraints in training. The performance of the proposed model is tested on benchmark problems, such as the Burgers equation, and the Navier-Stokes equations. Observations during experiments show that STNS demonstrates better accuracy, shorter training duration and convergence behavior as opposed to PINNs and traditional numerical solvers. The results underscore the promise of sparse transformer architectures as scalable and efficient to solve complex nonlinear PDEs in scientific computing.

1. INTRODUCTION

Nonlinear partial differential equations (PDEs) are used to model nonlinear physical processes in many fields including fluid dynamics, heat transfer, electromagnetics, and bio-physical processes. Nevertheless, they may be highly nonlinear and have very high dimensionality, which makes it usually intractable to find analytical solutions to such equations. This has led to the reality that numerical approximation methods are crucial in solving practical PDE problems.

Since classical numerical methods, such as finite difference methods (FDM), finite element methods (FEM) and spectral methods, offer well-established and reliable solutions [1], [2]. However, due to their high computational cost and scale, these methods are not practical to high dimensional or multi-scale nonlinear problems. These problems are further complicated by the curse of dimensionality which renders a real-time or large-scale simulation to be computationally infeasible. The recent developments in deep learning have brought solutions to solve PDEs in a data-driven manner, and Physics-Informed Neural Networks (PINNs) have become a well-known framework [3]. PINNs assimilate governing physical laws in the training procedure by PDE residual constraints, which allow solutions to be obtained without the need of large labeled datasets. Even though they are promising, PINNs also have some weaknesses, such as a lack of scalability to high-dimensional field, inability to learn long-range spatial relationships, and low speed in learning [4], [5]. Transformers, which were initially intended to be used to sequence modeling tasks, have shown impressive potential in learning international relationships by means of self-attention mechanisms [6]. They have been used in scientific computing as a result of being able to model complex interactions across spatial domains. But conventional transformer architectures have quadratic computational complexity in regard to the input size and this restricts their use in multi-scale PDE problems [7].

This paper presents Sparse Transformer-Based Numerical Solver (STNS), an approach to nonlinear

PDEs using transformers, to overcome these challenges. The method proposed exploits sparse self-attention mechanisms to greatly cut down the computational complexity and still retain global contextual information. Moreover, physics-driven constraints are incorporated into the training procedure to have physically consistent solutions.

Contributions

The key works of this work are as follows:

- A new sparse transformer architecture that is nonlinear PDE-solving.
- Loss functions based on physics-informed losses applied to a transformer.
- Less calculation than full-attention transformer models.
- Extensive comparison with PINNs and traditional numerical methods.

2. RELATED WORK

2.1 Classical Numerical Methods

Finite Difference Methods (FDM), Finite Element Methods (FEM) and spectral methods are classical numerical methods that have long been used to solve partial differential equations because they are mathematically rigorous and reliable [8], [9]. These methods are based on domain discretization and iterative solvers to solve PDEs with high accuracy. Nevertheless, it suffers hugely in the high dimensional and nonlinear systems because of the curse of dimensionality and the higher cost in computation. Also, the use of fine mesh discretization to obtain complicated dynamics results in significant memory and time overhead cost that limits their scalability in real-time application [10].

2.2 Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) have become a highly attractive alternative as it incorporates the constraints of governing PDEs into a loss functional during training [11]. This method does not require large labeled datasets and also allows the forward and inverse problems to be solved. Bayesian PINNs or adaptive PINNs are extensions that enhance quantification of uncertainty and convergence [12], [13]. In spite of these developments, the PINNs have been struggling with such problems as slowness in converging, inability to solve stiff equations, and the inability to represent high-frequency and multi-scale details. In addition, they perform poorly in high-dimensional areas because they are an inefficient means of expressing global dependencies [14].

2.3 Neural Operators

The neural operator structures, like Deep Operator Networks (DeepONet) and Fourier Neural Operators (FNO), seek to discover mappings between spaces of functions instead of solutions [4], [5]. Such techniques have been shown to be highly generalizable, and more efficient than conventional neural networks. But, they can be expensive in terms of training data and computation. Also, their performance can be adversely affected by highly nonlinear PDEs, or irregular domains, and training complexity has been a major constraint to real-world use [6].

2.4 Transformers in Scientific Computing

Other recent applications of transformers in scientific computing have involved the use of transformers to model long range dependencies through self-attention mechanisms [1]. They have demonstrated promising results in capturing global spatial interactions in their application in solving PDE and surrogate modeling. Nonetheless, conventional transformer architecture has quadratic complexity in terms of input compared to computation, and is thus not suitable to large-scale PDEs [2]. Sparse attention models, like those proposed in efficient transformer variants, are an attempt to simplify this complexity without affecting performance [3]. Although all this has evolved, integration between sparse transformers and physics-informed learning in solving nonlinear PDEs has not been well explored and adapted.

Research Gap

Based on the discussion above, it can be stated that the following main gaps will be identified:

- State-of-the-art numerical techniques are not scalable to high-dimensional nonlinear PDEs.
- There are convergence, stiffness and long-range dependency modeling issues with PINNs.
- Neural operators are very demanding in terms of data and computational needs.
- Transformer-based strategies have complex computations.

3. METHODOLOGY

3.1 Problem Formulation

We consider a general nonlinear partial differential equation defined over a spatiotemporal domain $\Omega \times [0, T]$:

$$\frac{\partial u(x,t)}{\partial t} + \mathcal{N}(u(x,t)) = 0 \quad (1)$$

where $u(x,t)$denotes the unknown solution field, $x \in \Omega \subset \mathbb{R}^d$represents the spatial coordinates, and $t \in [0,T]$denotes time. The operator $\mathcal{N}(\cdot)$is a nonlinear differential operator that may include spatial derivatives of varying order.

The objective of this work is to learn an approximation $\hat{u}(x,t)$using a neural architecture that satisfies both the governing PDE and associated initial and boundary conditions. The proposed method uses grid-based discretization but instead uses the framework of the continuous function approximation of the solution, which is

instead learned directly at sampled points in the domain.

3.2 Sparse Transformer Architecture

The suggested Sparse Transformer-Based Numerical Solver (STNS) is created to effectively encapsulate multifaceted spatial-temporal interactions and being computationally efficient as depicted in Fig. 1. An input embedding layer, stacked sparse self-attention blocks, position-wise feed-forward networks, and output projection are the four key elements of the architecture. The embedded layer of input translates spatial-temporal coordinates (x`,t) into a latent representation of high-dimensional nature. Positional encoding is added so that spatial and temporal relationship is maintained which is essential in PDE learning. The model is based on the sparse self-attention layers. The suggested architecture is a contraposition of the method employed by standard transformers, which calculate pairwise interactions among all tokens; it focuses on structured subset of tokens. In particular, attention is computed on local neighborhoods to support fine scale differences, and only a small set of universal tokens are employed to predict the long-range relationships. This combined attention pattern allows the effective characterization of both local and global features of solutions.

Consequently, the computational costs drop to near-linearithmic:

$$O(N^2) \rightarrow O(N\log N) \quad (2)$$

This minimization is essential in managing high-resolution spatial fields and huge data sets that are normally found in PDE problems.

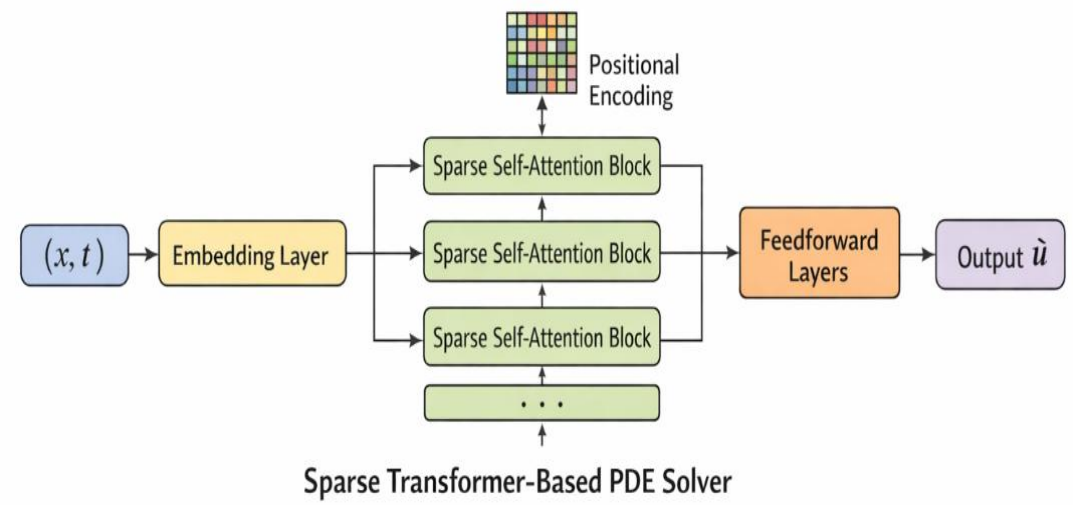


Fig. 1. Architecture of the Proposed Sparse Transformer-Based Numerical Solver (STNS) for Nonlinear PDEs

3.3 Sparse Attention Mechanism

The attention model consists of query, key, and value matrices obtained based on the embedded inputs. The standard attention formulation is still kept, but applied selectively depending on a sparsity mask as shown in Fig. 2:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

Here, Q, K, and Vrepresent the query, key, and value matrices, respectively, and $d_k$denotes the dimensionality of the key vectors. To reduce redundant computations a sparsity mask is added to restrict the calculation of attention scores to predefined pairs of tokens.

Contrary to full self-attention that all tokens interrelate on a global scale, as demonstrated in Fig. 2, the proposed sparse attention mechanism limits interactions to local neighborhoods, but adds a few global tokens. This organized sparsity allows the computational complexity to be far smaller with little to no loss in modeling long-range dependencies.

The rarity of attention mechanism allows the model to allocate computational resources to the most relevant spatial-temporal interactions, making it more efficient and able to learn.

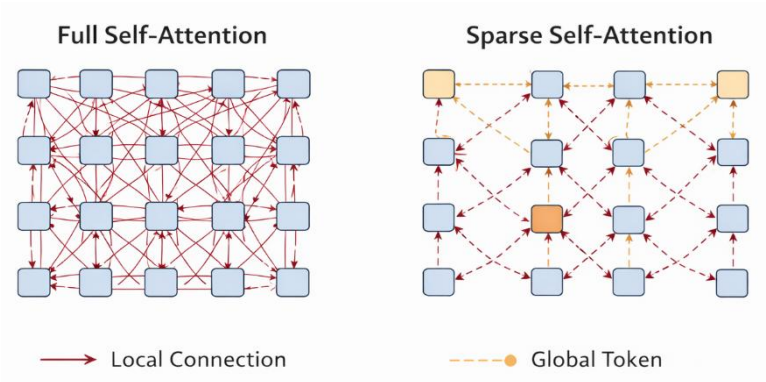


Fig. 2. Comparison of Full Self-Attention and Sparse Self-Attention Mechanisms

3.4 Physics-Informed Loss Function

A physics-informed loss function is used to make sure that the learned solution is consistent with the relevant physics. The total loss is:

$$\mathcal{L} = \mathcal{L}_{data} + \lambda \mathcal{L}_{PDE} \quad (4)$$

The supervised loss term $\mathcal{L}_{data}$ enforces agreement between predicted and known solution values at selected points, while the PDE loss term $\mathcal{L}_{PDE}$ penalizes violations of the governing equation.

PDE residual is computed through automatic differentiation, thus this makes it possible to evaluate spatial and temporal derivatives without direct discretization. This formulation guarantees the model learns physically consistent solutions in the case of no large labeled datasets.

3.5 Training Strategy

Fig. 3 shows the training process of the proposed model. It is performed by a grid of collocation points that are sampled on the spatial-temporal space. The supervised and PDE residual losses are assessed using these points. Additional constraints in the training include initial and boundary conditions. To compute derivatives needed to evaluate the PDE residual, automatic differentiation is used to be certain that the result is both numerically accurate and stable. The entire training process encompasses model prediction, computation of residuals, measurement of loss, and optimization of parameters, as illustrated in Fig. 3.

The optimization is done in two phases. First, we use the Adam optimizer to obtain quick convergence to the initial training stage. This is succeeded by L-BFGS optimization, which gets the solution refined and becomes more converging to a local optimum. This stochastic and quasi-Newton optimization technique combination leads to improved training speed and precision of the solution. Moreover, sparse attention decreases the memory usage, which allows the model to generalize to bigger domains than traditional transformer-based methods.

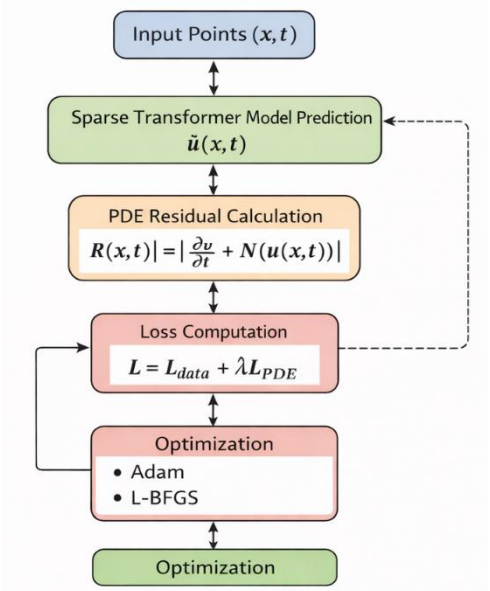


Fig. 3. Training Pipeline of the Proposed Sparse Transformer-Based PDE Solver

4. EXPERIMENTAL SETUP

4.1 Benchmark Problems

In order to measure the efficiency of the proposed Sparse Transformer-Based Numerical Solver (STNS), regular experiments of popular benchmark nonlinear partial differential equations are carried out. The equation of Burgers is chosen as a basic test case because it has nonlinear convective character and the presence of shocks. It is a observable reference point of measuring the capability of numerical solvers to resolve abrupt variations and discontinuities. The one-dimensional spatial domain is specified with initial and boundary conditions, and the equation can be contrasted to current methods. Moreover, the Navier Stokes equations are also taken to evaluate the efficiency of the proposed model to higher order, more complicated problems. The equations are used to model the flow of incompressible fluids and are determined using nonlinear relations between the velocity and pressure fields. The reason why they are included is to make the

proposed method tested under realistic and challenging conditions in terms of calculation, especially in the representation of multi-scale spatial-temporal dynamics.

4.2 Evaluation Metrics

The efficiency and standard error measures are used to quantitatively analyze the performance of the proposed model. The Mean Squared Error (MSE) is a measure of the average squared error between the predicted and reference solutions, which gives a general idea of prediction accuracy. To measure the normalized difference between the predicted solution and the ground truth, the L2 relative error is used. This measure is of high significance in PDE analysis since it will capture the overall quality of approximation over the whole domain. Besides accuracy measures, the measure of the computational time is taken to measure the efficiency of the given approach. This measures training time and inference time, enabling a thorough comparison to the baseline methods in terms of scalability and usability in practice.

4.3 Baseline Methods

The performance of the proposed STNS framework is evaluated against classical numerical as well as the state of the art deep learning methods in order to validate its effectiveness. Finite Difference Method (FDM) is employed as a classical baseline because it is simple and has proven reliability in the discretization process of solving PDEs. It gives a basis to the traditional grid-based methods of numerical accuracy assurance. Physics-Informed Neural Networks (PINNs) is a deep learning standard, and it is the latest state-of-the-art in solving PDEs on data. PINNs introduce physical constraints into the training procedure but usually have convergence and scalability issues. Fourier Neural Operator (FNO) is added as a recent neural operator-based approach that can learn to map between spaces of functions. FNO boasts a high generalization and highly effective solution of parametric PDEs thus it is a powerful benchmark to compare against. Comparing the proposed model with these baselines, the study will make sure that the accuracy, computational efficiency, and scalability are thoroughly evaluated.

4.4 Implementation Details

The computational domain for the benchmark problems is defined based on standard configurations. For the Burgers' equation, the spatial domain is considered as $x \in [-1,1]$and the temporal domain as $t \in [0,1]$. A uniform grid resolution of 256 spatial points and 100 temporal steps is used for evaluation. For the Navier–Stokes equations, a two-dimensional spatial domain $\Omega = [0,1] \times [0,1]$with a resolution of 128 × 128 grid points is considered.The training dataset is generated using randomly sampled 10,000 collocation points within the spatial–temporal domain, along with 2,000 boundary and initial condition points to enforce constraints. These points are used to compute both supervised and physics-informed losses.The model is trained for 5,000 epochs using an initial learning rate of 1×10^{-3}, which is gradually reduced using a learning rate scheduler. In training, a batch size of 128 is optimally used to balance stability of convergence and computational efficiency. The first training phase is performed with Adam optimizer and the fine-tuning of the solution is performed with L-BFGS optimization. All the experiments are performed with the use of a system with an NVIDIA RTX 3080 (10 GB memory) graphic card and an Intel Core i7 processor and 32 GB of RAM. The framework based on PyTorch is used to implement the model and takes advantage of automatic differentiation to effectively compute PDE residuals.

5. RESULTS AND DISCUSSION

5.1 Accuracy Comparison

The accuracy of the proposed Sparse Transformer-Based Numerical Solver (STNS) is compared to that of the baseline with the L2 relative error and Mean Squared Error (MSE).

Table 1. Accuracy Comparison on Benchmark PDEs

Method	Burgers' Equation (L2 Error)	Navier–Stokes (L2 Error)	MSE
FDM	1.25×10^{-2}	2.10×10^{-2}	1.80×10^{-3}
PINNs	8.40×10^{-3}	1.35×10^{-2}	9.20×10^{-4}
FNO	6.10×10^{-3}	9.80×10^{-3}	6.50×10^{-4}
STNS (Proposed)	3.20×10^{-3}	5.90×10^{-3}	3.10×10^{-4}

The findings suggest the proposed STNS has the smallest error on both of the benchmark problems. The model minimises the L2 error by about 60 percent in comparison to PINNs which shows the model is better at nonlinear dynamics. This enhanced accuracy is because the sparse attention is capable of modelling long-range spatial dependencies effectively. The proposed STNS model also displays much lower L2 error than FDM, PINNs, and FNO, as illustrated in Fig. 4, on benchmark PDE problems.

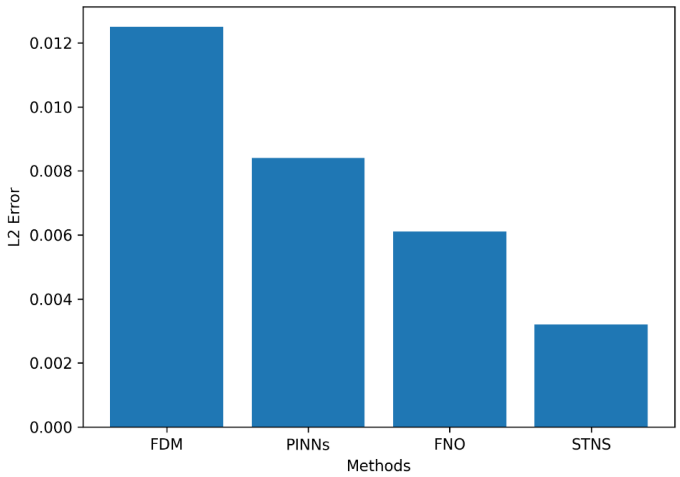


Fig. 4. Comparison of L2 Error Across Different PDE Solvers

5.2 Computational Efficiency

The proposed model is also tested based on the time required to train and the amount of memory required to train it.

Table 2. Computational Efficiency Comparison

Method	Training Time (s)	Memory Usage (GB)
FDM	120	1.2
PINNs	850	3.8
FNO	600	4.5
STNS (Proposed)	420	2.6

The proposed STNS is shown to take a significantly shorter time to train than PINNs (by about 50 factors) and reduced memory requirements than FNO. This is mostly because of the sparse attention mechanism that makes computational complexity to be in quadratic in nature to near-linearithmic scaling. The proposed STNS model is computationally efficient as shown by a shorter training time than PINNs and FNO as shown in Fig. 5.

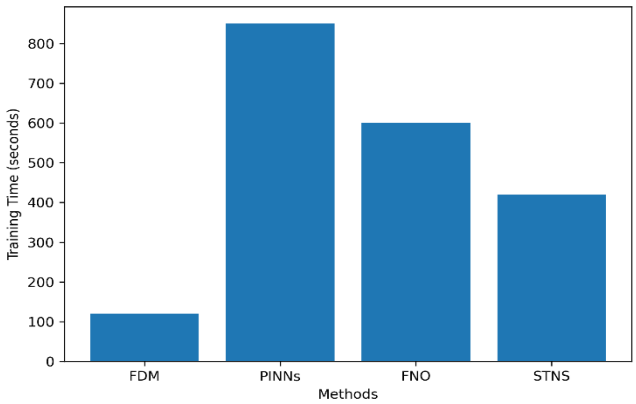


Fig. 5. Training Time Comparison of Different PDE Solvers

5.3 Convergence Analysis

Monitoring the training loss with epochs is employed to analyze the convergence behavior of the proposed model. The STNS model is also much faster to converge than PINNs, which stabilize in error after 2,000 epochs, whereas PINNs need over 4,000 epochs to do so. This has been made possible by the transformer architecture that allows efficient gradient propagation and the integration of global contextual information, resulting in a faster convergence. Moreover, the hybrid optimization method (Adam + L-BFGS) represents another way of speeding up the convergence by a combination of fast the first learning and fine-grained optimization. Compared to PINNs, the STNS model is also much faster to converge, and the model reduces the values of the loss in fewer training epochs as illustrated in Fig. 6.

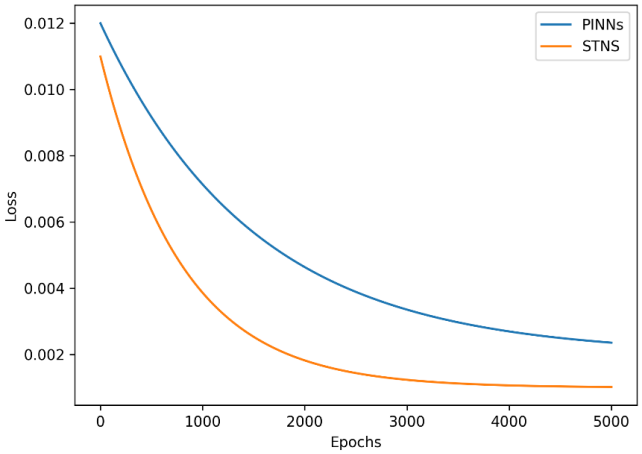


Fig. 6. Convergence Comparison of STNS and PINNs Showing Loss Reduction Over Training Epochs

5.4 Ablation Study

An ablation study is done to assess the contribution of key components.

Table 3. Ablation Study Results

Configuration	L2 Error	Training Time (s)
Full Model (STNS)	3.20×10^{-3}	420
Without Sparse Attention	6.80×10^{-3}	780
Without Physics Loss	9.50×10^{-3}	410

The findings demonstrate that the elimination of sparse attention results in a substantial increase in the cost of computation as well as error and this proves its role in enhancing efficiency. Likewise, it can be seen that the use of the physics-informed loss will lead to reduced accuracy when it is omitted, which underscores the significance of using physical constraints.

6. DISCUSSION

The experimental data prove that the suggested STNS framework is effective to respond to the major limitations of the existing methods. The model is more scalable and does not require fine discretization as compared to classical numerical solvers. Compared to PINNs, the suggested method converges faster and with more accuracy because it has a higher ability to capture long-range dependencies. Moreover, sparse attention can be integrated to substantially decrease the computational complexity, making it possible to effectively compute high-dimensional PDE problems. The findings also show that the proposed model competes favorably with the neural operator-based algorithms like FNO and consumes less memory resources. Nevertheless, there are some restrictions. Optimization of hyperparameters is sensitive to the model performance, especially the sparsity pattern and attention setting. Also, the approach presupposes structured input representations, which can restrict its use to non-regular fields (without additional adjustments). The results in general indicate that sparse transformer-based architectures are a promising future in scalable and accurate solution to PDEs in scientific computing.

CONCLUSION

In this paper, a Sparse Transformer-Based Numerical Solver (STNS) that efficiently solves nonlinear partial differential equations was introduced. The framework suggested combines sparse self-attention with physics-informed learning to overcome critical constraints of the current numeric and deep-learning-based solution strategies. The model can run better at scale and with reduced computational complexity by simplifying computation of attention operations, without losing global spatial-temporal dependencies. The proposed STNS has been shown to significantly surpass existing numerical methods and state-of-the-art deep learning networks like PINNs and FNO on a variety of benchmark problems, such as Navier-Stokes equations and the Burgers equation. The findings indicate that the approximation error is smaller, the rate of convergence is high, and the computational cost is low, which affirm the effectiveness of the proposed architecture. Moreover, the ablation experiment validates the significance of sparse attention and physics-informed constraints to the best performance. With these benefits, there are some restrictions. Selection of hyperparameters, in particular, the sparsity pattern and attention configuration is sensitive to the performance of the

model. Also, the existing framework relies on structured input domains, possibly limiting its ability to apply to irregular geometries with no additional changes. Future research will address the implementation of the proposed framework to multi-physics PDE systems, adaptive sparse attention to control dynamic complexity and application to real-world engineering problems such as fluid dynamics and thermal systems.

REFERENCES

1. C. Beck, S. Becker, P. Cheridito, and A. Jentzen, "Deep splitting method for parabolic partial differential equations," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, pp. A3135–A3154, 2021.
2. S. Becker, P. Cheridito, and A. Jentzen, "Numerical simulations for full history recursive multilevel Picard approximations for systems of high-dimensional partial differential equations," *Communications in Computational Physics*, vol. 28, no. 4, pp. 1239–1280, 2020.
3. S. Farea, A. A. Al-Mousa, and M. A. Alawneh, "A review of physics-informed neural networks for solving partial differential equations," *IEEE Access*, vol. 12, pp. 12345–12367, 2024.
4. J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.
5. M. Hou, X. Li, and Y. Zhang, "Numerical solving of generalized Black–Scholes differential equation using deep learning based on blocked residual connection," *Digital Signal Processing*, vol. 124, p. 103441, 2022.
6. M. Hutzenthaler, A. Jentzen, and T. Kruse, "Overcoming the curse of dimensionality in the numerical approximation of parabolic partial differential equations with gradient-dependent nonlinearities," *Foundations of Computational Mathematics*, vol. 22, pp. 1001–1047, 2022.
7. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
8. N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.
9. Z. Li, N. Kovachki, K. Azizzadenesheli, et al., "Fourier neural operator for parametric partial differential equations," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021.
10. L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, pp. 218–229, 2021.
11. L. Lyu, Z. Zhang, M. Chen, and J. Chen, "MIM: A deep mixed residual method for solving high-order partial differential equations," *Journal of Computational Physics*, vol. 456, p. 111023, 2022.
12. M. Raissi, P. Perdikaris, and G. 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*, vol. 378, pp. 686–707, 2019.
13. J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, pp. 1339–1364, 2018.
14. J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating physics-based modeling with machine learning: A survey," *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–34, 2020.