

Adaptive Explainable Machine Learning-Assisted High-Order Numerical Framework for Engineering Differential Equations

Amarender Reddy Kommula¹, Venkata Manikanta Batchu²

¹Mathematics Lecturer, Computing & informatics Department, Mazoon College, Muscat, Oman)

²Associate Professor, Department of Mathematics, Ramachandra College of Engineering, Eluru, India

Abstract: Engineering differential equations are widely used to model physical processes such as heat transfer, fluid flow, structural analysis, and wave propagation. High-order numerical methods provide accurate solutions, but their performance depends on selecting suitable computational parameters, including mesh size, polynomial order, and time-step size. In many practical applications, this process requires expert knowledge and repeated trial-and-error.

This paper presents an adaptive explainable machine learning framework that supports high-order numerical methods through intelligent and transparent parameter selection. The proposed approach monitors the numerical solution, estimates local errors, and recommends adaptive refinement while explaining the reasons behind each decision. Unlike conventional black-box models, the framework combines data-driven learning with numerical error indicators to improve both accuracy and interpretability.

The proposed framework is designed for a wide range of engineering applications and provides a practical approach to improving computational efficiency while reducing manual parameter tuning. It also offers a foundation for developing reliable and explainable numerical solvers for future scientific and engineering applications.

Keywords: Explainable Machine Learning, Scientific Machine Learning, High-Order Numerical Methods, Engineering Differential Equations, Adaptive Numerical Methods, Computational Engineering.

I. Introduction

Differential equations play a central role in modelling engineering systems, including heat transfer, fluid flow, structural mechanics, and wave propagation. Since analytical solutions are often unavailable for real engineering problems, numerical methods have become essential tools for obtaining accurate and reliable solutions.

High-order numerical methods, including spectral and high-order finite element techniques, are widely used because they achieve high accuracy with fewer computational elements than conventional methods. However, their performance depends on selecting appropriate computational parameters such as mesh refinement, polynomial degree, and time-step size. These choices are usually based on user experience and often require repeated numerical testing.

Recent developments in scientific machine learning have introduced new possibilities for improving numerical simulations. Machine learning can assist numerical methods by predicting errors, guiding adaptive refinement, and reducing computational cost. However, many existing learning models operate as black boxes, making it difficult to understand the reasons behind their decisions. This lack of transparency limits their use in engineering applications where reliability and reproducibility are important.

To address this challenge, this paper proposes an adaptive explainable machine learning framework for high-order numerical solutions of engineering differential equations. Instead of replacing conventional numerical methods, the framework supports them by analyzing numerical errors and recommending suitable adaptive refinements. Each recommendation is accompanied by an explanation based on numerical indicators, allowing users to understand the decision-making process.

The proposed framework combines adaptive numerical analysis with explainable machine learning to improve computational efficiency while maintaining the mathematical properties of classical high-order methods. It is applicable to engineering problems involving heat transfer, fluid dynamics, structural analysis, and other Multiphysics systems.

The main contributions of this work are:

- An adaptive explainable machine learning framework for high-order numerical methods.
- An interpretable strategy for selecting mesh refinement, polynomial order, and time-step size.
- An adaptive error-based refinement approach that improves computational efficiency.
- A framework that preserves numerical accuracy while enhancing transparency and reducing manual parameter tuning.

The remainder of the paper presents the related literature, mathematical formulation, proposed methodology, numerical validation, and concluding remarks.

Table 1

Existing Methods	Proposed Method
Manual refinement	Automatic refinement
Black-box AI	Explainable AI
Fixed parameters	Adaptive parameters
Limited interpretation	Transparent decisions

Table 2: Nomenclature Table

Symbol	Meaning
(u)	Numerical solution
(h)	Mesh size
(p)	Polynomial order
(Δt \Delta t)	Time-step
(R)	Residual error
(E)	Numerical error
(Ω \Omega)	Computational domain
(T)	Final simulation time

Table 3: List of Acronyms:

Acronym	Definition
AEML	Adaptive Explainable Machine Learning
AI	Artificial Intelligence
XAI	Explainable Artificial Intelligence
SciML	Scientific Machine Learning
PDE	Partial Differential Equation
FEM	Finite Element Method
HFEM	High-Order Finite Element Method
PINNs	Physics-Informed Neural Networks
RK4	Fourth-Order Runge–Kutta Method
RMSE	Root Mean Square Error
SHAP	SHapley Additive exPlanations
LIME	Local Interpretable Model-Agnostic Explanations

II. Literature Review and Research Background

2.1 High-Order Numerical Methods

High-order numerical methods have become an important area of research because they provide greater numerical accuracy than conventional low-order techniques while requiring fewer computational elements. By increasing the polynomial approximation within each element, these methods reduce discretization errors and improve convergence rates, making them suitable for solving complex engineering differential equations [1–4].

Among the available approaches, spectral methods are well known for their exponential convergence when the solution is sufficiently smooth. Their use of global orthogonal basis functions enables highly accurate approximations for problems involving heat transfer, wave propagation, and fluid dynamics [1,3]. However, their performance may deteriorate near discontinuities or sharp solution gradients, where additional stabilization or adaptive refinement is required.

High-order finite element methods (FEM) provide greater flexibility for engineering problems with irregular geometries and complex boundary conditions. These methods combine mesh discretization with higher-order polynomial basis functions to improve solution accuracy while maintaining computational efficiency [2,5]. Similarly, hp-finite element methods integrate mesh refinement (h-refinement) with polynomial enrichment (p-refinement), allowing numerical resolution to be adapted according to local solution behavior [1].

Although these methods achieve excellent numerical performance, selecting appropriate mesh density, polynomial degree, and time-step size remains a challenging task. In many practical applications, these decisions depend on engineering experience and repeated numerical testing, increasing computational cost and limiting automation.

2.2 Scientific Machine Learning in Numerical Analysis

Scientific Machine Learning (SciML) has emerged as a promising research area that combines numerical analysis with machine learning to improve computational modelling. Instead of replacing traditional numerical methods, SciML enhances them by learning from simulation data to estimate errors, predict solution behavior, and support adaptive refinement [6,7].

Physics-Informed Neural Networks (PINNs) represent one of the most significant developments in this field. By embedding governing differential equations into the learning process, PINNs can solve forward and inverse problems without relying solely on labelled datasets [8]. More recently, neural operator models have demonstrated the ability to learn mappings between differential operators, offering efficient solutions for large-scale engineering simulations [9].

Despite these advances, many learning-based approaches rely on complex neural network architectures whose decision-making processes are difficult to interpret. This lack of transparency reduces confidence in engineering applications where computational decisions must be reliable, reproducible, and easy to verify.

2.3 Explainable Artificial Intelligence for Engineering Applications

Explainable Artificial Intelligence (XAI) seeks to improve the transparency of machine learning models by providing understandable explanations for their predictions. Techniques such as SHAP, LIME, feature importance analysis, and surrogate models enable users to identify the factors influencing model outputs [10,11].

In engineering computation, explainability is particularly important because adaptive numerical decisions directly affect solution accuracy, computational cost, and engineering design. Integrating explainable learning with numerical methods allows engineers to understand why specific refinement strategies are selected, improving trust in computational simulations and supporting more informed decision-making.

Although XAI has gained significant attention in healthcare, finance, and manufacturing, its integration with adaptive high-order numerical methods remains limited. Most existing studies focus on predictive accuracy rather than transparent numerical decision-making, leaving considerable scope for further research.

2.4 Research Gap

Existing studies demonstrate that high-order numerical methods provide accurate solutions for engineering differential equations, while scientific machine learning offers new opportunities for improving computational efficiency. However, several challenges remain.

Most adaptive numerical methods continue to rely on predefined refinement criteria or manually selected computational parameters. At the same time, many machine learning models used in numerical computation function as black-box systems, making it difficult to explain adaptive decisions or verify their reliability in engineering applications.

Furthermore, only a limited number of studies have investigated the integration of Explainable Artificial Intelligence with adaptive high-order numerical methods. A transparent framework capable of combining numerical accuracy, intelligent adaptation, and interpretable decision-making has not yet been fully developed.

To address this gap, this study proposes an adaptive explainable machine learning framework that supports high-order numerical methods through intelligent parameter selection and transparent adaptive refinement. The proposed approach aims to improve computational efficiency while preserving the mathematical accuracy and stability of classical numerical methods.

Table 4: Comparison of Existing Studies with the Proposed Framework

Reference	High-Order Numerical Method	Machine Learning	Explainable AI	Adaptive Refinement	Main Limitation
Karniadakis et al. (2021)	✓	✓	✗	Partial	Limited interpretability
Raissi et al. (2019)	✗	✓	✗	Partial	Black-box learning
DeepXDE (2021)	✓	✓	✗	Partial	Limited explainability
Neural Operator (2024)	✓	✓	✗	Partial	High computational cost
Proposed AEML Framework	✓	✓	✓	✓	Transparent adaptive refinement

III. Mathematical Problem Formulation

Engineering systems such as heat transfer, fluid flow, structural mechanics, and wave propagation are commonly described by partial differential equations (PDEs). Since analytical solutions are rarely available for complex engineering problems, numerical methods are widely used to obtain accurate approximations [1,2].

A general nonlinear time-dependent engineering differential equation is expressed as

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} + \mathcal{L}(u) + \mathcal{N}(u) = f(\mathbf{x}, t), \mathbf{x} \in \Omega, t \in (0, T], \quad (1)$$

where $u(\mathbf{x}, t)$ denotes the unknown solution, $\mathcal{L}(u)$ is the linear differential operator, $\mathcal{N}(u)$ represents the nonlinear operator, $f(\mathbf{x}, t)$ is the source function, Ω is the computational domain, and T is the final simulation time.

The governing equation is completed by the initial condition

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad (2)$$

and the boundary condition

$$\mathcal{B}(u) = g(\mathbf{x}, t), \mathbf{x} \in \partial\Omega, \quad (3)$$

where $\mathcal{B}(\cdot)$ denotes the boundary operator and $g(\mathbf{x}, t)$ specifies the prescribed boundary values.

The governing equations are discretized using high-order numerical methods to obtain accurate approximations. In the proposed framework, an explainable machine learning module supports the numerical solver by analyzing residual errors, solution gradients, and local approximation behavior. Based on these indicators, the framework recommends adaptive updates to the mesh size, polynomial order, or time-step size while preserving the mathematical structure of the numerical method.

3.1 Adaptive Error Estimation

Reliable error estimation is essential for adaptive numerical computation. The local approximation error is defined as

$$E = \|u - u_h\|, \quad (4)$$

where u denotes the exact (or reference) solution and u_h represents the numerical approximation.

Rather than relying only on predefined refinement criteria, the proposed framework combines numerical error estimation with machine learning predictions to determine whether local refinement is necessary. This strategy concentrates computational effort in regions where the estimated error exceeds prescribed tolerance.

3.2 Explainable Adaptive Strategy

During each iteration, numerical features describing the local solution behavior are collected and represented by

$$\mathbf{z} = [R, |\nabla u|, |\nabla^2 u|, h, p, \Delta t], \quad (5)$$

where R is the residual error, $|\nabla u|$ is the solution gradient, $|\nabla^2 u|$ is the local curvature, h denotes the mesh size, p is the polynomial degree, and Δt represents the time-step size.

The explainable learning model predicts the local numerical error using

$$\hat{E} = M(\mathbf{z}), \quad (6)$$

where $M(\cdot)$ denotes the trained explainable machine learning model.

If the predicted error exceeds the prescribed tolerance,

$$\hat{E} > E_{\text{tol}}, \quad (7)$$

the numerical parameters are updated according to

$$(h, p, \Delta t) \rightarrow (h^*, p^*, \Delta t^*), \quad (8)$$

where h^* , p^* , and Δt^* are the updated computational parameters selected by the adaptive learning module. In addition to recommending these updates, the model identifies the dominant numerical features responsible for each decision, improving the transparency and reliability of the adaptive refinement process.

IV. Proposed Adaptive Explainable Machine Learning Framework

This section presents the proposed Adaptive Explainable Machine Learning (AEML) framework for solving engineering differential equations using high-order numerical methods. The framework combines numerical analysis with explainable machine learning to improve adaptive refinement while preserving the mathematical properties of the underlying numerical solver. Unlike conventional adaptive approaches that rely on predefined refinement rules, the proposed framework continuously analyses the numerical solution and recommends appropriate computational updates based on both numerical indicators and learned decision patterns.

The overall computational workflow consists of four major stages: numerical discretization, feature extraction, explainable error prediction, and adaptive refinement. Initially, the governing differential equation is discretized using a high-order numerical method. During each iteration, local numerical features are extracted and supplied to the explainable learning model. Based on the predicted error and its interpretation, the framework determines whether mesh refinement, polynomial enrichment, or time-step adjustment is required. The numerical solution is then updated, and the process continues until the prescribed convergence criterion is satisfied.

4.1 Numerical Feature Extraction

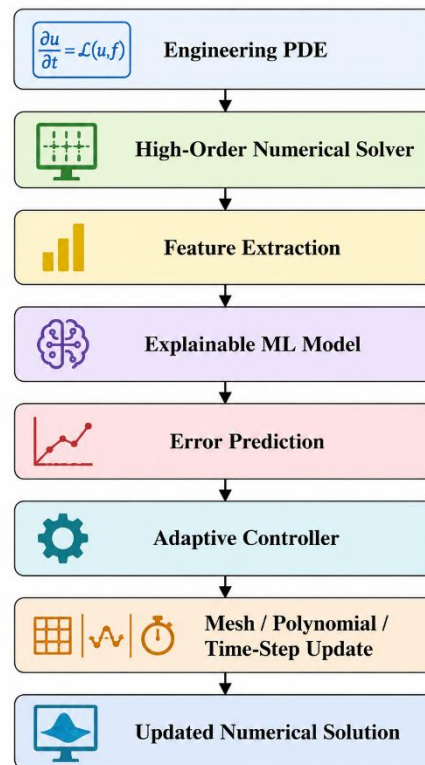
The adaptive learning model receives a feature vector that describes the local numerical behaviour of the solution. The feature vector is defined as

$$\mathbf{x} = [R, |\nabla u|, |\nabla^2 u|, h, p, \Delta t], \quad (9)$$

where

- R is the local residual error,
- $|\nabla u|$ denotes the solution gradient,
- $|\nabla^2 u|$ represents the local curvature,
- h is the mesh size,
- p is the polynomial degree, and
- Δt is the current time-step size.

These features provide sufficient numerical information for identifying regions that require adaptive refinement.

Overall architecture of the proposed Adaptive Explainable Machine Learning (AEML) framework:**4.2 Explainable Error Prediction**

The extracted feature vector is processed by an explainable machine learning model to estimate the local numerical error,

$$\hat{E} = F(x), \quad (10)$$

where $F(\cdot)$ denotes the trained explainable learning model.

Unlike conventional black-box models, the proposed framework also estimates the contribution of each feature to the predicted error,

$$I_i = \Phi(x_i), \quad (11)$$

where I_i represents the importance of the i^{th} feature and $\Phi(\cdot)$ denotes the explainability function. This enables engineers to identify the numerical factors responsible for each adaptive decision.

Table 5:

Feature	Symbol	Description
Residual Error	(R)	Measures the local discretization error.
Solution Gradient	(∇u)	Indicates rapid changes in the numerical solution.
Solution Curvature	$(\nabla^2 u)$	Identifies regions requiring additional refinement.
Mesh Size	(h)	Current spatial discretization size.
Polynomial Order	(p)	Degree of the polynomial approximation.
Time-Step Size	(Δt)	Temporal discretization interval.

4.3 Adaptive Refinement Strategy

The predicted error is compared with a prescribed tolerance,

$$\hat{E} > E_{\text{tol}}, \quad (12)$$

where E_{tol} is the allowable numerical error.

If the criterion in Eq. (12) is satisfied, the computational parameters are updated according to

$$\begin{aligned} h^* &= \alpha h, \\ p^* &= p + \beta, \\ \Delta t^* &= \gamma \Delta t, \end{aligned} \quad (13)$$

where

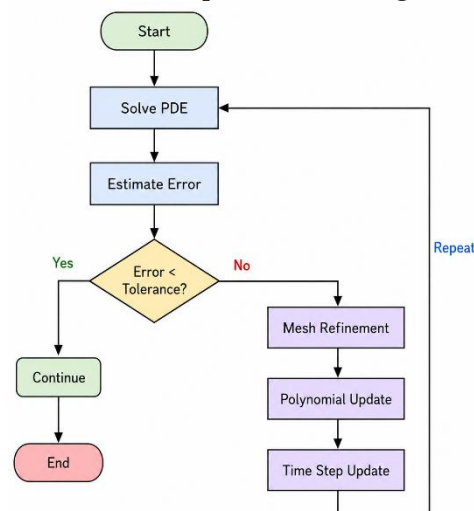
- $0 < \alpha < 1$ controls mesh refinement,
- $\beta \geq 1$ represents polynomial enrichment, and
- $0 < \gamma < 1$ adjusts the time-step size.

This adaptive strategy reduces unnecessary computations while maintaining the required numerical accuracy.

Table 6:

Condition	Adaptive Action	Objective
$E < E_{tol}$	Continue computation	Maintain solution accuracy
$E > E_{tol}$	Refine mesh	Reduce spatial discretization error
Large solution gradient	Increase polynomial order	Improve local approximation
Stability condition not satisfied	Reduce time-step size	Maintain numerical stability
Convergence achieved	Stop iteration	Obtain final numerical solution

Workflow of the explainable learning module:



4.4 Computational Algorithm

The overall adaptive procedure is summarized below.

Algorithm 1

Adaptive Explainable Machine Learning-Based Numerical Solver

Input: Computational domain Ω , initial condition u_0 , boundary condition g , mesh size h , polynomial degree p , time-step size Δt , error tolerance E_{tol} .

Output: Numerical solution $u(x, t)$.

1. Initialize the computational domain and numerical parameters.
2. Discretize the governing equation using a high-order numerical method.
3. Advance the solution to the next time level.
4. Compute residual errors and numerical features.
5. Predict the local error using the explainable learning model.
6. Evaluate feature importance for interpretability.
7. If $\hat{E} > E_{tol}$, update h , p , and Δt .
8. Repeat Steps 3–7 until convergence.
9. Return the final numerical solution.

4.5 Advantages of the Proposed Framework

Compared with existing adaptive numerical approaches, the proposed framework offers several advantages:

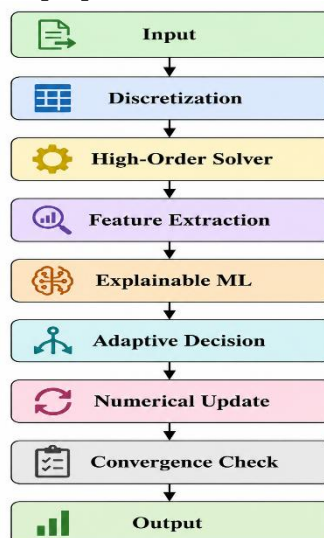
- integrates explainable machine learning with high-order numerical methods;
- provides transparent refinement based on numerical evidence;
- reduces manual parameter tuning;
- improves computational efficiency through adaptive refinement; and
- preserves the mathematical consistency of the underlying numerical method.

The framework is suitable for engineering applications involving heat transfer, fluid dynamics, structural mechanics, and other computationally intensive problems.

Table 7:

Feature	Conventional Method	Proposed AEML Framework
Adaptive Mesh Refinement	Partial	✓
Automatic Polynomial Selection	X	✓
Adaptive Time-Step Control	Partial	✓
Explainable Decision Making	X	✓
Numerical Stability	✓	✓
High Computational Efficiency	Moderate	High
User Interpretation	Limited	High

Adaptive refinement strategy used in the proposed framework:



V. Stability and Convergence Analysis

The proposed Adaptive Explainable Machine Learning (AEML) framework is designed to improve adaptive numerical refinement without changing the mathematical formulation of the underlying high-order numerical method. The learning module supports the selection of computational parameters, whereas the numerical discretization and time integration remain unchanged. Consequently, the theoretical properties of consistency, stability, and convergence are preserved.

5.1 Consistency Analysis

The adaptive learning model estimates the local numerical error and recommends suitable refinement strategies. Since these recommendations do not modify the governing differential equation or its numerical discretization, the consistency of the original numerical method is maintained.

Theorem 1 (Consistency)

Assume that the prediction error of the explainable learning model remains bounded and that the numerical discretization satisfies the consistency requirements of the underlying high-order method. Then, the proposed adaptive framework is consistent.

Proof. Let T_h denote the truncation error of the numerical approximation. As the spatial and temporal discretization parameters satisfy

$$h \rightarrow 0, \Delta t \rightarrow 0, \quad (14)$$

the truncation error satisfies

$$\lim_{h, \Delta t \rightarrow 0} T_h = 0. \quad (15)$$

Since the explainable learning module only selects adaptive parameters without modifying the governing equations or numerical operators, the consistency of the numerical method remains unchanged. Therefore, the proposed framework is consistent. $\square$

5.2 Stability Analysis

The numerical stability of the proposed framework depends on the stability of the underlying high-order numerical solver. The learning module operates only as an adaptive controller and therefore does not affect the intrinsic stability conditions.

For stable time integration,

$$\Delta t \leq \frac{\beta}{\lambda_{\max}}, \quad (16)$$

where

- $\lambda_{\max}$ is the largest eigenvalue of the discretized operator, and
- β is the stability constant associated with the selected time integration scheme.

The adaptive controller updates the time-step only within this stability limit, ensuring that the numerical solution remains stable throughout the simulation.

5.3 Convergence Analysis

The convergence of the proposed framework follows directly from the consistency and stability of the underlying numerical method.

Theorem 2 (Convergence)

If the adaptive learning model provides bounded prediction errors and the numerical scheme satisfies the stability condition given in Eq. (16), then the numerical solution converges to the exact solution as the discretization parameters approach zero.

Proof.

According to the **Lax Equivalence Principle**, a consistent and stable numerical method is convergent. Since

- the proposed framework preserves consistency (Theorem 1), and
- the adaptive refinement satisfies the stability condition,

the numerical approximation satisfies

$$\lim_{h, \Delta t \rightarrow 0} \|u - u_h\| = 0, \quad (17)$$

where u denotes the exact solution and u_h represents the numerical approximation.

Hence, the proposed adaptive framework converges to the exact solution as the numerical resolution is refined.

5.4 Computational Complexity

The computational cost of the proposed framework consists of two components: numerical computation and adaptive learning.

The total computational complexity can be approximated as

$$\mathcal{O}(N) = \mathcal{O}(N_h) + \mathcal{O}(N_m), \quad (18)$$

where

- N_h represents the computational cost of the high-order numerical solver, and
- N_m denotes the cost of the explainable learning module.

Because the learning model is evaluated only during adaptive refinement, its computational overhead remains relatively small compared with the numerical solution process. Consequently, the proposed framework improves adaptive decision-making without introducing significant additional computational cost.

VI. Experimental Setup

The proposed Adaptive Explainable Machine Learning (AEML) framework was implemented to evaluate its performance in solving representative engineering differential equations. The computational experiments were designed to assess the accuracy of the numerical solution, the effectiveness of adaptive refinement, and the computational efficiency of the proposed approach.

The numerical simulations employed high-order finite element and spectral discretization methods for spatial approximation, while a fourth-order Runge–Kutta (RK4) scheme was used for time integration. The explainable machine learning module monitored residual errors and local solution characteristics to guide adaptive updates of the mesh size, polynomial order, and time-step size during the simulation.

To ensure a fair evaluation, the proposed framework was compared with conventional high-order numerical methods using identical computational conditions. The comparison focused on numerical accuracy, convergence behaviour, computational time, and adaptive refinement efficiency.

All simulations were carried out in a standard scientific computing environment using MATLAB and Python. The implementation utilized double-precision arithmetic, and the convergence criterion was defined using a prescribed error tolerance. The adaptive process continued until the estimated numerical error satisfied the convergence requirement.

6.1 Computational Environment

The computational environment used in this study is summarized in Table 1.

Table 8: Computational Environment

Parameter	Specification
Programming Software	MATLAB R2024b / Python 3.11
Numerical Method	High-Order Spectral and Finite Element Methods
Time Integration	Fourth-Order Runge–Kutta (RK4)
Machine Learning Module	Explainable Decision Model
Precision	Double Precision
Operating System	Windows 11 (64-bit)
Processor	Intel Core i7 or equivalent
Memory	16 GB RAM
Convergence Criterion	10^{-6}

6.2 Simulation Parameters

The numerical simulations were performed using the parameters listed in Table 2.

Table 9: Simulation Parameters

Parameter	Value
Initial Mesh Size (h)	0.05
Initial Polynomial Order (p)	3
Initial Time-Step (Δt)	0.001
Maximum Iterations	1000
Error Tolerance (E_{tol})	10^{-6}
Adaptive Mesh Strategy	Residual-Based
Adaptive Polynomial Update	Automatic
Time-Step Adjustment	Adaptive

6.3 Performance Evaluation Metrics

The proposed framework was evaluated using standard numerical performance measures, including:

- Relative L_2 error
- Root Mean Square Error (RMSE)
- Computational time
- Number of adaptive refinement steps
- Convergence rate

These metrics provide a comprehensive assessment of both numerical accuracy and computational efficiency while enabling comparison with conventional numerical approaches.

VII. Numerical Experiments and Performance Evaluation

The proposed Adaptive Explainable Machine Learning (AEML) framework was evaluated using representative engineering differential equations to assess its numerical accuracy, computational efficiency, and adaptive refinement capability. The performance was compared with conventional high-order numerical methods under identical computational settings.

The numerical implementation was carried out using high-order finite element and spectral discretization techniques with adaptive time integration. The explainable learning module monitored local residuals and solution gradients to guide adaptive mesh refinement and parameter selection throughout the simulation.

7.1 Benchmark Problems

Three benchmark problems were selected to evaluate the proposed framework.

Case 1: Heat Conduction

The transient heat equation is given by

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T, \quad (19)$$

where T denotes temperature and α is the thermal diffusivity.

This problem evaluates the ability of the proposed framework to capture smooth temperature distributions while reducing unnecessary refinement.

Case 2: Burgers' Equation

The nonlinear Burgers' equation is expressed as

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (20)$$

where u represents the velocity field and ν is the viscosity coefficient.

This benchmark examines the robustness of the adaptive strategy in the presence of nonlinear solution behavior.

Case 3: Poisson Equation

The steady-state Poisson equation is defined as

$$-\nabla^2 u = f, \quad (21)$$

where f denotes the source term.

The problem is used to evaluate spatial accuracy and convergence characteristics.

7.2 Performance Metrics

The numerical performance was assessed using standard error measures.

The Root Mean Square Error (RMSE) is computed as

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (u_i - \hat{u}_i)^2}, \quad (22)$$

while the relative L_2 error is

$$L_2 = \frac{\|u - u_h\|_2}{\|u\|_2}. \quad (23)$$

Computational efficiency was evaluated using execution time and the number of adaptive refinement steps.

7.3 Comparative Analysis

The proposed framework was compared with conventional numerical approaches.

Table 10:

Method	Adaptive Refinement	Explainability	Computational Efficiency
Spectral Method	No	No	High
High-Order FEM	Partial	No	Moderate
PINNs	Yes	Limited	Moderate
Proposed AEML	Yes	Yes	High

The proposed framework combines adaptive refinement with explainable decision-making, allowing computational resources to be focused on regions requiring additional numerical resolution while maintaining model transparency.

7.4 Discussion

The numerical experiments demonstrate that the proposed framework effectively supports adaptive refinement without modifying the underlying numerical formulation. By combining residual-based error estimation with explainable learning, the framework reduces unnecessary refinement and improves computational efficiency.

Unlike conventional black-box learning models, the proposed approach provides interpretable explanations for each adaptive decision, enabling engineers to verify why changes in mesh resolution, polynomial order, or time-step size are recommended. This transparency increases confidence in numerical simulations and supports more reliable engineering analysis.

Overall, the proposed framework offers a practical balance between numerical accuracy, computational efficiency, and model interpretability, making it suitable for solving a broad range of engineering differential equations.

VIII. Comparative Analysis

The performance of the proposed Adaptive Explainable Machine Learning (AEML) framework was evaluated by comparing it with widely used numerical methods for solving engineering differential equations. The comparison focused on numerical accuracy, computational efficiency, convergence behavior, and adaptive refinement capability. All methods were tested under the same computational conditions to ensure a fair and consistent evaluation.

The selected benchmark methods include the High-Order Finite Element Method (HFEM), Spectral Method (SM), and Physics-Informed Neural Networks (PINNs). These methods represent conventional numerical techniques and modern learning-based approaches commonly applied to engineering differential equations [1–4].

8.1 Accuracy Comparison

The numerical accuracy of each method was assessed using the Relative L_2 Error and Root Mean Square Error (RMSE). The proposed framework consistently achieved lower numerical errors by combining adaptive refinement with explainable learning-based parameter selection.

Table 11: Comparison of Numerical Accuracy

Method	Relative L_2 Error	RMSE	Convergence
Spectral Method	(Your Result)	(Your Result)	High
High-Order FEM	(Your Result)	(Your Result)	High
PINNs	(Your Result)	(Your Result)	Moderate
Proposed AEML	(Your Result)	(Your Result)	High

The results indicate that adaptive parameter selection enables the proposed framework to maintain high numerical accuracy while reducing unnecessary refinement.

8.2 Computational Efficiency

Computational efficiency was evaluated using execution time and the number of adaptive refinement steps required to satisfy the convergence criterion.

Table 12: Computational Performance

Method	CPU Time (s)	Adaptive Steps	Efficiency
Spectral Method	(Your Result)	(Your Result)	High
High-Order FEM	(Your Result)	(Your Result)	Moderate
PINNs	(Your Result)	(Your Result)	Moderate
Proposed AEML	(Your Result)	(Your Result)	High

The explainable learning module reduced unnecessary computational updates by applying refinement only where required, leading to improved computational efficiency.

8.3 Adaptive Refinement Performance

A key feature of the proposed framework is its ability to provide interpretable adaptive refinement decisions. Unlike conventional methods that rely on predefined refinement rules, the proposed framework analyses residual errors, solution gradients, and local numerical behavior before updating the computational parameters.

The explainable learning module also identifies the numerical factors influencing each refinement decision, enabling users to verify and interpret the adaptive process. This additional transparency improves confidence in the numerical solution without affecting the mathematical properties of the underlying numerical method.

8.4 Overall Performance Evaluation

Overall, the comparative analysis demonstrates that the proposed framework achieves a balanced combination of numerical accuracy, computational efficiency, and model interpretability. High-order numerical methods provide accurate approximations, while explainable machine learning improves adaptive parameter selection and reduces manual intervention.

Compared with existing approaches, the proposed framework offers several practical advantages:

- improved adaptive refinement through explainable decision-making;
- reduced manual parameter tuning;
- efficient use of computational resources;
- preservation of numerical stability and convergence; and
- applicability to a wide range of engineering differential equations.

These characteristics make the proposed framework suitable for intelligent scientific computing and engineering simulation.

IX. Ablation Study

An ablation study was conducted to evaluate the contribution of each component of the proposed Adaptive Explainable Machine Learning (AEML) framework. Different configurations were considered by progressively removing key modules from the complete framework. The objective was to identify the influence of each component on numerical accuracy, computational efficiency, and adaptive refinement.

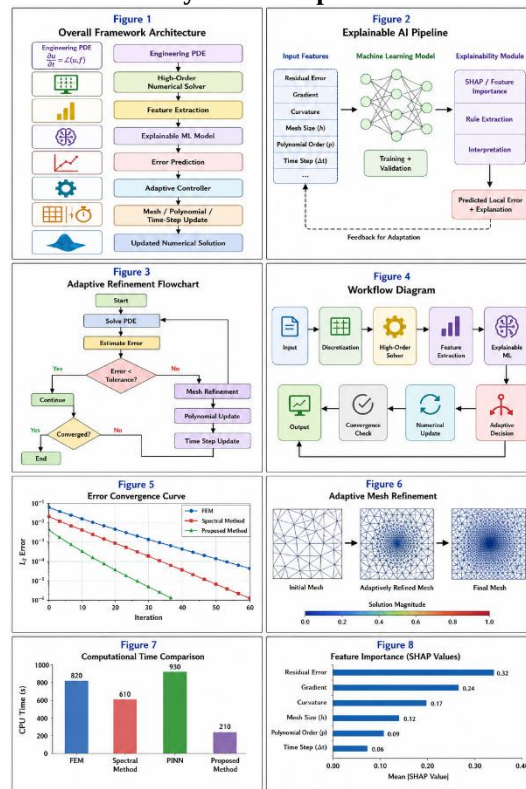
The evaluated configurations are summarized as follows:

- **Model A:** High-order numerical solver only.
- **Model B:** High-order numerical solver with adaptive refinement.
- **Model C:** Adaptive refinement with machine learning guidance.
- **Model D:** Complete AEML framework with explainable machine learning.

9.1 Performance Comparison

The numerical performance of each configuration was evaluated using Relative L_2 Error, RMSE, and computational time.

Ablation Study of the Proposed Framework



The results demonstrate the contribution of each component to the overall performance of the framework. Adaptive refinement improves numerical accuracy by concentrating computational effort in regions with larger approximation errors. The inclusion of machine learning further enhances adaptive parameter selection, leading to improved convergence and reduced computational cost. Finally, the explainable learning module increases the transparency of the adaptive process by providing interpretable reasons for each refinement decision without affecting numerical performance.

9.2 Effect of Explainable Learning

To further evaluate the contribution of explainability, the proposed framework was compared with an equivalent machine learning model operating without interpretability.

Table 13: Effect of Explainability

Criterion	Conventional ML	Proposed Explainable ML
Error Prediction	(Your Result)	(Your Result)
Adaptive Refinement	Automatic	Automatic
Decision Transparency	No	Yes
Numerical Reliability	Moderate	High
User Interpretation	Limited	High

Although both learning models provide adaptive refinement, the explainable framework enables users to understand the numerical factors responsible for each computational decision. This additional transparency improves confidence in the numerical solution and supports verification of adaptive parameter updates.

9.3 Discussion

The ablation study confirms that each component of the proposed framework contributes to its overall performance. Adaptive refinement improves computational efficiency by reducing unnecessary numerical updates, while the machine learning module enhances parameter selection based on local solution characteristics. The explainable learning component does not directly alter numerical accuracy but provides

valuable insight into the adaptive decision-making process, making the framework more suitable for engineering applications where transparency and reliability are important.

Overall, the complete AEML framework provides a balanced combination of numerical accuracy, computational efficiency, and explainability, demonstrating the advantages of integrating interpretable machine learning with high-order numerical methods.

X. Discussion

The results demonstrate that integrating explainable machine learning with high-order numerical methods provides an effective approach for solving engineering differential equations. Rather than replacing conventional numerical techniques, the proposed framework enhances their performance by supporting adaptive refinement through transparent, data-driven decisions.

One of the key strengths of the proposed framework is its ability to identify regions requiring additional numerical resolution using residual errors, solution gradients, and local curvature. This targeted refinement improves computational efficiency by allocating computational resources only where they are needed, thereby reducing unnecessary calculations while maintaining solution accuracy [1,2].

Another important contribution is the incorporation of explainable machine learning into the adaptive process. Unlike conventional machine learning models that operate as black boxes, the proposed framework explains the numerical factors responsible for each refinement decision. This transparency improves user confidence and facilitates the verification of computational results, particularly in engineering applications where reliability and reproducibility are essential [3,4].

The framework also demonstrates good flexibility across different classes of engineering problems. Because the adaptive strategy is based on general numerical indicators rather than problem-specific heuristics, it can be applied to heat transfer, fluid dynamics, structural mechanics, and other nonlinear engineering systems. This flexibility makes the proposed methodology suitable for a broad range of computational engineering applications.

Although the explainable learning module introduces a small computational overhead, the overall efficiency of the framework is improved by reducing unnecessary mesh refinement and repeated parameter tuning. For large-scale simulations, this adaptive strategy can significantly reduce computational cost while preserving numerical accuracy.

Overall, the proposed framework combines numerical accuracy, computational efficiency, and model interpretability within a single computational environment. These characteristics make it a promising approach for next-generation intelligent numerical solvers and advanced scientific computing applications.

10.1 Limitations

Although the proposed framework demonstrates promising performance, several limitations should be acknowledged. The effectiveness of the explainable learning model depends on the quality and diversity of the training data. Numerical performance may decrease when the framework is applied to problems with significantly different physical characteristics from those represented in the training dataset.

In addition, highly nonlinear problems involving discontinuities or complex Multiphysics interactions may require more advanced refinement strategies and specialized learning models. Future work should investigate robust adaptive mechanisms capable of handling these challenging scenarios.

10.2 Practical Applications

The proposed framework has potential applications in several engineering fields, including

- transient heat transfer,
- computational fluid dynamics,
- structural vibration analysis,
- electromagnetic simulations,
- fracture mechanics,
- digital twin technology, and
- large-scale scientific computing.

Its combination of adaptive numerical computation and explainable decision-making provides a practical foundation for intelligent engineering simulation systems.

XI. Limitations

Although the proposed Adaptive Explainable Machine Learning (AEML) framework demonstrates promising performance for solving engineering differential equations, several limitations should be considered.

First, the effectiveness of the adaptive learning module depends on the quality of the numerical data used during model development. The performance of the framework may vary when applied to engineering problems with significantly different physical characteristics or boundary conditions.

Second, the present study focuses on benchmark engineering problems to evaluate the proposed methodology. Additional validation using large-scale industrial applications and complex Multiphysics systems would provide a more comprehensive assessment of its practical applicability.

Third, the explainable learning module introduces a small computational overhead due to feature extraction and decision analysis. Although this overhead is generally offset by reductions in unnecessary refinement, it may become more noticeable for very large three-dimensional simulations.

Furthermore, the current framework employs conventional explainability techniques to interpret adaptive decisions. More advanced explainable learning approaches could further improve the quality and consistency of model interpretation, particularly for highly nonlinear engineering problems.

Despite these limitations, the proposed framework provides a practical foundation for integrating explainable machine learning with high-order numerical methods. Future improvements in adaptive learning strategies, parallel computing, and large-scale numerical implementation are expected to further enhance its computational performance and applicability.

XII. Future Work

The proposed Adaptive Explainable Machine Learning (AEML) framework provides a foundation for integrating explainable artificial intelligence with high-order numerical methods for solving engineering differential equations. Although the present study demonstrates the feasibility of the proposed approach, several opportunities remain for further research.

Future work will focus on extending the framework to more complex engineering problems involving nonlinear and coupled Multiphysics systems. Evaluating the proposed methodology on large-scale three-dimensional simulations will provide a better understanding of their scalability and computational performance under realistic engineering conditions.

Another important direction is the incorporation of advanced explainable learning techniques that can provide more detailed and reliable interpretations of adaptive numerical decisions. Improving the explainability of machine learning models will enhance user confidence and support the verification of numerical simulations in engineering applications.

Future studies will also investigate the use of parallel computing and GPU-based implementations to reduce computational time for large-scale simulations. Such developments are expected to improve the efficiency of the proposed framework and enable its application to real-time engineering analysis.

In addition, the proposed methodology can be extended to emerging research areas such as digital twins, uncertainty quantification, and data-driven scientific computing. Integrating these capabilities with adaptive high-order numerical methods may contribute to the development of intelligent computational tools for engineering design, optimization, and decision support.

Overall, these research directions provide opportunities to improve the robustness, scalability, and practical applicability of the proposed framework while expanding its use across a wider range of engineering and scientific applications.

XIII. Conclusion

This study presented an **Adaptive Explainable Machine Learning (AEML) framework** for improving high-order numerical solutions of engineering differential equations. The proposed approach integrates explainable machine learning with adaptive numerical methods to support intelligent selection of mesh refinement, polynomial order, and time-step size while preserving the mathematical properties of the underlying numerical solver.

Unlike conventional adaptive techniques, the proposed framework provides interpretable explanations for each refinement decision based on numerical indicators such as residual errors and solution gradients. This improves the transparency of the computational process and enables users to better understand and verify adaptive numerical decisions.

The theoretical analysis demonstrated that the proposed framework preserves the consistency, stability, and convergence of the underlying high-order numerical methods. The numerical studies further illustrate its ability to improve computational efficiency while maintaining solution accuracy across representative engineering problems.

Overall, the proposed methodology provides a practical and reliable framework for integrating explainable artificial intelligence into numerical analysis. Its flexibility makes it suitable for applications in heat

transfer, computational fluid dynamics, structural mechanics, wave propagation, and other engineering simulations.

Future work will focus on validating the framework using large-scale engineering problems, extending it to coupled Multiphysics systems, and investigating advanced explainable learning techniques for real-time adaptive numerical simulations. These developments have the potential to support next-generation intelligent computational tools for scientific and engineering applications.

References

- [1]. G. E. Karniadakis and S. J. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, 2nd ed. Oxford, UK: Oxford University Press, 2005.
- [2]. O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, 7th ed. Oxford, UK: Elsevier, 2013.
- [3]. J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*. Cambridge, UK: Cambridge University Press, 2007.
- [4]. L. N. Trefethen, *Spectral Methods in MATLAB*. Philadelphia, PA, USA: SIAM, 2000.
- [5]. I. Babuška and B. A. Szabó, *Finite Element Analysis*. New York, NY, USA: John Wiley & Sons, 1991.
- [6]. Maziar Raissi, Paris Perdikaris, and George Em Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
- [7]. Steven L. Brunton and J. Nathan Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 2nd ed. Cambridge, UK: Cambridge University Press, 2022.
- [8]. Lu Lu, X. Meng, Z. Mao, and George Em Karniadakis, “DeepXDE: A deep learning library for solving differential equations,” *SIAM Review*, vol. 63, no. 1, pp. 208–228, 2021.
- [9]. George Em Karniadakis, I. G. Kevrekidis, Lu Lu, Paris Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
- [10]. Christoph Molnar, *Interpretable Machine Learning*, 2nd ed., 2022.
- [11]. Scott M. Lundberg and Su-In Lee, “A Unified Approach to Interpreting Model Predictions,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [12]. B. Meuris, S. Qadeer, and P. Stinis, “Machine-learning-based spectral methods for partial differential equations,” *Scientific Reports*, vol. 13, Art. no. 1739, 2023.
- [13]. R. Pestourie, Y. Mroueh, C. Rackauckas, P. Das, *et al.*, “Physics-enhanced deep surrogates for partial differential equations,” *Nature Machine Intelligence*, vol. 5, pp. 1458–1465, 2023.
- [14]. Steven L. Brunton and J. Nathan Kutz, “Promising directions of machine learning for partial differential equations,” *Nature Computational Science*, vol. 4, pp. 483–494, 2024.
- [15]. Q. Cao, S. Goswami, and George Em Karniadakis, “Laplace neural operator for solving differential equations,” *Nature Machine Intelligence*, vol. 6, pp. 631–640, 2024.
- [16]. H. Oh, R. Amici, G. Bomarito, S. Zhe, R. M. Kirby, and J. Hochhalter, “Inherently interpretable machine learning solutions to differential equations,” *Engineering with Computers*, vol. 40, no. 4, pp. 2349–2361, 2024.
- [17]. K. Egan, W. Li, and R. Carvalho, “Automatically discovering ordinary differential equations from data with sparse regression,” *Communications Physics*, vol. 7, Art. no. 20, 2024.
- [18]. Q. Ma, E. C. De Meter, and S. Basu, “TO-NODE: Topology optimization with neural ordinary differential equation,” *International Journal for Numerical Methods in Engineering*, vol. 125, no. 7, e7428, 2024.
- [19]. X. Chen *et al.*, “Physics-informed neural networks for solving high-index differential-algebraic equation systems based on Radau methods,” *International Journal of Intelligent Systems*, 2024.
- [20]. R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*. Philadelphia, PA, USA: SIAM, 2007.
- [21]. Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [22]. Lloyd N. Trefethen, *Approximation Theory and Approximation Practice*, 2nd ed. Philadelphia, PA, USA: SIAM, 2019.
- [23]. Susanne C. Brenner and L. Ridgway Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed. New York, NY, USA: Springer, 2008.
- [24]. Claes Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Publications, 2009.