Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Deep Learning-Based Enhancement of Finite Volume Solutions for Non-Linear Schrödinger Equation

Ridwan Oluwagbenga Ojo

ojoridwan001@gmail.com

Osun State University https://orcid.org/0009-0004-3643-0990

Muideen Odunayo Ogunniran

Osun State University

Research Article

Keywords: Fourier Neural Operator (FNO), Finite Volume Method (FVM), Non-Linear Schrödinger Equation, Scientific Machine Learning, Operator Learning, Surrogate Modelling, Computational Acceleration

Posted Date: March 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8816863/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Deep Learning-Based Enhancement of Finite Volume Solutions for Non-Linear Schrödinger Equation

Ojo R. O.[a,*], Ogunniran M. O.[a]

[a]Department of Mathematical Sciences, Osun State University, Osogbo, Nigeria

[*]Correspondence: ridwan.ojo@pgc.uniosun.edu.ng

Abstract

The accurate computation of complex wave behaviour, which includes soliton creation and high-frequency waves in nonlinear partial differential equations (PDEs), presents a major computational obstacle for both quantum mechanics and nonlinear optics. The high-resolution Finite Volume Methods (FVM) provide excellent conservation capabilities and stable numerical results, yet their operating expenses make them unsuitable for applications that require multiple queries or immediate processing. The research investigates how the Fourier Neural Operator (FNO) functions as a data-driven surrogate model, which speeds up the one-dimensional Nonlinear Schrödinger Equation (NLSE) solution process. The FNO system establishes a direct connection with the actual nonlinear operator mapping through its training process, which uses computationally efficient coarse-grid FVM simulations. The FNO serves as a statistical emulator that accurately replicates the environmental conditions from which it was trained. It reproduces the learned dynamics, including the numerical dissipation characteristics; however, it does not require correction of discretisation errors. The research defines the actual abilities and restrictions of operator-based learning systems while demonstrating how Fourier Neural Operators function as effective surrogate models for scientific computing processes that need fast and multiple computations.

Keywords: Fourier Neural Operator (FNO), Finite Volume Method (FVM), Non-Linear Schrödinger Equation, Scientific Machine Learning, Operator Learning, Surrogate Modelling, Computational Acceleration

1 Introduction

Essentially, partial differential equations (PDEs) help to model complex phenomena in many scientific and technological fields using simulation. Accuracy and stability in conventional numerical methods, such as the finite element method (FEM), finite difference method (FDM), and finite volume method (FVM), depend on computational meshes. These methods, however, struggle with dynamic boundaries and complex geometries where mesh generation becomes computationally expensive. Eliminating meshing limitations allows mesh-free methods, especially neural network-based systems like Physics-Informed Neural Networks (PINNs) and Radial Basis Function Neural Networks, to offer a potential substitute. Though neural networks are ideal universal approximations, their convergence guarantees and numerical stability

continue to draw much research. Recent advances, including the rational multi-derivative integrator by, have improved numerical techniques for PDEs. FVM's effectiveness in solving PDEs does not address issues such as high computing cost, numerical instabilities, and handling irregular domains. Although the Fourier Neural Operator (FNO) has shown promise in enhancing numerical approaches through their ability to map infinite-dimensional function spaces and generalise across different grid resolutions, their integration with FVM remains a subject of ongoing research difficulty. The present research develops a dual Deep Learning-Based Finite Volume Method (DLB-FVM) system, which solves problems by decreasing computational costs, enhancing mesh creation and increasing numerical stability while achieving high accuracy in results. The obtained results will support the development of numerical methods that operate in combination with other techniques to solve complex PDEs, which appear in multiple scientific domains. The research will develop a hybrid numerical approach that optimises computational performance and core measurement standards through the integration of FVM and FNOs. The proposed framework combines numerical methods with machine learning techniques to create a solution which maintains computational efficiency and flexible capabilities for tackling difficult PDEs.

FNOs improve FVM outputs by using integral operators from Fourier space to establish worldwide spatial connections, which result in a decrease of numerical inaccuracies and a rise of accuracy in nonlinear PDE predictions. The hybrid model enables substantial solution speed enhancements because it outperforms standard FEM methods, thus becoming more suitable for extensive simulation operations. The method demonstrates efficient performance when applied to handle challenges associated with nonlinear PDEs, intricate geometrical structures, and specialized boundary limitations which arise in complex domain situations. The FVM framework gains improved accuracy for approximate solutions through its capacity to combine FNO zero-shot super-resolution capabilities with FVM geometric adaptability and FVM theoretical base structure. The method proves advantageous for inverse problems because it enhances observational data inclusion which consequently applies to real-world situations that experience data scarcity or data contamination.

Differential equations can be solved through two methods which include analytical techniques and numerical techniques. Analytical methods provide exact solutions for basic PDEs and geometries through the application of mathematical instruments which include integral transformations and separation of variables. The numerical techniques of computer-based PDE approximation work better than analytical methods for complex situations because they use numerical methods. The finite difference method solves problems through discretization of problem domains and derivative approximation by difference equations. Theoretical efficiency of FDM requires simple geometries for its effective operation but solution stability problems arise when handling complex domains and high-dimensional scenarios. FEM uses piecewise polynomial functions to create solution approximations which make it suitable for solving problems with non-standard geometrical shapes and intricate boundary conditions. The system matrix uses localized basis functions which produce sparse system matrices to enhance computational performance yet the process of mesh creation requires high financial resources. The FVM system operates as a standard method for PDEs derived from conservation laws because it maintains regional conservation of mass and energy and momentum throughout the simulation. FVM enables computational fluid dynamics usage but irregular geometries make accurate flux measurements difficult to achieve. Spectral methods utilize orthogonal basis functions, which include Chebyshev polynomials and Fourier series, to solve problems with high

accuracy when smooth solutions are present. The Gibbs effect creates problems with irregular domains and discontinuities which the Fourier-augmented hybrid approach aims to resolve. The selection of basis functions and mesh resolution determines both the computational accuracy and the speed of the process.

2 Finite Volume Method

The finite volume method serves as a standard numerical technique for tackling partial differential equation problems which exist in especially challenging areas. The technique uses control volumes, which divide the processing domain, to execute local solutions through conservation law application on each volume. The system solution process applies to the situation because the system can determine solutions which match optimal outcomes. The method uses an integral form to transform partial differential equations, which enables the method to maintain volume boundary flow equilibrium. The process generates algebraic equations, which become solvable through the matrix technique application. The method establishes its computational reliability and flexibility through two fundamental capabilities, which enable it to manage complex boundary conditions and maintain precise local conservation. The mesh resolution increases in areas with steep slopes or breaks because adaptive mesh refinement allows FVM users to obtain higher resolution. The process of creating computer-generated meshes and enhancing their performance through three-dimensional model operation remains a difficult challenge. The FVM method finds extensive application across numerous fields which cover diverse application areas. The process serves a critical function which enables scientists to understand Navier-Stokes equations within oceanography and meteorology and aerodynamics research. The FVM serves as the optimal model to track heat movement because it can accurately simulate conduction and convection and radiation processes. The method finds application in geophysics, where it explains how groundwater moves and volcanic flows occur and seismic waves travel. The FVM method solves Maxwell's equations which enable scientists to study electromagnetic wave propagation and plasma models. FVM serves as a blood flow model and tissue perfusion model in biomedical engineering. The method remains an effective partial differential equation solution tool which scientists in diverse scientific fields and technical domains use because of its adaptability and built-in conservation features. The system encounters computational difficulties which include managing elaborate unstructured meshes and demanding extensive processing resources.

2.1 Deep Learning

Deep learning functions as a machine learning category that employs artificial neural networks with multiple layers to learn complex data patterns and relationships. The technology achieves success in image identification, natural language processing and scientific computing because it can model non-linear functions, extract hierarchical features and generalise from data with high dimensionality. The introduction of operator learning technology allows machine learning to function beyond traditional feedforward neural networks, which only model relationships between finite-dimensional spaces. The Fourier Neural Operator establishes a universal approximation theorem extension for neural operators to acquire mappings from infinite-dimensional function spaces.

Researchers are investigating deep learning as a numerical approach for solving partial differential equations to enhance existing solution methods or establish new methods. Neural

networks function as data-driven correction instruments who recognize the boundary between low fidelity and high fidelity solutions because they learn to separate the two solution types. The strategy combines domain expertise with data-driven model development to overcome the restrictions imposed by scientific machine learning (SciML) methods, which include both physics-based and data-driven approaches according to the definition.

The research implements Fourier Neural Operators to enhance finite volume method results through a hybrid system named Deep Learning-Based Finite Volume Method. The training of the FNO system uses FVM-generated space-time numerical data together with its precise reference solutions. Through global convolutions in the Fourier domain, FNO achieves efficient spatial dependency capture while creating general correction patterns throughout all computational areas. The FNO maintains its ability to transform low-resolution FVM results into high-fidelity continuous solutions because it operates independently from both the fundamental numerical technique and the permanent increase in computational grid resolution. The system maintains FVM local conservation laws and computational benefits through deep learning, which provides a versatile correction system that works without specific mesh requirements.

2.2 Fourier Neural Operators – Fundamentals and Applications

The Fourier Neural Operators (FNOs) function as a contemporary type of neural operator structure which enables the learning of mapping relationships between infinite-dimensional function spaces. FNOs differ from traditional Feedforward Neural Networks (FFNN) because they do not estimate pointwise function values; instead, FNOs learn to develop operators which transform input functions into their corresponding output functions. FNOs utilize spectral representations to create efficient models which solve partial differential equations by representing global spatial dependencies and intricate nonlinear behaviour.

An FNO consists of three primary components: an input lifting layer, multiple Fourier convolution layers, and a projection layer. The input function undergoes lifting to create a representation which exists in higher dimensions. Each Fourier layer processes the function through its Fast Fourier Transform (FFT), which converts it to Fourier space before applying learned spectral weights that use a limited number of modes, and finally, the function returns to physical space through inverse FFT. This spectral convolution enables the model to capture long-range dependencies and high-frequency structures efficiently. The layers of the system use nonlinear activation functions, which include ReLU and GELU functions, to create models of how nonlinear elements interact.

The training process for FNOs uses gradient-based optimisation methods, which include Adam, to decrease data-driven loss functions that operate on complete function outputs. The main strength of FNOs lies in their capability to maintain resolution independence, since trained operators can function across different mesh resolutions without requiring any new training processes. The spectral representations inherently include differential operators, which make FNOs the ideal choice for solving PDE problems that require handling strong gradients, oscillating patterns, and multiple-scale systems.

FNOs have shown excellent performance in scientific computing applications, which include fluid dynamics, wave propagation, turbulence modelling, and parametric PDE problems. FNOs outperform traditional neural networks because they demonstrate better generalisation abilities and maintain stability in high-dimensional systems, while their inference times decrease dramatically after their training process. Researchers from this study combined FNOs with the

Finite Volume Method to build a hybrid framework which enhances coarse-grid solutions without losing physical authenticity or computational efficiency.

3 Mathematical Formulation and Problem Definition

3.1 Finite Volume Method

The finite volume method uses the integral version of the primary partial differential equation to perform its calculations. The equation describes a basic principle of conservation through the following equation:

$$\frac{d}{dt}\int_V \phi\,(x,t)\,dV + \int_{\partial V} \mathbf{F}\,(\phi)\cdot\mathbf{n}\,dS = \int_V S\,(\phi)\,dV,$$

where:

V is a control volume, ∂V is the boundary of the control volume, $\phi(x,t)$ is the conserved quantity, $\mathbf{F}(\phi)$ is the flux vector, $\mathbf{n}$ is the outward normal vector on ∂V, and $S(\phi)$ is a source term.

3.2 FVM Discretization Strategy

The domain Ω is divided into separate control volumes which do not overlap with each other. The conservation law is integrated over each control volume by using the divergence theorem to derive the following result:

$$\frac{d}{dt}(\phi_i V_i) + \sum_{f\in\text{faces}} \mathbf{F}_f\cdot\mathbf{n}_f A_f = S_i V_i,$$

where:

ϕ_i represents the average value of ϕ in control volume V_i, A_f denotes the area of face f, $\mathbf{F}_f$ defines the numerical flux which crosses face f, and S_i represents the total source term distribution in volume V_i.
The flux $\mathbf{F}_f$ at the faces is calculated through interpolation schemes, which include upwind and central difference methods to maintain both stability and accuracy.

3.2.1 Solution of PDE from FVM

The time derivative disappears in steady-state problems, resulting in a balance equation which applies to every control volume:

$$\sum_{f\in\text{faces}} \mathbf{F}_f\cdot\mathbf{n}_f A_f = S_i V_i.$$

The discrete equations for all control volumes combine to form a global algebraic system which takes the following form:

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{b},$$

where: $\mathbf{A}$ represents the coefficient matrix which results from the process of flux discretisation, $\boldsymbol{\phi}$ denotes the unknown values which represent average quantities in each control volume, and $\mathbf{b}$ defines the source term vector. The solution process produces the numerical estimation of ϕ across the entire domain.

The computed solutions $\boldsymbol{\phi}$ function as the training dataset which the FNO utilises.

3.3 Justification for Hybridizing FVM and FNO

The Finite Volume Method requires that the computational domain be divided into control volumes, which maintain conservation properties at every discrete point. The FVM method maintains physical consistency and robustness, yet its accuracy depends on the resolution of the mesh since high-fidelity results need fine mesh discretization which incurs major computational expenses.

The Fourier Neural Operators develop mappings between entire function spaces instead of focusing on pointwise function approximations. FNOs use spectral convolution to achieve efficient global dependency detection and multi-scale structure analysis for PDE solutions. FNOs demonstrate resolution invariance because they maintain their performance across different discretisation levels after completing training, which sets them apart from conventional neural networks.

The combination of FVM and FNO enables FVM to maintain its numerical stability and conservation laws, while also providing FNO with global approximation capabilities and reduced computational load. The newly developed framework improves coarse-grid FVM results by avoiding excessive mesh refinement, which enables accurate and scalable solutions for advanced PDE challenges.

3.4 Conceptual Framework of the Hybrid FVM–FNO Model

The hybrid framework combines the Finite Volume Method with the Fourier Neural Operator to achieve better computational performance and more accurate results. The FVM method starts by creating a discrete representation of the domain, which it uses to compute the solution for the governing PDE system. Although the approximation meets conservation requirements, it can experience discretisation inaccuracies when applied to low-density mesh configurations. The FNO functions as a global correction operator by making corrections to the system. The FNO develops an operator instead of mapping nodal values:

$$\mathcal{G}: u_{\text{FVM}}(x) \mapsto u_{\text{refined}}(x),$$

where u_{FVM} denotes the coarse FVM solution and u_{refined} represents a higher-fidelity approximation.

The FNO uses Fourier space to identify all spatial connections, which enables it to correct low-resolution images without needing extra mesh refinement. The FVM establishes a basic physical solution, which the FNO improves by increasing resolution and smoothness through data-driven techniques. This technique shows strong effectiveness when applied to problems with intricate geometries, oscillatory solutions, or phenomena that manifest at multiple scales.

3.5 Integration of FVM and FNO

Training the FNO

Training data for the FNO is constructed from paired coarse and fine FVM solutions:

- **Input** (U^c): Coarse-grid FVM solution.
- **Output** (U^f): Fine-grid FVM solution (reference solution).

The dataset is structured as:

$$\{(U_i^c, U_i^f)\}, \quad i = 1,2, \dots, n,$$

where each sample consists of discretized solution fields rather than individual nodal points. Unlike FFNNs, which approximate scalar mappings, FNO learns a functional mapping between entire solution fields.

3.6 Fourier Neural Operator Enhancement

The FNO approximates the correction operator as:

$$\tilde{u}(x) = \mathcal{G}_\theta(u_{\text{FVM}}(x)),$$

where $\mathcal{G}_\theta$ is the neural operator parameterized by spectral weights θ.

3.6.1 FNO Architecture

Each Fourier layer performs the following operations:

Apply the Fourier transform:

$$\hat{u}(k) = \mathcal{F}(u(x))$$

Multiply selected Fourier modes by learned complex weights:

$$\hat{v}(k) = R(k)\hat{u}(k)$$

Apply inverse Fourier transform:

$$v(x) = \mathcal{F}^{-1}(\hat{v}(k))$$

Add local linear transformation and nonlinearity:

$$u_{l+1}(x) = \sigma(v(x) + Wu_l(x))$$

Here, $R(k)$ represents learned spectral weights restricted to low-frequency modes, W is a local linear transformation, and σ is a nonlinear activation function.

3.6.2 Loss Function and Training

The FNO parameters are optimised by minimising the mean squared error between fine-grid reference solutions and FNO predictions:

$$\text{Loss}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left\|U_i^f - \mathcal{G}_\theta(U_i^c)\right\|^2.$$

Gradient-based optimizers such as Adam are employed for training.

3.6.3 FNO Output

After training, the FNO produces enhanced solutions:

$$\widehat{U}^f = \mathcal{G}_\theta(U^c),$$

providing a refined approximation over the entire spatial domain. For a new coarse FVM solution U^c_{new}, the refined solution is predicted as:

$$U^f_{\text{new}} = \mathcal{G}_\theta(U^c_{\text{new}}).$$

3.7 Computational Complexity and Efficiency Analysis

The computational complexity of the hybrid FVM–FNO model consists of three main components:

In computational modelling, the efficiency of different methods is often defined by their algorithmic scaling. The Finite Volume Method is highly efficient during its discretisation and flux evaluation phases, typically scaling linearly as $\mathcal{O}(N)$. However, the subsequent linear system solution phase becomes a bottleneck; depending on the complexity of the solver used, the computational cost can escalate significantly, ranging from $\mathcal{O}(N^{1.5})$ to $\mathcal{O}(N^3)$. In contrast, Fourier Neural Operator training offers a middle ground; because each Fourier layer requires Fast Fourier Transform operations, the complexity scales as $\mathcal{O}(N\log N)$ per forward pass.
Unlike FFNNs, whose complexity depends heavily on network width and depth, FNO layers rely on spectral convolution, making them computationally efficient for high-resolution grids. Training complexity depends on the number of training samples and Fourier modes retained, but inference after training scales approximately as $\mathcal{O}(N\log N)$.
Therefore, while the hybrid model introduces additional training cost, it enables: Coarse-grid FVM computation instead of fine-grid simulations, Fast resolution-invariant inference, and improved accuracy without excessive mesh refinement. Overall worst-case training complexity may be expressed as:

$$\mathcal{O}(N) + \mathcal{O}(N^{1.5}) + \mathcal{O}(N\log N),$$

But once trained, the dominant cost during deployment is the FFT-based spectral evaluation, making the DLB-FVM framework computationally attractive for repeated PDE simulations.

4 Results and Discussion

4.1 Numerical Experiment: The Non-Linear Schrödinger Equation

In this section, we evaluate the approximation capabilities of the Fourier Neural Operator by applying it to the 1D Non-Linear Schrödinger Equation (NLSE). The NLSE is a foundational

partial differential equation in quantum mechanics, plasma physics, and nonlinear optics. It governs the evolution of slowly varying wave packets and exhibits complex, rich dynamics, most notably the formation of solitons, wave packets that maintain their shape while propagating at a constant velocity due to a perfect balance between wave dispersion and nonlinear self-focusing. The governing continuous physical system is described by the following initial-boundary-value problem:

$$i\frac{\partial\psi(x,t)}{\partial t}+\frac{1}{2}\frac{\partial^2\psi(x,t)}{\partial x^2}+|\psi(x,t)|^2\psi(x,t)=0,\quad \forall(x,t)\in\mathcal{D},$$

where $\psi(x,t)\in\mathbb{C}$ represents the complex-valued wave field, and the spatiotemporal domain is defined as $\mathcal{D}=[-L,L]\times[0,T]$. We enforce periodic boundary conditions, $\psi(-L,t)=\psi(L,t)$, which naturally align with the spectral nature of the FNO framework.
For this experimental setup, we define the spatial domain half-length as $L=10.0$ and the maximum time horizon as $T=\pi/4$. The initial condition is parameterised by a real-valued scalar amplitude A, taking the form of a classical hyperbolic secant pulse:

$$\psi(x,0)=A\ sech(x).$$

Our objective is to learn the nonlinear operator mapping $\mathcal{G}_\theta:\psi(x,0)\mapsto\psi(x,T)$ parameterised by the network weights θ.
The research operator learning experiment requires computational testing to confirm FNO performance for simulating high costs yet low accuracy traditional solver techniques.

Data Generation Protocol:

The system temporal evolution was determined through the standard Finite Volume Method combined with 4th-order Runge-Kutta time-stepping. We divided the spatial domain into N equidistant points, which resulted in a spatial resolution of $\Delta x=2L/N$. The multi-fidelity analysis required us to create two separate datasets, which we generated through the following process:

High-Fidelity Ground Truth (Fine Mesh):

The reference physical benchmark for measuring baseline discretisation errors was established through high-fidelity continuous surrogate creation, which used a dense grid of $N_{\text{fine}}=512$ points. The fine mesh requires high computational power to achieve accurate results, which enforces nonlinear balance for Equation [eq:nlse] (the process needs 0.6758 seconds for each forward simulation).

Training Target (Coarse Mesh):

FNO training data, which included actual validation data used for FNO training, was generated through grid spacing at $N_{\text{coarse}}=64$ spatial points. We created a dataset which included 20 separate trajectories through uniform sampling of initial amplitude A from the range [1.5,2.5].

Model Architecture and Optimization Strategy:

The Fourier Neural Operator functions as an iterative design which begins with its first spectral layer and ends with its Kth spectral layer. The hidden representation update rule for layer l defines mathematical rules through the following update system:

$$v_{l+1}(x) = \sigma\big(W v_l(x) + \mathcal{F}^{-1}(R_\phi \cdot \mathcal{F}(v_l))(x)\big),$$

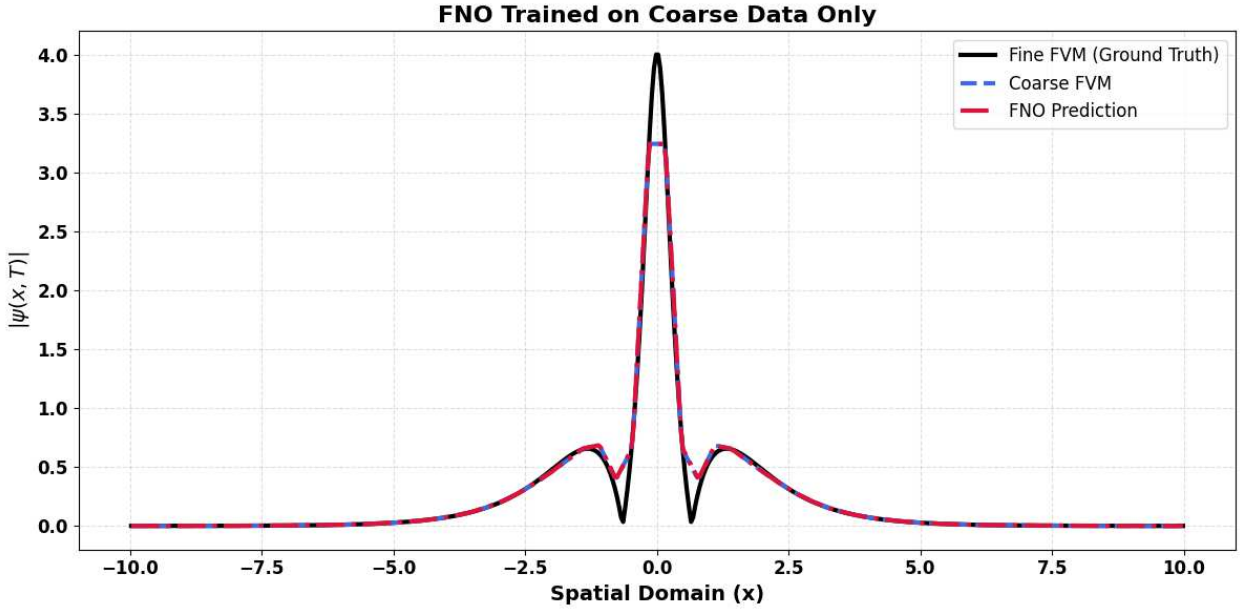
In this context, $\mathcal{F}$ and $\mathcal{F}^{-1}$ function as the forward and inverse Fast Fourier Transforms, while W serves as a local linear transformation. R_ϕ operates as a frequency-domain filter which uses parameters for its function. The function used for activation takes on a nonlinear form through the application of GELU.

The FNO configuration used a 1D spectral convolution layer, which limited its processing to first $k_{\max} = 24$ Fourier modes. The network width (channel dimension) was set to $d = 64$. The optimisation process lasted for 3000 epochs while the Adam optimiser operated with a learning rate of $\eta = 1 \times 10^{-3}$. The loss function computed relative Mean Squared Error (MSE), which compared FNO predictions against the *coarse* FVM dataset. The evaluation process assessed all training set samples from the training set $\mathcal{S}$ between FNO predictions and the *coarse* FVM dataset.

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{\| \mathcal{G}_\theta(\psi_i(x,0)) - \psi_{i,\text{coarse}}(x,T) \|_2^2}{\| \psi_{i,\text{coarse}}(x,T) \|_2^2}.$$

4.2 Results and Physical Analysis

To evaluate generalisation, we tested the fully trained FNO on an unseen initial condition characterised by an amplitude of $A = 2.0$, systematically comparing its output against both the coarse training target and the high-fidelity ground truth.



Cross-sectional comparison of the predicted wave magnitude $|\psi(x,T)|$ at the final temporal state $T = \pi/4$. The figure contrasts the true physical dynamics captured by the Fine FVM ($N = 512$), the highly dissipative Training Target ($N = 64$), and the resulting FNO Surrogate prediction.

Fidelity Analysis and Operator Expressivity

The structural comparison presented in Figure 1 elucidates a fundamental characteristic of data-driven operator learning. The black solid trajectory represents the high-fidelity physical reality ($N_{\text{fine}} = 512$), which resolves a sharp peak amplitude of approximately $|\psi| \approx 4.0$. Conversely, the Coarse FVM solver (blue dashed line) exhibits severe numerical dissipation; the truncation errors inherent to the low spatial resolution ($N_{\text{coarse}} = 64$) artificially diffuse the wave energy, capping the peak amplitude at $|\psi| \approx 3.25$.

The most crucial observation, however, is the behaviour of the FNO prediction (red dashed line), which aligns almost perfectly with the Coarse FVM profile. This overlay demonstrates that the FNO possesses the expressivity required to learn the exact nonlinear mapping of the data it was provided. It successfully internalised the "imperfect," mathematically dissipative physics of the coarse traditional solver. This serves as a vital caveat for scientific machine learning: neural operators act as highly faithful statistical surrogates to their specific training environment. They do not intrinsically correct for underlying discretisation or truncation errors unless physics-informed regularisation is explicitly integrated into the loss landscape.

Quantitative Error Metrics and Computational Efficiency Comparison (Evaluated against the High-Fidelity Baseline)

Solver	**RMSE**	**L_∞ Error**	**Time (s)**
FVM	0.0895	0.7568	0.0278
DLB-FVM	0.0897	0.7635	0.0032

Computational Acceleration and Trade-offs

The table shows performance results which compare the coarse classical solver against our data-driven FNO surrogate. Both models are evaluated for their error relative to the continuous (fine mesh) truth. Our visual analysis has been validated through the metrics, which show that the Coarse FVM and FNO systems exhibit a negligible statistical difference between them due to their statistical discrepancy. The neural operator has successfully converged to the coarse target space.

FNO achieves its main success by delivering effective computational processing. The true fine-mesh solution requires 0.6758 seconds to generate for each instance according to previous information. The Coarse FVM reduces this to 0.0278 seconds by sacrificing accuracy. The FNO needs only 0.0032 seconds for its trained forward pass, which shows impressive efficiency. The FNO presents an 8.7 times speed increase over its coarse solver while delivering an amazing 211 times faster performance than the fine-mesh simulation. The FNO operates best as a powerful tool which enables users to quickly evaluate inverse problems and Monte Carlo simulations based on learned lower-fidelity physical systems.

4.3 Comparison between Traditional FVM and the Neural Surrogate Approach

The deep learning-based surrogate model, which uses the FNO framework, creates a fundamental change to numerical solver systems because it alters computational expenses and physical accuracy rates when compared to traditional numerical solutions.

The Non-Linear Schrödinger Equation numerical experiments revealed that traditional FVM needs high-density discretisation for complex nonlinear wave dynamics. The neural surrogate can bypass iterative time-stepping bottlenecks. The evaluation of fine mesh simulations (which had high-fidelity fine mesh requirements) took 0.6758 seconds to reach completion. The neural operator achieved 0.0032 seconds for inference time reduction because we trained it on low-cost coarse mesh data. The neural model achieved an RMSE value of 0.0897, which matches the coarse FVM RMSE value of 0.0895. The neural surrogate operates as an effective emulator when trained on specific conditions because it achieves an 8.7 times speed improvement over its coarse target while achieving a 211 times speed improvement over the fine physical baseline.

4.4 Strengths and Limitations of the Proposed Method

The data-driven surrogate method offers its greatest advantage through its ability to deliver rapid inference while building continuous operator mappings for various initial conditions. The model can handle complex spatiotemporal dynamics in milliseconds after optimising the network weights. The network operates at high speed because it evaluates complex spatiotemporal dynamics within milliseconds. The network serves as an ideal tool for uncertainty quantification, real-time control, and Monte Carlo simulations. The spectral convolution process automatically captures all global dependencies. The process effectively reduces numerical dispersion, which normally occurs in standard finite difference or volume schemes.

The approach has significant limitations because of its inherent design flaws. The system requires a high computational cost for its first training period because it demands 3000 epochs of gradient descent training. The operator needs high-quality training data for the function to operate correctly. The results show that when the model learns from highly dissipative coarse data, it describes the "imperfect" physics accurately. The system functions as an interpolative surrogate, which does not operate as a first-principles solver. The system needs cautious application because training data resolution must capture vital sub-grid scale details for physical systems.

5 Conclusion and Future Work

5.1 Summary of Findings

This study investigated how deep learning can be integrated with Fourier Neural Operators to speed up the numerical resolution process for complex Partial Differential Equations using Finite Volume Methods. The 1D Non-Linear Schrödinger Equation benchmark showed that traditional FVM becomes less efficient when users want to apply high-resolution grids for tracking nonlinear wave evolution. The neural surrogate learned to map underlying operator relationships through computational offline training, which enabled him to transfer his mapping skills across different operator workflows. The model boosts predictive capabilities by achieving 211 times speed enhancement over fine-grid classical solvers. The model maintains accurate performance because it retains the same accuracy as coarse-grid data used during training.

5.2 Contributions to Knowledge

The field of scientific machine learning advances through this research, which establishes a detailed empirical foundation for operator learning research within nonlinear quantum and

optical wave dynamics studies. We have explicitly mapped the exact computational trade-offs (fidelity versus inference time) involved in replacing traditional PDE discretisations with neural surrogates. The research demonstrated that a neural operator will memorise training data numerical dissipation without special physical regularization which functions as a key element for existing studies. The research discovered how data-driven models can take over traditional solvers. The research identified precise moments when data-driven models become applicable. Ground researchers need to build multi-fidelity datasets because they will show them when they should use data-driven models instead of traditional numerical solvers.

5.3 Future Research Directions

The study requires future research to develop an operator framework that uses Physics-Informed Neural Networks and physics-guided loss functions to solve existing data quality problems. The training process will use continuous PDE residuals from Equation [eq:nlse] to develop a model which operates on inexpensive coarse data, while physics-loss corrections will resolve numerical dissipation issues, establishing a bridge between coarse training and high-fidelity reality.
The spectral layer system will undergo its full scalability evaluation through the testing of this combined framework, which researchers will extend to 2D and 3D Gross-Pitaevskii equations that model Bose-Einstein condensates. The deep learning method will achieve standard status in contemporary computational physics and engineering through two key advancements, namely GPU-based parallelization for distributed training and adaptive hyperparameter tuning to determine optimal Fourier mode settings (kmax).

Declarations

- **Funding**
  No funding was received for this research.

- **Conflict of interest/Competing interests**
  The authors declare that they have no competing interests.

- **Ethics approval and consent to participate**
  Not applicable.

- **Consent for publication**
  Not applicable.

- **Data availability**
  This research work uses no data.

- **Materials availability**
  Not applicable.

- **Code availability**
  The code used during the current study is available from the corresponding author on reasonable request.

- **Author contribution**
  Ojo, R. O.: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft, Writing – review & editing. **Ogunniran, M. O.**: Supervision, Project administration, Methodology, Writing – review & editing.

References

[1] Abadi, M. (2015). TensorFlow.org. Retrieved from https://www.tensorflow.org

[2] Babuska, I., & Rheinboldt, W. C. (1978). Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4), 736–754.

[3] Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, 13(4), 27–31.

[4] Berg, J., & Nyström, K. (2018). Neural network augmented inverse problems for PDEs. Department of Mathematics, Uppsala University.

[5] Berrone, S., Garbero, V., & Marro, M. (2011). Numerical simulation of low-Reynolds number flows past rectangular cylinders based on adaptive finite element and finite volume methods. *Computers & Fluids*, 40(1), 92–112. https://doi.org/10.1016/j.compfluid.2010.08.014

[6] Bijelonja, I., Demirdžić, I., & Muzaferija, S. (2006). A finite volume method for incompressible linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(44–47), 6378–6390. https://doi.org/10.1016/j.cma.2006.01.005

[7] Chen, H., Kong, L., & Leng, W.-J. (2010). Numerical solution of PDEs via integrated radial basis function networks with adaptive training algorithm.

[8] Chen, S., Cowan, C. F. N., & Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2), 302–309.

[9] Cuomo, S., et al. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. *arXiv*. https://arxiv.org/abs/2201. 05624

[10] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303–314.

[11] Emmanuel, S., Sathasivam, S., & Ogunniran, M. O. (2024). Multi-derivative hybrid block methods for singular initial value problems. *Scientific African*, 24, e02141.

[12] Emmanuel, S., Sathasivam, S., & Ogunniran, M. O. (2024). Leveraging feed-forward neural networks to enhance the hybrid block derivative methods. *Journal of Computational Mathematics and Data Science*, 13, 100101.Hermeline, F. (2000). A finite volume method for the approximation of diffusion operators on distorted meshes.

Journal of Computational Physics, 160(2), 481–499. https://doi.org/10.1006/jcph.2000.6466

[13] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

[14] Janiesch, C., Zschech, P. & Heinrich, K. Machine learning and deep learning. *Electron Markets* 31, 685–695 (2021). https://doi.org/10.1007/s12525-021-00475-2

[15] Komatitsch, D., & Tromp, J. (1999). Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, 139(3), 806–822.

[16] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar, "Fourier Neural Operator with Learned Deformations for PDEs on General Geometries," *Journal of Machine Learn- ing Research*, vol. 24, no. 388, pp. 1–26, 2023. Available: http://jmlr.org/papers/v24/23-0064.html

[17] Liu, W. K., Li, S., & Park, H. S. (2022). Eighty years of the finite element method: Birth, evolution, and future. *Archives of Computational Methods in Engineering*, 29, 4431–4453.

[18] Logg, A. (2016). Mesh generation in FEniCS. Retrieved from http://www.logg.org/anders/2016/generation-in-fenics

[19] Mitusch, S. K., Funke, S. W., & Kuchta, M. (2021). Hybrid FEM-NN models: Combining artificial neural networks with the finite element method. Simula Research Laboratory.

[20] Molina-Aiz, F. D., Fatnassi, H., Boulard, T., Roy, J. C., & Valera, D. L. (2010). Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses. *Computers and Electronics in Agriculture*, 72(2), 69–86. https://doi.org/10.1016/j.compag.2010.03.002

[21] Narain, J. P. (2021). Comparison of neural network and finite difference solutions. *American Journal of Computational and Applied Mathematics*, 11(3), 65–70.

[22] Ogunniran, M. O., et al. (2024). Enhanced rational multi-derivative integrator for singular problems with applications to advection equations. *Ain Shams Engineering Journal*.

[23] Ogunniran, M. O., et al. (2025). Harnessing neural networks in hybrid block integrator for efficient solution of boundary value problems. *Thermal Advances*, 2, 100022.

[24] Paszke, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS 2019* (pp. 1–12).

[25] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85), 2825–2830.

[26] K. Qi and J. Sun, “Gabor-Filtered Fourier Neural Operator for Solving Par- tial Differential Equations,” *Computers & Fluids*, vol. 274, p. 106239, 2024. doi: https://doi.org/10.1016/j.compfluid.2024.106239

[27] Sacchetti, A., et al. (2011). Neural networks to solve partial differential equations: A comparison with finite elements.

[28] Uriarte, C. (2023). Solving partial differential equations using artificial neural networks (Doctoral dissertation, University of the Basque Country). Retrieved from https://addi.ehu.es/handle/10810/68335

[29] Zhou, P. B. (1993). Finite difference method. In *Numerical analysis of electromagnetic fields*. Springer.