

From Neural Networks to Adaptive Physics-Informed Learning: A Review of Sampling Strategies, Spectral Bias, and Thermal Applications

Rishika Roy¹, Kanwal Preet Singh Attwal¹

¹*Department of Computer Science & Engineering, Punjabi University, Patiala, Punjab, India)*

Abstract:

Physics-Informed neural networks (PINNs) are one of the most active frontiers in scientific machine learning, used to solve the forward and inverse problems governed by partial differential equations. It incorporates governing physical laws and equations into neural network training i.e., loss function. This review discusses twenty-four primary sources, beginning with the biological inspiration and architecture of artificial neural networks, proceeds through the formal construction of PINNs, and then concentrates on the two forces that most strongly govern their training behavior – the placement of collocation points and spectral bias. Different sampling strategies are then reviewed, including residual-based adaptive refinement (RAR), residual-based adaptive refinement with distribution (RAR-D), residual-based adaptive distribution (RAD), residual-based adaptive node generation (RANG), the point-adaptive collocation method PACMANN, residual and gradient adaptive sampling strategies, reinforcement learning-driven sampling, importance sampling, Hessian-based provably accurate sampling, and imbalanced-learning-inspired approaches such as RSmote. We then discuss the neural tangent kernel (NTK), which explains why PINNs sometimes fail to train and why adaptive weighting and self-adaptive mechanisms restore balance among competing loss terms. Finally, we connect this theory to practice through gradient-enhanced PINNs, DeepXDE library and engineering applications in heat transfer and thermal nanoimprint lithography, before identifying open challenges and research directions.

Keywords: *Physics-informed neural networks; PINNs; adaptive sampling; collocation points; spectral bias; neural tangent kernel; scientific machine learning; heat transfer.*

Date of Submission: 01-10-2026

Date of Acceptance: 10-10-2026

I. INTRODUCTION

For several decades, partial differential equations have primarily been solved using established numerical techniques such as finite difference, finite element, finite volume, and spectral methods. These methods are mature, well-understood, and supported by rigorous error analysis, but they share a common structural cost: the domain must be meshed or gridded, and the accuracy of the solution largely depends on the level of detail used in the discretization. As the dimensionality of a problem grows, the number of grid points required to maintain accuracy grows exponentially, which can significantly increase computational cost and make the problem difficult to solve efficiently. In response to these developments, a new group of approaches known as scientific machine learning has emerged and gained increasing attention. One of the main approaches in scientific machine learning is the physics-informed neural network (PINN), which uses a collection of scattered points within the problem domain instead of relying on a conventional mesh. The governing physical equations, boundary conditions and initial conditions are incorporated into a differentiable loss function, which is then minimized during neural network training [2].

PINNs are attractive not only because of their practical benefits but also because of their underlying approach to solving physical problems. Since the governing physical laws are directly incorporated into the loss function, PINNs can be applied to both forward and inverse problems. In forward problems, the boundary and initial conditions are known and the objective is to determine the unknown physical field. In inverse problems, sparse measurements are available and unknown physical parameters must be inferred [2, 4]. PINNs do not necessarily require a training dataset generated by a traditional numerical solver. Instead, the governing differential equations provide the required physical information, which is evaluated at selected collocation points using automatic differentiation. This framework allows PINNs to address both forward and inverse problems and also makes it possible to incorporate measurement data, including data affected by noise, into the training process. As a result, PINNs have been explored in several engineering and scientific fields, including fluid mechanics, solid mechanics, electromagnetics, and heat transfer.

Yet the very flexibility that makes PINNs appealing also presents certain challenges that can affect their training and overall performance. The PDE residual is not evaluated everywhere; it is sampled at a finite set of points, and the neural network is trained to drive that sampled residual toward zero. If the points are poorly placed – if they cluster in regions where the solution is smooth and neglect the boundary layers, shocks, or sharp gradients where the residual is the largest – the network may converge to a solution that appears acceptable on a training set while being badly wrong elsewhere. The distribution of collocation points is therefore not a minor implementation detail but a first-order determinant of accuracy and computational cost [5, 15, 16]. A parallel difficulty arises from the optimization dynamics themselves. Neural networks trained by gradient descent exhibit spectral bias, learning low-frequency structure long before they capture high-frequency detail, and the multiple loss terms of a PINN frequently produce unbalanced gradients that stall training altogether [6, 8, 9].

This review is organized around those two governing forces. It begins, in Section 2, with the fundamentals of artificial neural networks, recovering the biological inspiration and the learning-by-example principle that underpin the entire field [1]. Section 3 develops the formal construction of PINNs and distinguishes forward from inverse formulations [2, 3, 4]. Section 4 explains why collocation-point sampling matters and surveys the non-adaptive baselines – uniform grids, random sampling, Latin hypercube, Halton, Hammersley, and Sobol sequences [5]. Section 5 is the analytical heart of the review, presenting the major adaptive sampling families in turn. Section 6 turns to the theoretical explanations supplied by spectral bias and the neural tangent kernel [6, 7, 9, 12], and Section 7 examines gradient-enhanced PINNs and self-adaptive weighting as complementary remedies [13, 17]. Section 8 discusses software infrastructure [4], Section 9 surveys applications in heat transfer and thermal engineering [10, 11, 18], and Sections 10 and 11 set out challenges, future directions, and conclusions.

II. FOUNDATIONS OF ARTIFICIAL NEURAL NETWORKS

Any serious treatment of physics-informed learning must begin with the building block on which it rests. An artificial neural network is, in the classic formulation, an information-processing paradigm inspired by the way biological nervous systems such as the brain process information [1]. The key element of this paradigm is the structure of the processing system itself: it is composed of a large number of highly interconnected processing elements, or neurons, working in unison to solve specific problems. The individual neuron is a simple device, but the collective behavior of many neurons arranged in layers is capable of representing extraordinarily complex functions. This division between the simplicity of the unit and the richness of the ensemble is the central insight of connectionism, and it is what allows a neural network to act as a flexible function approximator rather than a rigid rule-based system.

The defining characteristic of the artificial neural network, as Wankar and colleagues emphasize, is that it learns by example [1]. Rather than being programmed with explicit rules, a network is configured for a specific application – pattern recognition, data classification, regression, or control – through a learning process in which its internal parameters are adjusted in response to examples. This learning-by-example principle is precisely what carries over into the physics-informed setting. There, the "examples" are not merely labelled data points but the residual of a differential equation evaluated at collocation points, together with boundary and initial conditions. The network is still learning by example; the examples have simply been enriched with the structure of physical law.

Architecturally, a feedforward network organizes neurons in an input layer, one or more hidden layers, and an output layer. Each connection carries a weight, each neuron applies an activation function to a weighted sum of its inputs, and the composition of these operations yields the network's output. The activation function introduces the nonlinearity that gives the network the ability to learn non-linear relationships; without it, a deep network would collapse into a single linear transformation. Common choices include the logistic sigmoid, the hyperbolic tangent, and, in modern practice, the rectified linear unit and its smooth variants. The rectified linear unit deserves particular attention in the present context because its piecewise-linear, non-smooth character is intimately connected to the spectral bias discussed in Section 6 [6].

Training proceeds by backpropagation, in which the gradient of a loss function with respect to every network weight is computed by the chain rule and then the weights are updated. The loss measures the discrepancy between the network's output and a target; in supervised learning the target is a label, whereas in the physics-informed setting the target is encoded in a differential equation that the output must satisfy. The universal approximation property guarantees that a sufficiently wide network can approximate any continuous function on a compact domain to arbitrary accuracy, which is the theoretical license for using neural networks as surrogates for the solution fields of PDEs. What the universal approximation theorem does not guarantee, however, is that gradient descent will find the good approximation, nor that it will find it efficiently. Those questions belong to optimization theory and to the neural tangent kernel, and they are the reason adaptive sampling and spectral-bias mitigation have become such active areas of research.

Two characteristics of neural networks are particularly relevant to this discussion. First, their differentiable structure allows automatic differentiation to be used for evaluating the derivatives required in PDE-based loss functions. Second, neural networks are commonly trained using stochastic optimization with mini-batches, so the loss calculated during each training step is based on only a subset of the available samples. In PINNs, these samples correspond to collocation points used to evaluate the governing equations. Consequently, the selection and distribution of collocation points can influence how accurately the loss represents the physical behavior across the domain. This connection between neural network training and numerical sampling provides an important basis for understanding the development of adaptive collocation-point sampling methods.

III. PHYSICS-INFORMED NEURAL NETWORKS: THEORY AND FORMULATION

The physics-informed neural network, introduced by Raissi, Perdikaris, and Karniadakis, embeds a partial differential equation into the loss function of a neural network using automatic differentiation [2]. The construction is elegant in its simplicity. Consider a general nonlinear PDE of the form:

$$u_t + N[u; \lambda] = 0, \quad x \in \Omega, t \in [0, T],$$

where u is the unknown solution field, N is a nonlinear differential operator parameterized by λ , and subscripts denote differentiation. A neural network $\hat{u}(x, t; \theta)$ with parameters θ is trained to approximate the true solution. The residual of the PDE, defined as

$$r(x, t; \theta) = \hat{u}_t(x, t; \theta) + N[\hat{u}(x, t; \theta); \lambda],$$

is evaluated at a set of collocation points by differentiating the network output with respect to its inputs – a computation made feasible by automatic differentiation, which produces the exact derivatives of the network with respect to spatial and temporal coordinates without any finite-difference approximation. The total loss combines the PDE residual with terms enforcing initial and boundary conditions, and the network is trained to minimize the weighted sum.

This formulation offers two important advantages. First, it can be applied to a wide range of differential equations without being restricted to a particular equation type. Raissi and colleagues demonstrated that the same framework applies to integral-differential equations, fractional PDEs, and stochastic PDEs, because the only requirement is that the residual be differentiable with respect to the network parameters [2]. Second, it solves inverse problems as easily as forward problems. In a forward problem the coefficients λ are known and the field is sought; in an inverse problem the field is partially observed and the coefficients are sought. Because λ can be treated as trainable variables, the two cases differ only in which quantities are optimized, not in the structure of the algorithm. This unification is one of the principal reasons PINNs have been adopted so widely, and it is a recurring theme in the comprehensive review by Cuomo and colleagues, who characterize PINNs as a multi-task learning framework in which a network must simultaneously fit observed data and reduce a PDE residual [3].

Cuomo and colleagues situate the vanilla PINN within a broader family of collocation-based physics-informed architectures. They note that the field has spawned numerous variants, including physics-constrained neural networks, variational hp-VPINNs, and other collocation-based formulations, these approaches differ in their flexibility, computational requirements, and the way physical constraints are incorporated into learning process [3]. PINNs have been applied to a broad range of mathematical problems, including ordinary differential equations, partial differential equations, fractional equations, integral-differential equations and stochastic PDEs. At the same time, the review identifies several challenges associated with PINNs, such as high training costs, sensitivity to the weighting of different loss components, and difficulties in obtaining accurate solutions in some problems. The distribution of collocation points is another important factor, as an unsuitable sampling strategy can leave certain regions of the domain inadequately represented during training.

Lu et al. provide a practical perspective on PINNs through the development of the DeepXDE library [4]. They present PINNs as a simple and flexible approach for solving problems governed by differential equations and compare their formulation with conventional numerical methods such as the finite element method. In FEM, the computational domain is divided into a mesh and the governing equations are converted into a system of algebraic equations. In contrast, a PINN represents the solution using a neural network and determines its parameters by minimizing a loss function through non-convex optimization. These differences also lead to differences in the theoretical analysis and convergence properties of the two approaches. Methods such as the neural tangent kernel have therefore been used to study the behavior of neural-network-based solvers. DeepXDE further simplifies PINN implementation by providing support for different geometries, collocation-point sampling, automatic differentiation, and model training while keeping the implementation relatively close to the underlying mathematical formulation.

Collocation points play an important role in the training of a PINN because the PDE residual is evaluated only at selected points within the computational domain. The resulting PDE loss provides a discrete

approximation of the residual over the domain rather than evaluating it continuously at every location. Therefore, the quality of this approximation depends on how the collocation points are selected and distributed. A sufficiently large and well-distributed set of points can represent the behavior of the solution more effectively, whereas sparse or uneven sampling may leave important regions inadequately represented. In such cases, the network may achieve a low residual at the sampled points while still having relatively large errors in regions that were not sufficiently sampled. This dependence on collocation-point distribution provides the motivation for the adaptive sampling methods discussed in Section 5.

IV. THE CENTRAL ROLE OF COLLOCATION-POINT SAMPLING

The most comprehensive early treatment of collocation-point sampling is the study by Wu et al., who present a systematic comparison of non-adaptive uniform sampling and adaptive non-uniform sampling for PINNs [5]. They begin from the observation that the distribution of residual points is highly important to the performance of PINNs, yet that the existing literature had relied mainly on a few simple sampling methods. To remedy this, they examine six uniform sampling strategies: the equispaced uniform grid, uniformly random sampling, Latin hypercube sampling, the Halton sequence, the Hammersley sequence, and the Sobol sequence. They additionally consider a resampling strategy for uniform sampling, in which the collocation points are redrawn at intervals during training. The differences between these sampling approaches can be understood in terms of how effectively they distribute points throughout the computational domain. An equispaced uniform grid provides a simple and structured arrangement of points, but it is vulnerable to the curse of dimensionality problem. Uniform random sampling avoids a fixed grid structure and can provide a straightforward representation of the domain, although random points may sometimes cluster in certain regions and leave other areas less populated. Quasi-random sequences, such as Halton, Hammersley, and Sobol sequences, are designed to fill space more evenly than pure randomness, and reduce such clustering. Latin hypercube sampling uses a stratified strategy to improve the coverage of each coordinate across the domain, particularly for moderate-dimensional problems. Wu et al. reported that the choice of sampling strategy can influence both the accuracy and training efficiency of PINNs, with the effectiveness of each method depending on the problem being considered [5].

Wu et al. also proposed two adaptive sampling approaches that modify the distribution of collocation points according to the behavior of the PDE residual. The first approach, residual-based adaptive distribution (RAD), increases the likelihood of selecting points from regions where the residual is relatively high. The second approach, residual-based adaptive refinement with distribution (RAR-D), adds new collocation points while also considering the residual-based sampling distribution. In their study, Wu et al. evaluated ten sampling strategies using four forward problems and two inverse problems, providing a detailed comparison of different approaches to collocation-point selection [5]. Their results showed that residual-based adaptive strategies can improve PINN performance compared with uniform sampling in several of the problems considered, highlighting the importance of selecting collocation points according to the characteristics of the solution.

The intuition behind residual-driven sampling is straightforward, but the implementation is not, because the residual field changes as training progresses. A point that is difficult early in training may become easy later, and vice versa. Adaptive methods must therefore decide not only where to sample but when to resample, how many points to add, and how to avoid the accumulation of redundant points that inflate memory and computational cost. These practical concerns motivate the diversity of approaches reviewed in the next section, each of which strikes a different balance between accuracy, efficiency, and scalability to high dimensions.

V. ADAPTIVE SAMPLING METHODS FOR PINNs

Adaptive Sampling methods can be broadly grouped according to how they select and update collocation points during PINN training. This section explains these sampling methods, beginning with basic refinement-based methods and then moving towards more advanced strategies based on reinforcement learning, importance sampling, imbalanced sampling, and many more.

1. Residual-based adaptive refinement and distribution

The residual-based adaptive refinement (RAR) method was introduced alongside DeepXDE and the modern PINN framework [2, 4]. The basic procedure is straightforward. A small set of collocation points is first used to train the network for a few number of epochs. The PDE residual is then evaluated at a larger set of candidate points, and points with high residual values are added to the training set. By repeating this cycle, RAR progressively concentrates points in the regions where the network struggles most. So, RAR can improve the representation of difficult regions without requiring a dense set of collocation points from the beginning. RAR is attractive for its simplicity and has become a standard baseline against which newer methods are measured.

Wu et al. extended this idea with residual-based adaptive distribution (RAD) and residual-based adaptive refinement with distribution (RAR-D) [5]. RAD assigns a higher sampling probability to regions with larger

residuals rather than selecting only the points with the highest residual values. This allows new points to be sampled across regions where the residual is high. RAR-D combines the refinement idea of RAR with the residual-based sampling distribution used in RAD. Thus, RAR adds points based on high residual values, RAD changes the sampling distribution according to the residual, and RAR-D combines both techniques.

2. *Residual-based adaptive node generation (RANG)*

Peng et al. proposed the residual-based adaptive node generation (RANG) method to combine residual-based sampling with more uniform point distribution provided by quasi-uniform sampling [24]. The main idea is to avoid two common issues: residual-based methods may place too many points in a few difficult regions, while quasi-uniform sampling may allocate unnecessary points to regions where the PINN already performs well. RANG uses a variable-density node generation approach, originally developed for radial basis function finite difference (RBF-FD) methods, to adjust the density of collocation points according to the PDE residual. Thus, regions with larger residuals receive more points, while regions with smaller residuals are represented with fewer points. RANG also uses a memory mechanism to retain information from previous sampling steps and reduce repeated sampling of the same regions. In this way, the method aims to maintain higher point density in difficult regions while preserving relatively uniform coverage across the rest of the domain.

3. *PACMANN: Point-adaptive collocation via gradient motion*

A recurring limitation of refinement-based methods is their poor scaling to high-dimensional problems. Adding points, resampling, and retraining become expensive as the dimension grows, and the number of points needed to cover a high-dimensional domain explodes. The point adaptive collocation method for artificial neural networks, or PACMANN, addresses this limitation directly [16]. Rather than adding new points, PACMANN incrementally moves existing collocation points toward regions of higher residual using gradient-based optimization. Because the number of points remains fixed, the memory and computational cost do not grow with the number of refinement rounds, and the method scales far better to high dimensions than its refinement-based counterparts. The authors report that PACMANN outperforms existing adaptive methods for high-dimensional problems, precisely the regime in which classical approaches degrade. The idea of moving points rather than adding them is conceptually elegant and represents a genuine shift in the adaptive-sampling design space.

4. *Residual-and-gradient adaptive sampling for sharp solutions*

For problems whose solutions contain sharp features – steep gradients, moving fronts, or near-discontinuities – the residual alone may be an insufficient guide, because the residual can be small even where the solution changes rapidly. Mao et al. propose adaptive sampling methods that incorporate both the residual and the gradient of the solution [19]. Their first method, denoted ASM I, is residual-only: the network is trained with a small number of residual points, the computational domain is divided into sub-domains, and new points are added in the sub-domain with the largest mean absolute residual, with the points of largest absolute residual within that sub-domain selected for addition. Their second method, ASM II, augments this criterion with the gradient of the solution, so that points are placed not only where the residual is large but also where the solution varies steeply. The authors report that ASM II is superior for capturing sharp solutions, confirming that the gradient carries information that the residual alone does not. This finding anticipates the gradient-enhanced loss functions discussed in Section 7.

5. *Reinforcement learning-driven sampling (RL-PINNs)*

Conventional adaptive sampling methods share a structural inefficiency: they require multiple rounds of sampling and repeated retraining, and they accumulate redundant points and costly gradient computations, particularly in high dimensional or high-order-derivative settings. RL-PINNs, proposed by Song, reframes adaptive sampling as a sequential decision-making problem and solves it with reinforcement learning [21]. The method formulates sampling as a Markov decision process in which an agent observes the current training state and selects the sampling action according to the learned policy. The selected action determines how the collocation points are updated, and the resulting change in the training process is used to provide a reward. Through this process, the model learns which sampling actions are more effective under different training conditions rather than relying entirely on a fixed sampling rule. Thus, RL-PINNs provide different perspective on adaptive sampling.

6. *Importance sampling*

Another approach to collocation-point selection considers the problem from the perspective of variance reduction in stochastic optimization. Nabian et al. note that PINN training involves non-convex optimization, where stochastic gradient-based methods estimate gradient of the loss using batches of randomly selected collocation points [14]. Under uniform sampling, each point has the same probability of being selected. Although this provides an unbiased estimate of the gradient, the resulting estimate may have relatively high

variance, which can affect the efficiency of the optimization process. To address this issue, Nabian et al. proposed an importance-sampling approach in which points are assigned different sampling probabilities according to their importance. Points that contribute more significantly to the gradient estimate are therefore given a higher probability of being selected. In the context of PINNs, information such as the PDE residual can be used to characterize the importance of collocation points. This approach provides a statistical perspective on collocation-point selection by showing how a suitable sampling distribution can reduce the variance of stochastic gradient estimates and potentially improve training efficiency.

7. *Provably accurate hessian-based sampling*

Most adaptive-sampling methods are justified empirically rather than analytically. Caradot and colleagues take a different tack, introducing a provably accurate sampling method grounded in numerical quadrature theory [15]. They leverage a new quadrature method for approximating definite integrals and derive a sampling scheme based on the Hessian of the PDE residuals. Because the Hessian captures the local curvature of the residual field, it provides a principled measure of where the quadrature error is largest and where additional points are most needed. The resulting method carries accuracy guarantees that heuristic approaches lack, and comparative experiments on a set of one- and two-dimensional PDEs demonstrate its benefits. The work is significant because it begins to close the theoretical gap between the empirical practice of adaptive sampling and the rigorous error analysis that has long characterized classical numerical methods.

8. *RSmote: Imbalanced-learning-inspired sampling*

The RSmote method, introduced by Luo and colleagues, draws an unexpected but fruitful analogy between adaptive sampling and imbalanced classification [22]. In an imbalanced dataset, one class is vastly under-represented, and techniques such as the synthetic minority oversampling technique (SMOTE) generate synthetic examples of the minority class to restore balance. In a PINN, the regions of high residual are analogous to the minority class: they are under-represented in a uniform sample yet are precisely where the network needs the most attention. RSmote applies oversampling techniques from imbalanced learning to refine the sampling process, targeting high-residual regions and synthesizing additional points there. The authors provide a theoretical analysis supporting the method's efficiency in managing computational resources, and they benchmark it against RAD across a variety of dimensions and differential equations. The results show that RSmote matches or exceeds the accuracy of RAD while significantly reducing memory usage, an advantage that is especially pronounced in high-dimensional scenarios.

9. *h-adaptive collocation*

Trynda et al. proposed an h-adaptive collocation method for PINNs based on the idea of adaptive mesh refinement used in traditional numerical methods [23]. In adaptive mesh refinement, the parameter h generally represents the local size or spacing of the mesh elements, and h-adaptivity involves reducing this size in regions where greater resolution is required. In the PINN setting, this idea is applied to collocation points by increasing the point density in regions where the solution is not adequately represented, while using fewer points in regions where the solution can already be resolved sufficiently. In this way, the distribution of collocation points is adjusted according to the local characteristics of the problem. The method provides a connection between traditional adaptive mesh-refinement techniques and adaptive collocation-point strategies used in PINNs, showing how ideas from classical numerical methods can be adapted to mesh-free neural-network-based approaches.

10. *Hybrid residual-driven sampling*

A further refinement is offered by Zhao and colleagues, who propose a hybrid residual-driven adaptive sampling approach, sometimes denoted gpKP, that combines multiple sampling criteria to exploit their complementary strengths [20]. The hybrid philosophy recognizes that no single signal – residual, gradient, or curvature – is sufficient in all regimes, and that a method which adaptively blends them can be more robust than any one of them alone. This mirrors a broader trend in the field toward hybrid and ensemble strategies that hedge against the failure modes of individual heuristics.

VI. SPECTRAL BIAS AND THE NEURAL TANGENT KERNEL

Adaptive sampling addresses where the network is trained; a complementary body of theory addresses how it learns. The two are deeply connected, because the difficulty of learning certain features is precisely what makes certain regions hard and drives the residual to remain large there. This section reviews the spectral bias of neural networks, the Fourier-feature remedy, and the neural tangent kernel theory that explains both the successes and the failures of PINNs.

1. *The spectral bias of neural networks*

Rahaman and colleagues established that neural networks, despite being highly expressive functions capable of fitting even random input-output mappings with perfect accuracy, exhibit a systematic preference for low-frequency structure [6]. They termed this phenomenon spectral bias: during training, a network learns the low-frequency components of a target function first and only gradually, if at all, captures the high-frequency components. The bias is especially pronounced for networks built from the rectified linear unit, whose piecewise-linear structure is inherently inclined toward smooth, low-frequency approximations. The practical consequence for PINNs is severe. Many physically interesting solutions contain sharp gradients, boundary layers, or oscillatory structure – precisely the high-frequency content that spectral bias suppresses. A network that has learned only the low-frequency skeleton of such a solution will exhibit a large residual near the sharp features, which is exactly the behaviour that motivates residual-based adaptive sampling.

2. *Fourier features as a remedy*

Tancik and colleagues proposed a direct and elegant remedy: pass the input coordinates through a Fourier feature mapping before feeding them to the network [7]. They showed that a simple Fourier feature mapping enables a multilayer perceptron to learn high-frequency functions in low-dimensional domains, whereas a standard multilayer perceptron fails to do so both in theory and in practice. Using tools from the neural tangent kernel literature, they demonstrated that the Fourier mapping transforms the effective NTK into a stationary kernel with a tunable bandwidth, allowing the frequency content that the network can learn to be controlled explicitly. They further suggested an approach for selecting problem-specific Fourier features. The relevance to PINNs is immediate: Fourier features can be used to inject prior knowledge about the frequency content of the solution, mitigating spectral bias at its source rather than compensating for it after the fact through sampling. In effect, adaptive sampling and Fourier features attack the same problem – the network's difficulty in representing high-frequency structure – from two different directions, one by concentrating training effort and the other by enriching the representation.

3. *The neural tangent kernel*

The theoretical framework that unifies these observations is the neural tangent kernel, developed by Jacot and colleagues and extended by Du and colleagues. Du and colleagues proved that at initialization, artificial neural networks are equivalent to Gaussian processes in the infinite-width limit, connecting them to kernel methods, and that the evolution of a network during training can also be described by a kernel [12]. During gradient descent on the parameters, the network function evolves according to a kernel that captures the behaviour of the network in the infinite-width limit. Crucially, Du and colleagues showed that gradient descent finds global minima of deep neural networks under appropriate conditions, providing a theoretical guarantee that had long been elusive. The NTK thus supplies a linearized description of a highly nonlinear process, and it is this description that makes the training dynamics of PINNs analytically tractable.

4. *Why PINNs fail: An NTK perspective*

Wang, Yu, and Perdikaris applied the NTK to PINNs directly and derived the NTK of a PINN, proving that under appropriate conditions it converges to a deterministic kernel that remains constant during training in the infinite-width limit [9]. This result is powerful because it reduces the analysis of PINN training to the spectral properties of a fixed kernel. The convergence rate of each component of the solution is governed by the corresponding eigenvalue of the NTK: components associated with large eigenvalues are learned quickly, while those associated with small eigenvalues are learned slowly or not at all. When the eigenvalues of the PINN's NTK are highly imbalanced – which happens when the PDE residual term and the boundary-condition term produce kernels with very different spectra – training stalls, and the network fails to satisfy all the constraints simultaneously. This eigenvalue imbalance is the mechanistic explanation of the training failures that practitioners had long observed empirically. Wang and colleagues proposed eigenvalue-based weighting of the loss terms as a remedy, and the analysis has since become a standard lens for diagnosing and repairing PINN training.

5. *Gradient flow pathologies and numerical stiffness*

A related diagnosis was offered by Wang, Teng, and Perdikaris, who identified a fundamental mode of failure related to numerical stiffness [8]. They observed that the soft penalties used to impose physical constraints act as domain-specific regularizers of the empirical risk, and that the resulting optimization problem can be stiff, leading to unbalanced back-propagated gradients during training. When the gradients contributed by different loss terms differ by orders of magnitude, the optimizer effectively ignores the smaller ones, and the corresponding constraints are violated. Their remedy was a learning rate annealing algorithm that dynamically balances the gradient magnitudes of the competing loss terms, together with a more principled weighting derived from the NTK. The work complements the eigenvalue analysis of Wang, Yu, and Perdikaris: both

identify imbalance among loss terms as the root cause of failure, and both propose dynamic reweighting as the cure.

6. Self-adaptive PINNs and soft attention

McClenny and Braga-Neto generalize the reweighting idea to the level of individual training points [13]. In self-adaptive PINNs (SA-PINNs), the adaptation weights are fully trainable and are applied to each training point individually, so that the network learns autonomously which regions of the solution are difficult and is forced to focus on them. The self-adaptation weights specify a soft multiplicative soft attention mask, reminiscent of the attention mechanisms used in computer vision, and the network is trained to simultaneously minimize the losses and maximize the weights. This minimax structure ensures that points with large losses receive large weights, since increasing a weight reduces the corresponding loss term while the maximization pushes the weights upward. McClenny and Braga-Neto further show how to build a continuous map of self-adaptive weights using Gaussian process regression, which allows stochastic gradient descent to be used in problems where conventional gradient descent is insufficient. Finally, they derive the NTK matrix for SA-PINNs and use it to show that the self-adaptive weights produce a smooth equalization of the eigenvalues of the NTK matrix corresponding to the different loss terms. This is a beautiful result: it explains why self-adaptation works in precisely the terms that Wang and colleagues used to explain why PINNs fail. SA-PINNs succeeded in numerical experiments on several linear and nonlinear benchmark problems, outperforming other state-of-the-art PINN algorithms in L2 error while using fewer training epochs.

VII. GRADIENT-ENHANCED PINNs AND THE ROLE OF DERIVATIVE INFORMATION

The observation that the residual alone can be an insufficient guide, made by Mao and colleagues in the context of sampling [19], has a parallel in the design of the loss function itself. Yu and colleagues propose gradient-enhanced physics-informed neural networks (gPINNs), which leverage gradient information of the PDE residual and embed that gradient into the loss function [17]. The motivation is that a residual which is small at the collocation points does not guarantee that the solution is accurate between them; the residual may be small at the sampled points yet vary rapidly nearby. By penalizing the gradient of the residual, gPINNs enforce that the residual is not merely small but also smooth, which is a stronger and more robust condition. The authors tested gPINNs extensively and demonstrated their effectiveness in both forward and inverse PDE problems, finding that gPINN achieves better accuracy than the standard PINN with fewer training points. They further showed that gPINN can be combined with residual-based adaptive refinement to handle PDEs with steep gradients, where the synergy between a gradient-aware loss and a residual-aware sampling scheme is particularly valuable. The gradient-enhanced approach connects several threads of this review. It echoes the gradient-based sampling of ASM II, the curvature-based sampling of the Hessian method, and the NTK analysis, in which the kernel is constructed from the derivatives of the network output. In all these cases the underlying insight is the same: first-order residual information is a partial view, and incorporating higher-order derivative information yields a more faithful and more stable description of how well the network satisfies the physics. The cost is additional computation, since the gradient of the residual requires higher-order automatic differentiation, but the accuracy gains can justify the expense, particularly for problems with sharp features.

VIII. SOFTWARE INFRASTRUCTURE: DEEPXDE AND THE SCIENTIFIC MACHINE LEARNING ECOSYSTEM

The theoretical and algorithmic advances described above would be of limited practical value without accessible software. DeepXDE, presented by Lu and colleagues, was designed to serve both as an educational tool for the classroom and as a research tool for solving problems in computational science and engineering [4]. It can solve forward problems given initial and boundary conditions, as well as inverse problems given extra measurements, and it supports complex-geometry domains based on the technique of constructive solid geometry. Its design enables user code to be compact and to resemble closely the mathematical formulation of the problem, lowering the barrier to entry for researchers who are experts in their physical domain but not in deep learning. DeepXDE introduced residual-based adaptive refinement as a built-in capability and demonstrated the PINN framework across five different examples, contributing to the more rapid development of the emerging scientific machine learning field. The importance of such infrastructure should not be underestimated. The adaptive-sampling methods reviewed here are only reproducible and comparable when implemented in a common framework, and the availability of a well-documented library has enabled the systematic benchmarking studies, such as that of Wu and colleagues, that have advanced the field [5]. The broader scientific machine learning ecosystem, surveyed by Cuomo and colleagues, depends on this kind of shared tooling to translate theoretical insight into reproducible practice [3].

IX. APPLICATIONS: HEAT TRANSFER, THERMAL SYSTEMS, AND CONTROL

PINNs have been applied to multiple thermal and heat-transfer problems. Cai and colleagues studied forced and mixed convection, phase change problems, and thermal management in power electronics, including cases with unknown thermal boundary conditions [18]. Their work also showed how PDE residuals can support active sensor placement.

Park and Lee investigated the thermal behavior of a large-area STS304 hot plate for thermal nanoimprint lithography using CFD and experimental data [10]. Their study emphasized on achieving uniform temperature distribution and controlling thermal gradients, demonstrating the importance of accurate thermal modeling in engineering systems.

PINNs have also been combined with reinforcement learning for adaptive decision-making. RL-PINNs use reinforcement learning to learn sampling policies during PINN training [21], while deep reinforcement learning provides a broader connection between neural networks, control, and optimization [11]. These developments highlight the potential of combining PINNs with adaptive learning strategies for thermal systems and control applications.

Table no 1: Summary of the principal adaptive sampling methods reviewed, with their guiding signal and distinguishing mechanism.

Method	Guiding signal	Mechanism	Key strength
RAR/RAR-D	PDE residual	Add high-residual points	Simple, effective baseline
RAD	PDE residual	Residual-weighted distribution	Robust, avoids over-concentration
RANG	PDE residual	Variable-density node generation	Balances density and uniformity
PACMANN	PDE residual	Move points by gradient ascent	Scales to high dimensions
ASM I/ ASM II	Residual (+ gradient)	Sub-domain refinement	Captures sharp solutions
RL-PINNs	Learned utility	Reinforcement-learning policy	Single-round sampling
Importance sampling	Importance measure	Variance-reduced gradient	Statistical grounding
Hessian-based	Residual Hessian	Quadrature-guided placement	Provable accuracy
RSmote	PDE residual	Imbalanced-learning oversampling	Memory efficiency
h-adaptive	PDE residual	Local refinement	Analogy to adaptive FEM
gpKP (hybrid)	Residual + gradient	Hybrid criteria	Robustness across regimes

X. CHALLENGES AND FUTURE DIRECTIONS

PINN faces several challenges in spite of significant progress in its field. First, many sampling methods are supported mainly by empirical results, while rigorous theoretical guarantees are still limited [15]. Second, scalability remains difficult for high-dimensional PDEs, as adaptive sampling does not completely overcome the curse of dimensionality [16, 21].

Another challenge is understanding how adaptive sampling interacts with other techniques such as adaptive weighting, Fourier features, and gradient enhancement. Fair comparison is also difficult because different methods are often tested on different benchmark problems, highlighting the need for standardized datasets and evaluation procedures [5]. Finally, integration with experimental data and established numerical methods is needed. Reliable uncertainty estimation and interpretability are also important for improving trust in PINNs, particularly in engineering applications [10, 13, 18].

XI. CONCLUSION

This review discusses about twenty-four primary sources involving the introduction and development of physics-informed neural networks (PINNs), adaptive collocation points sampling, and neural network learning behavior. It began with the theory of artificial neural networks and the basic PINN formulation, where the PDE residual is incorporated into the loss function through automatic differentiation, allowing PINNs to solve both forward and inverse problems [1–4].

A strong emphasis is on the role of collocation points in PINN training. RAR, RAD, RANG, PACMANN, ASM, RL-PINNs, importance sampling, Hessian-based sampling and RSmote are different adaptive sampling strategies used for improving the PINN training [5, 14–16, 19–24]. The review also discussed spectral bias, NTK-based analysis, Fourier features, self-adaptive PINNs, and gradient-enhanced PINNs, which provide different perspectives on the learning behavior and training difficulties of PINNs [6–9, 12, 13, 17].

In the nutshell, the PDE residual serves as an important link between PINN training, adaptive sampling, and solution assessment. Applications in heat transfer, phase-change problems, and thermal management depict the growing potential of PINNs in engineering [10, 18]. However, challenges related to sampling, optimization,

accuracy, and computational cost still exist. Therefore, PINNs can be viewed as a complementary approach to established numerical methods, with continued research needed to improve their reliability and efficiency.

References

- [1] Maind, S. B., & Wankar, P. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1), 96-100.
- [2] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378, 686-707. DOI: 10.1016/j.jcp.2018.10.045. <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [3] Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of scientific computing*, 92(3), 88. DOI: 10.1007/s10915-022-01939-z. arXiv:2201.05624. <https://doi.org/10.1007/s10915-022-01939-z>.
- [4] Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM review*, 63(1), 208-228. DOI: 10.1137/19M1274067. arXiv:1907.04502. <https://doi.org/10.1137/19M1274067>.
- [5] Wu, C., Zhu, M., Tan, Q., Kartha, Y., & Lu, L. (2023). A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403, 115671. arXiv:2207.10289. <https://arxiv.org/abs/2207.10289>.
- [6] Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., ... & Courville, A. (2019, May). On the spectral bias of neural networks. In *International conference on machine learning* (pp. 5301-5310). PMLR. arXiv:1806.08734. <https://arxiv.org/abs/1806.08734>.
- [7] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., ... & Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33, 7537-7547. arXiv:2006.10739. <https://arxiv.org/abs/2006.10739>.
- [8] Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5), A3055-A3081. arXiv:2001.04536. <https://arxiv.org/abs/2001.04536>.
- [9] Wang, S., Yu, X., & Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449, 110768. arXiv:2007.14527. <https://arxiv.org/abs/2007.14527>.
- [10] Park, G., & Lee, C. (2019). Experimental and numerical study on the characteristics of the thermal design of a large-area hot plate for nanoimprint equipment. *Sustainability*, 11(17), 4795. DOI: 10.3390/su11174795. <https://doi.org/10.3390/su11174795>.
- [11] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529-533. DOI: 10.1038/nature14236. <https://doi.org/10.1038/nature14236>.
- [12] Jacot, S., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31. arXiv:1806.07572. <https://arxiv.org/abs/1806.07572>.
- [13] McClenney, L. D., & Braga-Neto, U. M. (2023). Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474, 111722.4.
- [14] Nabian, M. A., Gladstone, R. J., & Meidani, H. (2021). Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8), 962-977. arXiv:2104.12325. <https://arxiv.org/abs/2104.12325>.
- [15] Caradot, A., Emonet, R., Habrard, A., Mezidi, A. R., & Sebban, M. (2025, September). Provably accurate adaptive sampling for collocation points in physics-informed neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 19-37). Cham: Springer Nature Switzerland. arXiv:2504.00910. <https://arxiv.org/abs/2504.00910>.
- [16] Visser, C., Heinlein, A., & Giovanardi, B. (2026). PACMANN: Point adaptive collocation method for artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 452, 118723. arXiv:2411.19632. <https://arxiv.org/abs/2411.19632>.
- [17] Yu, J., Lu, L., Meng, X., & Karniadakis, G. E. (2022). Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering*, 393, 114823. DOI: 10.1016/j.cma.2022.114823. arXiv:2111.02801. <https://doi.org/10.1016/j.cma.2022.114823>.
- [18] Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6), 060801. DOI: 10.1115/1.4050542. <https://doi.org/10.1115/1.4050542>.
- [19] Mao, Z., & Meng, X. (2023). Physics-informed neural networks with residual/gradient-based adaptive sampling methods for solving partial differential equations with sharp solutions. *Applied Mathematics and Mechanics*, 44(7), 1069-1084. arXiv:2302.08035. <https://arxiv.org/abs/2302.08035>.
- [20] Zhao, Z., & Ren, B. (2025). PINNs with hybrid residual-driven adaptive sampling and weighted loss for the (2+ 1)-dimensional gpKP equation. *Physics Letters A*, 130998. DOI: 10.1016/j.physleta.2025.130998. <https://doi.org/10.1016/j.physleta.2025.130998>.
- [21] Song, Z. (2025). RL-PINNs: Reinforcement learning-driven adaptive sampling for efficient training of PINNs. *arXiv preprint arXiv:2504.12949*. arXiv:2504.12949. <https://arxiv.org/abs/2504.12949>.
- [22] Luo, J., Yang, Y., Yuan, Y., Xu, S., & Hao, W. (2025). An imbalanced learning-based sampling method for physics-informed neural networks. *Journal of Computational Physics*, 534, 114010. arXiv:2501.11222. <https://arxiv.org/abs/2501.11222>.
- [23] Trynda, J., Maczuga, P., Oliver-Serra, A., Garcia-Castillo, L. E., Schaefer, R., & Woźniak, M. (2025). An h-adaptive collocation method for Physic-Informed Neural Networks. *Journal of Computational Science*, 102684. DOI: 10.1016/j.jocs.2025.102684. <https://doi.org/10.1016/j.jocs.2025.102684>.
- [24] Peng, W., Zhou, W., Zhang, X., Yao, W., & Liu, Z. (2022). Rang: A residual-based adaptive node generation method for physics-informed neural networks. *arXiv preprint arXiv:2205.01051*. arXiv:2205.01051. <https://arxiv.org/abs/2205.01051>.