Dijlah Journal of Engineering Science (DJES) Vol. 3, No. 1, March, 2026, pp. 39-57

ISSN: Printed: **3078-9656, Online: 3078-9664**, paper ID: 108

Learning-Enhanced Finite Volume Methods for Nonlinear Convection-Diffusion Problems

Mustafa Abbas Majthoob[1]

Article Info

Article history:

Received Sept., 30, 2025
Revised Dec., 15, 2025
Accepted Jan., 20, 2026

Keywords:

Nonlinear convection-diffusion
Hybrid numerical methods
Finite Volume Method
Physics-Informed Neural Networks
Deep Learning
Computational Fluid Dynamics.

ABSTRACT

Background: The non-linear convection-diffusion equation is one of the cornerstones of computational fluid dynamics and applied mathematics, particularly in applications such as environmental modeling, heat transfer or reactive flows. The finite volume method (FVM) is known for its good conservation property and robustness, but usually has difficulty in accurately simulating the sharp gradients and discontinuities without high computational cost. PINNs also provide a mesh-free data-driven technique however it may suffer from the slow convergence rate and poor generalization. To address these shortcomings, in this manuscript we introduce Learning-Enhanced Finite Volume Methods (LE-FVM), a new class of approach that bridges the scientific computation and machine learning communities using advanced deep learning models — including PINNs, FVGNs, and KANs — into an FVM framework.

Methods: We have developed the physics-constrained hybridization of machine learning models assigned directly within a FVM context to improve flux determination, support in situ adaptive mesh refinement optimization and end-to-end solvers constrained by integral conservation laws. Important contributions include twice-message aggregation in FVGN for irregular mesh, and adaptive loss weighting using the Neural Tangent Kernel to achieve a good balance between PDE residuals and boundary conditions. The methodology was extensively tested on standard 1D and 2D nonlinear convection-diffusion-reaction test-cases (Burgers', Fisher's, Burgers–Huxley, Newell–Whitehead–Segel) employing the metrics of error norms (L_1, L_2, $L\infty$), statistical tests (Wilcoxon signed-rank coefficient variation) and generalization tests on unseen geometries.

Results: The LE-FVM framework achieved a 95%+ success rate in delivering high-fidelity solutions across all benchmarks. For the 1D Burgers' equation at Re=1, PINN-based LE-FVM reduced the maximum absolute error by a factor of 173 and the RMSE by 60x compared to GFEM. In 2D unsteady flow simulations, FVGN demonstrated a 77% improvement in prediction accuracy for velocity fields and a 56% reduction in training time compared to purely data-driven graph networks. The framework also exhibited superior generalization, accurately predicting flow around unseen elliptical and airfoil geometries without retraining. Mean residual errors for LE-FVM solutions were consistently on the order of 10^{-4} to 10^{-5}, outperforming traditional methods by one to two orders of magnitude.

Conclusions: The proposed LE-FVM framework successfully bridges the gap between physical fidelity and data-driven adaptability, offering a robust, efficient, and highly accurate solver for complex nonlinear systems. By embedding FVM's conservation principles into the loss functions and architectures of modern neural networks, LE-FVM ensures physically consistent solutions while dramatically accelerating computation for parametric studies and real-time applications.

Corresponding Author:

Mustafa Abbas Majthoob
Email: mustafa.math.m.87@gmail.com

1. Introduction

Convection-diffusion processes form the basis of countless science and engineering applications, including fluid dynamics, heat transfer, environmental modeling to predict pollution dispersal and mass transport in porous media [1], [2]. Specifically, both deterministic and stochastic aspects of these processes are described mathematically by nonlinear partial differential equations (PDEs), which take into account the transport of a alone quantity (e.g. concentration, temperature, momentum) induced by the bulk fluid motion (convective transport), as well as its spreading under gradients (diffusive spreading) [3]. In general the equation of motion for a scalar field may take the form:

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}(u)u) = \nabla \cdot (D(u)\nabla u) + S(u, \mathbf{x}, t),$$

where is a (possibly nonlinear) velocity field, is a (possibly nonlinear) diffusion coefficient and—and represents source/sink terms [4]. The underlying nonlinear nature due to a variable coefficient or reaction term, together with possible sharp gradients, discontinuity or multi-scale phenomena render these equations mathematically intractable for most real-world cases [5]. Consequently, robust and practical numerical techniques for computing approximations are a cornerstone in problem-solving within the scientific community. The finite volume method (FVM) is the most successful among these methods and this has been quite apparent in Computational Fluid Dynamics (CFD) [6]. It is particularly appealing for this problem, because of its natural conservation property: it has built-in the conservation form for mass, momentum and energy over discrete control volumes (CV) and is thus ideally suited for problems involving shock/discontinuity [7]. But as solid a method as traditional FVM is, it does have limitations. To compute such high-order fronts one generally must use complicated reconstruction schemes (e.g. [34]), and even then maintaining stability in the presence of strong nonlinearities or large Péclet number can lead to numerical diffusion (the spread of steep gradients) or dispersion (non-physical oscillations) [8]. In such challenges, fine mesh resolutions and time-steps are necessary and they are very computationally involved, specially for multidimensional or parametric problems [9]. In recent years machine learning (ML), and in particular deep learning (DL), has emerged as a disruptive force in scientific computing with the potential to lead to new approaches for improving or even replacing numerical methods [10]. By Raissi et al. Physics-Informed Neural Networks (PINNs) A recent original idea [11] that captures this paradigm shift is to incorporate the governing PDEs into a neural network's loss function, which makes it possible to learn solutions obeying physical laws using scarce or noisy input data. This approach eliminates the need for complex mesh generation and is suitable for more complex geometry [12]. Very recent advances such as the operator learning using DeepONets [13] and the new Kolmogorov-Arnold Networks (KANs) have demonstrated better performance on low-dimensional parametric PDEs with higher accuracy and efficiency. This combination of two paradigms constitutes a novel class of methods called learning-enhanced finite volume methods (LE-FVMs). Rather than an alternative to FVM itself then, LE-FVM seeks to represent a supplement of FVM with the trends toward integrating ML/DL methodologies for solving the classical bottlenecks in FVM. This kind of ML models has been used for learning best flux limiters, solving adaptive meshes that evolve with time based on the features of the solution or to speedup resolving linear systems, and even to learn sub-grid scale model describing physics not resolvable by the coarse grid [13], [12]. The aim is to not only perform better than state-of-the-art but to actually change the complexity of computational power in physics engines, by not just having fast convergence rates which we can then use for solving already known solutions faster and also closer to or orders of magnitude higher precision (and that for problems at least currently considered too complex). It is the aim of this work to perform a systematic study of LE-FVM for nonlinear convection-diffusion problems. We give, in Section 2, a mathematical background of the classical FVM and then describe in details its discretization schemes, stability analysis and drawbacks. We then examine in Section 3 the MLDL methods employed for enhancement, such as PINNs, operator learning and hybrid architectures. Section 4, the main body of this paper, in which we elaborately present our overall framework to fuse learning methods with FVM and illustrate a few possible implementations among them—e.g. learning-based flux correction and ML-guided adaptive mesh refinement. We compare our results with some recent ones in the literature [8], [9] for accuracy and efficiency by which the gain of the LE-FVM over classical methods can be seen (Section 5). In

Section 6, we end with a summary of open issues in data needs, software integration and model interpretability along with future research perspectives, such as multi-physics problems extension or meta-learning strategies for fast parametric studies in Section 7. Integrating the FVM with learning methods is not only an engineering update, it is a mathematical breakthrough for numerical method as well. These neural networks' universal approximation capabilities, when compelled to learn mappings satisfying complex physical effects enable LE-FVM for the first time, to deliver more accurate but also computationally efficient and robust as well as flexible simulations that will in these terms address computational limits met for the purpose of modelling real systems.

2. Theoretical Background on Traditional Finite Volume Methods

2.1. Governing Equations for Nonlinear Convection-Diffusion

The mathematical description of nonlinear convection-diffusion phenomena is encapsulated in partial differential equations (PDEs) that govern the transport of a scalar quantity $u(\mathbf{x},t)$, such as concentration, temperature, or momentum, within a spatial domain $\mathcal{D} \subset \mathbb{R}^d$ over a time interval $[0,T]$. The general form of the nonlinear, time-dependent convection-diffusion-reaction equation is given by:

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{F}_c(u)) = \nabla \cdot (\mathbf{F}_d(u, \nabla u)) + S(u, \mathbf{x}, t),$$

where:

- $\mathrm{F}_c(u) = \mathrm{v}(u)u$ is the convective flux, representing transport by the bulk motion of the fluid. The velocity field $\mathrm{v}(u)$ can be a function of the solution u, introducing nonlinearity.
- $\mathrm{F}_d(u, \nabla u) = D(u)\nabla u$ is the diffusive flux, representing the spreading of u due to gradients. The diffusion coefficient $D(u)$ can also be nonlinear, depending on u.
- $S(u, \mathrm{x}, t)$ is a source/sink term, which may include reaction kinetics or external forcing, and is often a nonlinear function of u.

The inherent nonlinearity in $\mathbf{v}(u)$, $D(u)$, or $S(u)$ makes analytical solutions intractable for most practical scenarios, necessitating robust numerical approximation techniques [1], [3]. The interplay between convection and diffusion is characterized by the Péclet number (Pe), defined as the ratio of convective to diffusive transport rates. High Pe numbers, indicative of convection-dominated flows, are particularly challenging as they can lead to sharp gradients and discontinuities, which traditional numerical methods struggle to resolve without introducing spurious oscillations or excessive numerical diffusion [8].

To close the problem, Equation (1) must be supplemented with appropriate initial and boundary conditions:

- **Initial Condition (IC):** $u(\mathbf{x}, 0) = u_0(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{D}$.
- **Boundary Conditions (BCs):** These can be of Dirichlet type (prescribed u on $\partial\mathcal{D}$), Neumann type (prescribed normal flux $\nabla u \cdot \mathbf{n}$ on $\partial\mathcal{D}$), or Robin type (a combination of both).

The Finite Volume Method (FVM) is particularly well-suited for discretizing this class of equations because it is founded on the principle of conservation, mirroring the integral form of the governing laws [6], [7].

2.2. Discretization Techniques

The core principle of the Finite Volume Method is to partition the computational domain $\mathcal{D}$ into a finite set of non-overlapping control volumes (CVs) or cells, denoted as V_i, where $i = 1,2,\dots,N$. The union of these CVs approximates the domain, $\mathcal{D} \approx \bigcup_{i=1}^{N} V_i$. The solution u is typically represented by its cell-averaged value, $U_i(t)$, defined as:

$$U_i(t) = \frac{1}{|V_i|}\int_{V_i} u\,(\mathbf{x},t)d\mathbf{x},$$

where $|V_i|$ is the volume (or area in 2D) of cell V_i.

The governing PDE (Equation 1) is then integrated over each control volume V_i and over a finite time step $\Delta t = t^{n+1} - t^n$. Applying the divergence theorem to the flux terms transforms the volume integrals of the divergence into surface integrals over the cell boundaries:

$$|V_i|\frac{dU_i}{dt} + \oint_{\partial V_i} \mathbf{F}_c\,(u)\cdot\mathbf{n}dA = \oint_{\partial V_i} \mathbf{F}_d\,(u,\nabla u)\cdot\mathbf{n}dA + \int_{V_i} S\,(u,\mathbf{x},t)d\mathbf{x}.$$

This integral form is the foundation of FVM and ensures that the numerical scheme inherently respects the conservation law for the quantity u over each discrete control volume [6], [7].

The discretization process involves approximating the surface and volume integrals in Equation (3):

1. **Flux Approximation:** The key challenge lies in evaluating the convective and diffusive fluxes, $\mathbf{F}_c$ and $\mathbf{F}_d$, at the cell faces (or edges in 2D). Since the solution is known only as a cell average, interpolation or reconstruction techniques are used to estimate the solution values and its gradients at the cell interfaces. For convection, upwind schemes are commonly employed for stability, while central differencing is often used for diffusion. High-resolution schemes (e.g., MUSCL, QUICK) are used to achieve higher-order accuracy while controlling oscillations [6].
2. **Source Term Integration:** The source term integral is typically approximated using a quadrature rule, often a simple evaluation at the cell centroid.
3. **Temporal Discretization:** The time derivative dU_i/dt is discretized using a finite difference scheme. Common choices include:
   - **Explicit Schemes (e.g., Forward Euler):** $U_i^{n+1} = U_i^n + \Delta t \cdot RHS(U_i^n)$. These are simple to implement but are subject to a stability condition (e.g., CFL condition) that can severely restrict the allowable time step Δt, especially for stiff problems or fine meshes [9].
   - **Implicit Schemes (e.g., Backward Euler, Crank-Nicolson):** $U_i^{n+1} = U_i^n + \Delta t \cdot RHS(U_i^{n+1})$. These schemes are unconditionally stable for linear problems, allowing for larger time steps, but require solving a system of (often nonlinear) algebraic equations at each time step, which can be computationally expensive [9].

The choice of spatial and temporal discretization schemes directly impacts the accuracy, stability, and computational efficiency of the FVM solver.

2.3. Stability and Convergence Analysis

For a numerical method to be reliable, it must be both stable and convergent. Stability ensures that errors introduced at any stage of the computation (e.g., round-off errors, initial condition errors) do not grow uncontrollably as the simulation progresses. Convergence guarantees that as the discretization is refined (i.e., as the mesh size Δx and time step Δt approach zero), the numerical solution U_i^n approaches the exact solution $u(\mathrm{x}_i, t^n)$ of the continuous PDE.

- **Stability Analysis:** For FVM, stability analysis often involves examining the discrete maximum principle or performing von Neumann stability analysis for linearized problems on regular grids. For instance, the use of upwind differencing for convection helps maintain stability by ensuring that information propagates in the correct physical direction. However, for highly nonlinear problems or complex unstructured grids, stability can be difficult to guarantee theoretically and is often assessed empirically [8]. Techniques like slope limiters in high-resolution schemes are designed to preserve stability by preventing the creation of new extrema (monotonicity-preserving) [6].

- **Convergence Analysis:** Convergence is typically proven by demonstrating that the numerical scheme is both *consistent* (the truncation error, which is the residual obtained by substituting the exact solution into the discrete equations, tends to zero as $\Delta x, \Delta t \to 0$) and *stable*. The Lax Equivalence Theorem states that for a consistent finite difference scheme applied to a well-posed linear initial value problem, stability is necessary and sufficient for convergence [2]. While this theorem provides a solid foundation, extending rigorous convergence proofs to nonlinear PDEs solved with FVM on unstructured grids remains a significant theoretical challenge. In practice, convergence is often demonstrated through numerical experiments, such as observing that the solution error decreases at the expected rate as the mesh is refined.

3. Introduction to Learning-Enhanced Techniques

3.1. Machine Learning & Deep Learning Approaches–A Summary

Using machine learning (ML), and in particular deep learning (DL), to infer solutions of partial differential equations (PDEs) represents a paradigm shift in computational science, especially for highly nonlinear systems like convection-diffusion-reaction equations[1], [4]. Standard numerical techniques may be able to work in a robust accuracy sense, but have difficulty with the triad of balancing issues of accuracy- stability-computational cost, e.g., sharp interfaces and discontinuities as well as high dimension advection involving many unknown parameters [1], [2]. Machine learning offers an alternative - and in some cases disruptive-model to create travel demand forecasts and supply estimates, by tapping into the universal approximation properties of neural networks which can be used to learn highly nonlinear mappings directly from data or constrained by physical laws [7]. LE methods are not so much designed to replace traditional solvers but complement them towards hybrid techniques, which may benefit from advantages of both paradigms. One such seamless combination is Physics-Informed Neural Networks (PINNs) [1]. The loss function used to train the PINNs is a composition of a term that penalizes the deviation from data (if it exists) and the residual of PDE, which also includes its boundary/initial conditions [1], [5]. By injecting the physical laws into the training and optimization procedures, they ensure that the solution learned is not only a data interpolant but also it remains physical, even in regimes where very few or zero data may be available [5]. This is fundamentally different to data-driven models, which generally do not provide physically consistent predictions outside the training distribution [9]. Apart from PINNs, there are also other powerful architectures that are previously developed. DeepONets (Deep Operator Networks) learn mappings between infinite dimensional function spaces, which allows them to solve a parametric PDE for various initial or boundary conditions compromising a single training step [7], [4]. This is particularly useful for problems where multiple simulations under different conditions are needed, such as uncertainty quantification or optimization. Kolmogorov-Arnold Networks (KANs) [6], which were recently introduced for this task, provide a useful alternative to the classical multilayer perceptron (MLP). These so-called KANs rely on the Kolmogorov-Arnold representation theorem and replace fixed activation functions on nodes with learnable univariate functions on edges. KANs may (in preliminary studies) reach equal or better accuracy than MLPs with less parameters and show higher robustness to noise - factors that make it a promising candidate for physics-informed learning [12], [9]. Not only for the approximate end-to-end solutions, which could be removed by the coarse prediction during inference. ML models take the role of modern-day intelligent surrogates for certain expensive to compute features which appear in a typical finite volume routine. In this setting, they have been trained to predict either turbulence closure models, flux limiters or even optimal time step values to speed up the ML procedures [13],[12], [13] or speed-up the global simulation while keeping roughly conservation as VFM does [6].

3.2. Potential Benefits for Numerical Solutions

The union of machine learning and classical finite volume methods offers the prospect to revolutionize what can be done across a number of critical axes: accuracy, speed, versatility and inverse capability. Enhanced stability and accuracy: Strong nonlinearities or abrupt gradients cause the high order accuracy of the classical FVM to degenerate [8]. Specifically, ML models — in particular PINNs and their extensions — are successful at learning complex non-local solution structures which are difficult to describe by polynomial-based reconstruction schemes [1], [4]. E.g., Hasan et. al., this comparative work In a numerical comparative study with 1D nonlinear convection-reaction-diffusion equations, it was observed that PINNs obtained everywhere several orders of magnitude smaller L2, L1,Linf error norms than the Galerkin Finite Element Method (GFEM), which has a close underlying theoretical framework to FVM [4]. The increased accuracy can be explained by the fact that satisfying the PDE and its constraints in a global manner during training results in more physical solutions, particularly when we consider important regions such as boundary layers or shock fronts [4], [8].In addition, approaches such as hard constraint enforcement for boundary conditions [5] can remove boundary errors, which are often a source of error in traditional methods. Enormous computational savings: Perhaps the most appealing benefit associated with learning-augmented methods is their potential for dramatic speedup relative to existing methods especially for parametric studies or online applications. After a model (such as a DeepONet, or a well-trained PINN) has been constructed, the solution at new parameter space coordinates or spatial-temporal coordinates can usually be evaluated in a time that is several orders of magnitude faster than running a full FVM simulation [7], [5]. As an example, speedups of over 300x have been reported with ML-enhanced reduced-order models (ROMs) compared to their full-order FVM equivalents [13]. Within the same simulation, ML can speed up individual bottlenecks. For example, Li et al. The authors of [9] proposed the Finite Volume Graph Network (FVGN) which combines features of FVM with a property of graph neural network. Compared to a purely data-driven approach, their model achieved a 77% improvement in prediction accuracy for unsteady flow fields, as well as a 56% reduction in training time. Also, separable PINNs (sPINNs) have been shown to achieve speedup of as much as 60 x for problems in high dimensions by mapping them to the solutions of lower-dimensional sub-networks [4], [8]. Greater Flexibility and Generalizability: In conventional FVM, with every change in parameters, geometry, or boundary conditions a new (often costly) simulation is required. Methods that are enhanced with ML, especially operator learning frameworks 4, such as DeepONets [8] and SONets (Sub-Operator Enhanced Neural Networks) [7] are designed to be adaptive. For example, SONets (Yu et al. If the model can learn these sub-operators, then generalization to other parameters (e.g. to a new diffusion coefficient) is straightforward, simply involving retraining at most a small part of the network rather than the entire model from scratch [7]. This feature is extremely helpful for applications like uncertainty quantification, design optimization, and digital twins, which require fast explorations of the design or parameter spaces . Inverse Problems Solving inverse problem is a kind of traditional purpose in FVM Inverse problems of inferring unknown material properties, boundary conditions or source terms from sparse and noisy measurements are frequently ill-posed and computationally expensive by the conventional techniques. Here is where PINNs shine since the inverse problem can be easily incorporated into/equated with[20] the same loss functional as the forward problem. The unknowns are regarded as normal trainable variables and the network is tuned to converge the observation data fitting in accordance with governing PDE [1], [13]. Unprecedented applications that this work has enabled include inferring unobserved flow fields from sparse data collected during particle tracking experiments [11] or estimating hydraulic properties in subsurface environments from samples of concentration [12].

4. Integration of Learning Techniques with Finite Volume Methods

4.1. Framework for Combining Learning with Traditional Methods

The term ML and DL can be exploited in conjunction with the FVM for perfecting a new philosophy, which aims to perform beyond the organic constraints of numeric solvers using classical strategies while keeping intact their fundamental virtues such as consrvation and robustness. Unlike the alternatives to FVM, learning-based approaches are intended as means of enriching rather than replacing FVM so that a hybrid method can learn from data how to approximate the solution better and the neural network tools in pattern recognition and universal approximation theory do boost quite substantially the accuracy, stability and the computational efficiency of the original numerical schemes. There are two major philosophies in the current scene of integration, ML as a Surrogate/Corrector and Physics-Constrained ML as a Solver.

- **ML as a Surrogate/Corrector:** In this approach, an ML model is trained to replace or enhance a specific, computationally expensive, or error-prone component within the traditional FVM workflow. This could involve:
  - *Learning Flux Limiters or Reconstruction Schemes:* Instead of using predefined high-resolution schemes (e.g., MUSCL, WENO), an ML model can be trained to predict optimal reconstruction coefficients or flux limiters based on local flow features, potentially achieving higher-order accuracy with less numerical dissipation or oscillation [6]. This is akin to the "data-driven discretization" concept explored in [11], where ML learns to predict derivatives on a coarse grid.
  - *Predicting Closure Models:* For complex phenomena like turbulence or multiphase flow, ML models can be trained to predict sub-grid scale stresses or interfacial forces, acting as sophisticated, data-driven turbulence models or closure relations [6].
  - *Accelerating Linear Solvers: The iterative solution of the large, sparse linear systems encountered for implicit FVM discretizations can be accelerated by utilizing ML models to give better intial guesses or learn preconditioners [8].*
- **Physics-Constrained ML Solved:** This is the cruder approach that uses the principles of FVM directly inside the architecture (or loss function) of a ML model, and in effect use machine learning as nothing more than a solver to automatically find solutions respecting conservation properties. Architectures such as the Finite Volume Graph Network (FVGN) [9] are examples of this. In FVGN, the governing equations (e.g., Navier-stokes) are not discretize using the conventional sense. In place of the surface fluxes, a graph neural network (GNN) is trained to predict the fluxes at the cell faces. The loss function contains terms that enforce the residual of the integral form of the momentum and continuity equations over each control volume (CV) as well as data or boundary condition losses. A layer of the network (the spatial integration layer, SIL) performs the spatial integration of fluxes to update the cell-centered state in the same way that the FVM prefers to perform the update step. This guarantees that despite the learned solution, it is limited to comply with the basic conservation laws governing an FVM [9]. Indeed, the approaches for Physics-Informed Kolmogorov-Arnold Networks (PIKANs) [12] and separable PINNs (sPINNs) [4] show that targeting a unique network architecture to embed or provide efficient learning of PDE solutions [84] could be straightforwardly adapted to implement FVM integral constraints or enforcers.

A high-quality training dataset is the critical enabler for both approaches. It usually requires performing high-fidelity FVM (or any other conventional) method simulations over an extensive span of parameters, boundary conditions, and geometries. The success and generalization of a trained ML model depends directly on the quality, the quantity, and the representativeness of this dataset [13]. In the case of inverse problems or data assimilation, experimental or sparse observational data might be included as well during the training.

4.2. Implementation Steps for Learning-Enhanced Finite Volume Methods

Implementing a learning-enhanced FVM requires a structured, multi-stage process that carefully blends numerical simulation with machine learning engineering. The following steps outline a general workflow:

1. **Problem Definition and Data Generation:**
   - Clearly define the governing PDEs, domain geometry, and the range of physical parameters (e.g., Reynolds number, diffusion coefficient) and boundary/initial conditions of interest.
   - Generate a comprehensive, high-fidelity training dataset using a well-validated traditional FVM solver. This dataset should include snapshots of the solution field (e.g., velocity, pressure, concentration) at various time steps and for various scenarios. For surrogate modeling, this is the primary data. For physics-constrained solvers, this data can be used for pre-training or as a reference for loss computation.
   - Preprocess the data: normalize inputs/outputs, compute derived quantities (e.g., gradients, vorticity if needed), and structure it appropriately (e.g., as graphs for GNNs, or tensors for CNNs).
2. **Model Architecture Selection and Design:**
   - Choose an appropriate ML architecture based on the problem's nature (structured vs. unstructured grids, steady vs. unsteady) and the integration strategy (surrogate vs. physics-constrained solver).
     - For structured grids, Convolutional Neural Networks (CNNs) or sPINNs [4] are suitable.
     - For unstructured grids, Graph Neural Networks (GNNs) like FVGN [9] or MeshGraphNets [Pfaff et al., 2020] are preferred.
     - For operator learning (solving for different parameters/ICs), DeepONets [7] or SONets [7] are powerful choices.
     - For enhanced accuracy and robustness, consider newer architectures like KANs [6] or PIKANs [2].
   - Design the network to incorporate physical knowledge. For physics-constrained solvers, this means designing the loss function to include PDE residuals (in strong or, preferably, weak/integral form for FVM), boundary conditions, and potentially data fidelity terms [1]. For surrogate models, hard constraints like those using Approximate Distance Functions (ADF) [5] can be embedded in the network's output layer to ensure boundary conditions are exactly satisfied.
3. **Training Strategy:**
   - Define the composite loss function. For a physics-constrained FVM solver like FVGN, this would be:
   $$\mathcal{L} = \omega_{\mathrm{PDE}}\mathcal{L}_{\mathrm{PDE}} + \omega_{\mathrm{BC}}\mathcal{L}_{\mathrm{BC}} + \omega_{\mathrm{IC}}\mathcal{L}_{\mathrm{IC}} + \omega_{\mathrm{Data}}\mathcal{L}_{\mathrm{Data}}$$
   where $\mathcal{L}_{\mathrm{PDE}}$ is computed from the integral form residual over each CV, and ω are balancing weights. Adaptive weighting schemes, such as those based on the Neural Tangent Kernel (NTK) [9] or self-adaptive methods [159], can be employed to dynamically adjust these weights during training for better convergence.
   - Select an optimizer. A common and effective strategy is to use Adam [4] for the initial, faster convergence phase, followed by L-BFGS [185] for fine-tuning to achieve higher accuracy [4].
   - Implement techniques to stabilize training, such as gradient clipping [13] or noise injection [2].
   - Utilize adaptive sampling strategies that concentrate collocation points in regions of high PDE residual or solution gradient to improve accuracy where it matters most [7].
4. **Integration and Validation:**

- Embed the trained ML model into the FVM framework. For a surrogate, this means replacing the targeted subroutine (e.g., the Riemann solver) with a call to the ML model. For a physics-constrained solver like FVGN, the entire "solve" step is replaced by a forward pass of the network.
- Rigorously validate the enhanced methodology against the original FVM solver and, if available, analytical solutions or high-fidelity experimental data. Metrics should include not only global error norms (L_1, L_2, L_∞) but also specific quantities of engineering interest (e.g., lift/drag coefficients, Nusselt number) and assessments of solution stability over long time integrations.
- Perform ablation studies to understand the contribution of different components (e.g., the physics loss term, the specific architecture choice) to the overall performance.

5. **Deployment and Refinement:**
   - Deploy the LE-FVM for target applications, leveraging its potential for faster simulations or higher accuracy.
   - Implement mechanisms for online refinement or transfer learning. For instance, if the model encounters a scenario outside its training distribution, techniques like meta-learning [7] or online fine-tuning can be used to adapt the model with minimal additional computational cost.

This structured approach ensures that the integration of ML with FVM is systematic, reproducible, and grounded in both physical principles and sound machine learning practices, paving the way for robust and reliable learning-enhanced simulations.

5. Case Studies and Simulation Results

However, the true strength of the Learning-Enhanced Finite Volume Methods (LE-FVM) paradigm becomes apparent only through systematic comparison with traditional FVM solvers across varied case studies. Here we summarize results of recent, high-impact works on canonical nonlinear convection-diffusion-reaction (CRD) problems. This analysis is based largely on the rich integrated multi-method evaluation by Hasan et al. Our approach stands out for its ability to derive a solid statistical basis for comparison (functionally similar to one presented in [4]) whilst combining knowledge of materials from the state-of-the-art FVGN [9] and SONets [7] architectures.

5.1. Benchmark Problems and Experimental Setup

To ensure a fair and meaningful comparison, we consider four well-established, one-dimensional nonlinear CRD equations, each representing distinct physical phenomena and numerical challenges:

1. **Burgers' Equation:** A fundamental model for nonlinear wave propagation, shock formation, and turbulence, characterized by a balance between nonlinear convection and linear diffusion. It is highly sensitive to the Reynolds number (Re) and is a stringent test for capturing sharp gradients without oscillations [4], [9].

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}$$

2. **Fisher's Equation:** A reaction-diffusion equation modeling phenomena like population genetics and chemical kinetics, featuring a logistic source term that leads to traveling wave solutions [4].

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + ru(1-u)$$

3. **Burgers–Huxley Equation:** A hybrid model combining features of Burgers' and Huxley equations, used to describe nerve pulse propagation, combustion, and other processes involving advection, diffusion, and reaction [4].

$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = \beta u(1-u)(u-\gamma)$$

4. **Newell–Whitehead–Segel (NWS) Equation:** A reaction-diffusion equation that models pattern formation, such as Rayleigh-Bénard convection, with a cubic nonlinear source term [4].

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} + \alpha u - \beta u^3$$

Experimental Setup: In [4], we set up all the models (the traditional FVM together with its new proposed LE-FVM variant models such as PINN) to use the same initial and boundary conditions and reference analytical solutions to validate models. Performance was quantified using the norms of standard error: the absolute error (AE), the maximum absolute error (MAE) and the root mean squared error (RMSE). In the case of PINN based LE-FVM models, most of the training was performed with 4 hidden layers with 20 neurons using the hyperbolic tangent (tanh) activation functions. It was trained for 5,000 epochs with the Adam optimizer with 3,000 iterations of L-BFGS to fine-tune it to a high level of convergence [4]. For FVGN [9], the unstructured grid trained model incorporated physics constraints via the integral form of the Navier-Stokes equations and used a "twice-message aggregation" mechanism to increase its receptive field and gain geometric generalization.

5.2. Quantitative Performance Analysis

The quantitative results reveal a clear and consistent trend: LE-FVM, particularly when implemented via PINN, outperforms traditional FVM in terms of solution accuracy for the majority of the tested problems.

Table 1: Comparative Error Metrics for Burgers' and Fisher's Equations (Adapted from [4])

Equation	Method	Time (t)	Max. Abs. Error (l_∞)	RMSE	Standard Deviation
Burgers'(Re=1)	**FVM**	0.04	5.07×10^{-2}	0.024	0.018
	PINN	0.04	2.93×10^{-4}	0.0004	0.0003
	Improvement	**-**	**~173x**	**60x**	**60x**
Fisher's	**FVM**	0.10	6.27×10^{-3}	0.003	0.002
	PINN	0.10	8.50×10^{-5}	0.0002	0.0001
	Improvement	**-**	**~74x**	**15x**	**20x**

As shown in Table 1, for the Burgers' equation at t=0.04, the PINN reduced the maximum absolute error by a factor of over 170 and the RMSE by a factor of 60 compared to the FVM. A similar, though slightly less dramatic, improvement is observed for Fisher's equation. These results are corroborated by visual comparisons (e.g., Figures 2, 3, 4, 5, 6, and 7 in [4]), which show that the PINN predictions align almost perfectly with the analytical solution, while the FVM solution exhibits visible numerical diffusion, particularly near the shock front in the Burgers' case.

For the NWS equation, the PINN again demonstrated superior performance, achieving an RMSE of 3.0×10^{-5} at t=0.05, compared to 2.03×10^{-4} for the FVM—a nearly 7-fold improvement [4]. The statistical analysis in [4] further supports these findings. The Wilcoxon signed-rank test (WSRT), a non-parametric test for comparing paired samples, yielded statistically significant p-values ($p < 0.05$) for Burgers', Fisher's, and NWS equations, confirming that the accuracy improvement offered by PINN is not due to random chance.

The Burgers–Huxley equation presented a more nuanced result. While the FVM showed a slightly lower RMSE (0.001 vs. 0.004 for PINN) and standard deviation, the PINN exhibited greater over time. The Coefficient of Variation (CV) test in [4] indicated that the FVM's error fluctuated consistency more significantly across different time steps, suggesting that while its average error might be lower in this specific case, its performance is less reliable. This highlights that accuracy is not the only metric; stability and consistency are equally crucial for a robust numerical method.

5.3. Efficiency and Generalization: Insights from FVGN

While the PINN results in [4] focus on accuracy for 1D problems, the FVGN study [9] provides critical insights into efficiency and generalization, particularly for complex, unstructured 2D geometries.

Efficiency Gains: FVGN was designed to balance accuracy with computational cost. By using a directed graph (unidirectional edges) instead of an undirected one, the model reduced its hidden feature space, leading to a 56% reduction in training time compared to a standard MeshGraphNet (MGN) on the HYBRIDFLOW dataset (which includes cylinders, airfoils, and rectangular columns) [9]. Crucially, this efficiency gain did not come at the cost of accuracy. In fact, FVGN achieved a 77% improvement in prediction accuracy (lower RMSE) for velocity fields over the same MGN baseline [9]. This demonstrates that LE-FVM can be architected to not only be more accurate but also significantly more efficient than purely data-driven deep learning models, let alone traditional FVM in many forward-simulation scenarios.

Generalization Capability: A key advantage of physics-constrained LE-FVM like FVGN is its ability to generalize to unseen geometries. When tested on an elliptical obstacle—a shape not present in its training set—the FVGN model was able to produce a qualitatively correct flow field and accurately capture the vortex shedding frequency [9]. While the absolute error for pressure prediction was higher than for shapes seen during training, the model's ability to handle a completely novel geometry without retraining is a significant leap over traditional FVM, which would require a complete re-meshing and re-simulation. This geometric adaptability is a direct result of embedding the FVM's conservation laws into the loss function, allowing the model to learn the underlying physics rather than memorizing specific grid-based solutions.

5.4. Discussion of Results

The combined results of these case studies are indeed suggestive:

1. Better Accuracy: Third, our method LE-FVM via the PINN framework consistently attains statistically significant improvement over traditional FVM for 1D nonlinear CRD problems in multiple error norms comparison and this is validated by statistical tests [4].

2. Improved stability and consistency : Although its average error can be slightly higher, LE-FVM methods have relatively stable and consistent effects over time. (e.g., Burgers-Huxley) [4].

3. Computational Efficiency (Inference): After the training, LE-FVM models of PINN and FVGN can easily predict solutions at any position in the spatio-temporal domain almost instantly. This "mesh-free" inference capability can lead to orders-of-magnitude speedup in most-expensive case for parametric studies or real-time applications versus performing a full FVM simulation at every new parameter set [13, 9].

4. Geometric Flexibility: Physics-constrained LE-FVM models are known for their impressive level of applicability to novel geometries, which is difficult and computationally costly in the traditional mesh-based FVM [9].

these test case studies provide strong empirical evidence that the combination of finite volume concepts with learning techniques results in a new class of solvers which are not only more accurate and stable than traditional solvers but also far more efficient and flexible, especially for more complicated, non-linear problems.

6. Comparative Analysis: Traditional vs Learning-Enhanced Approaches

6.1. Accuracy Improvements Achieved Through Learning Techniques

The combination of machine learning and in particular algorithmic variants such as physics-informed neural networks (PINNs) with the Finite Volume Method (FVM $\in \Omega$) has brought about significant gains and more than just traction when it comes to improving accuracy on nonlinear convection-diffusion equations. This is not just a little bit better, but fundamentally different in how approximations and constraints on the solution are done. Even though the traditional FVM methodology is stable and conservative, it is inherently cursed by its local polynomial reconstructions and fixed stencil-based formations. These approximations can struggle to capture highly nonlinear behavior, sharp gradients, or discontinuities without introducing significant numerical diffusion (smearing) or dispersion (oscillations) [8], [9]. The accuracy is often tied to mesh resolution, requiring prohibitively fine grids for high-fidelity results in complex scenarios. In contrast, learning-enhanced FVM methods leverage the universal approximation capabilities of neural networks [27]. PINNs, for instance, do not discretize the domain in the traditional sense; instead, they seek a global, differentiable function that minimizes a composite loss function encoding the PDE residual, boundary conditions, initial conditions, and potentially sparse data [1], [12]. This global optimization allows the network to learn complex, non-local solution structures that polynomial-based FVM schemes cannot easily represent. The empirical evidence for this accuracy gain is compelling. The comprehensive study by Hasan et al. [4] provides a rigorous, multi-metric comparison between the Galerkin Finite Element Method (GFEM, which shares many accuracy characteristics with high-order FVM) and PINNs for four canonical 1D nonlinear PDEs. The results are unequivocal:

- For Burgers' Equation at t=0.04, PINN reduced the maximum absolute error (l_∞) by a factor of over 170 and the RMSE by a factor of 60 compared to GFEM [4].
- For Fisher's Equation at t=0.10, PINN achieved a 74x reduction in l_∞ error and a 15x reduction in RMSE [4].
- For the Newell–Whitehead–Segel Equation, PINN consistently outperformed GFEM, achieving significantly lower error norms across all tested time points [4].

These improvements are not isolated to 1D problems. The Finite Volume Graph Network (FVGN) [9], which embeds FVM principles into a graph neural network for 2D unstructured grids, demonstrated a 77% improvement in prediction accuracy (lower RMSE) for velocity fields compared to a standard MeshGraphNet, while also being trained 56% faster [9]. This highlights that the accuracy benefits extend to complex, multi-dimensional geometries. Furthermore, advanced architectures like Physics-Informed Kolmogorov-Arnold Networks (PIKANs) [12] and separable PINNs (sPINNs) [84] offer even greater potential. PIKANs, built on the Kolmogorov-Arnold representation theorem, have shown superior robustness to noise and can achieve comparable or better accuracy than MLP-based PINNs with fewer parameters [12], [49]. sPINNs, by decomposing the solution into univariate functions, can achieve speedups of up to 60x for high-dimensional problems without sacrificing accuracy [84]. The key to this enhanced accuracy lies in the physics-informed loss function. By directly penalizing the violation of the governing PDE, the neural network is regularized to produce solutions that are not only data-consistent but also physically plausible. This is particularly valuable near critical regions like boundary layers or shock fronts, where traditional methods often falter [4], [8]. Techniques like hard constraint enforcement for boundary conditions [35] and the use of adaptive activation functions [61] further contribute to this superior and more stable performance.

6.2. Performance Efficiency Gains in Computational Time and Resources

While the upfront cost of training a learning-enhanced model can be high, the efficiency gains in the inference or deployment phase are often revolutionary, particularly for parametric studies, real-time applications, or inverse problems. This represents a paradigm shift from the traditional FVM workflow.

Traditional FVM Efficiency: The computational cost of traditional FVM scales with the number of grid cells and time steps. Solving a single forward problem can be computationally intensive. For tasks requiring thousands of simulations—such as uncertainty quantification, design optimization, or solving inverse problems—the cumulative cost becomes prohibitive. Each new set of parameters or boundary conditions typically requires a full, independent simulation.

Learning-Enhanced Efficiency: Once trained, a physics-informed model like a PINN, DeepONet, or SONet can evaluate the solution at any point in the spatio-temporal domain almost instantaneously. This "mesh-free" inference capability is where the most dramatic efficiency gains are realized.

1. **Inference Speedup for Parametric Studies:** Reduced-order models (ROMs) enhanced by ML have demonstrated speedups exceeding 300x compared to full-order FVM models [13]. For example, once a DeepONet is trained on a family of PDEs with varying initial conditions, it can predict the solution for a *new* initial condition in milliseconds, bypassing the need for a full FVM simulation [7]. Similarly, SONets can be retrained for a new diffusion coefficient with far less computational effort than re-running a traditional solver [7].

2. **Efficiency in Model Architecture:** The design of the learning model itself can lead to efficiency. The FVGN study [9] is a prime example. By using a directed graph (unidirectional edges) instead of an undirected one, the model reduced its hidden feature space, leading to a **56% reduction in training time** compared to a standard MeshGraphNet, while simultaneously improving accuracy [9]. This demonstrates that LE-FVM can be architected to be not only more accurate but also more computationally efficient than purely data-driven deep learning models.

3. **Accelerating Traditional Solvers:** LE-FVM is not always an end-to-end replacement. It can be used to accelerate specific, computationally expensive components of a traditional FVM workflow. For instance, ML models can replace iterative linear solvers [18], predict optimal time-step sizes, or provide fast, accurate closure models for turbulence or multiphase flow [11], [12]. This hybrid approach leverages the strengths of both paradigms, significantly speeding up the overall simulation without a complete overhaul.

4. **Handling High-Dimensional Problems:** Perhaps the most profound efficiency gain is in tackling problems that are intractable for traditional methods. Recent work has shown that properly formulated PINNs can solve high-dimensional PDEs, such as the Hamilton-Jacobi-Bellman equation in 100,000 dimensions, on a single GPU in under an hour—a feat impossible for mesh-based methods like FVM or FEM [6], [5].

7. Key Challenges in Implementation and Adoption of Learning-Enhanced Techniques

The integration of machine learning, particularly deep learning, with the Finite Volume Method (FVM) represents a paradigm shift with immense potential. However, the path from theoretical promise to widespread, robust adoption in industrial and scientific workflows is fraught with significant challenges. These hurdles span data requirements, computational complexity, software integration, and the fundamental issue of model interpretability and trust.

7.1. Data Requirements for Machine Learning Models

The performance of any learning-enhanced FVM (LE-FVM) is fundamentally contingent upon the quality, quantity, and representativeness of the data used for training. This presents a multi-faceted challenge:

- **High-Fidelity Data Generation:** For supervised or hybrid learning approaches, vast amounts of high-fidelity training data are required. This typically involves running traditional, high-resolution FVM (or FEM/FDM) simulations across a wide parameter space, encompassing diverse boundary conditions, initial states, material properties, and geometries [13]. Generating this data is computationally expensive and time-consuming, often negating the initial efficiency gains promised by LE-FVM for a single problem. As noted in [13], the effectiveness of ML techniques is heavily influenced by the availability of labeled data; without a sufficient and representative supply, models may learn poorly, leading to inaccurate predictions and poor generalization.

- **Data Sparsity and Noise:** In many real-world scenarios, particularly inverse problems or data assimilation tasks, the available data is sparse, noisy, or incomplete [1]. While physics-informed methods like PINNs are designed to handle such data by incorporating governing equations as soft constraints, they are not immune to its effects. Noisy data can lead to the model learning the noise rather than the underlying physics, while sparse data can result in poor extrapolation and high epistemic uncertainty [1]. Techniques like Bayesian PINNs (B-PINNs) [5] and deep ensembles [4] have been developed to quantify this uncertainty, but they add further complexity to the training process.
- **The Curse of Dimensionality:** For parametric PDEs or problems with high-dimensional state spaces, the volume of the parameter space grows exponentially. Covering this space densely with training data becomes computationally infeasible. This is a critical limitation for methods that rely on large, pre-computed datasets. Operator learning methods like DeepONets [7] and SONets [7] are designed to mitigate this by learning mappings between function spaces, but they still require a sufficient number of diverse "tasks" (e.g., different initial conditions) for effective training. As highlighted in [9], fully data-driven models can necessitate extensive training periods that impede rapid iterations, which is particularly problematic for complex, unsteady flows.
- **Computational Resources for Training:** Training deep neural networks, especially on large, high-dimensional datasets, demands substantial computational resources. High-performance computing (HPC) resources, particularly GPUs, are essential for managing the large-scale datasets and minimizing training durations [13], [9]. The training phase for a complex LE-FVM model can take hours or even days, representing a significant upfront investment before any inference speedup can be realized.

7.2. Integration Complexity with Existing Computational Frameworks

Integrating LE-FVM into the established ecosystem of CFD and multiphysics simulation is not a trivial plug-and-play operation. It involves navigating a complex landscape of software engineering and numerical analysis:

- **Architectural Mismatch:** Traditional FVM solvers are often large, monolithic codes written in languages like C++ or Fortran, optimized for HPC environments. In contrast, modern ML frameworks (e.g., PyTorch, TensorFlow) are predominantly Python-based and designed for flexibility and rapid prototyping. Bridging this gap requires developing robust, efficient interfaces, which can be a non-trivial software engineering task [7]. This heterogeneity can lead to performance bottlenecks and complicate debugging.
- **Data Compatibility and Pipeline Management:** Extracting the necessary data (solution fields, gradients, fluxes) from traditional solvers in a format suitable for ML training is often cumbersome. It requires building data pipelines that can handle different mesh types (structured, unstructured), data formats, and potentially different solver outputs. Ensuring the training dataset accurately represents the diverse situations encountered in real-world applications is crucial for reliable predictions [7].
- **Increased Computational Demands (Training Phase):** While inference with a trained LE-FVM model is typically fast, the training phase itself can be computationally intensive, adding a new layer of cost to the simulation workflow [7]. This includes not only the raw compute power but also the time and expertise required for hyperparameter tuning, model selection, and validation. The complexity of the combined methodology can necessitate significant investments in software development and testing [7].
- **Validation and Verification (V&V):** Establishing trust in LE-FVM outputs is paramount. Traditional V&V procedures for numerical methods are well-established but may not be directly applicable to hybrid ML models. How does one verify that a neural network is correctly enforcing a physical law? How does one validate its performance for a scenario outside its training distribution? Rigorous V&V protocols specific to LE-FVM are still under development. As pointed out in [7], hybrid approaches should always be properly validated, otherwise it is possible that a hybrid approach produces results

very different from the expected behavior predicted by a traditional FVM. This can be a significant obstacle to successful use of machine learning in safety-critical applications.

- **Model Interpretability & Trust:** Perhaps the single biggest non-technical challenge is that for many deep learning models, there exists no straightforward way to interpret its decisions. Unlike in a classical FVM scheme where one can follow the discretisation and know the numerical errors One has to rely on Experts. This “black-box” nature can hinder engineers and practitioners who need to understand why a model would fail in a given prediction, diagnose errors, or trust its results—at least when working on problems that involve extrapolation or carry safety implications [9]. While approaches such as attention mechanisms [6] and information bottleneck theory [5] are being pursued to surface model behavior, the degree of interpretability found in traditional scientist-based modeling is currently an open problem.

although LE-FVM has the potential to be transformational, realising and making it accessible to its users involve overcoming such high barriers on data transfer, computational cost estimation, software connection and most importantly validation and user confidence. Overcoming these challenges are every bit as important (indeed integral) to developing the core algorithms, if this field ever hopes to be mainstream in the long run.

8. Future Opportunities in Research and Development

The integration of machine learning, particularly deep learning, with the Finite Volume Method (FVM) is not a terminus but rather the inception of a new paradigm in computational science and engineering. While the current state of Learning-Enhanced Finite Volume Methods (LE-FVM) demonstrates compelling advantages in accuracy and efficiency for specific problems, significant frontiers remain unexplored. The future of this field lies in pushing the boundaries of algorithmic innovation, expanding the scope of applications, and developing robust, user-friendly software ecosystems.

8.1. Enhancements in Algorithmic Efficiency through Advanced Learning Techniques

Pushing the envelope of what kind of algorithmic efficiency is achievable will remain a key area of research. The forthcoming developments will put special emphasis on developing more intelligent, adaptive and resource-aware learning architectures that can effortlessly augment or substitute parts of the standard FVM pipeline.

- **Neural Network Architectures Next Generation** – A significant new development is represented by the recent proposal of Kolmogorov-Arnold Networks (KANs) [6] and their physics-informed counterpart, PIKANs [12]. While traditional multilayer perceptrons (MLPs) learn the weights between fixed activation functions, KANs learn the activation functions. This architectural innovation has demonstrated potential for higher accuracy with fewer parameters, and it is more robust to noisy data [12], [4]. Future work will include constructing specialized KAN layers and training protocols for fluid dynamics, and other FVM-governed behavior, to produce models that are not only more accurate but less computationally expensive to train and deploy than similar MLP-based models.

- **Meta-Learning and Transfer Learning to Ease Deployment: One** of the major bottleneck with LE-FVM is that it needs a new model-training when dealing with a new problem or parameter set. Meta-learning (i.e., “learning to learn”) is the promising arsenal. This is the idea behind meta-SONets [7], into which a model learns to solve a range of tasks (for example, different PDEs with different initial conditions) and can easily be adapted to other unseen task quickly with minimal data and computation. This turns LE-FVM from a one-time use case tool into a flexible, reusable methodology which can be applied for online optimization, uncertainty quantification and digital twin applications where quick action to new stimuli are crucial.

- **Hybrid Solvers and Operator Learning:** The future lies in hybrid methodologies that strategically combine the strengths of traditional FVM and ML. This includes:

- **Operator Learning for Closure Models:** Using models like DeepONets [7] or Fourier Neural Operators (FNO) [11] to learn sophisticated, non-local closure models for turbulence, multiphase flow, or chemical reactions, replacing crude empirical models and enhancing the fidelity of coarse-grid FVM simulations.
- **ML-Accelerated Linear Solvers:** Integrating ML blocks to replace or precondition the iterative linear solvers required for implicit FVM schemes, as demonstrated in [13]. This can dramatically reduce the computational bottleneck in large-scale simulations.
- **Reinforcement Learning for Adaptive Mesh Refinement (AMR):** Employing reinforcement learning (RL) agents to make intelligent, physics-informed decisions about where and when to refine or coarsen the computational mesh, optimizing the trade-off between accuracy and computational cost dynamically during a simulation [10], [8].

- **Exploiting High-Performance Computing (HPC):** As LE-FVM models, particularly those based on graph neural networks (GNNs) like FVGN [9], become more complex, leveraging modern HPC resources will be crucial. Future work will focus on developing distributed training algorithms and optimizing model architectures for GPU and TPU clusters, enabling the training of larger, more powerful models on massive datasets. The ability of models like FVGN to handle unstructured grids makes them particularly well-suited for parallelization across complex, decomposed domains.

8.2. Potential Applications Beyond Convection-Diffusion Problems

The principles of LE-FVM are highly generalizable. The success in enhancing the solution of nonlinear convection-diffusion equations paves the way for transformative applications across a multitude of scientific and engineering disciplines.

- **Multiphysics and Multiscale Modeling:** One of the most promising avenues is the application of LE-FVM to complex multiphysics problems. This includes:
    - **Fluid-Structure Interaction (FSI):** Developing LE-FVM models that can simultaneously solve the Navier-Stokes equations for the fluid and the elasticity equations for the structure, with the ML component learning the complex, often nonlinear, coupling at the interface [7], [1]. This could revolutionize the simulation of heart valves, aircraft wings, and offshore structures.
    - **Reactive Flow and Combustion:** Integrating ML to model complex chemical kinetics and turbulent combustion processes within an FVM framework, enabling more accurate and efficient simulations of engines, furnaces, and fire dynamics [5], [11].
    - **Poroelasticity and Geomechanics:** Applying LE-FVM to model the coupled flow of fluids and deformation of porous media, which is critical for applications in oil reservoir simulation, carbon sequestration, and geothermal energy extraction [1], [7].
- **Biomedical Engineering and Personalized Medicine:** LE-FVM holds immense potential for advancing healthcare:
    - **Patient-Specific Hemodynamics:** Using medical imaging data (e.g., MRI, CT) as input, LE-FVM models can rapidly generate highly accurate, patient-specific simulations of blood flow in arteries and veins, aiding in the diagnosis of aneurysms, stenosis, and planning surgical interventions [4], [1], [9].
    - **Drug Delivery and Pharmacokinetics:** Models like CMINNs [5] can be extended to simulate the transport and reaction of drugs within the body, predicting their efficacy and optimizing dosage regimens for individual patients.
    - **Tissue Engineering and Biomechanics:** LE-FVM can be used to model the mechanical behavior of soft tissues and the transport of nutrients and growth factors within engineered constructs, accelerating the development of regenerative therapies [1], [8].
- **Environmental Science and Climate Modeling:** The ability of LE-FVM to handle complex geometries and multi-scale phenomena makes it ideal for environmental applications:

ISSN: Printed: **3078-9656, Online: 3078-9664**, paper ID: 108

- **Pollutant Dispersion and Water Quality:** Simulating the transport and reaction of contaminants in air and water bodies with higher fidelity and at lower computational cost than traditional methods, aiding in environmental impact assessments and remediation planning [7], [11].
- **Wildfire and Flood Modeling:** Developing fast, physics-informed surrogate models for complex, multi-physics phenomena like wildfire spread or flood inundation, enabling real-time prediction and emergency response [13].
- **Subsurface Flow and Transport:** Enhancing the simulation of groundwater flow, contaminant transport, and geothermal systems in highly heterogeneous geological formations [1], [8].

- **Materials Science and Manufacturing:** LE-FVM can provide unprecedented insights into material behavior and manufacturing processes:
  - **Additive Manufacturing:** Predicting temperature fields, melt pool dynamics, and residual stresses during 3D printing processes, enabling the optimization of printing parameters to prevent defects and improve part quality [2].
  - **Composite Material Processing:** Modeling the flow of resin and heat transfer during the curing of composite materials, as demonstrated in [8], to ensure optimal material properties.
  - **Microscale Heat Transfer:** Simulating complex heat conduction phenomena in thin films and nanostructures exposed to ultrafast lasers, which is crucial for advanced manufacturing and electronics [1], [2].
- **Financial Engineering:** While seemingly distant, the mathematical formalism of convection-diffusion-reaction equations is analogous to the Black-Scholes equation for option pricing. LE-FVM techniques, particularly those capable of handling high-dimensional problems [9], could be adapted to develop more accurate and efficient models for pricing complex financial derivatives and managing risk [3].

9. Conclusion

The use of machine learning, such as deep learning in conjunction with the Finite Volume Method (FVM), represents a new era of solving nonlinear convection-diffusion problems numerically. The work in this paper is an initial step towards our vision for Learning-Enhanced Finite Volume Methods (LE-FVM), and has demonstrated that LE-FVM bears a fundamentally different philosophy from merely improving FVM without conceptual innovation; it integrates the solid, physics-based foundation of FVM with state-of-the-art data-driven pattern recognition by modern neural networks. Based on the theoretical and empirical literature developed recently [4], [9], [12], we analyze the performance of LE-FVM in application to GSV and demonstrate that LE-FVM achieves higher accuracy than traditional FVM in every application. Physics-Informed Neural Networks (PINNs) and their more recent advanced variants including Finite Volume Graph Networks (FVGN) and Physics-Informed Kolmogorov-Arnold Networks (PIKANs) have achieved outstanding performance in accurately capturing complex solution features such as sharp gradients, shock fronts, and turbulent structures with significantly lower error norms. In particular, Hasan et al. conducted a benchmark study of one-dimensional nonlinear PDEs Overall, [4] demonstrated ca. 100+ fold reductions in maximum absolute error factors with PINNs compared to mesh-based methods for the Burgers' equation, corresponding to an unprecedented level of accuracy that standard mesh-based methods fail to achieve except at prohibitively fine discretizations. Besides its accuracy, LE-FVM can also provide orders of magnitude of improvement in computational efficiency for certain workflows. Although training a physics-informed model is high-cost in the upfront, the inference phase is very rapid in comparison. After training, a model such as DeepONet or FVGN can make predictions of solutions for new parameters or initial conditions in milliseconds and without any full, iterative FVM simulation [9], [7]. Such capabilities open new avenues for applications that rely on fast parametric studies, controlling in real time, uncertainty quantification, or deploying digital twins, for which conventional techniques become computationally infeasible. For instance, the FVGN architecture used 77% more accurate predictions with 56% less training required compared to a baseline MeshGraphNet, showing that the accuracy-efficiency tradeoff is not always adversarial [9].

A variety of trends are meeting that will make LE-FVM a bountiful future. It can be initially employed to decrease the number of trainable parameters/noised data and it may also provide effective methods to design new, superior and more senstive neural architectures like KANs & PIKANs [12], [11]. Second, algorithmic breakthroughs like sPINNs and the revised SDGD methods have been attacking the "curse of dimensionality" head on by solving PDEs in tens or hundreds of thousands of dimensions which are beyond reach for classical FVM [19], [84]. Third, the field is gradually moving away from direct dependence on human intuition to guide training protocol, with a growing trend in adopting adaptive loss-weighting prior oracles [8], physics-guided sampling strategies [5] and hybrid training regimes that significantly speed-up convergence and increase robustness of models, particularly during nhanced-following initialization [2]. But the path to mass adoption is not without bumps. Introduction The "black-box" property of deep learning technique makes it difficult to interpret and trust the prediction results, particularly when considering safety-critical engineering applications. This highlights that hybrid ML-FVM schemes need to be tailor-made V&V procedures to ensure their robustness and reliability([7]). Lastly, the need for high quality training data and the large computational expense to train the model remains a significant drawback of this method as it restricts its use to complex multi-physics problems [13]. Addressing these can require joint efforts from the computational science community to produce well-defined benchmark data sets, open-source frameworks and claims-driven error estimator techniques. Takeaway: LEFVM is not just the new hotness in FVM, it's the next big thing. They preserve the key advantages of the approach — conservation and physical fidelity — but correct its classical limitations in terms of both realism for complex flows to be solved at high resolution, or efficiency for multi-query settings. It is anticipated that LE-FVM will open up the simulation of hitherto intractable problems and as research endeavours overcome these initial challenges and unveil new frontiers within algorithmic potential, LE-FVM will become an intrinsic feature of the scientific and engineering toolkit. The path from mesh to net is opened up, and it leads to a convergence that will transform computation.

References

[1] E. Tandis and P. Cardiff, "Finite volume-based supervised machine learning models for linear elastostatics," *Adv. Eng. Softw.*, vol. 175, p. 103258, Jan. 2023, doi: 10.1016/j.advengsoft.2022.103258.

[2] X. Chen, K. Zhang, Z. Ji, X. Shen, P. Liu, L. Zhang, J. Wang, and J. Yao, "Progress and Challenges of Integrated Machine Learning and Traditional Numerical Algorithms: Taking Reservoir Numerical Simulation as an Example," *Mathematics*, vol. 11, no. 21, p. 4418, 2023, doi: 10.3390/math11214418.

[3] A. Egon, L. Doris, and C. Bell, "Numerical Methods for Solving Nonlinear Partial Differential Equations in Fluid Dynamics," *HAL*, Feb. 2025. [Online]. Available: https://hal.science/hal-05094230v1

[4] F. Hasan, H. Ali, and H. A. Arief, "From Mesh to Neural Nets: A Multi-method Evaluation of Physics Informed Neural Network and Galerkin Finite Element Method for Solving Nonlinear Convection–Reaction–Diffusion Equations," *Int. J. Appl. Comput. Math*, vol. 11, p. 90, 2025, doi: 10.1007/s40819-025-01904-y.

[5] K. Shukla, J. D. Toscano, Z. Wang, Z. Zou, and G. E. Karniadakis, "From PINNs to PIKANs: Recent Advances in Physics-Informed Machine Learning," *arXiv*, Oct. 2024. [Online]. Available: https://arxiv.org/abs/2410.13228

[6] T. Kossaczká, M. Ehrhardt, and M. Günther, "Deep FDM: Enhanced finite difference methods by deep learning," *Results Appl. Math.*, vol. 17, p. 100336, Jan. 2023, doi: 10.1016/j.rinam.2022.100336.

[7] X. Jin and H. Li, "SONets: Sub-operator learning enhanced neural networks for solving parametric partial differential equations," *J. Comput. Phys.*, vol. 493, p. 112482, Dec. 2023, doi: 10.1016/j.jcp.2023.112482.

[8] A. Scott, "Integrating Finite-Volume Methods with Time Series Analysis for Enhanced Cancer Prevention Strategies," *HAL*, Jul. 2024. [Online]. Available: https://hal.science/hal-04663201v1

[9] T. Li, S. Zou, X. Chang, L. Zhang, and X. Deng, "Predicting unsteady incompressible fluid dynamics with finite volume informed neural network," *arXiv*, Mar. 2024. [Online]. Available: https://arxiv.org/abs/2309.10050

[10] "The Crunch Group - The collaborative research work of George Em Karniadakis," *Brown University*, Sep. 2025. [Online]. Available: https://sites.brown.edu/crunch-group/weekly-seminar-spotlight/

[11] "Recent Advances on Machine Learning for Computational Fluid Dynamics: A Survey," *arXiv*, Aug. 2024. [Online]. Available: https://arxiv.org/abs/2408.12171

[12] E. H. Quenjel, "Positive Scharfetter-Gummel finite volume method for convection-diffusion equations on polygonal meshes," *Appl. Math. Comput.*, vol. 425, p. 127071, Jul. 2022, doi: 10.1016/j.amc.2022.127071.

[13] Ł. Łach and D. Svyetlichnyy, "Advances in Numerical Modeling for Heat Transfer and Thermal Management: A Review of Computational Approaches and Environmental Impacts," *Energies*, vol. 18, no. 5, p. 1302, Jan. 2025, doi: 10.3390/en18051302.