Graph Neural Networks for Surrogate PDE Solvers in Computational Physics: A Survey

www.surveyx.cn

Abstract

Partial Differential Equations (PDEs) are integral to modeling complex systems across scientific domains, capturing dynamics in areas such as fluid mechanics and biological processes. Traditional numerical methods for solving PDEs face computational challenges, particularly with high-dimensional data and complex geometries. This survey explores the burgeoning role of Graph Neural Networks (GNNs) as surrogate PDE solvers, emphasizing their potential to address these challenges efficiently. GNNs leverage graph structures to model spatial dependencies, offering enhanced accuracy and scalability. Innovations such as PDEformer and physics-informed neural networks (PINNs) exemplify the integration of GNNs with traditional techniques, improving solution fidelity and computational efficiency. The survey highlights applications across fluid dynamics, quantum simulations, and high-dimensional PDEs, showcasing GNNs' versatility and robustness. Despite their promise, GNN-based solvers face challenges in scalability and data quality, necessitating ongoing research to refine algorithms and integrate domain-specific knowledge. The survey concludes by underscoring the transformative potential of GNNs in computational physics, advocating for continued development to enhance their applicability and performance across diverse scientific fields.

1 Introduction

1.1 Significance of PDEs in Modeling Physical Phenomena

Partial Differential Equations (PDEs) are essential for modeling parameterized, time-dependent nonlinear systems across various scientific and engineering fields. They form the backbone of computational physics, effectively simulating complex processes in fluid mechanics and other domains [1]. PDEs are instrumental in representing a wide array of physical phenomena, including heat transfer, fluid flow, electromagnetic waves, and calcium dynamics in neurons, underscoring their versatility in capturing the dynamic nature of the world.

However, analytic solutions to PDEs are often elusive, presenting significant challenges in their application [2]. The data-driven discovery of these equations is complicated by potential incompleteness in candidate libraries, necessitating innovative model discovery approaches [3]. Additionally, PDEs are crucial for addressing initial boundary value problems in linear evolution equations, which require precise characterization of unknown boundary values based on initial and boundary conditions [4].

In biological sciences, PDEs are employed to model intricate processes like calcium dynamics in neurons, highlighting their significance in understanding biological systems [5]. The incorporation of PDEs across diverse fields enhances the predictive capabilities of computational models, advancing our comprehension of complex systems [6].

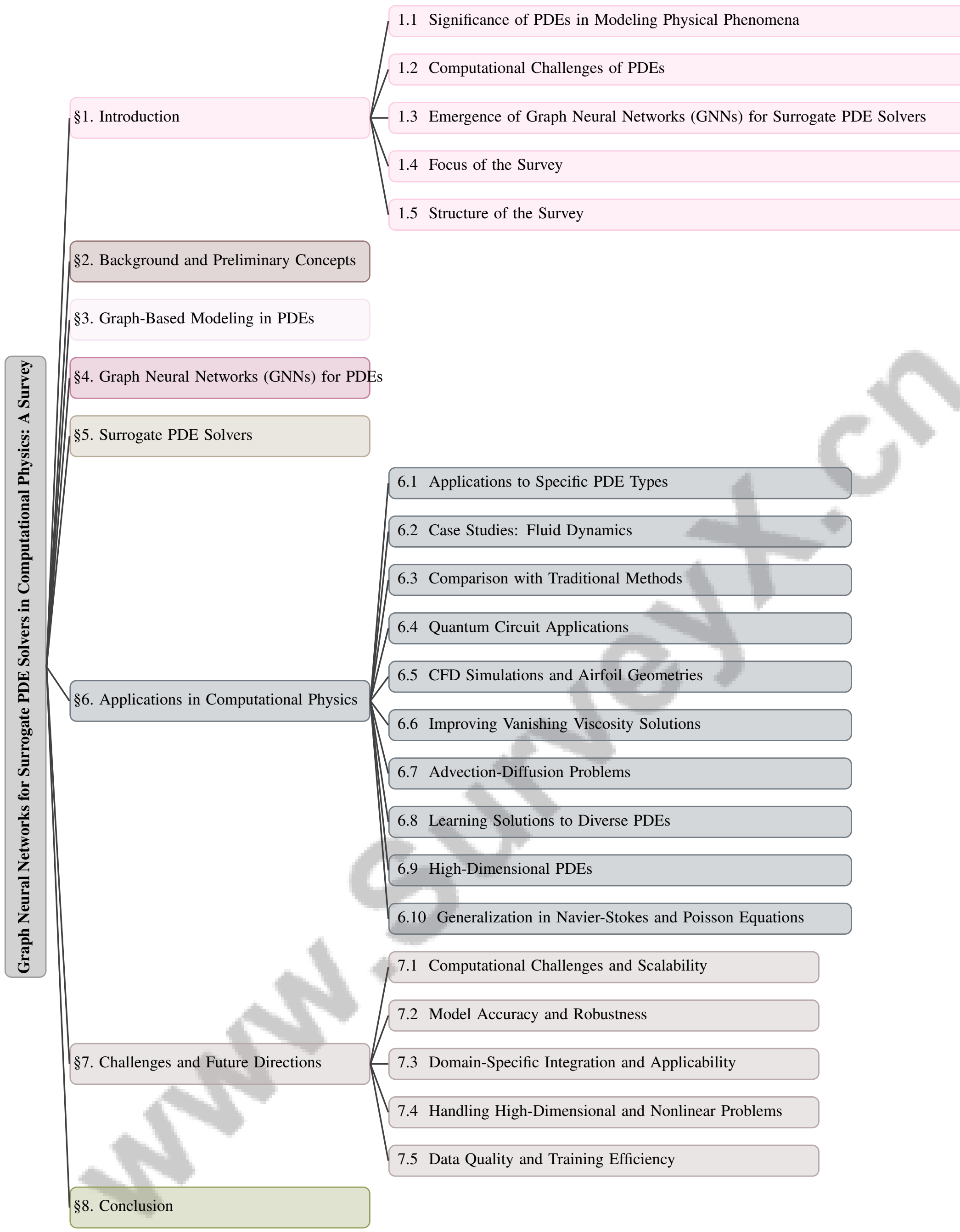


Figure 1: chapter structure

1.2 Computational Challenges of PDEs

The computational challenges of solving PDEs with traditional numerical methods are significant, primarily due to the high dimensionality of many physical problems, which leads to exponential increases in computational costs [7]. Conventional techniques like Finite Difference Methods (FDM) and Finite Element Methods (FEM) often struggle with accuracy and efficiency, particularly in irregular domains, necessitating fine discretization of spatial and temporal variables that can result in substantial computational overhead and inaccuracies [8].

These inefficiencies are exacerbated in complex geometries and boundary conditions, particularly in fluid dynamics [9]. Non-differentiability of projection operators over time introduces additional

AI-generated, for reference only.

challenges, leading to dissipation and instability in numerical solutions [10]. Moreover, traditional grid-based methods face difficulties when solving high-dimensional Hamilton-Jacobi (HJ) PDEs due to the curse of dimensionality [11]. Stochastic PDEs, especially parabolic SPDEs driven by a Wiener process, further heighten these computational demands [12].

Implementing boundary conditions accurately, such as homogeneous Neumann boundary conditions in reaction-diffusion equations, can lead to inaccuracies over long-time integrations [13]. The limitations of model order reduction techniques, particularly for Friedrichs' systems due to slow Kolmogorov n-width decay, present additional computational hurdles [9]. In low-data regimes, traditional numerical solvers become inefficient, requiring retraining for varying initial or boundary conditions [14].

Identifying PDEs from experimental data containing significant noise further complicates the effectiveness of traditional methods [15]. Additionally, traditional data-driven approaches for PDE discovery often falter with incomplete candidate libraries [3]. The fragmented literature on adjoint methods presents analogous computational challenges [6].

The computational inefficiency of traditional numerical solvers for high-fidelity PDE approximations necessitates faster evaluation techniques [16]. Furthermore, the complexity of biological processes complicates the analytical derivation of governing equations for ion channels, rendering traditional methods inadequate [5]. The quest for weak solutions in nonlinear parabolic equations necessitates advanced computational techniques to manage the dense matrices resulting from discretization.

These challenges highlight the urgent need for innovative computational techniques that can surpass the limitations of conventional methods, providing efficient and accurate solutions to PDEs across a spectrum of scientific and engineering domains. The development of such methods is vital for applications ranging from turbulence modeling in Reynolds-averaged Navier-Stokes simulations to the formulation of meaningful mathematical models that encapsulate complex interactions in non-moving systems [7].

1.3 Emergence of Graph Neural Networks (GNNs) for Surrogate PDE Solvers

Graph Neural Networks (GNNs) have emerged as a revolutionary approach for developing surrogate PDE solvers, addressing the computational challenges posed by traditional numerical methods. Leveraging their graph structure, GNNs effectively model complex spatial dependencies and manage high-dimensional data, making them particularly adept at approximating PDE solutions. These networks provide enhanced computational accuracy and efficiency, serving as a robust alternative to conventional methods [17].

The integration of GNNs with established numerical techniques is exemplified by models like PDEformer, which introduces a foundation model capable of addressing various PDE types through a computational graph representation that synthesizes symbolic and numerical information [18]. This approach mitigates the limitations of existing models, which often struggle with generalization across varying parameters and time scales [1].

Physics-informed neural networks (PINNs) have shown promise in overcoming traditional method constraints by effectively approximating nonlinear PDE solutions while incorporating physical constraints [14]. Innovative strategies, such as trapz-PiNNs, which merge physics-informed neural networks with a modified trapezoidal rule, illustrate novel approaches for solving non-local Fokker-Planck equations [19].

The rise of GNNs as surrogate PDE solvers not only enhances their applicability and effectiveness but also opens new avenues for computational modeling and simulation. By ensuring that neural network outputs conform to conservation laws through novel projection techniques like PINN-Proj, GNN-based approaches maintain solution fidelity [20]. The application of reinforcement learning principles to control unknown PDEs by dynamically estimating parameters further demonstrates the versatility and adaptability of GNNs in dynamic environments [12].

In high-dimensional contexts, where traditional methods often falter due to computational complexity, GNNs deliver efficient approximations [21]. This advancement underscores the significance of GNNs in tackling complex PDE challenges, offering scalable and accurate solutions to problems that were previously intractable with conventional methods. The role of adjoint operators in computational science further emphasizes the emergence of GNNs as a powerful tool in this field [6]. Additionally, the application of tensor decomposition for constructing trial functions for direct numerical integration represents a key innovation that reduces computational complexity and enhances accuracy [22]. The advent of deep learning has introduced new data-driven methodologies to enhance numerical methods for PDEs, as evidenced by recent advancements [23].

1.4 Focus of the Survey

This survey primarily focuses on the integration of Graph Neural Networks (GNNs) as surrogate solvers for Partial Differential Equations (PDEs) in computational physics. It aims to illuminate the transformative potential of GNNs in addressing the computational challenges associated with traditional numerical methods for large and complex PDEs, particularly in computational fluid dynamics (CFD), where conventional approaches are often computationally prohibitive [24]. By leveraging neural message passing, GNNs provide enhanced flexibility and generalization capabilities that surpass the limitations of existing numerical methods [25].

The survey emphasizes the utility of GNNs in developing data-driven time-stepping schemes that evolve dynamical systems across varying geometries, showcasing their adaptability and efficiency [26]. The GCN-FFNN model exemplifies how hybrid approaches that combine GNNs with traditional methods can significantly enhance the learning of PDE solutions, demonstrating the potential for superior accuracy and computational efficiency [2].

Additionally, this survey seeks to unify the understanding of adjoint applications across various scientific and engineering disciplines, aligning with the broader goal of advancing surrogate PDE solvers [6]. It also highlights innovative hybrid PDE-DNN models that merge PDEs with deep neural networks (DNNs) to simulate complex biological processes, such as calcium dynamics in neurons, thereby extending the applicability of GNN-based surrogate solvers across diverse scientific domains [5]. Through this comprehensive examination, the survey aims to provide a detailed understanding of the current landscape and future directions for GNN-based surrogate PDE solvers.

1.5 Structure of the Survey

The survey is structured into several key sections, each addressing different facets of Graph Neural Networks (GNNs) as surrogate solvers for Partial Differential Equations (PDEs) in computational physics. The introduction establishes the significance of PDEs in modeling physical phenomena and the computational challenges they present, followed by a discussion on the emergence of GNNs as a promising solution for surrogate PDE solvers. The background section offers an overview of PDEs, graph-based modeling concepts, and the role of GNNs, laying the foundational knowledge necessary for understanding subsequent discussions.

The core of the survey explores graph-based modeling in PDEs, illustrating how graphs can represent spatial relationships and dependencies, thereby enhancing computational efficiency and accuracy. An in-depth analysis of GNNs applied to solving PDEs follows, emphasizing key advancements in architectural design, customized approximation strategies, and innovative learning methodologies that incorporate physical principles. This includes the introduction of a fully differentiable GNN-based PDE solver that integrates finite volume methods for improved adaptability across various unstructured grids, as well as deep learning-based approximation techniques for high-dimensional PDEs, and a theoretical investigation into the convergence of physics-informed neural networks (PINNs) for linear second-order elliptic and parabolic PDEs, demonstrating their potential for robust applications in computational fluid dynamics and other complex physical problems [27, 28, 29].

The survey then shifts focus to surrogate PDE solvers, showcasing their development and computational advantages through applications and case studies across various domains within computational physics. It delves into specific applications such as fluid dynamics, quantum circuit simulations, and CFD, highlighting the adaptability and efficiency of GNN-based solvers. These applications are supported by innovative methodologies, including a fully differentiable GNN-based PDE solver that merges traditional numerical methods with deep learning, a hybrid model that combines graph convolutional networks with embedded CFD simulations for enhanced generalization, and a two-stream deep model that processes both graph and grid representations to effectively learn solutions to nonlinear PDEs. Additionally, novel techniques like NeuralStagger illustrate the potential for significant computational speed-ups while maintaining accuracy in complex simulations, further underscoring the versatility of GNN-based approaches in addressing a range of scientific challenges [2, 29, 24, 30, 31].

Finally, the survey addresses the challenges and future directions in the field, discussing issues such as computational scalability, model accuracy, and the integration of domain-specific knowledge. The conclusion synthesizes the main findings and emphasizes the transformative potential of Graph Neural Networks (GNNs) in solving PDEs within computational physics. It highlights the innovative integration of GNNs with traditional numerical methods, such as the finite volume method, which enhances adaptability across various unstructured grids and boundary conditions. The discussion underscores the necessity for ongoing research and development to refine these hybrid approaches, particularly given their demonstrated capabilities in improving mesh generalization, stability, and accuracy across diverse physical scenarios. This advancement not only paves the way for more efficient CFD applications but also lays the groundwork for future breakthroughs at the intersection of deep learning and numerical analysis [28, 29, 25].The following sections are organized as shown in Figure 1.

2 Background and Preliminary Concepts

2.1 Overview of Partial Differential Equations (PDEs)

Partial Differential Equations (PDEs) are integral to modeling complex physical systems, offering a framework for describing dynamics involving multiple variables and their partial derivatives [32]. They are categorized into elliptic, parabolic, and hyperbolic types, each addressing specific problems: elliptic PDEs for steady-state, parabolic for time-dependent, and hyperbolic for wave propagation. PDEs are pivotal across scientific fields like fluid dynamics, electromagnetic theory, and biology, where they model complex system behaviors, such as neuronal calcium dynamics [5].

Solving high-dimensional and nonlinear PDEs poses computational challenges due to the curse of dimensionality, necessitating advanced numerical techniques and substantial computational resources [33]. Iterative methods are essential for solving linear systems from discretized PDEs, yet their implementation requires specialized knowledge as no universal method exists [23]. Recent computational advancements include hybrid models integrating PDEs with deep neural networks (DNNs) to manage multiscale particle dynamics, enhancing solver efficiency and accuracy [34, 35].

The significance of PDEs in computational physics is underscored by their robust mathematical framework for modeling dynamics across various disciplines. Advanced methodologies, such as physics-informed neural networks (PINNs) and variational physics-informed neural networks (VPINNs), leverage PDEs to improve simulation accuracy in areas like climate modeling and biomedical phenomena. Techniques like automatic differentiation and graph neural networks help address challenges related to multiscale dynamics and data noise, enhancing predictive capabilities [36, 6, 37, 38]. As computational techniques evolve, PDEs remain fundamental to advancing theoretical and practical aspects of computational physics, driving innovation in simulation and modeling.

2.2 Graph-Based Modeling Concepts

Graph-based modeling is a robust approach for representing complex systems, particularly in the context of PDEs. It uses graph structures to encapsulate spatial relationships and dependencies, offering a framework for modeling various physical phenomena [39]. Graphs enable concise representation of properties and interactions among invariant sets, crucial for systems governed by maps and differential equations [39].

In computational physics, graph-based methods provide a unified framework for discrete calculus, enhancing local and nonlocal image processing techniques [40]. This framework efficiently handles manifold-valued data, broadening the applicability of graph-based techniques across scientific domains. Integration of reinforcement learning with graph neural networks (GNNs) has advanced the field, enabling optimization-based algebraic multigrid coarsening strategies that improve scalability and efficiency for large-scale PDEs [41]. These strategies, supported by adaptive learning theory, enhance the resolution of subgrid microscale interactions, paving the way for innovative approaches in computational physics [42].

2.3 Introduction to Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs) are specialized machine learning models designed for processing data organized in graph structures. They leverage graph connectivity to improve interpretability and performance, as seen in modeling nonlinear PDEs and integrating neural differential equations [16, 43, 2]. GNNs demonstrate versatility across data types and sizes, excelling in multifidelity applications by capturing complex relationships in graph-structured data. Their architecture utilizes node and edge connectivity patterns, facilitating learning of representations sensitive to both local and global graph structures.

A notable innovation is the Reaction Diffusion Graph Neural Network (RDGNN) architecture, which incorporates reaction diffusion principles to model phenomena on graphs, enhancing modeling capabilities for systems where diffusion processes are critical [43]. GNNs benefit from advancements like residual connections, which improve training and function approximation by providing shortcut connections [44]. This enhances GNNs' ability to learn deep representations, improving effectiveness across applications.

GNNs' potential applications are vast, including drug discovery, transportation networks, and social network analysis, where they model molecular interactions, optimize traffic flow, and enhance community structure identification [16, 43, 45, 2]. Their versatility and robustness make GNNs a powerful tool for addressing complex relational data problems, enabling innovative solutions across multiple domains.

2.4 Importance of Surrogate PDE Solvers

Surrogate PDE solvers are essential in computational physics, offering advantages over traditional methods by efficiently approximating solutions for high-dimensional systems. These solvers use advanced machine learning techniques, like autoencoder-based architectures, to enhance reduced order models (ROMs) through non-intrusive methods leveraging experimental data [16]. This adaptability allows surrogate solvers to adjust to various scenarios without extensive retraining.

Integration of GNNs with differentiable finite volume methods exemplifies innovative approaches that streamline solution processes, enhancing model generalization [29]. Surrogate solvers like HyperPNODE demonstrate significant speedups and minimal errors in training for parameterized PDEs [7]. Multifidelity surrogates in the Hierarchical Neural Hybrid (HNH) method adaptively use high-fidelity models, reducing computational costs while maintaining accuracy [46].

Parallel numerical tensor methods enhance surrogate solvers' efficacy, improving accuracy and efficiency in solving high-dimensional PDEs [33]. Techniques like Tensor Train Decomposition (TTD) facilitate direct numerical integration, enhancing stability and accuracy [22]. The Deep Energy Method (DEM) uses system energy as a loss function, integrating physical principles into the learning process, enhancing surrogate solvers' robustness [47]. Learnable multigrid solvers streamline PDE solving, enabling rapid convergence for large-scale problems [23]. Hybrid PDE-DNN models reduce computational complexity and improve adaptability, making them valuable surrogate PDE solvers [5].

Surrogate PDE solvers represent a transformative leap in computational physics by leveraging advanced techniques like GNNs and deep learning to provide flexible, efficient, and accurate solutions for complex systems governed by parameterized nonlinear PDEs. These models significantly reduce computational burdens, enabling rapid simulations while maintaining high accuracy. Developments like the CZP framework and PDEformer model efficiently handle geometric variability and high-dimensional random PDEs, improving predictive performance and generalization capabilities [26, 18, 48, 49]. Their ability to integrate advanced learning techniques underscores their role as powerful alternatives to traditional methods, paving the way for effective and scalable computational modeling across various domains.

2.5 Comparative Analysis with Traditional Numerical Methods

Surrogate PDE solvers, particularly those leveraging GNNs and advanced machine learning techniques, signify a transformative shift from traditional numerical methods. Conventional approaches, like Finite Difference and Finite Element Methods, face challenges with computational efficiency and accuracy, especially with high-dimensional data and complex geometries. These methods require fine

AI-generated, for reference only.

discretization, leading to significant computational overhead [50]. In contrast, surrogate solvers use data-driven techniques, allowing more flexible and efficient PDE approximations without extensive discretization.

Traditional methods struggle with interpolation techniques in multigrid algorithms, failing to capture high-frequency information, resulting in slower convergence and reduced accuracy [50]. Surrogate solvers enhance interpolation strategies and machine learning models, capturing complex patterns and dependencies, leading to faster convergence and improved solution fidelity.

Conventional methods demand substantial computational resources and face limitations due to the curse of dimensionality, restricting effectiveness in real-time applications and large-scale simulations. Artificial neural networks (ANNs) efficiently approximate high-dimensional functions, overcoming this challenge, particularly in problems like the Black-Scholes PDE [51, 45]. Surrogate solvers, with multifidelity models and reduced order techniques, offer computational savings while maintaining accuracy, advantageous in applications like CFD and structural analysis.

Surrogate PDE solvers represent a paradigm shift, offering enhanced flexibility, scalability, and accuracy over traditional methods. By harnessing GNNs and advanced machine learning, these solvers provide compelling alternatives to traditional numerical methods, addressing limitations like generalization difficulties across properties like resolution and boundary conditions. They use strategies like quantization-aware training and systematic model augmentation to enhance efficiency and robustness, facilitating faster convergence and improved accuracy across diverse scenarios. By integrating neural message passing and differentiable computation, they pave the way for efficient solutions to complex PDE problems, demonstrating superior performance in applications ranging from fluid dynamics to sparse linear systems [52, 23, 29, 53, 25].

In recent years, the application of graph-based modeling in Partial Differential Equations (PDEs) has gained significant attention due to its capacity to represent complex physical systems. As illustrated in Figure 2, this figure provides a comprehensive overview of the hierarchical categorization of graph-based modeling. It encompasses various aspects, including the representation of physical systems, dynamical systems modeling, and unified graph frameworks for discrete calculus. Furthermore, it addresses the concept of well-posedness in hyperbolic systems on networks. Each section of the figure not only highlights key concepts but also explores the integration of these models with neural networks and their practical applications. This emphasizes the connectivity and adaptability of graph-based approaches, underscoring their importance in the analysis of complex systems.

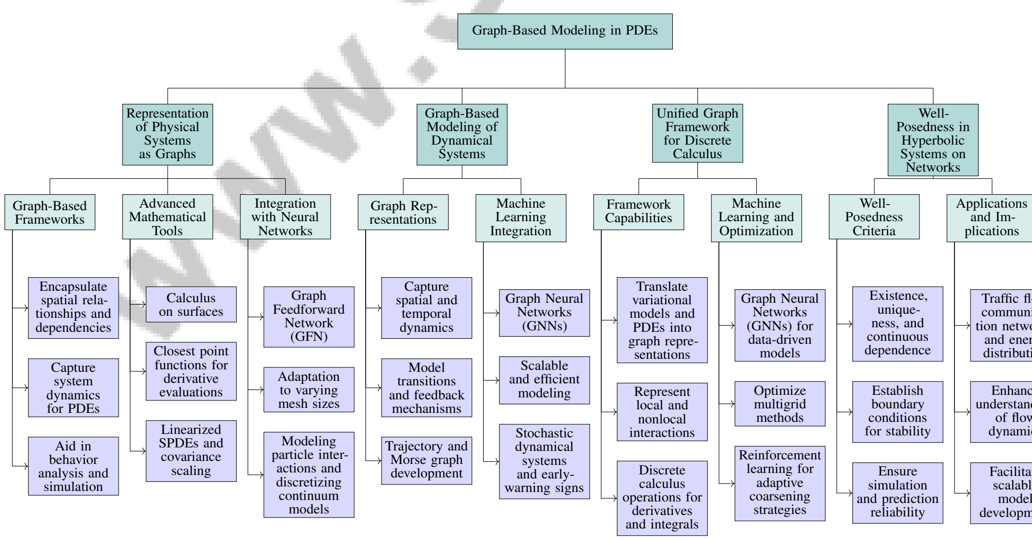


Figure 2: This figure illustrates the hierarchical categorization of graph-based modeling in Partial Differential Equations (PDEs), encompassing the representation of physical systems, dynamical systems modeling, unified graph frameworks for discrete calculus, and well-posedness in hyperbolic systems on networks. Each section highlights key concepts, integration with neural networks, and practical applications, emphasizing the connectivity and adaptability of graph-based approaches in complex systems analysis.

3 Graph-Based Modeling in PDEs

3.1 Representation of Physical Systems as Graphs

Graph-based representation of physical systems offers a powerful framework for modeling complex dynamical behaviors by leveraging the inherent connectivity of graphs to encapsulate spatial relationships and dependencies. This approach is particularly effective in Partial Differential Equations (PDEs), where the geometry of tangent bundles and properties of closed forms and vector fields are crucial for capturing system dynamics [54]. Graphs facilitate the representation of intricate interactions among components, aiding in behavior analysis and simulation.

Beyond simple connectivity, graph-based modeling incorporates advanced mathematical tools such as calculus on surfaces, introducing a broader class of closest point functions to enhance derivative evaluations, essential for accurately capturing PDE dynamics on complex geometries [55]. This is particularly valuable for systems with irregular boundaries or non-uniform material properties.

Graph-based frameworks also examine linearized Stochastic Partial Differential Equations (SPDEs) near bifurcation points, where covariance scaling serves as an early-warning indicator for pattern formation, highlighting the potential of graphs to capture critical transitions in dynamical systems [56]. A significant advancement is proving that every trajectory graph in a dynamical system is connected under mild hypotheses [39], emphasizing connectivity's role in understanding dynamical systems' structure and evolution.

Integrating graph-based frameworks with neural network approaches like the Graph Feedforward Network (GFN) enhances the learning of reduced-order models from multifidelity data, allowing for direct adaptation of neural network weights to varying mesh sizes [16]. This capability is particularly beneficial for high-fidelity simulations across diverse spatial resolutions.

By harnessing graphs' connectivity, researchers can create advanced models that improve simulation accuracy and efficiency for complex phenomena. This approach enables the representation and analysis of high-dimensional PDE solutions, facilitating the exploration of intricate systems. Integrating graph neural networks (GNNs) allows for modeling particle interactions and discretizing continuum models while addressing limitations like neglecting long-range interactions, thus enhancing computational efficiency and supporting data-driven approximation techniques for applications from climate modeling to fluid dynamics [57, 37].

3.2 Graph-Based Modeling of Dynamical Systems

Graph-based modeling is a pivotal approach for representing and analyzing dynamical systems, offering a versatile framework to encapsulate complex interactions and dependencies. This paradigm effectively captures spatial and temporal dynamics described by PDEs, where graph connectivity and structure represent intricate relationships between components [39]. Graphs facilitate intuitive representations, enhancing system behavior and evolution analysis.

In dynamical systems, graph-based models represent interactions between invariant sets, essential for understanding stability and bifurcation phenomena [39]. Graph structures model transitions and feedback mechanisms in complex dynamics, providing insights into underlying mechanisms and potential emergent behaviors.

Integrating graph-based models with machine learning techniques, such as GNNs, further enhances their applicability in dynamical system analysis. GNNs learn from graph-structured data, capturing local and global dependencies crucial for accurately modeling dynamics with complex interactions, offering scalable and efficient alternatives to traditional numerical methods [16].

Graph-based models extend to stochastic dynamical systems, analyzing covariance scaling near bifurcation points as early-warning signs for pattern formation [56]. This provides a powerful tool for predicting critical transitions and understanding behavior under stochastic influences.

Graph-based modeling offers a robust framework for representing and analyzing dynamical systems, deepening the understanding of complex behaviors and facilitating the development of accurate, efficient computational models. By leveraging graph theory's connectivity and adaptability, researchers can enhance the analysis and representation of dynamical systems, ranging from simple maps to complex ordinary and partial differential equations. This approach enables trajectory and Morse graph development, encapsulating system behavior and facilitating high-dimensional solution exploration in various fields. Consequently, this framework elucidates intricate phenomena and fosters innovative solutions to real-world problems by revealing underlying patterns and relationships [39, 57].

3.3 Unified Graph Framework for Discrete Calculus

The unified graph framework for discrete calculus offers a robust methodology for modeling and analyzing complex systems by leveraging graph structural properties to translate variational models and PDEs into graph representations. This facilitates insights into high-dimensional phenomena and manifold-valued data processing across diverse applications, including image processing and continuum physics [57, 40, 58]. This framework effectively represents spatial relationships and dependencies in systems governed by PDEs, enabling discrete calculus application to a wide range of scientific and engineering problems.

Central to this framework is representing local and nonlocal interactions within a unified structure, essential for accurately modeling phenomena such as diffusion and wave propagation. The graph framework supports discrete calculus operations, facilitating efficient computation of derivatives and integrals over complex domains [40]. This capability is valuable for applications in image processing and signal analysis, where manifold-valued data requires sophisticated mathematical tools.

Moreover, the unified graph framework enhances computational model scalability and flexibility by incorporating advanced machine learning techniques, such as GNNs. These networks learn from graph-structured data, capturing local and global dependencies crucial for accurately modeling complex interaction dynamics [16]. Integrating GNNs with the graph framework enables developing data-driven models that adapt to varying conditions and datasets, providing powerful tools for solving high-dimensional and nonlinear PDEs.

Applications of this framework extend to optimizing multigrid methods, where the graph structure facilitates adaptive coarsening strategies that enhance computational efficiency and accuracy [41]. By employing reinforcement learning principles, the framework dynamically adjusts to modeled systems' complexities, ensuring accurate and efficient PDE solutions.

The unified graph framework for discrete calculus represents a significant advancement in modeling and analyzing complex systems by integrating graph theory with variational models and PDEs, enabling high-dimensional computed state representation and exploration in a manageable low-dimensional format. This approach enhances intricate phenomena analysis across various fields, including continuum physics and manifold-valued data, facilitating new insights into complex networks and systems governed by PDEs [58, 39, 6, 57, 40]. By leveraging graph connectivity and adaptability, this approach provides a versatile and powerful tool for addressing modern computational physics challenges, paving the way for innovative solutions to complex real-world problems.

3.4 Well-Posedness in Hyperbolic Systems on Networks

Well-posedness in hyperbolic systems is fundamental for ensuring stability and reliability of solutions in network modeling, defined by existence, uniqueness, and continuous dependence on initial and boundary conditions. This concept is crucial for ensuring that hyperbolic PDEs, particularly under general or non-local boundary conditions, yield reliable and predictable results. Analyzing well-posedness involves establishing necessary and sufficient conditions related to boundary conditions and system coefficients to maintain specific subset invariance within the ambient space under the governing flow [9, 6, 59, 10]. This is vital when modeling complex systems on networks, where interactions and dependencies can lead to intricate dynamical behaviors.

Establishing appropriate boundary conditions is essential for achieving well-posedness in hyperbolic systems on networks. The proposed boundary condition framework establishes well-posedness for a broad class of hyperbolic systems, providing a foundation for future applications and theoretical developments [9]. This framework allows systematic treatment of boundary interactions, ensuring solutions remain stable and robust across various network configurations.

Well-posedness in network modeling is crucial for ensuring simulation and prediction reliability in complex systems, establishing existence, uniqueness, and continuous solution dependence on initial conditions and parameters. This foundational property is essential for accurately modeling PDE-described phenomena, guaranteeing consistent and robust simulation outcomes, enhancing the ability to analyze and predict intricate system behavior across various disciplines [60, 58, 6, 57, 61]. By ensuring well-posed solutions, researchers can confidently analyze hyperbolic systems on networks, facilitating accurate and efficient computational model development. This capability is valuable in applications such as traffic flow, communication networks, and energy distribution systems, where hyperbolic dynamics play a critical role.

Establishing well-posedness in hyperbolic systems on networks marks a significant advancement in modeling and analyzing complex dynamical systems, providing a robust framework for addressing one-dimensional PDEs under various boundary conditions. This development enhances understanding of flow dynamics within networks and facilitates necessary and sufficient condition application for maintaining system invariance, broadening the scope of mathematical techniques applicable to both theoretical and practical problems in engineering and applied sciences [9, 6, 58]. By providing a solid theoretical foundation, this concept enables reliable and scalable model development, advancing understanding and simulation of network-based systems across diverse scientific and engineering domains.

4 Graph Neural Networks (GNNs) for PDEs

Graph Neural Networks (GNNs) have emerged as a powerful tool for tackling the complexities of Partial Differential Equations (PDEs). Table 2 provides a detailed comparison of different architectural innovations and learning approaches in Graph Neural Networks (GNNs) tailored for solving Partial Differential Equations (PDEs). This section delves into the architectural innovations that enhance GNNs' performance and applicability in solving PDEs, showcasing how GNNs leverage their structural properties to improve computational efficiency and accuracy across various PDE contexts.

4.1 Architectural Innovations in GNNs for PDEs

Method Name	Architectural Innovations	Integration Techniques	Model Applications
Gen-FVGN[29]	Differentiable Methods	Finite Volume Method	Fluid Dynamics Scenarios
PDEformer[18]	Graph Transformer	Symbolic And Numeric	Zero-shot Predictions
DLGA-PDE[3]	Deep Neural Network	Deep Learning Integration	Pde Discovery
DRL[35]	Rule-based Self-learning	Physical Rules Critic	Nonlinear Differential Equations

Table 1: Table illustrating recent architectural innovations in graph neural networks (GNNs) applied to partial differential equations (PDEs), highlighting the integration techniques and specific model applications. The methods include Gen-FVGN, PDEformer, DLGA-PDE, and DRL, each employing distinct approaches to enhance accuracy and scalability in solving complex PDE problems.

Recent advancements in GNNs have led to architectural innovations specifically designed for solving PDEs. Table 1 provides a comprehensive overview of the architectural innovations in graph neural networks (GNNs) designed for partial differential equations (PDEs), detailing the integration techniques and applications of each method. The Generative Finite Volume Graph Network (Gen-FVGN) utilizes a fully differentiable finite volume method, enabling direct PDE solutions during training and integrating graph-based neural networks with traditional numerical methods [29]. The PDEformer model employs a graph Transformer to process computational graphs and an implicit neural representation (INR) for mesh-free PDE solutions, enhancing flexibility and scalability in complex PDE problems [18]. Additionally, architectures like ResNet, Fourier Neural Operators (FNO), and U-Net have been effective in modeling fluid mechanics, capturing multi-scale spatiotemporal dynamics [1]. The DLGA-PDE framework combines deep learning with genetic algorithms to discover PDEs from incomplete data, enhancing GNNs' ability to identify underlying structures [3]. Hybrid models that integrate deep neural networks with physical principles show promise in solving nonlinear differential equations, aligning deep learning with physical rules for enhanced accuracy and robustness [35]. Recent developments focus on improving the efficiency, accuracy, and scalability of neural network models for complex differential equations, including neural–numerical hybrid solvers that integrate classical methods with deep learning for improved generalization across various properties [2, 29, 53, 25, 62].

4.2 Tailoring GNNs for PDE Approximation

GNNs have been increasingly adapted for PDE approximation through advanced techniques and innovative architectures. The use of canonical and hierarchical tensor methods, combined with alternating least squares and hierarchical singular value decomposition, provides an effective strategy for high-dimensional PDEs [33]. Integrating Deep Neural Networks (DNNs) trained on datasets from traditional Ordinary Differential Equation (ODE) models enhances hybrid models' capabilities to approximate complex dynamics, such as calcium dynamics in neurons [5]. Temporal continuity through transfer learning significantly accelerates the solution process for nonlinear differential equations [35]. These methodologies illustrate significant advancements in PDE approximation using GNNs. By integrating GNN architectures with traditional numerical methods like finite volume techniques, researchers have developed a robust computational framework for solving PDEs such as the Poisson and Navier-Stokes equations. This framework employs a fully differentiable training algorithm utilizing GPU parallel computing, allowing direct PDE solutions during training without reliance on pre-computed data, demonstrating superior generalization across mesh configurations and boundary conditions [62, 29, 2].

4.3 Self-Supervised and Physics-Informed Learning Approaches

Self-supervised and physics-informed learning approaches have significantly enhanced GNN capabilities in solving PDEs. These methodologies leverage the inherent properties and symmetries of PDEs to improve model accuracy, interpretability, and convergence rates. Extended Physics-Informed Neural Networks (XPINNs) utilize sub-networks to independently solve PDEs in decomposed subdomains, optimizing training through a self-supervised framework [63]. Physics-Informed Neural Networks (PINNs) incorporate known dynamics of physical systems into the loss function during training, allowing networks to approximate PDE solutions more effectively [64, 14]. The consistency of PINNs is validated by proving that the sequence of minimizers of a regularized loss converges uniformly to the PDE solution, establishing reliability [28]. Innovative methods, such as introducing Lie point symmetries into PINN training, enable networks to learn not only single solutions but also neighboring solutions generated by these symmetries [65]. The use of generative adversarial networks (GANs) for generating time-sequences of compressed states and model parameters significantly reduces computational burdens compared to methods like Markov Chain Monte Carlo (MCMC) [60]. The application of adjoint methods in backpropagation and optimization techniques highlights their relevance to GNN architecture for PDE applications, facilitating efficient gradient computation and enhancing scalability and performance [6]. These approaches establish a powerful framework that enhances both performance and interpretability of GNNs in addressing PDEs, paving the way for advanced solutions in scientific computing and engineering applications [14, 66].

4.4 Numerical and Computational Techniques

GNNs for PDEs have been significantly enhanced through advanced numerical and computational techniques, improving efficiency and accuracy in solving complex differential equations. Parameterization of spectral transformations using neural networks minimizes the spectral norm of the residual function, enhancing solution accuracy and computational efficiency [67]. The Deep Galerkin Method (DGM) employs deep neural networks to approximate solutions by minimizing a loss function incorporating the differential operator, boundary, and initial conditions, providing a practical approach for high-dimensional PDEs [30]. Incorporating explicit diffusion schemes into residual networks has demonstrated stability and reduced computational complexity, offering a stable framework for solving PDEs [62]. The Generative Multiscale Solver (GMS) utilizes a GAN for data-driven interpolation, achieving faster convergence in solving multiscale PDEs [50]. An R-adaptive strategy dynamically adjusts the radius of compact support during training, optimizing computational resources for solving PDEs [68]. Multi-level neural networks measure numerical error reduction after each iteration, ensuring enhanced accuracy and convergence in PDE solutions [69]. The Quadrature Rules for Neural Networks (QRNN) method improves integral approximation accuracy in neural network solutions to PDEs, addressing challenges in numerical integration [70]. The effectiveness of the deflation algorithm is assessed through distinct solution counts and average Krylov iterations for convergence [59]. Inf-SupNet demonstrates stability and accuracy in solving high-dimensional elliptic PDEs, validating its contributions to the field [71]. The DRNS method minimizes squared residuals over the computational domain, showcasing parallelizability on GPUs and effectiveness in representing solutions analytically [49]. These numerical and computational techniques highlight the transformative potential of GNN-based PDE solvers, demonstrating their ability to effectively integrate traditional numerical methods, such as finite volume techniques, with advanced deep learning frameworks. These hybrid solvers exhibit superior adaptability to complex geometries and boundary conditions, showcasing enhanced generalization capabilities across diverse physical problems, including fluid dynamics. Innovative training algorithms utilizing GPU parallel computing and systematic model augmentation strategies significantly improve model robustness and accuracy, underscoring the promise of GNN-based approaches in revolutionizing the solution of partial differential equations [53, 29, 25].

Feature	Architectural Innovations in GNNs for PDEs	Tailoring GNNs for PDE Approximation	Self-Supervised and Physics-Informed Learning Approaches
Integration Technique	Graph-based Neural Networks	Tensor Methods, Dnns	Physics-informed Networks
Application Context	Fluid Mechanics Modeling	High-dimensional Pdes	Scientific Computing
Key Advantage	Improved Efficiency, Accuracy	Superior Generalization	Enhanced Interpretability

Table 2: This table presents a comparative analysis of various methodologies employed in Graph Neural Networks (GNNs) for solving Partial Differential Equations (PDEs). It highlights the integration techniques, application contexts, and key advantages of architectural innovations, tailored GNNs for PDE approximation, and self-supervised and physics-informed learning approaches. The table underscores the advancements in efficiency, generalization, and interpretability achieved through these methodologies.

5 Surrogate PDE Solvers

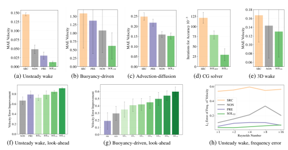
5.1 Applications and Case Studies

Surrogate Partial Differential Equation (PDE) solvers have demonstrated their efficacy across multiple scientific and engineering domains by addressing complex computational challenges. In fluid dynamics, the Reaction Diffusion Graph Neural Network (RDGNN) enhances modeling by capturing both homophilic and heterophilic interactions, underscoring the potential of surrogate solvers for accurate fluid dynamics solutions [43]. The DiffOAS method, evaluated on datasets like Darcy Flow and Solute Diffusion, showcases versatility in tackling various PDE problems [72]. For time-dependent parametric PDEs, the Model-Consistent Reduced Order Model (MC-ROM) offers superior generalization for diffusion- and convection-dominant problems, outperforming traditional methods like DL-ROM and Proper Orthogonal Decomposition (POD) [73].

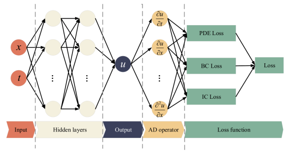
Incorporating uncertainty quantification, UQ-PredGAN aligns numerical simulations with observed data, integrating uncertainty into surrogate modeling [60]. A data-driven turbulence model using Variational Convolutional Neural Networks (VCNN) approximates turbulent flow in complex geometries, highlighting surrogate solvers' ability to model intricate fluid behaviors [74]. Quantum algorithms have simulated linear PDEs with spatially varying parameters, validated through acoustic and heat equation experiments [75]. The kernel-based collocation method for stochastic PDEs demonstrates surrogate solvers' adaptability in stochastic scenarios [76].

The Trapz-PiNNs approach effectively solves non-local Fokker-Planck equations by combining physics-informed neural networks with a modified trapezoidal rule [19]. The Locality-Preserving Sparse Model (LPSM) improves convergence rates in elliptic PDEs [11], while the DLGA-PDE framework applies to classical PDEs like Korteweg-de Vries, showcasing versatility in computational physics [3]. Neural Parameter Regression (NPR) efficiently learns one-dimensional PDE solutions, such as the heat and Burgers equations [77]. These case studies illustrate surrogate PDE solvers' transformative impact, leveraging advanced techniques for efficient, accurate, and scalable solutions to complex problems, particularly in scenarios involving geometric variability and parameter-dependent spatial domains. Rigorous numerical experiments affirm these approaches' effectiveness, addressing challenges conventional methods face [26, 11].

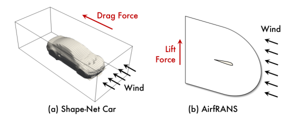
As shown in Figure 3, surrogate PDE solvers are pivotal in computational fluid dynamics and numerical simulations, enhancing efficiency and accuracy in solving complex physical problems. The first case study compares velocity prediction accuracy and computational efficiency across various wake types and solver configurations, offering insights into performance optimization. The second example features a neural network architecture for solving PDEs with boundary and initial conditions, emphasizing a structured approach to spatial and temporal data processing. Lastly, the Shape-Net



(a) Comparison of velocity prediction accuracy and computational efficiency across different wake types and solver configurations[78]



(b) A neural network architecture for solving partial differential equations (PDEs) with boundary conditions (BCs) and initial conditions (ICs)[79]



(c) Shape-Net Car and AirfRANS: Exploring Aerodynamic Forces[80]

Figure 3: Examples of Applications and Case Studies

Car and AirfRANS study highlights applications in the automotive and aerospace industries by simulating aerodynamic forces, providing valuable data for designing aerodynamically efficient vehicles. Collectively, these examples demonstrate surrogate PDE solvers' transformative impact on advancing computational methods across diverse scientific and engineering domains.

5.2 Advantages and Computational Benefits

Surrogate Partial Differential Equation (PDE) solvers, particularly those utilizing Graph Neural Networks (GNNs) and advanced machine learning techniques, offer substantial computational advantages over traditional numerical methods. These solvers excel in high-dimensional PDEs, enhancing accuracy, scalability, and efficiency. The tensor discretization method exemplifies these benefits by achieving optimal convergence and accuracy in high-dimensional contexts without the computational burden typical of conventional methods [22].

PDE-DNN models demonstrate superior flexibility and accuracy in simulating complex dynamics, such as calcium dynamics, compared to traditional Ordinary Differential Equation (ODE) methods, facilitating faster convergence through transfer learning [5, 35]. The Hierarchical Neural Hybrid (HNH) method offers significant cost reductions while maintaining accuracy through hierarchical surrogates, showcasing adaptability to various engineering challenges [46].

The GFN-ROM architecture enhances efficiency and adaptability for multifidelity applications in model order reduction, achieving performance on par with or superior to existing state-of-the-art methods [16]. The energy-based Deep Energy Method (DEM) improves flexibility in managing complex boundary conditions and simplifies training by minimizing system energy, essential for efficiently solving PDEs with intricate geometries [47].

Tensor methods in parallel numerical settings reduce computational complexity and increase accuracy, effectively addressing high-dimensional problems previously deemed intractable [33]. Integrating PDEs with differential geometric methods allows applicability across a wide range of PDEs, operating within original physical coordinates without cumbersome transformations [32]. This, combined with learnable multigrid solvers, reduces reliance on expert knowledge and generalizes efficiently across different problem sizes and coefficient distributions [23].

The advantages of surrogate PDE solvers arise from their capacity to integrate advanced machine learning techniques with domain-specific knowledge, enabling flexible, accurate, and computationally efficient solutions to intricate PDE problems. These solvers manage geometric variability, optimize polynomial approximations using localized functions, and enhance robustness in data-scarce environments, yielding significant performance improvements across diverse scenarios and complexities [52, 53, 25, 26, 11]. They effectively address the limitations of conventional methods, paving the way for more effective computational modeling across various domains.

6 Applications in Computational Physics

The development of innovative methodologies in solving Partial Differential Equations (PDEs) has gained momentum in computational physics, with Graph Neural Networks (GNNs) emerging as pivotal tools. GNNs significantly enhance computational efficiency and solution accuracy across various applications by addressing complex mathematical challenges inherent in PDEs.

6.1 Applications to Specific PDE Types

GNN-based surrogate solvers have shown remarkable efficacy across diverse PDE types, underscoring their robustness in computational physics. Notably, the Deep Galerkin Method (DGM) has outperformed traditional solvers in tackling one-dimensional heat equations and systems of ordinary differential equations [30]. In fluid dynamics, GNN-based solvers efficiently manage incompressible flow scenarios, such as laminar flow past a 2D cylinder and turbulent channel flow, parameterized by Reynolds numbers, providing accurate flow field approximations [81]. Furthermore, neural operators offer scalability and adaptability in approximating solutions to diverse PDEs, reinforcing GNN-based solvers' role in advancing computational modeling [45]. These advancements highlight GNNs' potential to transform computational physics by providing efficient alternatives to traditional finite element methods [26, 29, 82].

6.2 Case Studies: Fluid Dynamics

GNN-based surrogate solvers have demonstrated significant promise in fluid dynamics, offering efficient and accurate solutions to complex flow problems. For instance, they have been used to simulate incompressible turbulent channel flows, capturing intricate turbulence dynamics and aligning closely with high-fidelity simulations while reducing computational costs [81]. Additionally, a generative learning framework has effectively modeled laminar flow past a two-dimensional cylinder, showcasing adaptability to varying flow conditions and geometries [81]. These studies underscore the transformative impact of GNN-based solvers in fluid dynamics, managing geometrical variability and parameter-dependent spatial domains with robust predictions across complex geometries and mesh topologies [26, 83].

6.3 Comparison with Traditional Methods

Benchmark	Size	Domain	Task Format	Metric
GDL-Benchmark[84]	180	Fluid Dynamics	Regression	Mean Squared Error, Wall Shear Stress
PDEbench[1]	1,000,000	Fluid Mechanics	Time Series Prediction	Mean Squared Error, Rollout Loss

Table 3: This table presents a comparison of representative benchmarks used for evaluating GNN-based surrogate solvers in the context of fluid dynamics and fluid mechanics. It highlights the size, domain, task format, and evaluation metrics for each benchmark, providing insights into their applicability and relevance for assessing computational efficiency and accuracy.

GNN-based surrogate solvers have reshaped PDE solving by offering computational advantages and enhanced accuracy compared to traditional methods. Classical approaches like Finite Difference and Finite Element Methods often struggle with efficiency and scalability in high-dimensional problems. Innovations such as the Matrix-Oriented Finite Element Method and Universal Mesh Movement Network (UM2N) improve computational performance and mesh movement efficiency [22, 17, 85]. Table 3 provides a detailed overview of representative benchmarks utilized for evaluating the performance of GNN-based surrogate solvers in fluid dynamics and fluid mechanics, illustrating their size, domain, task format, and evaluation metrics. GNN-based solvers leverage data-driven techniques and graph structures to provide flexible and efficient PDE approximations, handling complex geometries and dynamic systems more effectively than traditional methods [85, 70, 2, 29, 86, 16].

6.4 Quantum Circuit Applications

GNN-based solvers are increasingly applied in quantum circuit simulations, addressing challenges associated with high-dimensional state spaces and intricate entanglement structures. These solvers model quantum states and operations as graph-structured data, improving scalability and accuracy by capturing complex interactions among quantum states [2, 37, 29, 43, 16]. This capability enhances quantum circuit design and optimization, facilitating efficient simulation and analysis [41, 58, 6, 37, 57]. By integrating with quantum algorithms, GNN-based solvers drive advancements in quantum computing technologies, offering insights into optimal configurations and minimizing error rates [87, 6, 75, 57, 88].

6.5 CFD Simulations and Airfoil Geometries

In computational fluid dynamics (CFD), GNN-based solvers have advanced the simulation of complex flow phenomena, particularly around airfoil geometries. These solvers capture complex spatial dependencies and interactions, achieving superior accuracy and efficiency compared to conventional methods. By integrating traditional graph convolutional networks with differentiable fluid dynamics simulators, these approaches enhance generalization capabilities while reducing computational costs [37, 24, 85, 89, 78]. GNNs efficiently handle unstructured mesh data, representing complex geometrical features and flow conditions essential for modeling aerodynamic parameters [90, 29, 24]. This adaptability is particularly beneficial in scenarios where traditional methods face convergence challenges or demand substantial resources [45, 6, 91, 86, 51].

6.6 Improving Vanishing Viscosity Solutions

GNN-based techniques significantly advance the numerical solution of PDEs, particularly in capturing fluid dynamics behavior characterized by small viscosity effects. Traditional methods struggle with resolving sharp gradients and discontinuities in vanishing viscosity scenarios. GNN-based approaches offer robust alternatives by modeling complex spatial dependencies and integrating advanced machine learning algorithms to accurately approximate vanishing viscosity solutions [10]. The integration of physics-informed neural networks (PINNs) with GNN architectures enhances the ability to enforce physical constraints and conservation laws, improving solution fidelity and interpretability [14]. This approach provides a more interpretable framework for modeling vanishing viscosity phenomena, achieving superior accuracy and robustness [92, 14, 45, 2, 29].

6.7 Advection-Diffusion Problems

GNN-based solvers effectively address the complexities of advection-diffusion problems, prevalent in environmental modeling, fluid dynamics, and heat transfer. These problems involve simultaneous transport and diffusion of quantities characterized by intricate spatial and temporal dynamics. GNNs leverage their ability to model complex graph-structured data, capturing spatial dependencies and interactions inherent in these systems [6, 34, 93, 89]. By representing the physical domain as a graph, GNNs efficiently handle irregular geometries and heterogeneous media, allowing accurate solution approximations even with complex boundary conditions [41]. Integrating PINNs with GNN architectures ensures solutions align with underlying physical laws, improving accuracy and interpretability [14]. This innovative approach enhances flexibility and precision in capturing transport and diffusion dynamics, establishing a robust platform for solving intricate PDEs in CFD [2, 29, 24, 25].

6.8 Learning Solutions to Diverse PDEs

GNNs have demonstrated exceptional adaptability in learning solutions to diverse PDEs, significantly enhancing computational modeling across various scientific fields. Recent advancements include developing hybrid models like GCN-FFNN, integrating graph convolutional networks with feed-forward neural networks to process both graph and grid data simultaneously, effectively tackling nonlinear PDEs [2, 37, 29, 24, 43]. GNNs' capability to handle complex graph-structured data allows them to approximate solutions to PDEs with varying characteristics effectively, particularly in scenarios where traditional numerical methods may struggle with scalability or require extensive computational resources. Incorporating advanced machine learning techniques, such as transfer learning and domain adaptation, further enhances GNNs' ability to learn solutions to diverse PDEs, allowing rapid adaptation to new problems and datasets [35]. The application of GNNs to a wide range of PDEs highlights their potential to revolutionize computational modeling by providing scalable, accurate, and efficient solutions [2, 37, 29, 25, 26].

6.9 High-Dimensional PDEs

High-dimensional PDEs pose significant challenges in computational physics due to the curse of dimensionality, leading to an exponential increase in computational requirements. While traditional methods struggle with complexity and cost, GNNs offer a promising solution by leveraging their ability to model complex dependencies in high-dimensional spaces. Recent advancements, such as multilevel Picard (MLP) approximation methods, have demonstrated potential for overcoming these challenges, offering a rigorous framework for approximating solutions with polynomially bounded computational effort [94, 95]. GNNs' graph-based approach enables efficient representation and computation of spatial relationships across multiple dimensions, providing a scalable and flexible framework for approximating solutions to high-dimensional PDEs [16]. Integrating advanced machine learning techniques, such as transfer learning and domain adaptation, further enhances GNNs' ability to tackle high-dimensional PDEs by enabling rapid adaptation to new problems and datasets [35]. This approach allows GNNs to maintain consistency with the theoretical foundations of the equations while efficiently handling the computational complexity associated with high-dimensional problems [14]. By overcoming traditional methods' limitations, GNNs pave the way for innovative solutions in high-dimensional computational physics, enabling more accurate and scalable simulations across diverse scientific and engineering applications [53, 29, 25].

6.10 Generalization in Navier-Stokes and Poisson Equations

GNNs have demonstrated substantial promise in effectively generalizing solutions for complex PDEs, such as the Navier-Stokes equations governing fluid dynamics and the Poisson equations foundational in electrostatics. Recent advancements include a novel computational framework that combines GNNs with the finite volume method, enabling direct PDE solutions during training without pre-computed data, enhancing adaptability to various unstructured grids and boundary conditions [29, 2]. The inherent ability of GNNs to capture spatial dependencies on graph-structured data makes them particularly suited for these PDEs, which often involve intricate boundary conditions and nonlinear dynamics. The Navier-Stokes equations pose significant challenges due to their nonlinear nature, but novel approaches utilizing Coupled Forward-Backward Stochastic Differential Systems offer a probabilistic framework for solutions [96]. For the Poisson equation, GNNs offer a flexible framework to approximate solutions efficiently across different geometries and boundary conditions, crucial for achieving precise predictions in complex scenarios [37, 29]. By integrating advanced machine learning techniques with graph-based modeling, GNNs provide scalable and efficient solutions for tackling challenges associated with these fundamental PDEs, paving the way for more accurate and reliable simulations in fluid dynamics and electrostatics [37, 29].

7 Challenges and Future Directions

Exploring the challenges and future directions of Graph Neural Network (GNN)-based solvers for Partial Differential Equations (PDEs) is vital in computational physics. This section addresses the multifaceted issues researchers face, including computational challenges, model accuracy, robustness, and domain-specific integration. A systematic approach to these challenges is crucial for advancing GNN methodologies.

7.1 Computational Challenges and Scalability

GNN-based solvers for PDEs encounter significant computational and scalability challenges due to high dimensionality and problem complexity. The curse of dimensionality leads to exponential increases in computational resources for high-dimensional PDEs, making traditional numerical methods impractical [35]. This is exacerbated by the need for numerous collocation points to enforce physics-informed constraints, especially in higher dimensions [14].

Performance degradation with unseen PDE parameters limits GNN solvers' scalability, as seen in frameworks like HyperPNODE [5]. The structural complexity of matrices in linear hyperbolic systems significantly influences outcomes [97]. Methods like DLGA-PDE require systematic hyperparameter tuning for optimal results [16].

Infinite-dimensional aspects, particularly involving first-order derivatives, create verification obstacles, leading to inefficiencies [39]. The exponential growth in computational effort with increasing dimensionality remains a critical challenge [33]. Additionally, reliance on static priors and noise in simulations complicates scalability [23].

Regions of vanishing magnitude can result in larger relative errors, especially in non-local Fokker-Planck equations, indicating a need for innovative management strategies [46]. The complexity of PDEs increases with their order, complicating differential condition satisfaction.

AI-generated, for reference only.

Future research should focus on optimizing GNN architectures, enhancing data utilization, and extending results to various PDE types while addressing generalization errors. Recent advancements, such as auxiliary-task learning and multi-level graph frameworks, have improved accuracy and generalization capabilities, allowing GNN solvers to efficiently manage diverse boundary conditions and mesh refinements. This evolution highlights the potential of GNNs to surpass traditional numerical methods and provides a robust foundation for tackling complex physical problems across fields like fluid dynamics and climate modeling [45, 37, 29, 25, 79].

7.2 Model Accuracy and Robustness

The accuracy and robustness of GNN-based solvers for PDEs are critical for their effectiveness in computational physics. A primary challenge is the sensitivity to the quality and quantity of training data, significantly impacting generalization across different PDE types and conditions [39]. The reliance on large datasets poses challenges, especially when data is scarce or costly to obtain [35].

GNN solvers' approximation capabilities are often limited by the underlying neural network architectures, which may struggle to accurately capture the complex dynamics of high-dimensional PDEs [33]. Noise in training data can lead to overfitting and reduced robustness [23]. While integrating physics-informed constraints benefits adherence to physical laws, it also introduces complexity that can affect accuracy if not managed carefully [14].

Robustness is further challenged by variability in PDE parameters and boundary conditions, causing significant deviations in model performance [16]. The need for hyperparameter tuning and the sensitivity of predictions to these settings complicate robust solution development [5]. Structural complexities in certain PDEs, such as those with non-local interactions or stochastic components, exacerbate the difficulties in achieving accurate solutions [46].

Future research should prioritize developing advanced GNN architectures capable of capturing the nuances of complex PDEs. Innovative techniques to enhance data efficiency and improve model generalization are essential. This could include hybrid neural-numerical approaches utilizing message passing for stability across boundary conditions and leveraging unsupervised training algorithms to optimize discretization. Strategies like the Systematic Model Augmentation for Robust Training (SMART) should also be explored to mitigate data scarcity issues and enhance robustness, ultimately leading to more accurate predictions across diverse PDE applications [53, 25, 29, 2]. Improving GNN solvers' accuracy and robustness will ensure their effectiveness in solving a wide range of PDE problems across various scientific and engineering domains.

7.3 Domain-Specific Integration and Applicability

Integrating domain-specific knowledge into GNN-based solvers is pivotal for enhancing their applicability and performance in solving PDEs across diverse scientific and engineering fields. Incorporating domain insights can significantly improve the accuracy and robustness of surrogate models, enabling them to effectively capture the intricate dynamics of complex systems [98]. One promising direction is enhancing frameworks like Mesh-Informed Neural Networks (MINN) by incorporating additional physical knowledge, aligning model predictions with known physical laws [99].

Advancing competitive numerical solutions for complex PDEs can be achieved by combining high-performance deep learning methods with domain-specific approaches, as demonstrated in solving elliptic equations using Brownian motion techniques [100]. This integration not only enhances model fidelity but also enables handling a broader range of parametric PDEs, as shown by methods like the Meta-AutoDecoder (MAD), which benefit from incorporating domain-specific knowledge [101].

Furthermore, extending the applicability of GNN-based solvers to complex classes of stochastic PDEs (SPDEs) involves exploring adaptive techniques and integrating richer classes of multivariate Gaussian Random Fields (GRFs) into broader statistical frameworks. This approach underscores the importance of domain-specific integration in managing the variability and complexity inherent in real-world conditions. Advanced methodologies such as Transolver, which utilizes PhysicsAttention for adaptive segmentation of discretized domains based on intrinsic physical states, and D3M, a deep domain decomposition method for formulating PDE solutions as constrained optimization problems, significantly enhance solver adaptability and effectiveness. These innovations capture intricate physical correlations in complex geometries and facilitate efficient computations, improving performance in large-scale industrial applications and engineering designs [80, 102].

Future research should also emphasize exploring inverse problems using methodologies like Annealed Adaptive Importance Sampling (AAIS), which can benefit from integrating domain-specific knowledge to improve solution accuracy and reliability [87]. Extending the theory to include non-monotone graphs may provide deeper insights into solution behavior, further illustrating the critical role of domain-specific integration in advancing GNN solvers' applicability [97].

Integrating domain-specific knowledge is crucial for enhancing the effectiveness and versatility of GNN-based solvers in tackling complex PDEs. This approach improves generalization across diverse physical scenarios—such as varying geometries, boundary conditions, and discretization methods—and fosters the development of innovative computational techniques in computational physics. Leveraging advances in neural message passing, differentiable methods, and robust training strategies enables GNN frameworks to achieve superior performance and adaptability, paving the way for groundbreaking solutions in scientific and engineering applications [53, 29, 63, 25].

7.4 Handling High-Dimensional and Nonlinear Problems

Addressing high-dimensional and nonlinear PDEs necessitates innovative strategies that integrate advanced mathematical and computational techniques. Future research should enhance the robustness of methods for chaotic systems and explore their application to problems lacking analytical or numerical solutions [35]. This approach could significantly improve solution efficiency and accuracy, broadening the applicability of deep learning methods to a wider class of PDEs.

Exploring center manifold methods presents a promising avenue for tackling complex PDEs, particularly in understanding the implications of various noise types on center manifold existence and structure. This could yield significant insights into the intricate dynamics of complex systems by leveraging graph-theoretic representations and learning frameworks, enhancing our capability to manage high-dimensional problems across disciplines such as physics, engineering, and economics [6, 103, 57]. Additionally, developing adaptive frameworks for solving initial value problems, especially in nonlinear hyperbolic conservation laws, can systematically improve regularization and performance, offering flexibility in high-dimensional challenges.

Enhancing the computational efficiency of current numerical methods and investigating their applications to intricate physical systems are essential for expanding numerical solvers' capabilities, particularly in advancing physics-informed neural networks (PINNs). Recent developments, such as auxiliary-task learning-based PINNs, have shown promise in addressing limitations like low accuracy and non-convergence, which hinder effectiveness in complex scenarios. Integrating auxiliary-task learning modes has led to significant accuracy improvements, showcasing the potential for these enhanced methods to tackle sophisticated PDEs and broaden numerical applications across scientific and engineering disciplines [6, 79]. The integration of adjoint methods in emerging fields like machine learning and complex systems may further enhance GNN applicability for PDEs, providing new innovation opportunities. Optimizing surrogate model training within the Hierarchical Neural Hybrid (HNH) method and exploring extensions to broader stochastic problem classes could significantly enhance current approaches.

Implementing advanced strategies, such as the infinite-dimensional deflation algorithm for identifying distinct solutions and the robust deep learning-genetic algorithm (R-DLGA) for robust PDE discovery, can significantly improve the ability to tackle high-dimensional and nonlinear PDEs. These methodologies enhance convergence rates and stability in complex scenarios—such as those involving sparse data, high noise, and shock waves—while facilitating systematic exploration of diverse solutions across scientific and engineering fields, ultimately leading to innovative problem-solving approaches [36, 59].

7.5 Data Quality and Training Efficiency

The development of GNN-based solvers for PDEs is intrinsically linked to the quality of training data and the efficiency of the training process. High-quality data is essential for ensuring model accuracy and robustness, influencing generalization across various PDE scenarios. This dependency is particularly pronounced in cases with limited datasets, where data heterogeneity and sparsity can adversely affect GNN performance [104]. Refining models to account for multidimensional interactions highlights the critical role of data quality in enhancing GNN solvers' predictive capabilities.

Training efficiency is equally crucial, determining the scalability and practicality of deploying GNN-based solvers in real-world applications. Integrating advanced machine learning techniques, such as adversarial multitask learning, presents a promising avenue for enhancing training efficiency by enabling simultaneous optimization of multiple tasks, thereby improving computational scalability [82]. This approach enhances the training process and facilitates incorporating diverse datasets, improving model generalization.

Future research should focus on developing effective loss functions to control relative errors in low-magnitude regions and exploring advanced techniques like trapz-PiNNs for PDEs with complex boundary conditions [19]. Extending methods to broader classes of nonlinear PDEs and Backward Stochastic Differential Equations (BSDEs), along with optimizing computational aspects, are critical areas for investigation [21]. Addressing these challenges can significantly enhance GNN-based solvers' capabilities to handle complex PDE problems, paving the way for innovative solutions in computational physics and beyond.

8 Conclusion

The exploration of Graph Neural Networks (GNNs) within computational physics underscores their transformative capacity in solving Partial Differential Equations (PDEs). GNNs offer a sophisticated framework for capturing complex spatial dependencies, thereby surpassing the limitations of traditional numerical solutions. By integrating with cutting-edge machine learning methodologies such as physics-informed neural networks (PINNs) and self-supervised learning, GNNs significantly enhance the precision and computational efficiency of PDE solvers across various applications, including fluid dynamics, quantum simulations, and high-dimensional systems.

Notable advancements include the development of surrogate PDE solvers leveraging GNN architectures, which facilitate efficient approximations in intricate systems. These solvers provide considerable computational benefits, especially in handling high-dimensional and nonlinear PDEs, effectively addressing the challenges posed by the curse of dimensionality. Furthermore, the application of GNNs in modeling stochastic PDEs highlights their ability to manage uncertainties in complex environments, offering robust solutions that were previously unattainable with classical methods.

The survey emphasizes the groundbreaking potential of methodologies like DeepONet in PDE solutions, highlighting the necessity for continued research in operator learning techniques. The XPINN approach demonstrates substantial improvements in training efficiency and solution accuracy, indicating its significant impact on solving nonlinear PDEs in computational physics. Additionally, the DiffOAS method accelerates PDE dataset generation while maintaining high precision, showcasing its potential in advancing neural operator training.

The ongoing refinement of algorithms to accommodate complex interaction dynamics and the exploration of applications across diverse domains, including biological modeling and traffic flow, remain crucial. Future research should focus on enhancing the robustness and applicability of GNN-based solvers, ensuring their validation through comprehensive numerical experiments. This continued innovation is essential for advancing the role of GNNs in computational physics, paving the way for new breakthroughs at the intersection of deep learning and numerical analysis.

References

[1] Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling, 2022.

[2] Onur Bilgin, Thomas Vergutz, and Siamak Mehrkanoon. Gcn-ffnn: A two-stream deep model for learning solution to partial differential equations, 2022.

[3] Hao Xu, Haibin Chang, and Dongxiao Zhang. Dlga-pde: Discovery of pdes with incomplete candidate library via combination of deep learning and genetic algorithm, 2020.

[4] Athanasios S. Fokas, Beatrice Pelloni, and Baoqiang Xia. Evolution equations on time-dependent intervals, 2019.

[5] Abel Gurung and Qingguang Guan. Hybrid pde-deep neural network model for calcium dynamics in neurons, 2024.

[6] Tan Bui-Thanh. Adjoint and its roles in sciences, engineering, and mathematics: A tutorial, 2023.

[7] Tianshu Wen, Kookjin Lee, and Youngsoo Choi. Reduced-order modeling for parameterized pdes via implicit neural representations, 2023.

[8] Reimar H. Leike and Torsten A. Enßlin. Towards information optimal simulation of partial differential equations, 2017.

[9] Marjeta Kramar Fijavž, Delio Mugnolo, and Serge Nicaise. Linear hyperbolic systems on networks, 2020.

[10] Kouta Sekine, Mitsuhiro T. Nakao, and Shin'ichi Oishi. A new formulation for the numerical proof of the existence of solutions to elliptic problems, 2019.

[11] Wouter van Harten and Laura Scarabosio. Exploiting locality in sparse polynomial approximation of parametric elliptic pdes and application to parameterized domains, 2023.

[12] Alessandro Alla, Agnese Pacifico, Michele Palladino, and Andrea Pesare. Online identification and control of pdes via reinforcement learning methods, 2023.

[13] Rambod Mojgani and Maciej Balajewicz. Physics-aware registration based auto-encoder for convection dominated pdes, 2020.

[14] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[15] Wenbo Cao and Weiwei Zhang. Machine learning of partial differential equations from noise data, 2021.

[16] Oisín M. Morrison, Federico Pichi, and Jan S. Hesthaven. Gfn: A graph feedforward network for resolution-invariant reduced operator learning in multifidelity applications, 2024.

[17] Massimo Frittelli and Ivonne Sgura. Matrix-oriented fem formulation for stationary and time-dependent pdes on x-normal domains, 2021.

[18] Zhanhong Ye, Xiang Huang, Leheng Chen, Hongsheng Liu, Zidong Wang, and Bin Dong. Pdeformer: Towards a foundation model for one-dimensional partial differential equations, 2025.

[19] Senbao Jiang and Xiaofan Li. Solving non-local fokker-planck equations by deep learning, 2022.

[20] Anthony Baez, Wang Zhang, Ziwen Ma, Subhro Das, Lam M. Nguyen, and Luca Daniel. Guaranteeing conservation laws with projection in physics-informed neural networks, 2024.

[21] Weinan E, Martin Hutzenthaler, Arnulf Jentzen, and Thomas Kruse. On multilevel picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations, 2017.

[22] Yangfei Liao, Yifan Wang, and Hehu Xie. Solving high dimensional partial differential equations using tensor type discretization and optimization process, 2022.

[23] Yan Xie, Minrui Lv, and Chensong Zhang. Mgcnn: a learnable multigrid solver for sparse linear systems from pdes on structured grids, 2024.

[24] Filipe de Avila Belbute-Peres, Thomas D. Economon, and J. Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction, 2020.

[25] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers, 2023.

[26] Nicola Rares Franco, Stefania Fresca, Filippo Tombari, and Andrea Manzoni. Deep learning-based surrogate models for parametrized pdes: handling geometric variability through graph neural networks, 2023.

[27] Christian Beck, Martin Hutzenthaler, Arnulf Jentzen, and Benno Kuckuck. An overview on deep learning-based approximation methods for partial differential equations, 2022.

[28] Yeonjong Shin, Jerome Darbon, and George Em Karniadakis. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes, 2020.

[29] Tianyu Li, Yiye Zou, Shufan Zou, Xinghua Chang, Laiping Zhang, and Xiaogang Deng. A fully differentiable gnn-based pde solver: With applications to poisson and navier-stokes equations, 2024.

[30] Georgios Is. Detorakis. Practical aspects on solving differential equations using deep learning: A primer, 2024.

[31] Xinquan Huang, Wenlei Shi, Qi Meng, Yue Wang, Xiaotian Gao, Jia Zhang, and Tie-Yan Liu. Neuralstagger: Accelerating physics-constrained neural pde solver with spatial-temporal decomposition, 2023.

[32] Naghmana Tehseen and Geoff Prince. Integration of pdes by differential geometric means, 2013.

[33] A. M. P. Boelens, D. Venturi, and D. M. Tartakovsky. Parallel numerical tensor methods for high-dimensional pdes, 2018.

[34] Mildred Aduamoah, Benjamin D. Goddard, John W. Pearson, and Jonna C. Roden. Pde-constrained optimization models and pseudospectral methods for multiscale particle dynamics, 2021.

[35] Shiyin Wei, Xiaowei Jin, and Hui Li. General solutions for nonlinear differential equations: a rule-based self-learning approach using deep reinforcement learning, 2019.

[36] Hao Xu and Dongxiao Zhang. Robust discovery of partial differential equations in complex situations, 2021.

[37] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020.

[38] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*, 2019.

[39] Chirag Adwani, Roberto De Leo, and James A. Yorke. What is the graph of a dynamical system?, 2024.

[40] Ronny Bergmann and Daniel Tenbrinck. A graph framework for manifold-valued data, 2017.

[41] Ali Taghibakhshi, Scott MacLachlan, Luke Olson, and Matthew West. Optimization-based algebraic multigrid coarsening using reinforcement learning, 2022.

[42] A. J. Roberts. Resolve subgrid microscale interactions to discretise stochastic partial differential equations, 2010.

[43] Moshe Eliasof, Eldad Haber, and Eran Treister. Graph neural reaction diffusion models, 2024.

[44] Jonas Baggenstos and Diyora Salimova. Approximation properties of residual neural networks for kolmogorov pdes, 2021.

[45] Hrishikesh Viswanath, Md Ashiqur Rahman, Abhijeet Vyas, Andrey Shor, Beatriz Medeiros, Stephanie Hernandez, Suhas Eswarappa Prameela, and Aniket Bera. Neural operator: Is data all you need to model the world? an insight into the impact of physics informed machine learning, 2023.

[46] Ke Li, Kejun Tang, Jinglai Li, Tianfan Wu, and Qifeng Liao. A hierarchical neural hybrid method for failure probability estimation, 2019.

[47] Esteban Samaniego, Cosmin Anitescu, Somdatta Goswami, Vien Minh Nguyen-Thanh, Hongwei Guo, Khader Hamdia, Xiaoying Zhuang, and Timon Rabczuk. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 362:112790, 2020.

[48] Andrew Cohen, Weiping Dou, Jiang Zhu, Slawomir Koziel, Peter Renner, Jan-Ove Mattsson, Xiaomeng Yang, Beidi Chen, Kevin Stone, and Yuandong Tian. Sample-efficient surrogate model for frequency response of linear pdes using self-attentive complex polynomials, 2023.

[49] Mohammad Amin Nabian and Hadi Meidani. A deep neural network surrogate for high-dimensional random partial differential equations, 2018.

[50] Francisco Holguin, GS Sidharth, and Gavin Portwood. Multigrid solver with super-resolved interpolation, 2021.

[51] Philipp Grohs, Fabian Hornung, Arnulf Jentzen, and Philippe Von Wurstemberger. *A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations*, volume 284. American Mathematical Society, 2023.

[52] Winfried van den Dool, Tijmen Blankevoort, Max Welling, and Yuki M. Asano. Efficient neural pde-solvers using quantization aware training, 2023.

[53] Yunpeng Gong, Yongjie Hou, Zhenzhong Wang, Zexin Lin, and Min Jiang. Adversarial learning for neural pde solvers with sparse data, 2024.

[54] D. J. Saunders, O. Rossi, and G. E. Prince. Tangent bundle geometry induced by second order partial differential equations, 2015.

[55] Thomas März and Colin B. Macdonald. Calculus on surfaces with general closest point functions, 2012.

[56] Karna Gowda and Christian Kuehn. Early-warning signs for pattern-formation in stochastic partial differential equations, 2014.

[57] R. Banerjee, K. Sagiyama, G. H. Teichert, and K. Garikipati. A graph theoretic framework for representation, exploration and analysis on computed states of physical systems, 2019.

[58] Fernando A Morales and Sebastián Naranjo Álvarez. The interaction between pde and graphs in multiscale modeling, 2016.

[59] Patrick E. Farrell, Ásgeir Birkisson, and Simon W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations, 2015.

[60] Vinicius L. S. Silva, Claire E. Heaney, Nenko Nenov, and Christopher C. Pain. Generative network-based reduced-order model for prediction, data assimilation and uncertainty quantification, 2023.

[61] Silvia Bertoluzza, Erik Burman, and Cuiyu He. Best approximation results and essential boundary conditions for novel types of weak adversarial network discretizations for pdes, 2024.

[62] Tobias Alt, Karl Schrader, Matthias Augustin, Pascal Peter, and Joachim Weickert. Connections between numerical algorithms for pdes and neural networks, 2022.

[63] Ameya D Jagtap and George Em Karniadakis. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2020.

[64] Edward Small. An analysis of physics-informed neural networks, 2023.

[65] Tara Akhound-Sadegh, Laurence Perreault-Levasseur, Johannes Brandstetter, Max Welling, and Siamak Ravanbakhsh. Lie point symmetry and physics informed networks, 2023.

[66] Grégoire Mialon, Quentin Garrido, Hannah Lawrence, Danyal Rehman, Yann LeCun, and Bobak T. Kiani. Self-supervised learning with lie symmetries for partial differential equations, 2024.

[67] Yiheng Du, Nithin Chalapathi, and Aditi Krishnapriyan. Neural spectral methods: Self-supervised learning in the spectral domain, 2024.

[68] Yaohua Zang and Gang Bao. Particlewnn: a novel neural networks framework for solving partial differential equations, 2023.

[69] Ziad Aldirany, Régis Cottereau, Marc Laforest, and Serge Prudhomme. Multi-level neural networks for accurate solutions of boundary-value problems, 2023.

[70] Jon A. Rivera, Jamie M. Taylor, Ángel J. Omella, and David Pardo. On quadrature rules for solving partial differential equations using neural networks, 2021.

[71] Xiaokai Huo and Hailiang Liu. Inf-sup neural networks for high-dimensional elliptic pde problems, 2024.

[72] Huanshuo Dong, Hong Wang, Haoyang Liu, Jian Luo, and Jie Wang. Accelerating pde data generation via differential operator action in solution space, 2024.

[73] Chen Cui, Kai Jiang, and Shi Shu. Solving time-dependent parametric pdes by multiclass classification-based reduced order model, 2022.

[74] Ruiying Xu, Xu-Hui Zhou, Jiequn Han, Richard P. Dwight, and Heng Xiao. A pde-free, neural network-based eddy viscosity model coupled with rans equations, 2022.

[75] Yuki Sato, Hiroyuki Tezuka, Ruho Kondo, and Naoki Yamamoto. Quantum algorithm for partial differential equations of non-conservative systems with spatially varying parameters, 2025.

[76] Igor Cialenco, Gregory E. Fasshauer, and Qi Ye. Approximation of stochastic partial differential equations by a kernel-based collocation method, 2012.

[77] Konrad Mundinger, Max Zimmer, and Sebastian Pokutta. Neural parameter regression for explicit representations of pde solution operators, 2024.

[78] Kiwon Um, Robert Brand, Yun, Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers, 2021.

[79] Junjun Yan, Xinhai Chen, Zhichao Wang, Enqiang Zhou, and Jie Liu. Auxiliary-tasks learning for physics-informed neural network-based partial differential equations solving, 2023.

[80] Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries, 2024.

[81] Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. Generative learning of the solution of parametric partial differential equations using guided diffusion models and virtual observations, 2024.

[82] Pongpisit Thanasutives, Masayuki Numao, and Ken ichi Fukui. Adversarial multi-task learning enhanced physics-informed neural networks for solving partial differential equations, 2021.

[83] James Duvall, Karthik Duraisamy, and Shaowu Pan. Discretization-independent surrogate modeling over complex geometries using hypernetworks and implicit representations, 2022.

[84] Florent Bonnet, Jocelyn Ahmed Mazari, Thibaut Munzer, Pierre Yser, and Patrick Gallinari. An extensible benchmarking graph-mesh dataset for studying steady-state incompressible navier-stokes equations, 2022.

[85] Mingrui Zhang, Chunyang Wang, Stephan Kramer, Joseph G. Wallwork, Siyi Li, Jiancheng Liu, Xiang Chen, and Matthew D. Piggott. Towards universal mesh movement networks, 2024.

[86] Philipp Grohs, Fabian Hornung, Arnulf Jentzen, and Philippe von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations, 2023.

[87] Zhengqi Zhang, Jing Li, and Bin Liu. Annealed adaptive importance sampling method in pinns for solving high dimensional partial differential equations, 2024.

[88] Tim De Ryck, Florent Bonnet, Siddhartha Mishra, and Emmanuel de Bézenac. An operator preconditioning perspective on training in physics-informed machine learning, 2024.

[89] Gideon Dresdner, Dmitrii Kochkov, Peter Norgaard, Leonardo Zepeda-Núñez, Jamie A. Smith, Michael P. Brenner, and Stephan Hoyer. Learning to correct spectral methods for simulating turbulent flows, 2023.

[90] Hamidreza Eivazi, Mojtaba Tahani, Philipp Schlatter, and Ricardo Vinuesa. Physics-informed neural networks for solving reynolds-averaged navier$x$2013stokes equations, 2021.

[91] Handi Zhang, Langchen Liu, and Lu Lu. Federated scientific machine learning for approximating functions and solving differential equations with data heterogeneity, 2024.

[92] Jérôme Darbon and Tingwei Meng. On some neural network architectures that can represent viscosity solutions of certain high dimensional hamilton–jacobi partial differential equations, 2020.

[93] Fredrik Hellman, Tim Keil, and Axel Målqvist. Numerical upscaling of perturbed diffusion problems, 2019.

[94] Martin Hutzenthaler, Arnulf Jentzen, Thomas Kruse, Tuan Anh Nguyen, and Philippe von Wurstemberger. Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations, 2020.

[95] Michael B. Giles, Arnulf Jentzen, and Timo Welti. Generalised multilevel picard approximations, 2019.

[96] Freddy Delbaen, Jinniao Qiu, and Shanjian Tang. Forward-backward stochastic differential systems associated to navier-stokes equations in the whole space, 2014.

[97] Miroslav Bulíček, Josef Málek, and Erika Maringová. On nonlinear problems of parabolic type with implicit constitutive equations involving flux, 2021.

[98] Nicola R. Franco, Andrea Manzoni, and Paolo Zunino. A deep learning approach to reduced order modelling of parameter dependent partial differential equations, 2022.

[99] Nicola Rares Franco, Andrea Manzoni, and Paolo Zunino. Mesh-informed neural networks for operator learning in finite element spaces, 2023.

[100] Cameron Martin, Hongyuan Zhang, Julia Costacurta, Mihai Nica, and Adam R Stinchcombe. Solving elliptic equations with brownian motion: Bias reduction and temporal difference learning, 2020.

[101] Xiang Huang, Zhanhong Ye, Hongsheng Liu, Beiji Shi, Zidong Wang, Kang Yang, Yang Li, Bingya Weng, Min Wang, Haotian Chu, Fan Yu, Bei Hua, Lei Chen, and Bin Dong. Meta-auto-decoder for solving parametric partial differential equations, 2022.

[102] Ke Li, Kejun Tang, Tianfan Wu, and Qifeng Liao. D3m: A deep domain decomposition method for partial differential equations, 2019.

[103] Yingtao Luo, Chang Xu, Yang Liu, Weiqing Liu, Shun Zheng, and Jiang Bian. Learning differential operators for interpretable time series modeling, 2022.

[104] Krzysztof Burdzy, Jeremy G. Hoskins, and Stefan Steinerberger. From pinned billiard balls to partial differential equations, 2024.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.

AI-generated, for reference only.