

Learning PDE Solvers with Physics and Data: A Unifying View of Physics-Informed Neural Networks and Neural Operators

Yilong Dai¹, Shengyu Chen², Ziyi Wang³, Xiaowei Jia², Yiqun Xie³, Vipin Kumar⁴, Runlong Yu^{1*}

¹The University of Alabama, Tuscaloosa, Alabama, United States

²University of Pittsburgh, Pittsburgh, Pennsylvania, United States

³University of Maryland, College Park, Maryland, United States

⁴University of Minnesota, Minneapolis, Minnesota, United States

{ydai17, ryu5}@ua.edu, {shc160, xiaowei}@pitt.edu, {zoewang, xie}@umd.edu, kumar001@umn.edu

Abstract

Partial differential equations (PDEs) are central to scientific modeling. Modern workflows increasingly rely on machine learning-based components. Despite the emergence of various physics-aware data-driven approaches, the field still lacks a unified perspective to uncover their relationships, limitations, and appropriate roles in scientific workflows. To this end, we propose a unifying perspective that places two dominant paradigms, Physics-Informed Neural Networks (PINNs) and Neural Operators (NOs), within a shared design space. We organize existing methods from three fundamental dimensions: what is learned, how physical structures are integrated into the learning process, and how the computational load is amortized across problem instances. In this way, many practical challenges can be best understood as consequences of these structural properties of learning PDEs. By analyzing recent advances through this unifying view, our survey aims to facilitate the development of reliable learning-based PDE solvers and help catalyze a deeper synthesis of physics and data.

1 Introduction

Partial differential equations (PDEs) are the core language of scientific modeling. They are widely used in fields such as fluid dynamics, electromagnetics, climate modeling, materials science, and biological systems. Although classical numerical solvers are effective for single high-fidelity simulations, many modern scientific workflows increasingly require repeated solutions to PDEs with varying parameters, boundary or initial conditions, geometries, or partial observational data. Such multi-query scenarios naturally arise in design optimization, uncertainty quantification (UQ), inverse problems, and data assimilation. In these cases, the cost of repeatedly executing traditional discretization solution processes can become a bottleneck, especially for high-dimensional, multiscale, or geometrically complex problems, and when sparse or noisy data are incorporated.

These bottlenecks motivated research into *learning-based PDE solvers*, methods that move from computing solutions per instance to learning reusable representations through physics, data, or both. Early purely data-driven approaches can achieve good interpolation performance, but lack generalizability, especially when geometry, boundary conditions, or regimes change. In addition, they may violate physical constraints, including conservation laws. Consequently, these limitations led to two dominant physics-aware paradigms: *Physics-Informed Neural Networks (PINNs)* and *Neural Operators (NOs)*. PINNs approximate individual solutions as neural functions, trained to satisfy governing equations and auxiliary conditions, optionally incorporating observational data [Raissi *et al.*, 2019]. NOs, in contrast, learn the mapping between instance specifications and solution fields covering a family of PDEs, enabling rapid amortized inference after offline training [Lu *et al.*, 2021; Li *et al.*, 2021].

Typically, the community views PINNs and NOs as competing methods for learning-based PDE solving. In this survey, however, we take a different perspective. We view these two paradigms as complementary methods within a shared *physics–data continuum* and a common design space shaped by workflow requirements (see Figure 1). From this perspective, PINNs and NOs have a common objective: approximating the mapping from PDE specifications to their solutions. Therefore, both paradigms face inherent constraints imposed by PDE solution families, including multiscale representation, geometry and boundary feasibility, and trustworthiness under contract shift. Meanwhile, the two paradigms also have systematic differences, because they stem from their fundamental structural decisions: what is being learned (an instance-specific solution versus a family-level operator), how physical structure is incorporated into learning (physics-as-supervision versus paired supervision with physics injection), and where computational cost is amortized (instance-by-instance optimization versus amortized inference across instances). These decisions lead to unique regimes of difficulty and failure patterns for each.

Scope and contributions. This survey reviews the literature from 2017 to 2026 (primarily from top AI and machine learning conferences and mathematics and physics journals). It covers the emergence and recent development of PINNs, NOs, and hybrid variants. Our contributions have three as-

*Runlong Yu is the corresponding author.

pects: (i) a unifying taxonomy based on structural design choices and workflow contracts; (ii) a cross-paradigm analysis to examine structural properties of learning PDEs; and (iii) application-oriented method choice, as well as open challenges and opportunities towards scientific deployment.

Organization. Section 2 defines a general PDE formulation and the unified design space. Following this, Section 3 characterizes core structural properties shared across two learning paradigms. Next, Sections 4 and 5 separately review paradigm-specific developments of PINNs and NOs. After that, Section 6 discusses applications and method choice. Then, Section 7 outlines open challenges and future opportunities. Finally, Section 8 concludes the survey.

2 Problem Setup and Design Space

We introduce a consistent problem setup for learning-based PDE solvers. Our goal is to clarify what is being solved, what is learned, and how learning formulations relate within a shared design space.

PDE problems. We consider PDEs defined on a spatial domain $\Omega \subset \mathbb{R}^d$ and, when applicable, a time interval $t \in [0, T]$. The unknown is a field

$$u : \Omega \times [0, T] \rightarrow \mathbb{R}^m, \quad (1)$$

with d spatial dimensions and m physical variables. Many PDEs can be written abstractly as

$$\mathcal{F}(u(\mathbf{x}, t), \nabla u(\mathbf{x}, t), \dots, D^k u(\mathbf{x}, t); \mathbf{x}, t; \boldsymbol{\mu}, \boldsymbol{\theta}) = 0, \quad (2)$$

where $\mathcal{F}$ denotes the governing differential operator; $\boldsymbol{\mu}$ collects *known* instance specifications (e.g., coefficients, forcing, geometry descriptors, boundary/initial data); and $\boldsymbol{\theta}$ denotes *unknown* quantities to be inferred in inverse settings.

For time-dependent problems, the PDE is complemented by initial and boundary conditions

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathcal{B}[u](\mathbf{x}, t) = g(\mathbf{x}, t) \quad \text{on } \partial\Omega, \quad (3)$$

where $\mathcal{B}$ is a boundary operator and $\partial\Omega$ is the boundary of Ω .

Well-posedness and learning. Classical analysis studies whether Eq. (2)–(3) is well-posed. In learning-based settings, these conditions are often weakened or effectively violated under sparse/noisy observations, unknown parameters, or out-of-contract deployment. Stability is then promoted through inductive bias, regularization, and diagnostics rather than guaranteed purely by the PDE. These departures from classical well-posedness are precisely what motivate learning-based formulations under partial observation and task-specific workflow regimes.

Observations. When data are available, they are often partial and noisy:

$$y_i = \mathcal{H}[u](\mathbf{x}_i, t_i) + \varepsilon_i, \quad (4)$$

where $\mathcal{H}$ is an observation operator and ε_i denotes noise.

Workflow regimes. We will refer to: *forward solving* (given $\boldsymbol{\mu}$, infer u), *inverse/identification* (parts of $\boldsymbol{\theta}$ unknown, infer u and $\boldsymbol{\theta}$ from y), and *data assimilation* (recover spatiotemporal state under sparse/noisy measurements with stability requirements).

Solution learning (PINN baseline). Solution-learning approaches represent the unknown field by $u_\phi(\mathbf{x}, t)$ (typically a coordinate network). Physics enters via residuals, e.g.,

$$r_{\mathcal{F}}(\mathbf{x}, t) = \mathcal{F}(u_\phi(\mathbf{x}, t), \nabla u_\phi(\mathbf{x}, t), \dots; \mathbf{x}, t; \boldsymbol{\mu}, \boldsymbol{\theta}), \quad (5)$$

with analogous residuals for Eq. (3). A canonical formulation minimizes a composite objective

$$\min_{\phi, \boldsymbol{\theta}} \lambda_{\mathcal{F}} \mathcal{L}_{\mathcal{F}} + \lambda_{\mathcal{B}} \mathcal{L}_{\mathcal{B}} + \lambda_{\text{IC}} \mathcal{L}_{\text{IC}} + \lambda_{\text{data}} \mathcal{L}_{\text{data}}, \quad (6)$$

where $\mathcal{L}_{\mathcal{F}}$ measures the PDE residual in Eq. (5), $\mathcal{L}_{\mathcal{B}}$ and $\mathcal{L}_{\text{IC}}$ measure boundary and initial-condition violations in Eq. (3), and $\mathcal{L}_{\text{data}}$ measures the mismatch to observations through $\mathcal{H}$ (Eq. (4)); each term is typically an average squared residual (or misfit) over sampled collocation, boundary/initial, and measurement points, and the weights λ control their relative importance. This baseline corresponds to the core PINN template [Raissi *et al.*, 2019]: learn a *single instance* by treating physics (and optional observations) as supervision.

Operator learning (NO baseline). Operator learning targets a family of PDE instances. Let a denote an *instance specification* encoding $\boldsymbol{\mu}$ as one or more input functions. An operator is a map

$$u = \mathcal{G}(a), \quad (7)$$

trained on paired data $\{(a_j, u_j)\}_{j=1}^N$. Canonical realizations include DeepONet-style factorization [Lu *et al.*, 2021] and Fourier Neural Operators (FNO) [Li *et al.*, 2021]. A defining deployment goal is often *discretization-robust inference*: evaluating $\mathcal{G}$ on discretizations (and sometimes geometries) not identical to those used in training.

A unifying taxonomy. We organize the field through three structural choices.

(T1) Object of learning. PINNs learn an instance-specific solution $u_\phi(\mathbf{x}, t)$. Neural operators learn a family-level operator $\mathcal{G}$ mapping specifications a to solutions.

(T2) Supervision interface. PINNs treat physics constraints as primary supervision. Neural operators treat paired data as primary supervision; physics is injected through architecture, regularization, refinement, or diagnostics [Shu *et al.*, 2023; Bastek *et al.*, 2025].

(T3) Amortization locus. PINNs amortize *supervision* but typically require per-instance training. Neural operators amortize *inference compute* but require data generation and offline training.

Workflow contracts. These structural choices are induced by the *workflow contract* in which a solver is embedded, and they directly shape the dominant failure modes encountered. For PINNs, physics-as-supervision combined with per-instance optimization places numerical conditioning and stiff multi-objective training at the center of the workflow. For neural operators, paired supervision with amortized inference shifts the risk toward distribution shift, discretization mismatch, and generalization of geometry at deployment.

More generally, a learning-based PDE solver can be interpreted relative to a *workflow contract*, which specifies the perturbations permitted at deployment, the invariants that must

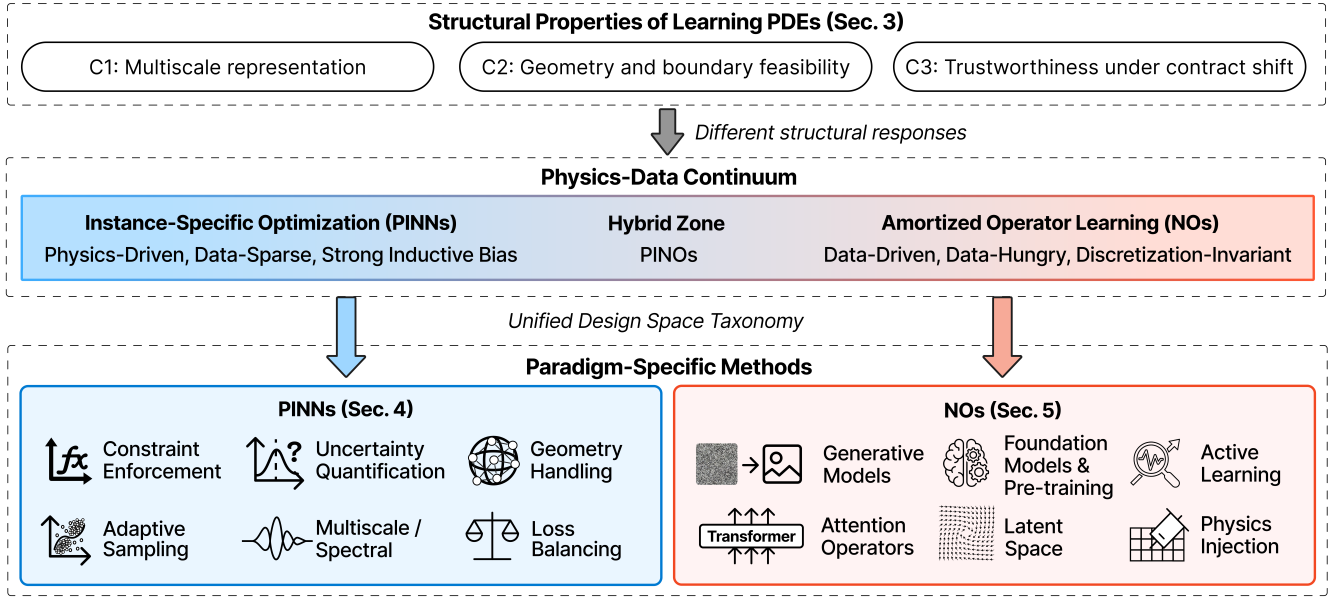


Figure 1: A unifying view of learning-based PDE solvers. Top: three structural properties (C1–C3) that challenge all learning paradigms (Section 3). Middle: a continuum shaped by what is learned and how physics enters supervision, organized within a unified design space taxonomy. Bottom: paradigm-specific innovations for PINNs (left, Section 4) and neural operators (right, Section 5).

be preserved, and the operational signals required (e.g., latency, reliability, and diagnostics). In scientific computing pipelines, such contracts are typically characterized along a small set of recurring axes: (i) *instance variability*, including changes in coefficients, forcing terms, boundary or initial conditions, and unknown parameters θ across queries; (ii) *geometry and discretization variation*, encompassing changes in domain shape, boundary configuration, and grid or mesh representations between training and evaluation; (iii) *temporal horizon*, ranging from single-step inference to long-horizon rollout where small biases can accumulate over time; (iv) *available supervision*, spanning paired simulation data, sparse or noisy observations via $\mathcal{H}$ (Eq. (4)), and physics constraints alone; and (v) *deployment validity and trust signals*, such as boundary feasibility, conservation properties, uncertainty estimates, out-of-distribution detection, and fallback or refinement mechanisms.

Section 3 analyzes structural properties of PDE solution families that make satisfying these workflow contracts challenging in practice, while Sections 4 and 5 examine how different learning paradigms expose distinct *interfaces* for meeting a given contract.

3 Structural Properties of Learning PDEs

This section is deliberately paradigm-agnostic. Rather than enumerating algorithms, we characterize shared *structural properties* of PDE solution families that act as persistent pressure points across learning-based workflows. These properties explain *why* learning PDEs is difficult and *which evaluation signals* are necessary to assess reliability beyond average accuracy. Paradigm-specific mechanisms and design choices are reviewed in Sections 4 and 5. We focus on three recur-

ring properties: (C1) *multiscale representation*, (C2) *geometry and boundary feasibility*, and (C3) *trustworthiness under workflow contract shift*.

3.1 C1: Multiscale representation

PDE solutions are inherently multiscale: sharp boundary layers, intermittent structures, and broadband spectra can coexist with smooth regions. Multiscale difficulty is not solely a matter of expressivity; it arises from the interaction between (i) available learning signals, (ii) finite representations or basis choices, and (iii) the deployment horizon over which predictions are evaluated.

Uneven learnability across scales. Common learning signals (such as residuals, paired supervision, or partial observations) typically allocate error unevenly across frequencies. Low-frequency components are often learned first, while high-frequency content may be delayed, smeared, or aliased depending on representation and discretization.

Pathologies of differential and nonlocal priors. Strong-form differential supervision can amplify high-frequency noise through derivatives, while operator architectures relying on global mixing may blur localized structure when fine-scale content is poorly represented. In long-horizon settings, small scale-dependent biases can accumulate and alter qualitative behavior.

Beyond global MSE, multiscale performance should be evaluated using signals aligned with the workflow contract, including: (i) spectral or band-limited error metrics, (ii) localized errors near interfaces or boundaries, (iii) error growth under rollout (one-step versus multi-step), and (iv) task-level invariants sensitive to fine-scale structure when relevant.

3.2 C2: Geometry and boundary feasibility

Geometry and boundary conditions define the feasible solution set through the domain Ω , boundary $\partial\Omega$, and boundary operator $\mathcal{B}$ (Eq. (3)). Under deployment perturbations, geometry and boundary variation are among the most common sources of workflow contract violation.

Feasibility is an interface problem. A solver must specify how geometric and boundary information enters computation. This interface determines whether feasibility is enforced by construction, weakly encouraged, or left implicit. Concretely, this requires consistent choices for: (i) *domain representation* (grid, mesh, graph, or manifold coordinates), (ii) *boundary representation* (boundary identification and prescribed conditions), and (iii) *feasibility enforcement* (how constraints are imposed or encoded).

Mismatch can be silent. When geometry or boundary information is under-specified or inconsistently represented, violations may not appear in the average training error. Instead, they surface abruptly under deployment perturbations, such as new domain shapes, boundary types, or discretizations.

Geometry and boundary feasibility should be evaluated using: (i) boundary violation metrics (e.g., Dirichlet or Neumann mismatch), (ii) conservation or compatibility metrics tied to boundaries when relevant, (iii) sensitivity to geometry perturbations, and (iv) stability under boundary variation.

3.3 C3: Trustworthiness under contract shift

Scientific deployment requires deciding when predictions remain trustworthy under shifts in the workflow contract. Mean accuracy alone is insufficient, as failures can be silent and operationally costly. Trustworthiness is therefore a system-level property, encompassing both predictions and the mechanisms used to assess their validity.

Distribution shift is structured. Contract shift typically occurs along a small number of structured axes, including parameters or regimes, geometry or boundary types, discretization or resolution, rollout horizon, and observation operators. Methods that perform well under interpolation may fail under these shifts without explicit warning signals.

Operational signals and fallback mechanisms. Trustworthy systems surface uncertainty, detect out-of-contract inputs, and couple predictions to validity checks, such as residual or feasibility diagnostics, that can trigger refinement or fallback.

Trustworthiness evaluation should include: (i) calibration and sharpness when probabilistic outputs are available, (ii) out-of-distribution detection and selective prediction, (iii) constraint violation rates and stability indicators, and (iv) the ability to trigger refinement or fallback under stress tests. These signals should be aligned with the workflow contract, not only the average error.

4 Physics-Informed Neural Networks

This section reviews developments implied by the PINN structural choice: learning an instance-specific solution via physics-based supervision and per-instance optimization (Eq. (6)). The shared pressures (*why* learning PDEs is hard) and the recommended evaluation signals are defined once in

Section 3. Here we focus strictly on the *PINN interfaces*—residual construction, constraint enforcement, representation, optimization, and sampling—that determine whether a PINN meets a given workflow contract.

4.1 Baseline recap and canonical failure modes

Physics-Informed Neural Networks, popularized by Raissi et al. [Raissi et al., 2019], train an instance-specific neural solution by minimizing a weighted sum of physics residuals, auxiliary-condition violations, and optional data mismatch (Eq. (6)). Their appeal is data efficiency, mesh-free evaluation, and natural support for inverse problems and data assimilation (unknown θ and partial observations in Eq. (4)).

Empirically, the PINN research agenda is shaped by training-loop failure modes: (i) strong-form residuals can be poorly conditioned for stiff PDEs and long-time dynamics; (ii) soft penalties can converge to low average loss but leave localized feasibility violations of Eq. (3); (iii) multi-term objectives exhibit gradient conflicts across $\mathcal{L}_{\mathcal{F}}$, $\mathcal{L}_{\mathcal{B}}$, $\mathcal{L}_{\text{IC}}$, $\mathcal{L}_{\text{data}}$; (iv) uniform collocation under-allocates samples to localized complexity; (v) automatic differentiation makes residual evaluation expensive for high-order derivatives and large collocation sets. Most PINN variants can be interpreted as design knobs targeting these mechanisms.

4.2 Design knobs for practical PINNs

We group the literature into five knobs that map directly to the terms in Eq. (6) and the failure modes above.

(K1) Residuals: strong vs. weak

Pointwise strong-form residuals are simple but can be stiff or noisy. Weak/variational/integral objectives improve conditioning by aligning the learning signal with Galerkin/FEM reasoning [Saleh et al., 2024; Gao et al., 2022].

(K2) Constraints: soft vs. hard

To meet feasibility requirements under geometry/boundary variation (contract axis C2), many PINNs move beyond soft penalties. Hard/semi-hard satisfaction can be achieved by constraint-satisfying parameterizations, projection, or physics-embedded discretized operators [Chen et al., 2021; Chalapathi et al., 2024]. These designs directly target boundary and auxiliary operators $\mathcal{B}$ in Eq. (3).

(K3) Representation: spectral and multiscale

Rather than re-motivating multiscale difficulty (Section 3.1), we focus on PINN-side remedies that modify the *training interface*: representations that reduce derivative amplification and accelerate learning of high frequencies. This includes Fourier encodings and multiresolution architectures [Kang et al., 2025; Fang et al., 2024].

(K4) Optimization: balance and curvature

PINN training behaves like a numerical procedure over a discretized objective: sampling, resolution, and loss weighting act like discretization and step-size choices. Dynamic balancing mitigates gradient conflicts across loss terms [Hwang and Lim, 2024; Liu et al., 2025a]. Curvature-aware and preconditioned strategies stabilize stiff dynamics and long-time integration under PDE constraints [Dangel et al., 2024].

Failure-mode analyses relate difficulties to loss landscape structure [Rathore *et al.*, 2024]. Meta-learned optimization transfers update rules across instances [Wandel *et al.*, 2025].

(K5) Sampling: adaptive collocation

Collocation is the discretization of the training signal in space-time. Residual-based refinement and physics-aware sampling allocate collocation where needed near violations [Wu *et al.*, 2023]. Information acquisition views collocation as experimental design [Lau *et al.*, 2024].

4.3 Scaling and geometry for PINNs

We detail the *PINN-side geometry interface*: how domain and boundary information enters the *training loop* through residual evaluation, constraint enforcement, and sampling.

Residual evaluation on discretizations. On meshes, point clouds, or manifolds, the central question is how to evaluate the operator signal consistently. This includes discretized differential operators, weak forms on elements/patches, intrinsic coordinates on manifolds, and geometry-aware quadrature rules. Graph-based PINNs and message passing can be viewed as transporting local operator structure onto discrete neighborhoods [Zeng *et al.*, 2025; Jia *et al.*, 2023].

Boundary and constraint enforcement. Geometry enters PINNs not only through $\mathcal{F}$ but through the practical representation of $\partial\Omega$ and the operator $\mathcal{B}$. Hard/semi-hard approaches build feasibility into parameterizations or enforce it via projection, reducing reliance on sampling density near boundaries [Chen *et al.*, 2021; Chalapathi *et al.*, 2024].

Sampling on geometry. Adaptive strategies must respect geometry: boundary sampling, curvature-aware sampling on manifolds, and refinement near interfaces where feasibility violations concentrate spatially. Manifold-aware PINNs use intrinsic coordinates and curvature-aware designs to reduce coordinate-induced artifacts during training [Sahli Costabal *et al.*, 2024; Liao *et al.*, 2025].

4.4 Computational efficiency

A central bottleneck is repeated residual evaluation with automatic differentiation. Strategies include domain decomposition and parallelization [Shukla *et al.*, 2021], hybrid AD and numerical differentiation [Sharma and Shankar, 2022], and hybrid solver/operator-splitting designs [Chen *et al.*, 2024].

4.5 Inverse problems, uncertainty, and diagnostics

For PINNs, trustworthiness (Section 3.3) is largely mediated by the *inference interface*: unknowns θ and sparse observations can render learning ill-posed, so reliable deployment requires uncertainty estimates and diagnostics tied to the contract. Bayesian/variational methods and robust regression variants quantify uncertainty under sparse/noisy data [Yang *et al.*, 2021; Hamelijnck *et al.*, 2024]. Generative formulations provide distributional field inference [Daw *et al.*, 2021]. Benchmarks and certification support diagnosis and reproducibility [Hao *et al.*, 2024; Eiras *et al.*, 2024].

4.6 Bridging toward reuse

Although vanilla PINNs are per-instance solvers, hypernetworks and parameter-conditioned PINNs move toward reuse across instances [Cho *et al.*, 2024], and meta-learning amortizes adaptation across PDE families [Penwarden *et al.*, 2023; Psaros *et al.*, 2022; Chen *et al.*, 2023]. These directions connect naturally to hybrids by shifting part of per-instance optimization into reusable representations.

5 Neural Operators

This section reviews developments implied by the operator-learning structural choice: learning an instance-to-solution mapping with paired data (Eq. (7)) for amortized inference and discretization-robust deployment. The shared pressures and evaluation signals are defined once in Section 3. Here we focus strictly on *operator interfaces*: how a and u are represented, how nonlocal mixing is implemented, and how robustness is engineered under the workflow contract.

5.1 Baseline recap and dominant risks

A neural operator learns a map $u = \mathcal{G}(a)$ from an input function space to a solution function space, trained on paired instances. A defining goal is to evaluate $\mathcal{G}$ on discretizations different from those used in training, shaping how inputs/outputs are parameterized and how nonlocal mixing is implemented. Canonical templates include DeepONet [Lu *et al.*, 2021] and FNO [Li *et al.*, 2021], but most later operators can be interpreted as modifying: (i) function parameterization (how a and u are discretized/encoded), (ii) nonlocal interaction mechanisms (how information mixes globally/locally), and (iii) geometry/boundary interfaces (how domains and boundary data enter a).

The dominant risks differ from PINNs because operators are deployed as surrogates: (i) *dataset coverage* and distribution shift in a , (ii) *discretization and geometry transfer* (contract axis C2), and (iii) *long-horizon error accumulation* in dynamical rollouts (C3). Accordingly, much of the literature focuses on representation and interfaces robust under contract transfer, and on system-level mechanisms (diagnostics/refinement/fallback) for safe deployment.

5.2 Geometry, discretization, boundary interfaces

We detail the *operator-side geometry interface*: how to encode a and decode u so that $u = \mathcal{G}(a)$ remains well-defined across grids/meshes/domains, and how to inject boundary information without breaking discretization robustness.

Grid vs mesh/graph operators as interface choices. Grid-based operators (e.g., FNO-style) assume a structured discretization for efficient global mixing. Mesh/graph operators generalize mixing to irregular discretizations and can encode conservation and symmetry through message passing [Wang *et al.*, 2024b; Xu *et al.*, 2024]. This is to make the *mapping* stable when discretizations change.

Domain variation and coordinate alignment. When domains vary, operators often introduce alignment mechanisms: reference mappings/diffeomorphisms, intrinsic bases on manifolds, or kernel/integral approximations that operate

in coordinate-free ways [Lingsch *et al.*, 2024]. Adaptive discretization complements model capacity with classical meshing choices [Hu *et al.*, 2024].

Boundary injection. Boundary conditions can enter a as input channels, as separate boundary fields, or via decompositions that separate boundary-driven and interior effects. Analytical constrained representations encode boundary satisfaction by construction [Dalton *et al.*, 2024], while boundary-aware decompositions (e.g., Green’s-function-inspired) separate boundary and interior contributions [Wang *et al.*, 2024b].

5.3 Multiscale and high-frequency structure

We focus on *operator-side multiscale interfaces*: how global mixing and basis choices handle (or fail to handle) fine-scale content under resolution and distribution shifts.

Spectral truncation and aliasing. Fourier-style mixing is efficient for smooth fields but truncation can smooth localized high-frequency features, and aliasing can dominate error under resolution shifts after downsampling. Spectral operator variants extend FNO via factorization and constrained mixing to reduce sensitivity [Tran *et al.*, 2023; Xiao *et al.*, 2024].

Locality and scale separation. Wavelet/multiwavelet and multigrid-inspired operators introduce locality and explicit scale separation across scales to better capture sharp features and localized dynamics [Hu *et al.*, 2025a; He *et al.*, 2024a]. On non-Euclidean domains, spectral bases generalize beyond Fourier (e.g., spherical harmonics) [Bonev *et al.*, 2023].

5.4 Latent and attention-based operators

Latent token operators compress inputs into fixed-size latent states for scalable mixing [Alkin *et al.*, 2024]. Attention can be interpreted as learned nonlocal kernels [Yu *et al.*, 2024], while cross-attention enables few-shot operator learning by conditioning on context pairs [Jiao *et al.*, 2025]. State-space models offer linear-complexity alternatives for long sequences [Hu *et al.*, 2025b].

5.5 Physics injection

Physics can be incorporated via architecture, objectives, or correction steps to improve robustness under shift. In this operator-side *augmentation interface*, physics is not the primary supervision (as in PINNs) but an added mechanism for validity and transfer. Importantly, these mechanisms also define the *hybrid (PI-operator)* regime: an operator provides fast, amortized *proposals* (Eq. (7)), while physics constraints provide *correction* and *validation* signals (e.g., residuals and boundary operators, Eq. (5)–(3)), optionally triggering lightweight per-instance refinement. Representative examples include Physics-Informed Neural Operators (PINO) [Li *et al.*, 2024], as well as residual-regularized and refinement-based operator pipelines.

Conservation and symmetry. Concretely, flux-form message passing enforces conservation via antisymmetric exchange across edges [Horie and Mitsume, 2024], while equivariant constructions stabilize transfer by preserving physical symmetries under shifts [Shumaylov *et al.*, 2025].

Boundary handling. Similarly, boundary-aware decompositions separate boundary-driven and interior effects [Wang *et al.*, 2024b]. Analytical constrained representations encode boundary satisfaction by construction [Dalton *et al.*, 2024]. Physics-informed DeepONet variants add boundary channels for improved enforcement [Cho and Son, 2025].

Residual-regularized operators and refinement. Residual penalties and physics-informed refinement connect operator learning to the hybrid zone: the operator proposes, physics corrects and certifies, and refinement can be invoked when diagnostics indicate out-of-contract behavior.

5.6 Data efficiency, robustness, system-level cost

Beyond model architecture, data acquisition and system cost often dominate operator deployment. Active learning selects informative PDE instances to reduce data-generation costs [Musekamp *et al.*, 2025]. End-to-end efficiency is further achieved via data generation recycling [Wang *et al.*, 2024c] and memory-stable mixed-precision implementations [Tu *et al.*, 2024].

5.7 Probabilistic and generative neural operators

Probabilistic and generative operators learn distributions over solution fields for inverse design, partial observations, and long-horizon reliability. Complementing efficiency, probabilistic operators using proper scoring rules yield calibrated function-space uncertainty [Bülte *et al.*, 2025]. Diffusion-based refinement recovers fine-scale structure lost under truncation or coarse representations, supports iterative error correction in long rollouts, and can incorporate physics residuals during generation as guidance [Lippe *et al.*, 2023; Dai *et al.*, 2026b; Oommen *et al.*, 2025]. Generative operator pipelines support inverse design and joint inference from partial observations at test time [Huang *et al.*, 2024].

5.8 Foundation models for PDE operators

At a larger scale, operator foundation models aim to transfer representations across multiple physics domains [Alkin *et al.*, 2024]. Physics-guided fine-tuning in spectral norms improves high-frequency correctness during transfer [Cao *et al.*, 2025], while residual-based objectives allow adaptation via symbolic PDE expressions as losses [Zhang *et al.*, 2025]. To ensure deployment efficiency, spatial modulation mechanisms further reduce fine-tuning costs under resolution and domain shifts [Wang *et al.*, 2025].

6 Applications and Method Choice

This section answers a practical question: *when should one use PINNs, NOs, or hybrids?* Method choice is mainly about supervision availability, amortization payoff, and the required generalization contract under concrete workflow constraints.

Decision axes. We highlight five axes: (i) availability of paired simulation data versus only sparse/indirect observations (Eq. (4)); (ii) forward surrogate vs. inverse/assimilation objectives (unknown θ in Eq. (2)); (iii) number of queries over a fixed PDE family (amortization payoff for Eq. (7)); (iv) geometry/boundary complexity and how much it varies at

deployment (Eq. (3)); (v) deployment requirements (latency, differentiability, robustness, uncertainty).

A recurring pattern. When simulation coverage exists and many-query evaluation is central (design loops, UQ, digital twins), NOs are often preferred as fast surrogates. When data are sparse, unknown parameters must be inferred, or feasibility under constraints is paramount, PINNs provide a strong physics prior. Hybrids increasingly implement proposal–correct–validate with uncertainty-aware diagnostics: an operator proposes (Eq. (7)), physics refines and validates (Eq. (5)).

Domain evidence. *Fluid dynamics* often involves many-query workloads in design and UQ, favoring operator surrogates in turbulence-rich flows; geometry-aware variants enable prediction on complex flow geometries such as hypersonic blunt bodies [Peyvan *et al.*, 2025]. PINNs/hybrids are frequently used for sensor-limited, boundary-sensitive, or regime-shifted inference.

Solid mechanics and materials often split similarly: operators amortize repeated finite-element solves across shape/loading variation, while inverse identification of latent parameters, defects, or damage states favors physics-regularized inference; history dependence motivates hybrids for fracture, damage [He *et al.*, 2024b; Zhang *et al.*, 2022].

Geophysics and climate emphasize non-Euclidean geometry, scale separation, sparse observations, and long rollouts. Geometry-aware operators enable fast ensemble surrogates, while PINNs address sensing-limited inverse problems. Long-horizon stability motivates conservative hybrids in data assimilation [Boney *et al.*, 2023; Chen *et al.*, 2023].

Energy systems foreground deployment constraints: low-latency monitoring/dispatch/control favors surrogates, while inverse state estimation favors physics-regularized inference; constraint-aware hybrids maintain feasibility under shift for real-time operation [Wang *et al.*, 2024a].

Biomedical and molecular applications combine expensive physics with scarce data: operator learning accelerates screening, while physics-informed variants stabilize ill-posed inference in medical imaging under measurement noise [Balcerak *et al.*, 2024]. At the molecular scale, equivariant neural dynamics and plane-wave operators enable high-fidelity modeling of protein-ligand interactions and electron densities [Liu *et al.*, 2025b; Kim and Ahn, 2024].

Engineering design and control embed PDE models inside decision loops: fast differentiable surrogates favor operators; PINNs/hybrids support monitoring and constrained inference under sparse sensing where feasibility matters more than average error [Matada *et al.*, 2025; Zhu *et al.*, 2024].

7 Open Challenges and Future Opportunities

Method choice clarifies what each paradigm can deliver; open challenges clarify what reliability still requires in deployment. As learned PDE solvers enter scientific workflows, predictability under contract shift, physical validity, algorithmic stability, and cost transparency become central concerns.

Controlled extrapolation via explicit contracts. Most evaluations emphasize interpolation, whereas deployment rarely does. Boundary conditions change, geometries vary,

parameters cross regimes, and long rollouts amplify small biases. The most damaging failures are often silent. A central challenge is therefore *controlled extrapolation*: specifying a generalization contract and instrumenting solvers with diagnostics. Proposal–correct–validate workflows are promising, where operators generate fast candidates (Eq. (7)), and physics-based diagnostics determine whether to accept, refine, or reject them (Eq. (5)). A concrete direction is to standardize stress tests along geometry, resolution, and rollout axes, aligned with the declared contract.

Physics-first design as an interface. Residual regularization is insufficient: PDEs encode conservation, symmetry, admissibility, and stability constraints whose violations invalidate downstream decisions even when pointwise errors are small. An emerging opportunity treats physics as an explicit interface, enforcing constraints through architecture, parameterization, and projection, with hard/semi-hard mechanisms ensuring correctness by construction [Chen *et al.*, 2021; Chalapathi *et al.*, 2024; Dai *et al.*, 2026a].

Training as a numerical method. Complementarily, many limitations are algorithmic rather than representational. PINNs involve stiff, multi-objective optimization; neural operators learn nonlocal mappings sensitive to discretization, truncation, and data coverage. Both can fail due to poor conditioning. Accordingly, a more principled view treats training as a numerical process: sampling, resolution, and loss weighting play roles analogous to discretization and step-size selection in classical solvers across workflows.

Scaling laws and end-to-end cost transparency. Amortization motivates operator learning, but reported benefits can be incomplete. True cost is end-to-end: data generation, training, adaptation under shift, and debugging. The field lacks explicit cost models and scaling laws relating error to data volume, resolution, and compute across PDE families. Such transparency clarifies when operator learning outperforms reduced-order modeling, when physics-constrained learning outperforms adjoint-based inversion, and when hybrid approaches are necessary. Progress likely requires reporting a single cost ledger that includes simulation, training, fine-tuning, and deployment latency.

8 Conclusion

This survey presented a unifying view of learning PDE solvers with physics and data. By placing PINN and NO within a shared, workflow-driven design space, we elucidate how differences in what is learned, how physics and data enter supervision, and where computation is amortized give rise to distinct strengths, limitations, and failure modes. The significance of this unification lies in the growing reliance of modern scientific computing on reusable, data-integrated, and decision-driven PDE solvers that operate across heterogeneous workflows. Progress in this area depends on moving towards shared abstractions, contracts, and evaluation signals that connect physics and data in a principled way. We hope that this survey can serve as a common reference that enables community-wide contributions towards the development of reliable learning-based PDE solvers for scientific discovery.

Acknowledgments

This work was supported in part by NSF grants 2126474, 2147195, 2203581, 2239175, 2313174, 2316305, 2425844, 2425845, 2530609, 2530610; NSF NCAR Derecho HPC system; USGS G22AC00266; NASA grants 80NSSC24K1061, 80NSSC25K0013, 80NSSC25K7221; Google AI for Social Good Impact Scholars; and the Alabama Center for the Advancement of Artificial Intelligence Faculty Fellowship.

References

- [Alkin *et al.*, 2024] Benedikt Alkin, Andreas Fürst, et al. Universal physics transformers: A framework for efficiently scaling neural operators. In *NeurIPS*, 2024.
- [Balcerak *et al.*, 2024] Michal Balcerak, Tamaz Amiranashvili, et al. Physics-regularized multi-modal image assimilation for brain tumor localization. In *NeurIPS*, 2024.
- [Bastek *et al.*, 2025] Jan-Hendrik Bastek, WaiChing Sun, et al. Physics-informed diffusion models. In *ICLR*, 2025.
- [Bonev *et al.*, 2023] Boris Bonev, Thorsten Kurth, et al. Spherical Fourier neural operators: Learning stable dynamics on the sphere. In *ICML*, 2023.
- [Bülte *et al.*, 2025] Christopher Bülte, Philipp Scholl, et al. Probabilistic neural operators for functional uncertainty quantification. *TMLR*, 2025.
- [Cao *et al.*, 2025] Shuhao Cao, Francesco Brarda, et al. Spectral-refiner: Accurate fine-tuning of spatiotemporal fourier neural operator for turbulent flows. In *ICLR*, 2025.
- [Chalapathi *et al.*, 2024] Nithin Chalapathi, Yiheng Du, et al. Scaling physics-informed hard constraints with mixture-of-experts. In *ICLR*, 2024.
- [Chen *et al.*, 2021] Yuntian Chen, Dou Huang, et al. Theory-guided hard constraint projection (hcp): A knowledge-based data-driven scientific machine learning method. *J. Comput. Phys.*, 2021.
- [Chen *et al.*, 2023] Shengyu Chen, Yiqun Xie, et al. Physics-guided meta-learning method in baseflow prediction over large regions. In *SDM*, 2023.
- [Chen *et al.*, 2024] Shengyu Chen, Shihang Feng, et al. Hossnet: An efficient physics-guided neural network for simulating micro-crack propagation. *Comput. Mater. Sci.*, 2024.
- [Cho and Son, 2025] Sung Woong Cho and Hwijae Son. Physics-informed deep inverse operator networks for solving PDE inverse problems. In *ICLR*, 2025.
- [Cho *et al.*, 2024] Woojin Cho, Minju Jo, et al. Parameterized physics-informed neural networks for parameterized PDEs. In *ICML*, 2024.
- [Dai *et al.*, 2026a] Yilong Dai, Shengyu Chen, et al. Flow learners for pdes: Toward a physics-to-physics paradigm for scientific computing. *arXiv preprint arXiv:2604.07366*, 2026.
- [Dai *et al.*, 2026b] Yilong Dai, Yiming Sun, et al. Flowrefiner: Flow matching-based iterative refinement for 3d turbulent flow simulation. *arXiv preprint arXiv:2604.17149*, 2026.
- [Dalton *et al.*, 2024] David Dalton, Alan Lazarus, et al. Boundary constrained gaussian processes for robust physics-informed machine learning of linear partial differential equations. *JMLR*, 2024.
- [Dangel *et al.*, 2024] Felix Dangel, Johannes Müller, et al. Kronecker-factored approximate curvature for physics-informed neural networks. In *NeurIPS*, 2024.
- [Daw *et al.*, 2021] Arka Daw, Maruf Maruf, and Anuj Karpatne. Pid-gan: A gan framework based on a physics-informed discriminator for uncertainty quantification with physics. In *KDD*, 2021.
- [Eiras *et al.*, 2024] Francisco Eiras, Adel Bibi, et al. Efficient error certification for physics-informed neural networks. In *ICML*, 2024.
- [Fang *et al.*, 2024] Shikai Fang, Madison Cooley, et al. Solving high frequency and multi-scale PDEs with gaussian processes. In *ICLR*, 2024.
- [Gao *et al.*, 2022] Han Gao, Matthew J. Zahr, et al. Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems. *Comput. Methods Appl. Mech. Eng.*, 2022.
- [Hamelijnck *et al.*, 2024] Oliver Hamelijnck, Arno Solin, et al. Physics-informed variational state-space gaussian processes. In *NeurIPS*, 2024.
- [Hao *et al.*, 2024] Zhongkai Hao, Jiachen Yao, et al. PINNacle: A comprehensive benchmark of physics-informed neural networks for solving PDEs. In *NeurIPS*, 2024.
- [He *et al.*, 2024a] Juncai He, Xinliang Liu, et al. MgNO: Efficient parameterization of linear operators via multigrid. In *ICLR*, 2024.
- [He *et al.*, 2024b] Junyan He, Seid Koric, et al. Geom-deepnet: A point-cloud-based deep operator network for field predictions on 3d parameterized geometries. *Comput. Methods Appl. Mech. Eng.*, 2024.
- [Horie and Mitsume, 2024] Masanobu Horie and Naoto Mitsume. Graph neural pde solvers with conservation and similarity-equivariance. In *ICML*, 2024.
- [Hu *et al.*, 2024] Peiyan Hu, Yue Wang, et al. Better neural PDE solvers through data-free mesh movers. In *ICLR*, 2024.
- [Hu *et al.*, 2025a] Peiyan Hu, Rui Wang, et al. Wavelet diffusion neural operator. In *ICLR*, 2025.
- [Hu *et al.*, 2025b] Zheyuan Hu, Nazanin Ahmadi Daryakenari, et al. State-space models are accurate and efficient neural operators for dynamical systems. *Neural Netw.*, 2025.
- [Huang *et al.*, 2024] Jiahe Huang, Guandao Yang, et al. DiffusionPDE: Generative PDE-solving under partial observation. In *NeurIPS*, 2024.
- [Hwang and Lim, 2024] Youngsik Hwang and Dong-Young Lim. Dual cone gradient descent for training physics-informed neural networks. In *NeurIPS*, 2024.
- [Jia *et al.*, 2023] Xiaowei Jia, Shengyu Chen, et al. Physics-guided graph diffusion network for combining heterogeneous simulated data: An application in predicting stream water temperature. In *SDM*, 2023.
- [Jiao *et al.*, 2025] Anran Jiao, Haiyang He, et al. One-shot learning for solution operators of partial differential equations. *Nat. Commun.*, 2025.
- [Kang *et al.*, 2025] Namgyu Kang, Jaemin Oh, et al. PIG: Physics-informed gaussians as adaptive parametric mesh representations. In *ICLR*, 2025.
- [Kim and Ahn, 2024] Seongsu Kim and Sungsoo Ahn. Gaussian plane-wave neural operator for electron density estimation. In *ICML*, 2024.
- [Lau *et al.*, 2024] Gregory Kang Ruey Lau, Apivich Hemachandra, et al. PINNACLE: PINN adaptive collocation and experimental points selection. In *ICLR*, 2024.

- [Li *et al.*, 2021] Zongyi Li, Nikola Borislavov Kovachki, et al. Fourier neural operator for parametric partial differential equations. In *ICLR*, 2021.
- [Li *et al.*, 2024] Zongyi Li, Hongkai Zheng, et al. Physics-informed neural operator for learning partial differential equations. *ACM/IMS J. Data Sci.*, 2024.
- [Liao *et al.*, 2025] Yunfeng Liao, Jiawen Guan, et al. Curvature-aware graph attention for PDEs on manifolds. In *ICML*, 2025.
- [Lingsch *et al.*, 2024] Levi E. Lingsch, Mike Yan Michelis, et al. Beyond regular grids: Fourier-based neural operators on arbitrary domains. In *ICML*, 2024.
- [Lippe *et al.*, 2023] Phillip Lippe, Bas Veeling, et al. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. In *NeurIPS*, 2023.
- [Liu *et al.*, 2025a] Qiang Liu, Mengyu Chu, et al. ConFIG: Towards conflict-free training of physics informed neural networks. In *ICLR*, 2025.
- [Liu *et al.*, 2025b] Shengchao Liu, Weitao Du, et al. A multi-grained symmetric differential equation model for learning protein-ligand binding dynamics. *Nat. Commun.*, 2025.
- [Lu *et al.*, 2021] Lu Lu, Pengzhan Jin, et al. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 2021.
- [Matada *et al.*, 2025] Sharath Matada, Luke Bhan, et al. Generalizable motion planning via operator learning. In *ICLR*, 2025.
- [Musekamp *et al.*, 2025] Daniel Musekamp, Marimuthu Kalimuthu, et al. Active learning for neural PDE solvers. In *ICLR*, 2025.
- [Oommen *et al.*, 2025] Vivek Oommen, Aniruddha Bora, et al. Integrating neural operators with diffusion models improves spectral representation in turbulence modelling. *Proc. R. Soc. A*, 2025.
- [Penwarden *et al.*, 2023] Michael Penwarden, Shandian Zhe, et al. A metalearning approach for physics-informed neural networks (pinns): Application to parameterized pdes. *J. Comput. Phys.*, 2023.
- [Peyvan *et al.*, 2025] Ahmad Peyvan, Varun Kumar, et al. Fusion-deepnet: A data-efficient neural operator for geometry-dependent hypersonic and supersonic flows. *J. Comput. Phys.*, 2025.
- [Psaros *et al.*, 2022] Apostolos F Psaros, Kenji Kawaguchi, et al. Meta-learning pinn loss functions. *J. Comput. Phys.*, 2022.
- [Raissi *et al.*, 2019] Maziar Raissi, Paris Perdikaris, et al. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 2019.
- [Rathore *et al.*, 2024] Pratik Rathore, Weimu Lei, et al. Challenges in training PINNs: A loss landscape perspective. In *ICML*, 2024.
- [Sahli Costabal *et al.*, 2024] Francisco Sahli Costabal, Simone Pezzuto, et al. Δ -pinns: Physics-informed neural networks on complex geometries. *Eng. Appl. Artif. Intell.*, 2024.
- [Saleh *et al.*, 2024] Ehsan Saleh, Saba Ghaffari, et al. Learning from integral losses in physics informed neural networks. In *ICML*, 2024.
- [Sharma and Shankar, 2022] Ramansh Sharma and Varun Shankar. Accelerated training of physics-informed neural networks (pinns) using meshless discretizations. In *NeurIPS*, 2022.
- [Shu *et al.*, 2023] Dule Shu, Zijie Li, et al. A physics-informed diffusion model for high-fidelity flow field reconstruction. *J. Comput. Phys.*, 2023.
- [Shukla *et al.*, 2021] Khemraj Shukla, Ameya D. Jagtap, et al. Parallel physics-informed neural networks via domain decomposition. *J. Comput. Phys.*, 2021.
- [Shumaylov *et al.*, 2025] Zakhar Shumaylov, Peter Zaika, et al. Lie algebra canonicalization: Equivariant neural operators under arbitrary lie groups. In *ICLR*, 2025.
- [Tran *et al.*, 2023] Alasdair Tran, Alexander Mathews, et al. Factorized fourier neural operators. In *ICLR*, 2023.
- [Tu *et al.*, 2024] Renbo Tu, Colin White, et al. Guaranteed approximation bounds for mixed-precision neural operators. In *ICLR*, 2024.
- [Wandel *et al.*, 2025] Nils Wandel, Stefan Schulz, et al. Metamizer: A versatile neural optimizer for fast and accurate physics simulations. In *ICLR*, 2025.
- [Wang *et al.*, 2024a] Fujin Wang, Zhi Zhai, et al. Physics-informed neural network for lithium-ion battery degradation stable modeling and prognosis. *Nat. Commun.*, 2024.
- [Wang *et al.*, 2024b] Haixin Wang, Jiaxin LI, et al. BENO: Boundary-embedded neural operators for elliptic PDEs. In *ICLR*, 2024.
- [Wang *et al.*, 2024c] Hong Wang, Zhongkai Hao, et al. Accelerating data generation for neural operators via krylov subspace recycling. In *ICLR*, 2024.
- [Wang *et al.*, 2025] Honghui Wang, Shiji Song, et al. Gridmix: Exploring spatial modulation for neural fields in PDE modeling. In *ICLR*, 2025.
- [Wu *et al.*, 2023] Chenxi Wu, Min Zhu, et al. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.*, 2023.
- [Xiao *et al.*, 2024] Zipeng Xiao, Siqi Kou, et al. Amortized fourier neural operators. In *NeurIPS*, 2024.
- [Xu *et al.*, 2024] Minkai Xu, Jiaqi Han, et al. Equivariant graph neural operator for modeling 3d dynamics. In *ICML*, 2024.
- [Yang *et al.*, 2021] Liu Yang, Xuhui Meng, et al. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *J. Comput. Phys.*, 2021.
- [Yu *et al.*, 2024] Yue Yu, Ning Liu, et al. Nonlocal attention operator: Materializing hidden knowledge towards interpretable physics discovery. In *NeurIPS*, 2024.
- [Zeng *et al.*, 2025] Bocheng Zeng, Qi Wang, et al. PhyMPGN: Physics-encoded message passing graph network for spatiotemporal PDE systems. In *ICLR*, 2025.
- [Zhang *et al.*, 2022] Enrui Zhang, Ming Dao, et al. Analyses of internal structures and defects in materials using physics-informed neural networks. *Sci. Adv.*, 2022.
- [Zhang *et al.*, 2025] Zecheng Zhang, Christian Moya, et al. Deepnet as a multi-operator extrapolation model: Distributed pre-training with physics-informed fine-tuning. *J. Comput. Phys.*, 2025.
- [Zhu *et al.*, 2024] Hanqing Zhu, Wenyan Cong, et al. Pace: Pacing operator learning to accurate optical field simulation for complicated photonic devices. In *NeurIPS*, 2024.